A SIMPLE LEARNING ENVIRONMENT IMPROVES MATHEMATICAL REASONING

Mitchell J. Nathan
Learning Research and Development Center
University of Pittsburgh

To appear in Nwana, H.S. (Ed.), *Mathematical Intelligent Learning Environments*. Intellect Books: Oxford, and *Intelligent Tutoring Media.*

A SIMPLE LEARNING ENVIRONMENT IMPROVES MATHEMATICAL REASONING

Mitchell J. Nathan
Learning Research and Development Center
University of Pittsburgh

ABSTRACT

ANIMATE, an interactive, computer animation-based tutor, has been developed as part of an ongoing test of a theory of word problem comprehension. Tutor feedback is unobtrusive and interpretive: Unexpected behavior in the equation-driven animated situation highlights equation errors which the student resolves through iterative debugging. The responsibility for learning, goal-setting, and diagnosis is placed on the student. Experimental controls (n=96) with Motion problems show improvement cannot be solely attributed to practice, computer use, or use of the situation-based method. Concurrent think aloud protocols of students (n=7) solving Motion, Work, and Investment problems over two days (in a pretest-posttest design) uncover specific changes that underlie these improvements. ANIMATE is an effective problem-solving aid, and there is transfer of learning. Problems with impossible situations were acknowledged by median level subjects (posttest scores between 77% and 85%), but solved blindly by high-level subjects (posttest scores >= 95%), suggesting an automaticity-controlled processing dichotomy. On day 2, subjects spent more time reviewing problem texts and correcting flawed expressions. They developed self-directed debugging skills without relying on tutor feedback--behaviors reminiscent of expert problem solving in many domains. The system is unintelligent by ITS

standards but communicates knowledge to the student, helping them teach themselves approaches for mathematics problem solving.

A SIMPLE LEARNING ENVIRONMENT IMPROVES MATHEMATICAL REASONING

Mitchell J. Nathan
Learning Research and Development Center
University of Pittsburgh

In the design of a learning environment, the central issues are to determine what to tutor and how. The *what* of tutoring addresses both selection of the content area and the level at which information in the content area will be presented. The *how* specifies the pedagogical approach -- the manner in which the student will interact with the tutor. For environments which support mathematics learning, the system must, in particular, teach students to use formal representations that model situations which are often presented informally, e.g., in a story format. This chapter presents research on the design of a learning environment, theoretical work on the comprehension of word algebra problems, and empirical results showing changes in students' problem solving performances and methods. Great gains in performance scores and important changes in students' mathematical reasoning are shown to occur after training with a learning environment which provides little in the way of knowledge-based guidance. These results seem dependent upon two factors: the ability of the tutor to ground the mathematics it teaches in a representation that is familiar and meaningful to the student; and placing the student in a learning context where exploration is encouraged but not unbridled.

ANIMATE is a learning environment built to support introductory word algebra problem solving. It facilitates within each student development of solution-enabling mental representations and strategies. This is done by helping students to think about the situations described by word problems and to explicitly address the means by which these situations can be formally modelled (cf. White & Frederiksen, 1990). In this view, the *comprehension* aspects of word algebra problem solving are central: the way problem information is represented; the integration of additional knowledge -- inferences and elaborations -- that make a story problem coherent; and the nature of the formal expressions that describe that knowledge and produce a quantitative solution.

The pedagogical view -- the *how* -- embodied within ANIMATE is in contrast to the typical approach of ITSs, whereby one sets out to impart to the student, through practice, the condition-action rule-like behavior extracted from experts and stored in the system's expert module (Anderson, 1988). ANIMATE's didactic approach is to engage the student and treat learning as a constructive process (Resnick, 1989). In this context the student must exercise goal-setting and strategy-selection abilities, and apply self-monitoring and knowledge organization skills (cf. Scardamalia, Bereiter, McLean, Swallow, & Woodruff, 1989). The environment is *student-centered*. That is, ANIMATE provides the structures which enable students to use their intelligence and knowledge, but it does not provide any intelligence to guide the learning. It is the *student*, not the tutor, that sets the goals and evaluates the progress made toward these goals.

