Knowledge and Situational Feedback in a Learning Environment for Algebra Story Problem Solving

MITCHELL J. NATHAN

Inquiry into interactive learning environments (ILEs) such as computer-based instructional systems, is necessarily rooted in both the nature of learning environment system design and the study of learning and problem-solving. This article addresses both facets in an effort to explore the role that theories of learning and competence play in the design of a system for training students to reason about mathematical story problems. Many successful computer-based tutoring environments are based on theories of cognitive skill acquisition and assimilation of expert knowledge. ANIMATE, an ILE for mathematical story problem solving, is cast from a different theoretical framework which highlights the role that comprehension-based processes play in problem solving within knowledge-rich domains, while underplaying the role of technologically intensive student modeling and knowledge-based feedback. Performance with it is compared to a variant that provides knowledge-based feedback in a learning experiment. The nature of the performance differences elicited by the two systems brings up basic issues of the role that learning theories play in the design of learning environments.

THE IMPACT OF THEORIES OF LEARNING ON LEARNING ENVIRONMENT DESIGN

Inquiry into interactive learning environments (ILEs) such as computer-based instructional systems, is necessarily rooted in both the study of learning and problem-solving and the nature of learning environment system design. This article addresses both facets in an effort to explore the role that theories of learning and competence play in the design of a system for training students to reason about mathematical story problems.

The issues surrounding design choices of learning environments are complex and not easily addressed. But from where does the designer obtain the design, or the principles which can guide, suggest, and constrain it? Psychological theories of learning and competence in a

Interactive Learning Environments
Volume 5, pages 135-159

ISSN: 1049-4820

Mitchell J. Nathan, University of Colorado, Campus Box 249, Boulder, CO 80309-0249; Tel: (303)492–612; E-mail: mitch.nathan@colorado.edu.

domain are seemingly natural sources of principles for the ILE designer, and the benefits often listed are manifold. First, theories of learning help to point the way toward effective designs in an instructional design space that is extremely large and difficult to formalize (cf. Miller, 1982). In such a large space it just is not likely that designers will "stumble" onto the set of optimal design characteristics without constraints to aid that search. Psychological theories of learning and competence play a heuristic role in the generation of instructional principles and ILE designs. Second, when the design process is tied directly to a theory of competence, the principles employed in the system are seen as drawn from a larger theoretical picture of the workings of the mind and the complexity of the domain (Anderson, 1993). Third, design principles rooted in a theory have the promise of being internally consistent. Finally, there is the added reliability expected from systems based on principles that have been tested empirically (Polson & Richardson, 1988). Following this rationale, there have been repeated calls by leaders in the field of CAI and cognitive psychology for ILE designs to be rooted in some theory of learning and competence in a domain (Anderson, 1989b; Cummings, 1990; LeBlanc, 1993; Koedinger & Anderson, 1990; Nathan, Johl, Kintsch & Lewis, 1989; Ohlsson, 1986; Yazdani, 1989). This is in contrast to an early (and continued!) tendency to base ILE designs primarily on intuitions of how learning proceeds in a field, or on novel uses of the available technology (Regian & Shute, 1992).

TWO VIEWS OF LEARNING ENVIRONMENT DESIGN

Intelligent Tutoring Systems (ITSs)

Some of the most theoretically well-founded tutoring system design efforts have been used to develop Intelligent Tutoring Systems (ITSs). These are systems that tutor novices in a particular domain of study by using expert problem-solving knowledge as a means to classify a learner's response and provide feedback. Error detection (and in some cases correction) is typically performed by the tutoring environment as the chief means of instruction. In the best of situations, stored expert knowledge is augmented with common student misconceptions in the domain to facilitate error diagnosis. Many systems use the interactions of the system with the learner to infer deficiencies in the student's knowledge base. They build from this a student model which is used to adapt instruction to the student's individual needs.

The most fruitful and well-tested of these tutoring systems have come from Carnegie Mellon's Advanced Computer Tutoring Project (ACTP) program, headed by John Anderson. Central to the designs of the ACTP tutors for LISP, geometry and algebra is the notion that the problem-solving knowledge possessed by domain experts (and ultimately to be acquired by the learner) can be naturally represented by the rules of a production system. Learning is characterized by the acquisition of the expert's production rules, or the replacement of the novice's buggy rules with those of experts. Analyses have shown clearly that the learning that emerges from using the CMU LISP Tutor, for example, can be modeled most effectively by the acquisition of the specific production rules that underlie particular skills in LISP programming, and that the knowledge elements represented by production rules are learned independently of one another (Anderson, 1989a; Pirolli, 1991).

Many of the basic instructional principles which shape the designs of the ITSs developed by Anderson and his colleagues have direct roots in Anderson's ACT* theory of cognitive skill acquisition (Anderson 1983, 1987; Anderson et al., 1984). For example, the prediction of the ACT* theory that excessive working memory load causes learners to lose goals leads to the instructional principle to minimize the working memory load placed on the learner by providing partial products, and (visually) making the goal structure available to the learner (also see Glaser and Bassok, 1984).

Challenges to the ITS Approach

Intelligent tutoring systems rely on the ability of the developers to construct a model of competent problem-solving performance in a domain and to project onto this model the behavior of a learner. It has been argued that ITS technologies based on student modeling may be fundamentally limited because of a number of reasons. While the success of some ITSs has been well-established (e.g., Anderson, 1993; Lesgold, Ivill-Friel, & Bonar, 1989), several issues have been raised which bring into question the generality of systems designed for knowledge-based feedback in open domains (e.g., Koegel, Lakshmipathy, & Schlesinger, 1989; Nathan, Kintsch, & Young, 1992). For example, expert modules, as with expert systems, can only be constructed for domains where expertise exists and is codifiable (Waterman, 1986). Another concern that has been levied is for the appropriateness for performing student modeling at all. As noted in Derry and LaJoie (1993, p. 3), student modeling necessarily limits the student ". . . to follow solution paths that the machine can recognize." It also may rob the student of the need to perform the diagnostic task–an aspect of problem-solving which appears to be basic to competency in a variety of domains as diverse as mathematics, text editing, and writing (Nathan, 1991). Relegating diagnosis to the tutoring environment may implicitly teach students that evaluation and reflection upon a solution is not considered relevant to the problem-solving process. This is particularly bothersome since many students already perceive problem solving as an immediate, single-step process, rather than the deliberate and reflective process described by cognitive scientists and educational reformers, and exhibited by practitioners.

There is also empirical evidence that the "intelligence" centered design of ITS systems is not the only vehicle for the learning gains enjoyed by some ITS users. In one classroom assessment of the CMU geometry tutor, for example, much of the learning was ascribed to social factors which extend well beyond the scope of the ACT* learning theory upon which the tutor design is based. Schofield, Evans-Rhodes and Huber (1990) noted that one of the major factors that improved subjects' performance was an increase in students' efforts based on the peer-competitive atmosphere that emerged around the tutor. The ACT* theory attributes learning to prove geometry theorems to changes that occur to individual students' mental representations of geometry knowledge at the level of production rules (Anderson, Boyle, & Yost, 1985; Koedinger & Anderson, 1993). This includes changes in students' declarative and procedural knowledge that are specific to geometry theorem-proving: analyzing diagrams, planning a proof, setting subgoals, accessing and applying relevant theorems and postulates, learning through the analogy to prior examples, and so on. These changes, in turn, come about through the direct acquisition of rules and rule composition, changes to the activation of rules, and manipulations of the goal structure (Anderson, 1993). Social changes such as those at the community level, and shifts in students' competitiveness may ultimately impact an individual's learning processes through underlying mechanisms posited by the (now) ACT-R theory. But the theory itself, through its set of instructional principles (e.g., Anderson et al., 1984), is silent about these social factors.

Student modeling and the expert-like feedback that ITSs attempt to provide are intended to bring about the large gains brought about by human tutoring (Bloom, 1984). Recent analyses of human tutoring discourse, however, suggest that these instructors make little use of such a dynamically evolving student model (Fox, 1993; Graesser, 1993; Lepper et al., 1993). Approximately 70% of the speech acts from the human tutor in Graesser's corpus were found to be guided by predetermined curriculum scripts, while "only 8% of the tutorial interaction was devoted to the correction of student bugs and misconceptions" (Graesser, 1993, p. 127). Behavior guided by preconceived examples and questions are far more characteristic of the human tutoring experience that yields the large learning gains noted by Bloom (1984) than are the ITS interactions customized to address each student's specific question, error, and misconception.

UnIntelligent Tutoring Systems (UnITSs)

An alternative view of cognitively based instructional technology has been offered that shifts away from the centrality of knowledge-based feedback in error diagnosis and relies instead on the learner as the diagnostician. The LOGO programming environment is the premiere case (Harel, 1990; Papert, 1980). However, several other investigators have developed learning environments which present alternatives to student modeling (e.g., diSessa, 1985; Roschelle, 1987; Self, 1988). The essential property is that the system is able to reflect back to the learner observable and meaningful ramifications of the learner's actions in such a way that the learner can use her prior knowledge to identify solution errors, re-examine prior conceptions, and propose and test hypotheses about the causes of errors (Nathan, 1991).[1] The learner then corrects the error, perhaps over multiple iterations, and thereby verifies her understanding of the new material and its relation to her prior knowledge.

A computer-based learning environment, ANIMATE, has been developed which promotes learning in this way. It is designed to enhance learners' mathematical story problem-solving performance by supporting their problem comprehension processes (Nathan et al., 1992). ANIMATE is cast from a theoretical framework which highlights the role that comprehension-based processes play in problem solving within knowledge-rich domains, while underplaying the role of technologically intensive student modeling. It does so through a technologically modest design which has been termed "unintelligent" because it supports learning without student modeling or knowledge-based feedback (Derry & Lajoie, 1993; Kintsch, 1991; Nathan et al., 1989). In place of tailored, knowledge-based feedback from the tutor's expert module, the unintelligent tutoring system uses state-based feedback which engages the student's own knowledge so that the student may provide self-assessment and error correction.[2] Specifically, students construct algebraic equations which drive an animation of the story problem situation (e.g., workers painting a fence at different rates). Because of the direct causal link between the animation and the formal expressions of the algebraic solution, unexpected behaviors in the animation–actions that are inconsistent with the student's model of the story situation–suggest errors in the mathematics of the proposed solution, the nature of which is highly constrained by the type of misbehavior. Students debug their solutions and test them until an acceptable story situation is depicted. By explicitly connecting the mathematics to the situation, the theory hypothesizes, students will learn how to interpret the mathematical formalisms they construct and manipulate.