ANIMATE does not try to understand in any deep way the student’s actions, or give intelligent or remedial feedback; in fact it cannot, possessing no domain expert module, no knowledge of the problems being solved, and no student model. To bring about the intended learning, the tutor presents each solution attempt to the student in a way that allows the student to evaluate his or her own performance -- in the form of a simple computer animation. The feedback supplied by the tutor is not evaluative and must be interpreted by the student. Thus, the student is an active participant, in fact the *central* participant, in the interaction, controlling not only the problem-solving process, but solution assessment, error diagnosis and recovery. Such a system is *empowering* to the students who use it since they are given the opportunity to approach each problem in their own style and pace, and to assess their own performance (Nathan, 1990).

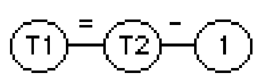
As will be shown, this pedagogical approach also supports the learning of some principles of problem-solving often exhibited by expert problem solvers in a variety of domains. The tutoring approach presented here is a general one. It has been applied to word algebra problem solving because this domain has some special characteristics: the integrated nature of these problems draws on reading and language comprehension skills as well as ones' knowledge of formal mathematics.

Learning Mathematics with ANIMATE

Word problem-solving performance depends upon reading comprehension skills. Recently, an analysis by Nathan (1991; Nathan, Kintsch, & Young, in press) showed this in a dramatic way. Relations that were unstated in the original problem texts yet which were needed for a solution proved to be the largest sources of solution errors. The difficulty arises because these relations have to be inferred by the students. The inference-making process is taxing and translation of these inferences into mathematical expressions is error-prone. Of the four hundred and eight classifiable pretest errors made by the ninety-six students who participated in a recent word algebra problem solving study, 258 errors (about 63%) were labelled as these inference-based errors. The frequency of these errors was significantly greater than that of any other class of errors at the 95% level (Nathan et al., in press). Several other studies have similarly identified language comprehension as a critical component to word problem solving (Carpenter, Corbitt, Kepner, Lindquist, & Reys, 1980 ; Cummins, Kintsch, Reusser, & Weimer, 1988; DeCorte, Verschaffel, & DeWin, 1985; Lewis & Mayer, 1987).

ANIMATE was designed to test a theory of word algebra problem comprehension (Nathan et al., in press). The critical feature of this model is that the mental representations of the semantics of a problem text (the *textbase*) and the underlying problem situation (the *situation model*) must be linked to the student's knowledge of the mathematics in order to guide selection of the proper problem-solving strategies. Solution attempts, specified algebraically, can be evaluated if the mathematics has meaning for the student and is not simply a string of abstract symbols. By grounding the mathematical symbols and expressions in a meaningful depiction of the problem situation, the student can interpret the mathematics situationally. The tutor does this explicitly, using computer animation. Students constructs algebraic expressions and a simple representation of the situation causally-linked to the mathematics. The validity of the mathematics is assessed by "Running" the equation-driven animation. Unexpected behaviors in the animation highlight equation errors which the student resolves through iterative debugging. Assuming the student understands the situation correctly, search for algebraic errors is constrained by the manner in which the animation deviates from the student's situation model.

Figures 1 through 4 show a subject producing a solution for a typical Work problem. Figure 1 contains the problem statement and the initial solution attempt. Several errors are present and they are removed successively. In Figure 1, Tom leaves before Huck which is counter to the problem situation presented in the text. In Protocol 1 Subject 6 hypothesizes that the Time equation shown in bubbles and labelled arcs in the lower right hand corner of Figure 1, is flawed.



Protocol 1

[*Initial formation of the equation*]

And, uh, the time, let me see, what we know about the time is that if Tom arrives one hour late from fishing, so we can ... Let's see, [Huck starts] an hour early, which meant that [Tom] ... Now if Tom arrives one hour late, so it would have to be a negative 1....

[*After running the animation*]

... He [Tom] had an extra hour to [paint], so we'll say that this is a plus 1.

Work Problem

Huck and Tom agree to paint a fence. Tom can paint the entire fence in two hours while it takes Huck four hours. If Tom arrives one hour late from fishing, how long will it take the two boys to complete the job? (Problem text adapted from Hall, 1990).