Design of an ILE for Mathematical Story Problem Solving

Experimental results with this learning environment have been promising: ANIMATE users show significantly higher posttest scores than do users of a variety of control learning environments (Nathan et al., 1992). The design of the ANIMATE system is based on a theory of mathematics story problem solving that accounts for several aspects of problem-solving difficulty and differences between high and low competency solvers.

The core theory: problem comprehension. The theory draws directly on constructs from the reading comprehension framework of Kintsch and van Dijk (Kintsch, 1988, 1991; van Dijk & Kintsch, 1983) as well as earlier work on the comprehension processes and problem-solving difficulties that students experience when solving mathematics story problems (Carpenter et al., 1980; Cummins et al., 1988; De Corte & Vershaffel, 1980; Kintsch & Greeno, 1985; Reusser, 1985).

In their theoretical analysis of story problem comprehension, Nathan and colleagues (1992) showed that reading for certain tasks such as problem solving made use of multiple mental models in order to represent all of the critical aspects of the problem situation. The reader, for example, must necessarily elaborate on the given text, draw inferences, and so on, to produce a more complete account of the intended situation (e.g., Bransford, Barclay, & Franks, 1972; Fletcher & Chrysler, 1990; Schmalhofer & Glavanov, 1986; Weaver & Kintsch, 1987). This representation, which draws on the reader's prior knowledge of events and semantic knowledge as a source for the elaboration has been termed the *situation model* (van Dijk & Kintsch, 1983; Kintsch, 1988). A model which represents the mathematical relations among the quantities of the story, including those that were not explicitly mentioned, captured quite another aspect of the situation. This latter representation has been termed the *problem model* since it represents the mathematical structures needed to solve the problem.[3] (Kintsch, 1991; Kintsch & Greeno, 1985; Mertz, 1993; Reusser, 1990, 1993).

Problem solving can occur using exclusively the situation model or the problem model. For example, many beginning mathematics students and those unschooled in formal methods often think of a problem purely in situational terms (e.g., Baranes, Perry, & Stigler, 1989; Carraher, Carraher & Schliemann, 1987; Rogoff & Lave, 1984). Conversely, researchers have found students representing story problems in formal problem model terms to the near exclusion of the situation to which the equations and values refer (cf. the tactical solution approach of naive physics problem solvers in Larkin, 1983, 1985). Students using this approach often disregard the meaning associated with the equations of the values and thus may provide solutions which are physically or situationally implausible (e.g. Hinsley, Hayes, & Simon, 1977; Paige & Simon (1966); Silver, 1988). Even experienced problem solvers may rely on problem-solving approaches that rely exclusively on situation model or problem model based solution approaches (e.g., Hall et al, 1989; Tabachneck, Koedinger, & Nathan, 1994).

In contrast, observations of highly competent problem solvers show the power of using the situation model and the problem model in an integrated fashion. The coordinated use of a situation model with one's formal problem model appears to be fundamental to problem solving with understanding in a variety of domains. Pennington (1987), for example, showed a similar pattern among professional programmers. She observed that overall, low-comprehenders tended to think about computer programs either in terms of the referent situation or in formal, programming terms (e.g., the datastructures and algorithms), but they rarely did they make connections between the two views. Similar patterns can be seen among students in physics, writing, and geometric theorem proving (Koedinger & Anderson, 1990; Larkin & Reif, 1983; Scardamalia, Bereiter, & Steinbach, 1984).

Comprehension-based problem solving. Why be concerned with comprehension processes during problem-solving? First, comprehension failures have been shown to be a major source of poor problem-solving performance. Second, instructional interventions aimed at supporting the problem solver's comprehension processes have been empirically shown to enhance problem-solving performance.

Cummins and her colleagues (1988) showed that the solution performance of first-graders was associated with their ability to comprehend the original problem statements. Nathan & Resnick (1993) showed empirically that "failure to make and algebraically specify necessary story inferences contributes greatly to [students'] poor problem-solving performance." (p. 120). Approximately 60% of all errors were attributable to failures of students' inference-making processes because the story problems were poorly or incompletely comprehended. This indicates the importance played by learners' prior knowledge of the world, and need for support in this area.

When comprehension problems arise, good learners and problem solvers try to make sense of the task. Tabachneck, Koedinger and Nathan

(1994) found in their analysis of the verbal protocols of story problem solvers how the need to comprehend a story problem mediates strategy changes during story problem solving. A coordinated use of unschooled strategies which enhance comprehension of the problem, and calculation-intensive schooled strategies such as algebra effectively doubled students' performance. Several studies have shown that those who learned the most tended to produce explanations which tied the new material to their prior knowledge (Bielaczyc & Pirolli, in press; Pirolli & Bielaczyc, 1989; Chi et al., 1989, in press).

In order to test the hypothesis that instructional support for one's comprehension processes enhances problem-solving performance, Nathan and colleagues (1992) developed ANIMATE. The learning environment aimed at improving subjects' awareness and skill at integrating their mathematical knowledge with their knowledge of the situation of the story problem (Nathan, 1991; Nathan et al., 1992).

Previous Results with ANIMATE

In an earlier empirical evaluation of the ANIMATE learning environment (Nathan, 1991; Nathan et al., 1992), subjects were pre-tested for their ability to solve mathematics story problems from the latter stages of the Algebra I curriculum (e.g., Foerster, 1984). All of the problems involved constructing systems of interrelated linear equations to determine values for intensive quantities (cf. Kaput, 1985).

Subjects were assigned randomly to one of five treatments, summarized in Table 1. ANIMATE users were able to set up a simple depiction of the situation for a problem on the computer screen, such as Problem 1 below.

> In a race to save the passenger train from possible destruction, a helicopter was sent out to warn the engineer of a bridge and radio tower that had been washed out only hours before. The train left Central City two hours before the helicopter operator had been notified of the incident. With the train traveling at an average speed of seventy-five miles per hour, the helicopter operator was ordered to fly at full speed—nearly three hundred miles per hour! If the train is 60 miles from the broken bridge, can the helicopter notify the engineer in time?

The animation served as an externalization of the situation model hypothesized to be constructed by the solver reading the problem (Kintsch, 1988, 1991). Additionally, the solver was encouraged to construct an interconnected set of algebraic equations intended to capture the quantitative relations of the problem. This served as an externalization of the problem model—the representation of the problem situation in terms of the solver's formal mathematical knowledge. The animation could be run at anytime during the problem-solving process, but it would behave only in the manner specified by the formal relations of the problem model. Mismatches that subjects perceived between the animation and the expected behavior of the characters (as determined by subjects' situation models) reflected errors in the formal specification of the solution (the formal problem model), and could only be resolved through changes in the mathematical equations.

For example, consider the attempt to solve Problem 1 above using the network of algebraic equations shown in the right-hand corner of figure 1 and animation as feedback. The movement of the helicopter in the upper window part of the screen incorrectly precedes the movement of the train, as is clear from a careful reading of the story problem. This action is governed, however, not by the story, but by the mathematics provided by the learner in the right-hand window. The learner builds a network of values and operators. In previous studies (Nathan et al., 1992; Weaver & Kintsch, 1992) students have been shown to translate from this network representation to a conventional set of algebra equations with ease.

The learner then enters particular values in order to instantiate the events and values of a particular story problem. As is typical of many learners that have been observed, the phrase "the helicopter leaves two hours after the train" has been interpreted as the mathematical expression T1 (for the train) equals T2 (for the helicopter) *minus 2* (Figure 1). This misunder-

Table 1. Treatment differences and the associated performance improvement

Training Groups	*Algebra Review*	*Network Formalism*	*Link Situation to Network*	*Set Up and Run Animation*	*???*
ANIMATE	•	•	•	•	+50
No Running	•	•	•		+24
Network-Only	•	•			+30
Situation-Only	•			•	+1
Control (no tutor)	•				

$^{*}p < .05$

standing of marked terms in story problems has been well-documented by A. Lewis (1989; Lewis & Mayer, 1987). Graphically this is constructed as,

$$T1 = T2 - 2$$

Situationally, however, the travel times of the train (T1) and the helicopter (T2) are mathematically related by the expression,

$$T1 = T2 + 2$$

Despite the negative relation as conveyed by the cover story, in order to faithfully represent the situation as described by the problem, one must add two (hours) to the travel time of the helicopter (T2) in order to equate the travel times of the two vehicles (since the helicopter actually will travel less time at the point that it overtakes the train).

The presence of the –2 (minus two) in the algebraic network causes the helicopter to commence travel two hours before the train (contrary to the story), as depicted by the animation and associated clock of Figure 1. It is now up to the ANIMATE user to evaluate the animation

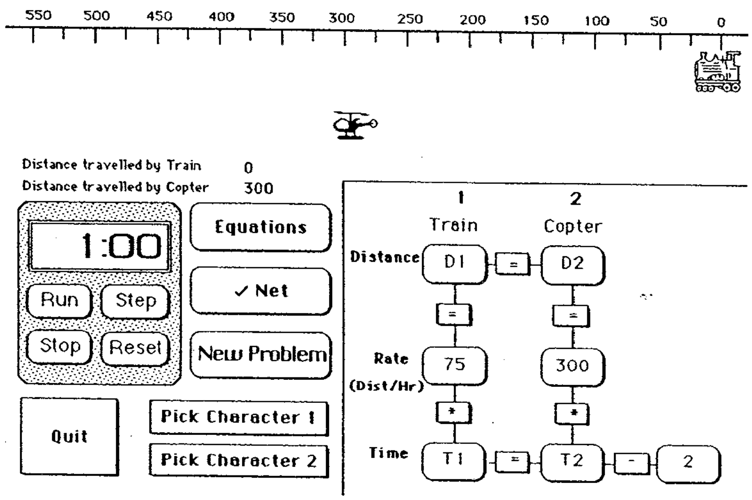

Figure 1. The ANIMATE Screen During a Solution Attempt of an Overtake Problem. The Learner is to Use the Animation (top pane) as a Source of Feedback to Highlight Conceptual Errors in the Solution Attempt (right-most pane).

with regard to the intended situation and, if necessary, to resolve any contradictions or expectation violations. Changing the equation to T1 = T2 + 2 results in the train leaving two hours before the helicopter, and thus providing an animation consistent with the overtake situation described.

This form of directed user control serves to explicitly link the external forms of the situation model and the problem model in a causal manner. In line with the theory, problem comprehension was facilitated to the extent that subjects coordinated their internal, mental representations of the situation model and the problem model. This internal coordination was hypothesized to provide a situational interpretation for mathematical structures, while supporting the (situation model-mediated) inference-making process needed to fully specify a formal solution.

Subjects in the remaining treatments received successively less support from their learning environments in order to experimentally control for the effects attributable to practice, computer use, or use of the situation-based problem-solving method. Those in the Stop-Condition treatment received an identical learning environment to the ANIMATE system, except that they could not actually run the animation. Subjects in the Situation-Only treatment could set up and run the animation of the problem situation, but were provided no explicit link to the mathematics. Subjects in the Network-Only treatment could set up the mathematics but had no link to the animated depiction of the problem situation. Finally, the control group solved the problem using paper and pencil.

As the results of Table 1 indicate, subjects in the ANIMATE condition showed the largest improvement and the highest posttest scores. Furthermore, an analysis of the problem-solving process indicated that providing support for subjects' situation model construction helped in their inference-making abilities during problem solving. Solution components were either based on necessary inferences and elaborations, or based on information present in the problem statement. Students' most frequent pretest solution errors, 63%, were errors of omission and commission for algebraic expressions based on inferences. Although inference-based errors were still the most numerous at posttest time, ANIMATE users showed the largest decrease in such errors, at the 1% level. Control subjects who developed the computer animation which was not driven by mathematical equations (i.e., the Situation-Only group) showed the second largest decrease in inference-based errors, further reinforcing the importance of situation model development on once inference-making and reasoning performance.

The Theory of Unintelligent Tutoring

Unintelligent tutoring system (UnITS) design, as it has come to be called, appears to some to exist primarily as a reaction to ITS research and development (e.g., Lajoie & Derry, 1993). The term "Unintelligent tutoring" certainly suggests this; but the label is, in actuality, something of a misnomer. Most systems which fall under this category (Nathan et al., 1992; Papert, 1980; Reusser, 1993) are not accurately portrayed if considered to be totally devoid of intelligence, since they utilize "off-line" knowledge supplied by the system developer (cf. Koschmann, 1994). This can take the form of checks on the syntactic structure of a user's input based on stored knowledge, often in the form of rules. Even more substantive aspects of a user's entries may be viewed through a knowledge filter within the system. These systems also rely on knowledge provided by the user "on-line" to determine their behavior.

Furthermore, the approach of unintelligent tutoring is undermined if it is considered to be chiefly as a system lacking expert knowledge.[4] The approach is, rather, a philosophy of cognitively based educational technology and tutor design that focuses on ways to elicit knowledge from the user during problem-solving and instructional activities. With this approach comes a set of basic properties for such a system and principles for the instruction and learning that emerge. These properties are articulated later in this section.

In contrast to the ITS approach, unintelligent tutors such as ANIMATE cannot assess the student's performance because they typically possess no expert module or student model and, in fact, have no knowledge of the correct answer, or even of the problem being solved. Instead, unintelligent tutors provide *situation-based feedback which the student must interpret.* The interpretation process forces the student to integrate mathematical and situation-based knowledge by making predictions and comparing the computer animation to expectations drawn from the text. When the animation deviates from the expected behavior, the student must look to the mathematics which drives it and decide how to modify the equations to produce the intended animation. A correspondence between the situation and components of the solution are formed by the learner—thus creating situation-based meaning for the equations which, empirical results to date suggest, enhance students' mathematical reasoning about algebra story problems.

Within the UnITS approach, the intention is, as with the ITS approach, to bring about procedural, declarative, and metacognitive learning; and this is done, as with ITSs, through a knowledge-rich interaction between the tutor and the learner. Unlike the typical ITS approach, the intelligence used for knowledge assessment is provided chiefly by the learner. Rather than trying to "understand" the learner, the UnITS reflects the learner's own performance back in a meaningful way. The aim here is to encourage learners to be self-evaluative and reflective, to assess their own problem-solving performance, and to detect and correct their errors. The claim here is that it is not necessary that the tutor understands the learner. This is merely a means to an end. The central goal is for the learner to achieve greater understanding of the problem-solving process and the underlying principles of the domain through knowledge elicitation, knowledge integration, and reflection. Consequently, the emphasis in UnITS development is on a thorough analysis of the task domain (cf. Reusser, 1993) and on the *user interface* which provides interpretable feedback, rather than on knowledge engineering and expert module development which directs knowledge-based feedback.

The approach used by ANIMATE also makes an important aspect of the problem-solving process overt: Reflecting on a solution is vital to robust problem solving (cf. Schoenfeld 1989). Making errors is part of complex skill development. Presenting a solution attempt in a form that is meaningful to the student engages solution assessment and correction (i.e., debugging) skills. All of these aspects of unintelligent tutoring have suggested a set of principles which govern their design and use.

Principles of Unintelligent Tutoring System Design

Achieve a correspondence between a known situation and the to-be-learned material. The goal is for a correspondence to be achieved between the situation within which a task is embedded and the components of the solution process. This creates a meaningful interpretation of the material which must ultimately be learned to support the solution process. ANIMATE achieves this by grounding the interpretation of the algebraic notation to a running animation depicting the situation implied by the cover story of a problem. The situation is known to the student and the student can apply her knowledge of the situation to reason about the constraints and outcomes. As noted, behaviors in the animation, expected or not, tie directly back to the specific mathematics constructed by the student. The behavior of the animation can only be changed by making adjustments to algebraic terms and operators. In this way, the mathematical notation takes on a situation-based meaning which supports learning and comprehension as well as novel reasoning activities.

Elicit the user's knowledge. Knowledge-based feedback is an effective way to foster content learning (e.g., Anderson, 1993). But it is based fundamentally on a model of knowledge (or cognitive skill) refinement, rather than knowledge elicitation and knowledge building. Working with an UnITS is intended to be an interaction which puts the student's knowledge to the forefront. The goal is for the student to use the knowledge she feels is relevant, observe the application of that knowledge, notice its relevance, and reflect on the process by which this particular knowledge came to be selected and applied.[5] When that knowledge is elicited from the user rather than from some unfamiliar source, this process can additionally foster higher-order thinking—metacognitive skills such as comprehension monitoring (Chi et al., 1989) and internal learner control (Lin, 1993). Results with the ANIMATE system also indicate a greater likelihood of making the inferences needed to fully and correctly specify solutions for many story problems (Nathan & Resnick, 1993).

Make a causal link between the student-entered solution representation and a student-meaningful representation. The effective presentation of multiple external representations for a phenomenon is a tricky task. In principle, we are providing the learner with a wonderful opportunity: To see several perspectives (often in the abstract sense of the word) of a phenomenon or concept in order to enhance its understanding (e.g., Comfrey, 1991; Courey & Pietras, 1989). When multiple representations are used to provide an understanding of the behavior of a phenomenon or formal system, there is a danger, however, that the user will have his attention misdirected, fail to see the relevance of these different representations, or fail to see how they correspond.[6] Consequently, these representations must clearly be linked to the intended material (see LeBlanc, 1993). The focus on establishing a single, uni-directional causal link from the solution representation (i.e., the algebraic formalism) and a situational depiction of that representation allows the system to reflect back a meaningful interpretation to the student. This is why there needs to be such a large emphasis on the interface design and user-testing of these systems (Nathan, 1990).

While the incorporation of this principle does not distinguish UnITS from some ITS designs, the relative importance it enjoys in the different approaches does. In ANIMATE, for example, representations for the mathematics and the situation are causally related. It is paramount to the learning experience that what one presents in the mathematical representation drives what will occur in the situation-based representation. In this way, attention of the learner is clearly directed toward planning activities early on, as one constructs an initial skeleton of the solution, and diagnosis as one checks the accuracy of the mathematics with one's situation model.

Error-making in a meaningful context provides learning opportunities. Students' difficulties for retrieving knowledge relevant to task performance are well known (e.g., Bereiter, 1984; Bransford, Franks, Vye, & Sherwood, 1989; Phye, 1992). However, when the subject is operating within a meaningful context, access to prior knowledge is facilitated and enables one to diagnose and correct errors. Children show enhanced mathematical problem-solving behaviors when the problems are couched in familiar money terms, for example (Baranes et al., 1989). Problem-solving activities in mathematics and science are also enhanced when they are "anchored" to rich stories that engage students (Bransford, et al., 1989; Sherwood et al., 1987).

Educational programs which help students perform their own causal reasoning for diagnosis by exploiting their small but powerful set of causal reasoning heuristics (e.g., Lewis, 1986) allow the student to take charge of the learning process and provide the tutoring system designer with a helpful ally in error diagnosis.

Support for error detection and recovery is central to developing competence in a domain. Critical to the learning process is the role of the student's knowledge in evaluating each solution attempt. While addressing the credit assignment problem (e.g., Samuel, 1963) is a notoriously difficult problem for AI systems, humans, drawing on their vast prior knowledge of situations, can exploit the associations they form between erroneous behaviors in the referent situation and its cause in the formal domain. Performance assessment and error recovery are crucial skills when teaching any domain.

Observations of expert performance in a variety of areas such as physics (Larkin & Reif, 1983), mathematics (Allwood & Montgomery, 1981, 1982; Lewis, 1981), writing (Scardamalia, Bereiter & Steinbach, 1989), for example, show that competent behavior in a domain is not characterized by flawless performance, but by effective recovery from errors. Few real-world learning situations inform a learner of the nature of his error, the correct behavior to perform in the future, and the place one can go to review the correct behavior for the future. Few real world environments are so brittle that they admit only a narrow set of behaviors that are considered to be adequate.

Permitting students to perform their own error-recovery is both more natural in human interactions, and may lead to greater learning. Observations of everyday conversation as well as interactions of learners with human tutors show quite clearly a preference for self-correction on the part of the one making the error (Fox, 1993; Schegloff, Jefferson, & Sacks, 1977). When given sufficient time from a human tutor, students will often correct their own mistakes. Empirical findings from two computer tutoring studies that compared immediate feedback supplied by an expert module, with delayed feedback that occurred at the end of a learning session, show that immediate feedback, while speeding up the lessons, can interfere with students' thought processes and block students' own diagnostic reasoning (Lee, 1989; Schmalhofer, Kuehn, Messamer, & Charron, 1989). Immediate feedback can lead to "guessing strategies" which rely too heavily on tutor feedback and subsequently impair one's learning (Lee, 1989). In addition to these findings, it has been shown (e.g., Shute, Woltz, & Regian, 1989) that providing well-crafted hints thought to be beneficial to the learner can have a detrimental effect on the user's learning process. While learning may initially be aided by the knowledge-based support, over time it was found to adversely affect both the learning time and the amount of knowledge acquired. As an alternative to expert module support, Ohlsson and Rees (1991) demonstrate how conceptual learning and self-correction in the area of mathematics can occur without external feedback and instruction.

Place the to-be-learned material along the path of least resistance. We must acknowledge as learning environment designers that the task we are analyzing is multifaceted. The student operating a tutoring environment is simultaneously developing mental representations of, minimally, the domain of interest, the tutoring system with which she interacts, the task put before her, and her own reasoning and problem-solving processes. There are many aspects of the task of solving a set of algebra problems with an algebra tutoring program which have little or nothing to do with algebra itself. Students learn to find the path of least resistance through an assignment. Successive guessing, for example, may be the easiest way to elicit the answer to a problem from an ITS or human tutor. In a comparison of immediate and delayed feedback in a genetics tutor, Lee (1989) replicated the finding by Anderson and colleagues that the immediate feedback condition led to more rapid learning. However, subjects in the delayed feedback condition provided more correct answers in the posttest. Delayed feedback subjects thought more about the test questions during training and took more time to notice incorrect responses. Their post-test performances benefited from the opportunities to observe change incorrect responses to correct ones (Lee, 1989). Error identification has been shown to improve with delayed feedback in other domains as well, such as navigation games (Lewis & Anderson, 1985). In contrast, the greater guessing used in training by the immediate feedback subjects appeared to be harm their post-test performance.

Reasoning by analogy from given examples in a textbook may be another (cf. Anderson, 1993;

Chi et al., 1989). In the ANIMATE system, a conscious effort is made to make algebraic modeling through equation construction the gateway into the system. Focus of the instructional interaction is on the intended material (e.g. the algebraic formalism). Feedback is provided by the student and has meaning only with respect to the appropriateness of the mathematics for describing a situation.

EXPERIMENTAL INVESTIGATION

Initial findings with ANIMATE use (described above) are very encouraging for the underlying comprehension-based theory of learning and problem solving, and for further exploration of the UnITS approach. But the findings to date are incomplete. In earlier studies none of the other treatment groups received feedback comparable to that provided by animation in the ANIMATE system. This may have been the vital component that lead to superior performance. In order to test this hypothesis, a new experiment was conducted where all subjects received feedback on their solution attempts. The comparison was made between the type of feedback received by subjects. Subjects received either the situation-based feedback suggested by the comprehension model (Nathan et al., 1992), or knowledge-based feedback of the type employed by ITSs (Anderson, 1993; Koedinger & Anderson, 1993; Singley et al., 1989).

METHOD

Subjects

Forty undergraduates from the University of Pittsburgh signed up to be in this study in order to meet the requirements for their introductory psychology course. Seven subjects were dropped from the experiment due to problems with their respective computer-based learning environments. Two subjects failed to follow proper directions. Thus, data from thirty-one subjects are used in the final analysis.

Materials and Design

The materials consisted of a test booklet, a pen, and an Apple Macintosh II workstation which ran the ANIMATE (version 2.1) or the ANIMATE+K (ANIMATE plus Knowledge-based feedback) learning environment under Hypercard™ 2.0. The test booklet contained a pretest section, a set of training tasks that were used with the computer, and a posttest section. Each section was introduced with a page of instructions. The pretest, training task and posttest each contained three story algebra problems—a travel problem scenario, an investment scenario, and a combined work scenario—typed one problem to a page.

Subjects were assigned to either the knowledge-based feedback group (knowledge FB) or the situation-based feedback group (situation FB) according to the order in which they signed up for participation in the experiment. Subjects in both groups received the same problem-solving tasks and the same task instruction. The two groups differed in the learning environment that they used to solve the training problems.

Situation-based feedback. Situation FB subjects used the ANIMATE system, as described above. The animation is driven by the values and operators subjects provide in the equations that they construct during their solution attempts (for detailed description of the system, see Nathan et al., 1992). Subjects are free to revise the situation or solution at any time. The animation provides a form of situation-based feedback for the formal mathematical elements of a solution.

Knowledge-based feedback. Subjects in the knowledge FB group used ANIMATE+K, a variant of the ANIMATE system which was augmented with knowledge about the sequences of equation construction which would ensure a legitimate solution. Animation of the situation as a source of feedback was disabled for this treatment. Instead, subjects were given immediate feedback in the form of pop-up messages when they strayed from one of the allowable

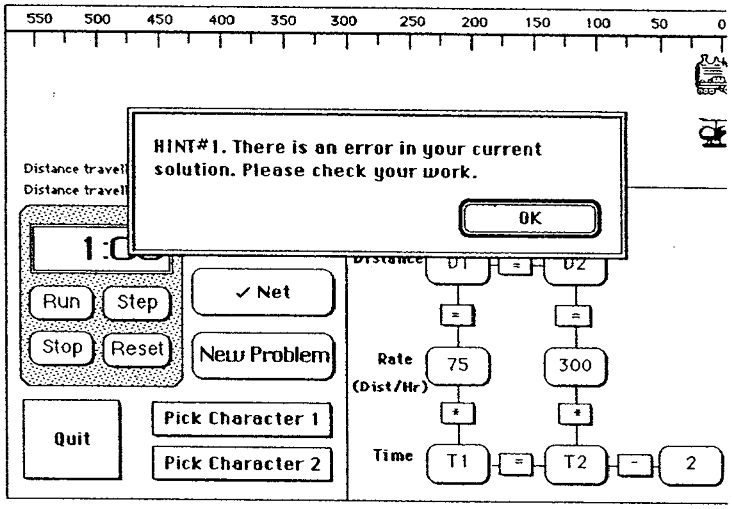

Figure 2. The Knowledge-Based Feedback in ANIMATE+K Initially Provides a Hint That Something is Wrong with the Student's Solution

solution sequences. Upon their first error, a subject in the knowledge feedback group was told that the equation, value, variable, or operator that was just entered into the workspace was incorrect (Figure 2). At the second error, the subject was given a hint for a next step in the solution (Figure 3). Upon the third error for the same state in the solution search space, a subject received a short description of the relevant algebraic concept that could be used to address that aspect of the problem.

Procedure

Each subject was seated in front of a computer workstation and received a booklet and a pen. Subjects were told that they would be presented with a novel way to reason about and solve algebra story problems. The training involved use of an interactive computer program. It was stated that prior to their use of the computer program it was necessary to see how they would perform on a preliminary test. The pretest was then administered without the use of a computer, and was terminated only when the subject was finished with all test problems.

Subjects then entered the training phase. The training phase had three distinct components: Subjects were familiarized with the use of the computer workstation; they were shown how to use the learning environment appropriate to their treatment group; and finally, subjects solved problems in the training section of their booklets using the assigned learning environment.

At the beginning of the training phase, subjects were introduced to the general workings of the computer interface. This involved teaching and/or reviewing the use of the rolling mouse cursor, entering numbers and letters with the keyboard, and using the mouse to select (i.e., highlight) objects and push buttons presented on the screen.

After going over the basic interface functionality, subjects were shown how to set up an example algebra solution and an associated problem scenario. Subjects learned how to select components of an algebra expression from an equation palette and build up a set of algebraic equations which model a problem scenario. Subjects also learned how to select characters that served as central figures in the problem and how to arrange them on the screen so they model the initial situation expressed in the problem text. Subjects were shown that they could construct the mathematical aspects of the solution and the situation aspects in any order

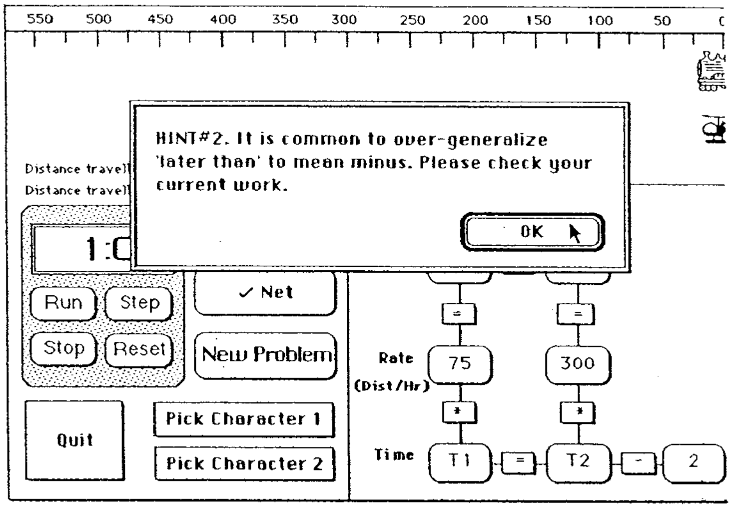

Figure 3. When an Error Continues in ANIMATE+K, the Second Hint is Directed to the Underlying Missing Concept or Misconception that is Believed to be the Cause of the Error

they liked. They were then given an opportunity to construct several simple scenarios on their own to explore the flexibility and utility of the interface. All subjects received immediate feedback for simple syntactic errors, such as producing unbalanced, incomplete, or algebraically illegal equations.

ANIMATE users then learned how the animation was causally linked to the mathematical equations that they constructed. By selecting the Run button (Figure 1) subjects saw the animation being executed in accord with the values and relations contained in the current set of equations. The animation and its associated gauges provided feedback for the equations. Changes in the equations resulted in changes in the behavior of the animation. Their goal was to get the animation to depict the events of the problem situation as they understood it, based on the cover story of the problem. If the animation failed to do this, then it must be due to an error in their current algebra solution.

ANIMATE+K users did not have the simulation capability provided to ANIMATE users, but were instead shown how their system could detect erroneous solution steps. Subjects were shown the various levels of feedback they would receive as soon as they made an error. They were then able to use the information provided by the system to correct mistakes.

In the final training component, subjects were instructed to solve three algebra story problems–a travel problem, an investment problem, and a work problem. The problems were presented in random order for each subject. All subjects were free to determine when they were done with each problem. Subjects were instructed that their final answers (often a set of equations) were to be recorded onto the appropriate page of the training section of the booklet.

Scoring

Problem solutions were scored against a series of solution templates which contained all of the steps needed to produce a quantitative solution to a given problem. A score for a problem solution reflected the proportion of the steps or solution components that overlapped with one of the templates. Novel solution approaches were compared to newly formed templates, which were constructed as needed. Solutions which contained a set of solution-enabling equations with no intermediate solution steps (i.e., a complete solution where no work was shown) received full credit. Each problem was scored out of one total point. The pretest, training tasks, and posttest each had a maximum score of three points.

RESULTS AND CONCLUSIONS

Problem-solving Performance

Subjects in the two different treatment groups participated in identical pretests, with no experimental manipulations present. Subjects in the situation group received an average pretest score of 1.65 (out of a possible three points), while those in the knowledge group scored 1.74 points on average. A one-way ANOVA of students' performances using pretest score as the dependent measure shows no significant treatment differences on the pretest, $t(30) < 1.0$, p = n.s. From this, we can safely assume that, as a group, subjects arrived with similar knowledge of the algebra subject matter that is investigated in the experiment.

A one-way analysis of covariance (ANCOVA), using posttest score as the dependent measure and pretest score as the covariate reveals that, on average, subjects' scores improved significantly from pretest to posttest, $F(30, 1) = 17.0$, $p = .0003$.

While subjects all improved on average, there were important differences attributable to treatment and problem type. In general, subjects using the ANIMATE system with situation-based feedback showed greater gains than subjects receiving knowledge-based feedback, though this advantage did not hold across all of the posttest problems. As Table 2 shows, situation FB subjects showed greater improvement for travel problems and investment problems, but showed inferior gains on work problems. An ANCOVA with posttest score on the travel problem as the dependent measure, pretest performance and total SAT (with 2 missing values) as covariates, and treatment as a between-subjects factor, showed a significant difference due to treatment, $F(27, 1) = 4.77$, MSe = 0.4, $p < .05$. The means (see Table 2) reveal that situation subjects had greater test gains than knowledge subjects. A similar analysis for investment problem-solving gains (with total SAT scores as the sole reliable covariate) also showed a significant effect for treatment, $F(28, 1) = 5.7$, MSe = 1.06, $p < .025$. As with travel problems, the means indicate that situation FB subjects achieved greater posttest performance than knowledge FB subjects. Situation-based feedback, as provided by ANIMATE, produced more effective training than did the knowledge-based feedback on solving later travel and investment problems without use of the computer.

Subjects did not show this same pattern of results for work problems, however. An ANCOVA with posttest score on the work problem as the dependent measure, total SAT (with 2 missing values) as the covariate, and treatment as a between-subjects factor, failed to show a significant difference due to treatment, $F(28, 1) < 2.24$, MSe = 0.21, p = n.s. Thus, per-

Table 2. Group size (n), and mean proportional performance improvement (along with standard deviations) for the travel, investment, and work problems as a function of treatment

Problem Type	*Treatment*	*n*	*Improvement (standard deviation)*
Travel	Situation feedback	17	.76* (.07)
	Knowledge-based feedback	14	.57 (.08)
Investment	Situationfeedback	17	.76* (.11)
	Knowledge-based feedback	14	.43 (.14)
Work	Situation feedback	17	.34 (.07)
	Knowledge-based feedback	14	.51 (.09)

*$p < .05$

formance differences of the two groups due to differences in training on this task was not distinguishable.

Analyses of the quantitative structure that underlies various algebra story problems has shown important differences across problems (Hall et al., 1989; Shalin & Bee, 1985). Work problems such as the one used differ markedly from motion and investment problems in their use of the intensive quantity (i.e., the rate at which a character completes a job when working alone). In their solutions, students needed to think about and manipulate the *reciprocal* of the given quantity (the most common error of commission for work problems during the pretest). Thus, given the fact that `A small hose can drain a pool in six hours by itself`, the solver must appreciate that the relevant quantity for an algebraic solution is not `Rate = 6 hours per pool`, but rather, `Rate = 1/6 pool per hour`. The system that provided subjects with knowledge-based feedback may be naturally better than situation-based feedback at conveying this useful fact. Situation-based feedback relies on the transparency of an error with respect to the associated situational animation. Many forms of error are properly conveyed this way: improper relative rates, delay information, the non-linearity of compound interest, and so on. These are aspects of the situation which map directly to the mathematical concepts that serve to model these events. Certain aspects of the algebraic problem-solving method, however, are more effectively learned when they are directly told to the student. The subjects in this experiment who made the work-rate error (e.g., entering a rate of 6 instead of 1/6) while working with the ANIMATE system saw that the relative rates were incorrect (the wrong character was fastest at completing the task), but often could not glean from the situation feedback that the reciprocal was the necessary quantity to rectify the error. There was no direct means by which the system could communicate this to the subject. Users of the ANIMATE+K system, however, were told by the system to examine the rates entered, and eventually as the error persisted, to use the reciprocal so that the units were consistent with the other quantities entered into the equation. This direct lesson carried over to the posttest problem-solving performances of these subjects.

Problem-solving Process

An analysis of subjects' solution errors was also conducted which focused on the frequency of errors made for information explicitly given in the problem statement, and needed information that was alluded to but not explicitly provided.

Inference-making

The most frequent type of error made during pretest was for information that was alluded to but not explicitly provided by the problem text, and consequently was dependent upon the solvers' use of inference-making. Of the 338 classifiable pretest errors made by the 31 subjects solving three problems, 247 of them (73%) were tied to information that had to be inferred from the cover story (Table 3). Unstated but necessary components of the problem solution tended to be omitted from the solution protocol more often (84% of the time) than they were included but improperly formulated. Those errors tied to given problem information (91 errors in all, or 27% of all classifiable pretest errors) tended to be misformulated in subjects' solution attempts (62%) more often than they were omitted from the solution protocols. This indicates that subjects are generally less successful at elaborating on the given (but incomplete) problem statements they are provided with in order to produce a correct and complete solution. Given information is a strong signal for its inclusion in a solution, although it may not be properly mathematized (i.e., translated) by the subject into an expression that correctly models the problem situation. These findings are consistent with prior results reported elsewhere (e.g., Nathan, 1992).

The reduction of these errors provides insight on the differential effects of the two treatments. While subjects in both conditions enjoyed similar improvement in providing and properly for-

Table 3. · Frequency (and percentages) of pretest solution errors or all subjects of omission and misformulation as a function of their explicit mention in the original problem statement

	Information Explicitly Given in Problem Statement	*Information Inferred from Problem Statement*	*Total*
Omission	35 (38%)	208 (84%)	243
Misformulation	56 (62%)	39 (16%)	95
All	91 (100%)	247 (100%)	338

mulating information that was given in the problem, subjects in the knowledge FB condition showed superior improvement for information absent in the problem statement. As Table 4 shows, subjects receiving knowledge-based FB as a form of direct instruction, reduced omission errors by 42% across all problems, while situation subjects reduced them by 27%. In previous work comparing ANIMATE users to users receiving no comparable form of feedback, the reduction of errors elicited from the situational feedback was sufficient to significantly elevate ANIMATE users' performance (Nathan et al., 1992). Here, too, the reductions achieved without knowledge directed feedback is still impressive. However, the impact of direct instruction is greater.

While direct FB on the source of solution errors supported the inclusion of unstated problem information best, situation-based FB was far more effective at reducing errors of misformulation for that same unstated (and so inferred) information. ANIMATE users reduced errors of misformulation by 45% while those receiving knowledge-based FB reduced them by only 11%. Thus, knowledge-based FB tended to be more effective in getting subjects to notice missing but needed aspects of their solutions, while the situation-based FB tended to support more correct modeling of this information once it was included in subjects' solution attempts. It is because of this that ANIMATE users ultimately achieved superior performance.

DISCUSSION

The earlier review of this article was highlight the essential role that learning theories play in the design and development of learning environments. However, as we strive to develop the science of learning environment design we must acknowledge the inherent limits of learning theories in prescribing the final implementation of learning environments. Theories of learning are not theories of instruction (Cobb, in press). It may also be necessary to contrast theories of learning and instruction in order to make clear the assumptions of a design, and to empirically counterpose alternative designs to better see their strengths and limitations.

The experiment presented here took this perspective by comparing changes in the problem-solving processes and performances of students exposed to two tutoring environments conceived within different learning approaches. The

Table 4. Proportion of error reduction (from pretest to posttest) acheived by the treatments for errors of omission and misformulation as a function of their explicit mention in the original problem statement

Treatment	*Error Types*	*Explicitly Given in Prblem Statement*	*Inferred from Problem Statement*
Situation	Omission	.50	.27
	Misformulation	.64	.45
Knowledge	Omission	.4	.42
	Misformulation	.75	.11

results suggest that training with the ANIMATE system, and its use of a particular form of interaction—whereby the learner is encouraged to interpret feedback at a situational level in order to assess mathematical performance—is sufficient to produce large and lasting learning gains in the complex area of algebra story problem instruction. These gains may also be understood in relation to the comparison treatment, for these subjects, too, progressed in ways that are not to be dismissed. Perhaps most importantly, there were trade-offs found between the two treatments. The results suggest that, while the gains are superior with the ANIMATE system, there are additional and worthwhile gains to be had from direct and immediate feedback provided by embedded domain knowledge that are not obtained from situational feedback. Specifically, subjects' were able to correct and avoid certain types of errors involving the use of intensive quantities that were not readily reflected back to them by the animation. Such findings are extremely valuable in supporting the attempt to improve our understanding of learning environment design.

Noticing and Direct Instruction

The nature of the differences in problem solving produced by these systems suggests that the ANIMATE system could benefit by making errors more salient to the learner. Novice-students do not always perceive the situation as experts would have them perceive it, and thus, may not notice important differences between situations. In this study, for example, Work problems look a lot like Travel problems, so many subjects in this study thought that the work rates (number of hours to perform a job) may treated similarly to the motion rates (e.g., added, subtracted, plug in the given values from the problem description). However, this was not the case since the rates provided in the problem were reciprocals of what subjects actually needed to solve the Work problems.

When novices first encounter the application of a new principle they don't necessarily have the discrimination skills to notice what they need to notice in order to apply it or recognize its misapplication (e.g., Garner, 1974; Randolph & Evertson, 1992). Specific cues which direct the learner's attention to the new concept play a critical role in the development of the pattern recognition skills necessary to notice relevant features in new situations (Bransford, Franks, Vye & Sherwood, 1989). Such cues can be provided by learned others, or by domain knowledge embodied in a computer-based tutor.

The ANIMATE system provided animation-based feedback to support students in noticing that "something is wrong," as when the incorrect character works faster. But the system apparently provided no direct way to focus subjects' attention to more subtle things such as the need to work with the reciprocal of the given rates for some problems. As vivid as the animation may be for aspects of the problem-solving process, it was too indirect to support subjects' awareness of the units involved and the need to create a new quantity (portion of job/hour) to properly reason about the quantities of Work problems. In contrast, the knowledge-based feedback of the ANIMATE+K system provided direct instruction on the need to invert the given work-rate quantity. Subjects were explicitly made aware of the units in the problem statement, and the need to have them altered for the solution schema. This direct instruction left little to be misunderstood or missed by the learner. Prior to their noticing, students could do little with any information about the role or different nature of the rate variable in the Work problem. Once learned in the context of the problem-solving activity, this information remained accessible to most learners during the posttest when no outside tutor support was available.

Providing Support for Error Detection and Correction

As the field of computer-based instruction matures, researchers and practitioners are realizing the great complexity of the task that lies ahead. The goal of the developer of a learning environment is to foster learning that will transfer to conditions outside of the instructional environment. In this more authentic setting there may be no dedicated expert monitoring one's progress, signaling errors and giving advice. The environment may not be so cooperative as to provide feedback which is salient and meaningful. Many authentic learning situations are similarly unsupportive.

Alternative 1: Feedback from a Domain Expert Module

Generally, the view has been that for learning to occur, feedback must come from some source external to the learner which is capable of interpreting the original output behavior and of comparing it to an ideal. The assumption that fosters this approach is, presumably, that the learner is not able to perform this comparison function; that this is, after all, the sought-after behavior and the reason that learning needs to occur. Thus, as ambitions grew for computer-based learning environments, the sophistication of the comparator capable of assessing the learner's output with respect to some ideal necessarily grew in sophistication, until it achieved the status "intelligent." As knowledge-engineering techniques were refined and knowledge representation tools became more powerful, faster, more accessible, and more flexible, the attainability of knowledge-based feedback increased. We now live in an age where a number of ITSs exist.

We are also at a point where it is worthwhile to reflect on the prospects of intelligent systems for education. Certainly there is cause to applaud these efforts. The instructional impact of a few well-tested systems developed under large research and development grants is worth noting. However, it is also valuable to ask why there are not more of them. Certainly, artificially intelligent tutoring is not yet a practical consideration from a financial point of view for most educational programs (Chipman, 1993). For this reasons, it is worthwhile to keep in mind low-tech options.

One of the major goals of the UnITS research program is to see how far computer assisted instruction can get without constructing a knowledge base and a student model. This project has yielded several important insights into the construction process of learning environments, several of which have been articulated in this article. Now we can reflect on more recent empirical results which acknowledge the value of knowledge-based feedback to support learning where an exclusive use of situation-based feedback cannot. The goal is still to hold onto the principles developed for the student-centered learning experience articulated earlier that encourages knowledge elicitation from the user. But we now can explore how knowledge-based feedback can most effectively be introduced into the system design when knowledge elicitation techniques prove to be insufficient.

One approach, clearly, is a selective development of domain knowledge packets that may be introduced into the system feedback mechanism and provide direct feedback of the kind demonstrated by the prototype. This approach inherits some of the advantages of large-scale ITSs, such as immediate and direct knowledge-based feedback. It inherits some of the limitations of ITSs articulated earlier in this article as well.

Since the earliest stages in its design, the ANIMATE system was intended to draw from the students themselves the monitoring and evaluation processes that are needed for meaningful feedback. Feedback closes a loop. In systems theory, its purpose is to provide a way to monitor the actual output of a system and compare it to the intended output. By comparing these two values, an appropriately designed system can correct itself. It has long been known that feedback for behavioral and cognitive systems can promote learning of certain behaviors and speed up their acquisition.

Alternative 2: Student Self-monitoring and Self-assessment

An alternative approach is to consider the larger environment within which the "unintelligent" system is embedded, and to ask whether or not that environment can provide the necessary scrutiny for performance assessment and correction. Two different educational approaches provide instances of such environments that are worth considering: Prompted self-explanation, and reciprocal teaching.

Researchers studying successful and less successful learners have identified active self-explanation processes that enhance comprehension monitoring and knowledge building during problem solving (Chi et al., 1989; Pirolli & Bielaczyc, 1989). Comprehension monitoring in particular has the advantage of focusing the student's attention on behaviors and outcomes that appear to be inconsistent with one's knowledge-based expectations. More recently, it has been shown that these self-explanation processes may be taught to experimental subjects (Bielaczyc, Pirolli & Brown, in press; Chi et al., 1993; Nathan, Mertz, and Ryan, 1994). Nathan and colleagues showed that while successful, training advantages seem limited to declarative knowledge acquisition. Bielaczyc and colleagues (in press) have argued that teaching self-regulation strategies is a necessary component in the successful teaching of self-explanation strategies. Lin (1993) has demonstrated that users of a computer-based learning environment can be alerted on-line to the need for comprehension monitoring and reflection on their cognitive activities, and that this can lead directly to enhanced learning and transfer.

Group discussion methods such as Reciprocal teaching (Brown & Palincsar, 1989; Palincsar & Brown, 1984) can also lead to improved comprehension monitoring on the part of the student and foster greater learning. In this approach, students internalize a structured study strategy that identifies knowledge gaps and supports their clarification. They explicitly work on the task of comprehension through collaborative efforts. Students share their knowledge and help the group to achieve a clear sense of what is understood and what is not, and identifies ways in which confusions may be resolved. The incorporation of this approach into larger classroom projects has met with great success in the "Schools For Thought" program (e.g., Lamon et al., in press). The introduction of such comprehension monitoring practices should also be explored within learning environment designs.

A variety of methods, knowledge elicitation from the individual learner, from a group of learners, and from an expert system provide ways for the learner to gain knowledge in the face of solution errors and provide for learning gains. All of these approaches enjoy empirical and theoretical support, and shape the character of the learning process offered by a learning environment. As we organize the findings put forth by researchers in the cognitive and learning sciences, we may find that certain domains and learner populations have an affinity to certain learning approaches, and that these may be manifest in particularly effective learning environments. The path from domain knowledge to learner knowledge is a complex one, and we must ultimately examine all of the facets of learning and instruction as we develop the science of learning environment design.

Acknowledgments: The author wishes to express the warmest appreciation for the many insights that emerged from discussions with Clayton Lewis, Baruch Schwartz, Kurt VanLehn, Thad Crews, and Lauren Resnick regarding this research. The work reported herein was performed while the author was funded by a post-doctoral fellowship the James S. McDonnell Foundation to the Learning Research and Development Center at the University of Pittsburgh. Both the McDonnell Foundation and LRDC are graciously thanked for providing such a valuable opportunity to complete this work.

NOTES

1. Many tools, such as interpretive programming languages (e.g., BASIC and LISP) and popular video games have this feature.
2. State-based feedback is a form of feedback governed solely by the state of the system. Consequently, the user can expect to receive the same response from the system whenever it is in that particular state. This is contrasted with performance-based feedback which is influenced by some (inferred) state of the user based on the user's prior performance with the system. This distinction grew out of discussions with Thad Crews.

3. The notion that a problem model is a distinct representational form from the situation model is based on the finding that it is possible to independently manipulate solvers' problem models by varying textual cues while still showing agreement in all other respects in their mental representation for the problem (Mertz, 1993).

4. Of course, once one starts to examine the nature of the knowledge embedded in tutoring systems and their manner of use, it becomes readily clear that the term "intelligent tutor" is also quite misleading. The existence of such systems has not, for example, settled the debate as to whether computer systems can actually be intelligent (e.g., Dreyfus & Dreyfus, 1986; Wertheimer, 1985). For most systems it is far more meaningful to characterize the nature of the interactions that emerge between system and user.

5. This is consistent with the concept of *procedural facilitation*, where learning is entrusted to the student, and the role of instruction is to facilitate the learning (Brown & Palincsar, 1989; Collins, Brown & Newman, 1989).

6. Correspondence, either spatial or conceptual, is a key aspect for integrating multiple views of an object, scene, or concept. The process, however, is computationally very demanding (cf. Magee & Nathan, 1987).

REFERENCES

Allwood, C. H., & Montgomery, H. (1981). Knowledge and technique in statistical problem solving. *European Journal of Science Education, 3*, 431-450.

Allwood, C. M. (1982) Use of knowledge and error detection in statistical problem solving. *Goteborg Psychological Reports, 12*(1).

Anderson, J. R. (1982). Acquisition of complex skills. *Psychological Review, 89*, 369-406.

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

Anderson, J. R. (1987). Skill acquisition: Compilation of weak method problem solutions. *Psychological Review, 94*, 192-210.

Anderson, J. R. (1989a). Analysis of student performance with the LISP tutor. In Frederickson, Glaser, Lesgold & Shafto (Eds.), *Diagnostic Modeling of Skill and Knowledge Acquisition*. Hillsdale, NJ: Erlbaum.

Anderson, J. R. (1989b, May). Psychology and intelligent tutoring [Summary]. *Proceedings of the 4th International Conference on AI and Education* (p. 1). Amsterdam, The Netherlands: IOS.

Anderson, J. R. (1993). *Rules of the mind*. Hillsdale, NJ: Erlbaum.

Anderson, J. R., Boyle, C. F., Farrell, R., & Reiser, B. (1984). Cognitive principles in the design of computer tutors. Carnegie-Mellon Technical Report #ONR-84-1.

Anderson, J. R., Boyle, C. F., & Yost, G. (1985). The geometry tutor. In A. Joshi (Ed.), *Proceedings of the Ninth International Joint Conference on Artificial Intelligence* (pp. 1-7). Los Altos, CA: Morgan Kaufmann.

Baranes, R., Perry, M., & Stigler, J. W. (1989). Activation of real-world knowledge in the solution of word problems. *Cognition and Instruction, 6*, 287-318.

Bereiter, C. (1984). How to keep thinking skills from going the way of all frills. *Educational Leadership, 42*, 75-77.

Bielaczyc, K., Pirolli, P., & Brown, A. (1995). Training in self-explanation and self-regulation strategies for learning computer programming. *Cognition and Instruction, 13*, 221-260.

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

Bransford, J. D., Barclay, J. R., & Franks, J. J. (1972). Sentence memory: A constructive versus interpretive approach. *Cognitive Psychology, 3*, 193-209.

Bransford, J. D., & Vye, N. J. (1989). A perspective on cognitive research and its implications for instruction. In L. B. Resnick & L. E. Klopfer (Eds.) *Toward the Thinking Curriculum: Current Cognitive Research*. (pp. 173-205). 1989 Yearbook of the Association for Supervision and Curriculum Development.

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

Brown A. L., Bransford, J. D., Ferrara, R., & Campione, J. (1983). Learning, understanding, and remembering. In J. Flavell & E. Markman (Eds.), *Mussen Handbook of Child Psychology* (Vol. 1). Somerset, N.J.: John Wiley and Sons.

Brown, A. L., & Palincsar, A. S., (1989). Guided, cooperative learning and individual knowledge acquisition. In L. B. Resnick (Ed.), *Knowing, Learning, and Instruction: Essays in Honor of Robert Glaser*. (pp. 393-452). Hillsdale: Erlbaum.

Carpenter, T. P., Corbitt, M. K., Kepner, H. S., Lindquist, M. M., & Reys, R. E. (1980). Solving verbal problems: Results and implications for national assessment. *Arithmetic Teacher, 28*, 8-12.

Carraher, T. N., Carraher, D. W., & Schliemann, A. D. (1987). Written and oral mathematics. *Journal for Research in Mathematics Education, 18*, 83-97.

Chi, M. T. H., Bassok, M., Lewis, M. W., Reiman, P., & Glaser, R.(1989). Self-explanations: How students study and use examples in learning to solve problems. *Cognitive Science, 13*, 145-182.

Chipman, S. (1993). Gazing once more into the silicon chip. In S. P. Lajoie & S. J. Derry (Eds.), *Computers As Cognitive Tools*. (pp. 341-368) Hillsdale, N.J.: Lawrence Erlbaum Associates.

Collins, A. (1985). Teaching reasoning skills. In S. F. Chipman, J. W. Segal & R. Glaser (Eds.), *Thinking and Learning Skills* (Vol. 2, pp. 579-586). Hillsdale, NJ: Erlbaum.

Collins, A. (1988). Different goals of inquiry teaching. *Questioning Exchange, 2*, 39-45.

Collins, A., Brown, J. S., & Newman, S. E. (1989). Cognitive apprenticeship: Teaching the craft of reading, writing, and mathematics. In L. B. Resnick (Ed.), *Knowing, Learning, and Instruction: Essays in Honor of Robert Glaser*. Hillsdale: Erlbaum.

Coury, B. G., & Pietras, C. M. (1989). Alphanumeric and graphic displays for dynamic process monitoring and control. Special Issue: Current Methods in Cognitive Ergonomics. *Ergonomics, 32*(11), 1373-1389.

Cummings, G. (1990, July). Invited talk. IFIP World Conference on Computers in Education (Sydney).

Cummins, D., Kintsch, W., Reusser, K., & Weimer, R. (1988). The role of understanding in solving word problems. *Cognitive Psychology, 20*, 439-462.

De Corte, E., & Vershaffel, L. (1980). The effect of semantic structure on first graders' strategies for solving addition and subtraction word problems. *Journal of Research in Mathematics Education, 18*, 363-381.

Derry, S. J., & Lajoie, S. P. (1993). A middle camp for (un)intelligent instructional computing: In introduction. In S. P. Lajoie & S. J. Derry (Eds.), *Computers As Cognitive Tools*. (pp. 1-14) Hillsdale, N.J.: Lawrence Erlbaum Associates.

diSessa, A. (1985). A principled design for an integrated computational environment. *Human-Computer Interaction, 1*, 1-47.

Dreyfus, H., & Dreyfus, S. (1986). *Mind Over Machine*. New York: Macmillan.

Eisenberg, M., & Fischer, G. (1993). Learning on demand: Why is it necessary and why does it make a difference? In *Proceedings of the Fifteenth Annual Conference of The Cognitive Science Society*. (Boulder, Co.) pp. 180-181. Lawrence Erlbaum Associates.

Ericsson, K. A., & Simon, H. (1984). *Protocol analysis*. Cambridge, MA: MIT Press.

Fletcher, C. R., & Chrysler, S. T. (1990). Surface forms, textbases, and situation models: Recognition memory for three types of textual information. *Discourse Processes. 13*(2), 175-190.

Firsching, J. T. (1986). Conversational implicatures: Mathematics word problems and teacher explanations (Volumes I and II). Unpublished doctoral dissertation, Georgetown University.

Foerster, P. A. (1984). *Algebra I: Expressions, equations, and applications*. Menlo Park, CA: Addison-Wesley.

Fox, B. A. (1993a). Correction in tutoring. In *Proceedings of the Fifteenth Annual Conference of The Cognitive Science Society*. (Boulder, Co.) pp. 121-125. Lawrence Erlbaum Associates.

Fox, B. (1993b). *The human tutorial dialogue project*. Hillsdale, NJ: Erlbaum.

Garner, W. R. (1974). *The processing of information and structure*. Potomac, MD: Erlbaum.

Glaser, R., & Bassok, M. (1989). Learning theory and the study of instruction. *Annual Review of Psychology, 40*, 631-666.

Graesser, A. C. (1993). Dialogue patterns and feedback mechanisms during naturalistic tutoring. In *Proceedings of the Fifteenth Annual Conference of The Cognitive Science Society*. (Boulder, Co.) pp. 126-130. Lawrence Erlbaum Associates.

Greeno, J. G. (1989). Situation models, mental models, and generative knowledge. In D. Klahr & K. Kotovsky (Eds.), *Complex information processing: The impact of Herbert A. Simon*. Hillsdale, NJ: Erlbaum.

Hall, R., Kibler, D., Wenger, E., & Truxaw, C. (1989). Exploring the episodic structure of algebra story problem solving. *Cognition and Instruction, 6*, 223-283.

Harel, I. (1990). *Constructionist Learning*. Cambridge, MA: MIT Laboratory.

Hinsley, D., Hayes, J., & Simon, H. A. (1977). From words to equations. In M. Just & P. Carpenter (Eds.), *Cognitive processes in Comprehension*. Hillsdale, N.J.: Erlbaum.

Kaput, J. J. (1985). Multiplicative word problems and intensive quantities an integrated software response (Technical Report 85-19). Cambridge, MA: Harvard University.

Katz, & Lesgold, (1993). The role of the tutor in computer-based collaborative learning situations.

In S. P. Lajoie & S. J. Derry (Eds.), *Computers As Cognitive Tools.* (pp. 289-318) Hillsdale, N.J.: Lawrence Erlbaum Associates.

Kintsch, W. (1988). The use of knowledge in discourse processing: A construction-integration model. *Psychological Review, 95,* 163-182.

Kintsch, W. (1991). A theory of discourse comprehension: Implications for a tutor for word algebra problems. In M. Carretero, M. Pope, R-J. Simons & J. I. Pozo (Eds.), *Learning and Instruction: European Research in an International Context* (Vol. 3). New York: Pergamon.

Kintsch, W., Britton, B. K., Fletcher, C. R., Kintsch, E., Mannes, S. M., & Nathan, M. J. (1993). A comprehension-based approach to learning and understanding. *The Psychology of Learning and Motivation, 30,* 165-214.

Kintsch, W., & Greeno, J. G. (1985). Understanding and solving word arithmetic problems. *Psychological Review, 92,* 109-129.

Koedinger, K. R., & Anderson, J. R. (1990). Abstract planning and perceptual chunks: Elements of expertise in geometry. *Cognitive Science, 14,* 511-550.

Koedinger K., & Anderson, J. R. (1993). Reifying implicit planning in geometry: Guidelines for model-based intelligent tutoring system design. In S. P. Lajoie & S. J. Derry (Eds.), *Computers As Cognitive Tools.* (pp. 15-46) Hillsdale, N.J.: Lawrence Erlbaum Associates.

Koedinger, K. R., & Tabachneck, H. J. M. (1994). *Two strategies are better than one: multiple strategy use in word problem solving.* Annual meeting of the American Educational Research Association, New Orleans, LA.

Koegel, J., Lakshmipathy, N., & Schlesinger, J. D. (1989). A tutoring system with incomplete domain expertise. In D. Bierman, J. Breuker & J. Sandberg (Eds.), *AI and Education: Synthesis and Reflection* (pp. 123-128). Amsterdam: IOS.

Koschmann, T. D. (1994). The dialectics of instructional technology. *Educational Researcher, 23*(7), 38-40.

Lamon, M., Secules, T., Petrosino, A. J., Hackett, R., Bransford, J. D., & Goldman, S. R. (in press). Schools for thought: Overview of the project and lessons learned from one of the sites. In L. Schauble & R. Glaser (Eds.). *Contributions of Instructional Innovation to Understanding Learning.* Hillsdale, NJ: Lawrence Erlbaum.

Larkin, J. (1983). The role of problem representation in physics. In D. Gentner & A. L. Stevens (Eds.), *Mental Models.* Hillsdale, N.J.: Lawrence Erlbaum Associates.

Larkin, J. (1985). Understanding, problem representations, and skill in physics. In S. F. Chipman & R. Glaser (Eds.), *Thinking and Learning Skills* (Vol. 2: Research and Open Questions). Hillsdale, N.J.: Erlbaum.

LeBlanc, M. (1993). From natural language to mathematical representations: A model of "Mathematical Reasoning." In H. S. Nwana (Ed.), *Mathematical Intelligent Learning Environments.* (pp. 126-144). Intellect Books: Oxford.

Lee, A. (1989). Timing of feedback in tutoring systems. Unpublished Masters Thesis. University of Colorado.

Lepper, M., Woolverton, M., Mumme, D., & Gurtner, J. (1993). Motivational techniques of expert human tutors: Lessons for the design of computer-based tutors. In S. P. Lajoie & S. J. Derry (Eds.), *Computers As Cognitive Tools.* (pp. 341-368) Hillsdale, N.J.: Lawrence Erlbaum Associates.

Lesgold, A., Ivill-Friel, J., & Bonar, J. (1989). Toward intelligent systems for testing. In L. B. Resnick (Ed.), *Knowing, Learning, and Instruction: Essays in Honor of Robert Glaser.* (pp. 337-360). Hillsdale: Erlbaum.

Lewis, A. B. (1989). Training students to represent arithmetic word problems. *Journal of Educational Psychology, 81,* 521-531.

Lewis, A. B., & Mayer, R. E. (1987). Students' misconceptions of relational statements in arithmetic word problems. *Journal of Educational Psychology, 79,* 363-371.

Lewis, A., & Nathan, M. J. (1991). A framework for improving students' comprehension of word arithmetic and word algebra problems. In *Proceedings of the International Conference on the Learning Sciences.* (Evanston, Ill.) (pp. 305-313). The Association of for the Advancement of Computing in Education.

Lewis, C. H. (1981). Skill in algebra. In J. R. Anderson (Ed.), *Cognitive Skills and Their Acquisition.* Hillsdale, NJ: Erlbaum.

Lewis, C. H. (1986) Why and how to learn why: Analysis-based generalization of procedures. *Cognitive Science, 12,* 211-256.

Lewis, M., & Anderson, J. R. (1985). Discrimination of operator schemata in problem solving: Learning from examples. *Cognitive Psychology, 17,* 26-65.

Lin, X. D. (1993). Far transfer problem solving in a nonlinear computer environment: The role of selfregulated learning processes. Unpublished doctoral dissertation, Purdue University.

Magee, M., & Nathan, M. (1987). A viewpoint independent modeling approach to object recognition. *IEEE Journal of Robotics and Automation, 3.*

Mertz, K. (1993). Effects of explicit relations on representing and solving proportional reasoning problems. Unpublished Masters' Thesis, University of Pittsburgh.

Miller, M. L. (1982). A structured planning and debugging environment for elementary programming. In D. Sleeman & J. S. Brown (Eds.), *Intelligent Tutoring Systems.* New York: Academic Press.

Nathan, M. J. (1990). Empowering the student. *Proceedings of Computer-Human Interaction* (pp. 407–414). Association of Computing Machinery.

Nathan, M. J. (1991). A simple learning environment improves mathematical reasoning. *Intelligent Tutoring Media, 2*(3&4), 101–111.

Nathan, M. J. (1992). Interactive depiction of mathematical constraints can increase students word problem solving. In W. Geeslin & K. Grahm (Eds.), *Proceedings of the Sixteenth Psychology of Mathematics Education (PME) Conference,* Vol. 2 (pp. 160–169). Program Committee of the 16th PME: USA. (August, 1992, Durham, NH)

Nathan, M. J., Johl, P., Kintsch, W., & Lewis, C. (1989). An unintelligent tutoring system for solving word algebra problems. In D. Bierman, J. Breuker, & J. Sandberg (Eds.), *AI and Education: Synthesis and Reflection* (pp. 169–176). Amsterdam: IOS.

Nathan, M. J., Kintsch, W., & Young, E. (1992). A theory of algebra word problem comprehension and its implications for the design of computer learning environments. *Cognition and Instruction, 9*(4), 329–389.

Nathan, M. J., Mertz, K., & Ryan, R. (1994). Learning through self-explanation of mathematics examples: effects of cognitive load. Paper presented at the Annual Meeting of the American Educational Research Association (New Orleans).

Nathan, M. J., & Resnick, L. B. (1993). Inference-making during word problem solving. Presentation to the European Association For Research on Learning and Instruction (EARLI), Aug., 1993 (Aix-en-Provence, France). Newell, A., & Simon, H. A. (1972). *Human problem solving.* Englewood Cliffs, NJ: Prentice-Hall.

Ohlsson, S. (1986). Some principles of intelligent tutoring. Special Issue: Artificial intelligence and education. *Instructional Science, 14*(3-4), 293-326.

Ohlsson, S., & Rees, E. (1991). The function of conceptual understanding in the learning of arithmetic procedures. *Cognition & Instruction 8*(2), 103–179.

Paige, J. M., & Simon, H. A. (1966). Cognitive processes in solving algebra word problems. In B. Kleinmuntz (Ed.), *Problem solving.* (pp. 51–118). New York: Wiley.

Palincsar, A. S., & Brown, A. L. (1984). Reciprocal teaching of comprehension-fostering and monitoring activities. *Cognition and Instruction, 1,* 117–175.

Papert, S. (1980). *Mindstorms: Children, Computers, and Powerful Ideas.* New York: Basic Books.

Pennington, N. (1987). Comprehension strategies in programming. In G. Olson, S. Sheppard & E. Soloway (Eds.), *Empirical studies of programmers: Second workshop.* Norwood, NJ: Ablex.

Phye, G. D. (1992). Strategic transfer: A tool for academic problem solving. *Educational Psychology Review, 4*(4), 393–421.

Pirolli, P. (1991). Effects of examples and their explanations in a lesson on recursion: A production system analysis. *Cognition and Instruction, 8,* 207–259.

Pirolli, P., & Bielaczyc, K. (1989). Empirical analyses of self-explanation and transfer in learning to program. In G. Olson & E.E. Smith (Eds.) *Proceedings of The Eleventh Annual Conference of the Cognitive Science Society,* 450–457.

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

Pressley, M., & Levine, J. R. (1983). *Cognitive strategy research: educational applications.* New York: Springer-Verlag.

Putnam, R. T. (1987). Structuring and adjusting content for students: A live and simulated tutoring for addition. *American Educational Research Journal, 24,* 13–48.

Randolph, C. H., & Evertson, C. M. (1992). Enhancing problem solving in preservice teachers' approaches to classroom management using video technology. Paper presented at the Annual Meeting of the American Educational Research Association (San Francisco).

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

Reusser, K. (1985). From text to situation to equation. Institute of Cognitive Science Technical Report. University of Colorado, Boulder.

Reusser, K. (1990). From text to situation to equation: Cognitive simulation of understanding and solving mathematical word problems. In H. Mandl, E. De Corte, N. Bennet & H. F. Friedrich (Eds.), *Learning and Instruction in an International Context* (Vol. II). New York: Pergamon Press.

Reusser, K. (1993). Tutoring systems and pedagogical theory: Representational tools for understanding, planning, and reflection in problem solving. In S. P. Lajoie & S. J. Derry (Eds.), *Computers As Cognitive Tools.* (pp. 143–178) Hillsdale, N.J.: Lawrence Erlbaum Associates.

Rogoff, B., & Lave, J. (1984). *Everyday Cognition: Its Development in Social Context.* Cambridge, MA: Harvard University Press.

Roschelle, J. (1987). Mental models, qualitative reasoning, and computer simulations. *Third International Conference on Artificial Intelligence and Education* (Pittsburgh, PA). p. 46.

Samuel, A. L. (1963). Some studies in machine learning using the game of checkers. In E. A. Feigenbaum & J. Feldman (Eds.), *Computers and thought.* New York: McGraw Hill.

Scardamalia, M., Bereiter, C., & Steinbach, R. (1984). Teachability of reflective processes in written composition. *Cognitive Science, 8,* 173–190.

Schegloff, E., Jefferson, G., & Sacks, H. (1977). The preference for self-correction in the organization of repair in conversation. *Language, 53,* 361–382.

Schmalhofer, F., & Glavanov, D. (1986). Three components of understanding a programmer's manual: Verbatim, propositional and situational representations. *Journal of Memory and Language, 25,* 279–294.

Schmalhofer, F., Kuehn, O., Messamer, P., & Charron, R. (1989). An experimental evaluation of different amounts of receptive and exploratory learning in a tutoring system. *Proceedings of the 19th Annual Meeting of the Society for Computers in Psychology.* (Atlanta, GA).

Schofield, J. W., Evans-Rhodes, D., & Huber, B. R. (1990). Artificial Intelligence in the classroom: The impact of a computer-based tutor on teachers and students. *Social Sciences Computing Review, 8,* 24–41.

Self, J. (1988). Bypassing the intractable problem of student modeling. *ITS–8 Proceedings.* Montreal: University of Montreal.

Shalin, V., & Bee, N. (1985). *Analysis of the semantic structure of a domain of word problems.* (Tech. Rep. No. APS-20) Pittsburgh: University of Pittsburgh, Learning Research and Development Center.

Sherwood, R. D., Kinzer, C. K., Hasselbring, T. S., & Bransford, J. D. (1987). Macro-contexts for learning: Initial findings and issues. *Applied Cognitive Psychology. 1*(2), 93–108.

Shute, V., Woltz, D. J., & Regian, J. W. (1989). An investigation of learner differences in an ITS environment: There's no such thing as a free lunch. *Proceedings of the 4th International Conference on AI and Education* (p. 260–266). Amsterdam, The Netherlands: IOS

Silver, E. A. (1988). Solving story problems using division and remainders: The importance of semantic processing and referential mapping. In M. Behr, C. B. Lacampagne & M. M. Wheeler (Eds.), *Proceedings of the Tenth Annual Meeting of PME-NA.* (pp. 127–133). DeKalb, IL: Author.

Singley, M. K., Anderson, J. R., Gevins, J. S., & Hoffman, D. (1989). The algebra word problem tutor. In D. Bierman, J. Breuker & J. Sandberg (Eds.), *Proceedings of the 4th International Conference on AI and Education* (pp. 267-275). Amsterdam, The Netherlands: IOS.

Tabachneck, H. T., Koedinger, K., & Nathan, M. J. (1994). Toward a theoretical account of strategy use and sense-making in mathematics problem solving. *Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society.* August, 1994 (Atlanta, Georgia, USA).

van Dijk, T. A., & Kintsch, W. (1983). *Strategies of discourse comprehension.* New York: Academic Press.

VanLehn, K., & Jones, R. (1993). What mediates the self-explanation effect? Knowledge gaps, schemas, or analogies? *Proceedings of the Fifteenth Annual Conference of the Cognitive Science Society.* Hillsdale, NJ: Erlbaum.

Waterman, D. A. (1986). *A Guide To Expert Systems.* Reading, MA: Addison-Wesley.

Weaver, III, C.A., & Kintsch, W. (1987). Reconstruction in the recall of prose. *Text, 7,* 165–180.

Weaver, III C. A., & Kintsch, W. (1992). The conceptual structure of word algebra problems. *Journal of Educational Psychology, 84*(4). 419–428.

Wertheimer, M. (1985). A Gestalt perspective on computer simulations of cognitive processes. *Computers in Human Behavior, 1,* 19–33.

Yazdani, M. (1989, May). Second Generation ICAI Systems [Summary]. *Proceedings of the 4th International Conference on AI and Education* (p. 338–339). Amsterdam, The Netherlands: IOS