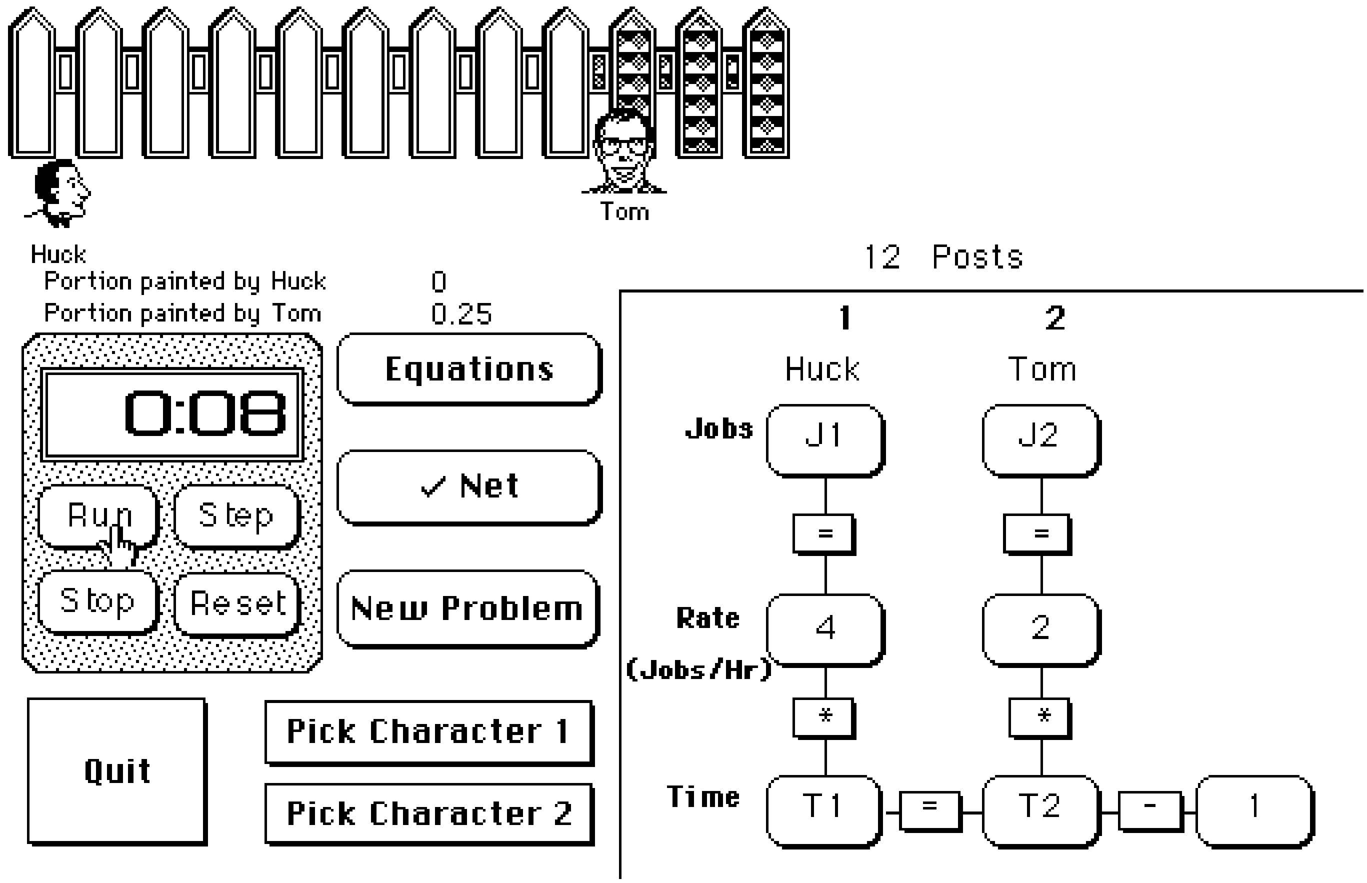


Figure 1. Subject 6's initial solution to the Work problem in the ANIMATE learning environment. Tom starts first while Huck waits one hour.

By changing the minus one to a plus one, the relative starting times of the two characters in the animation is now in accord with the original problem statement, though other errors still exist. As Figure 2 shows, Huck works too quickly. Subject 6 reasons in Protocol 2,

Protocol 2

[*After running the animation:*]

That seems to be moving up fast ... So hours can't, um, and then, okay, so it's four over one so it would be a quarter? Okay. All right. Rate, R1 is a quarter, T is T1, uh .... R [2] is a half ...

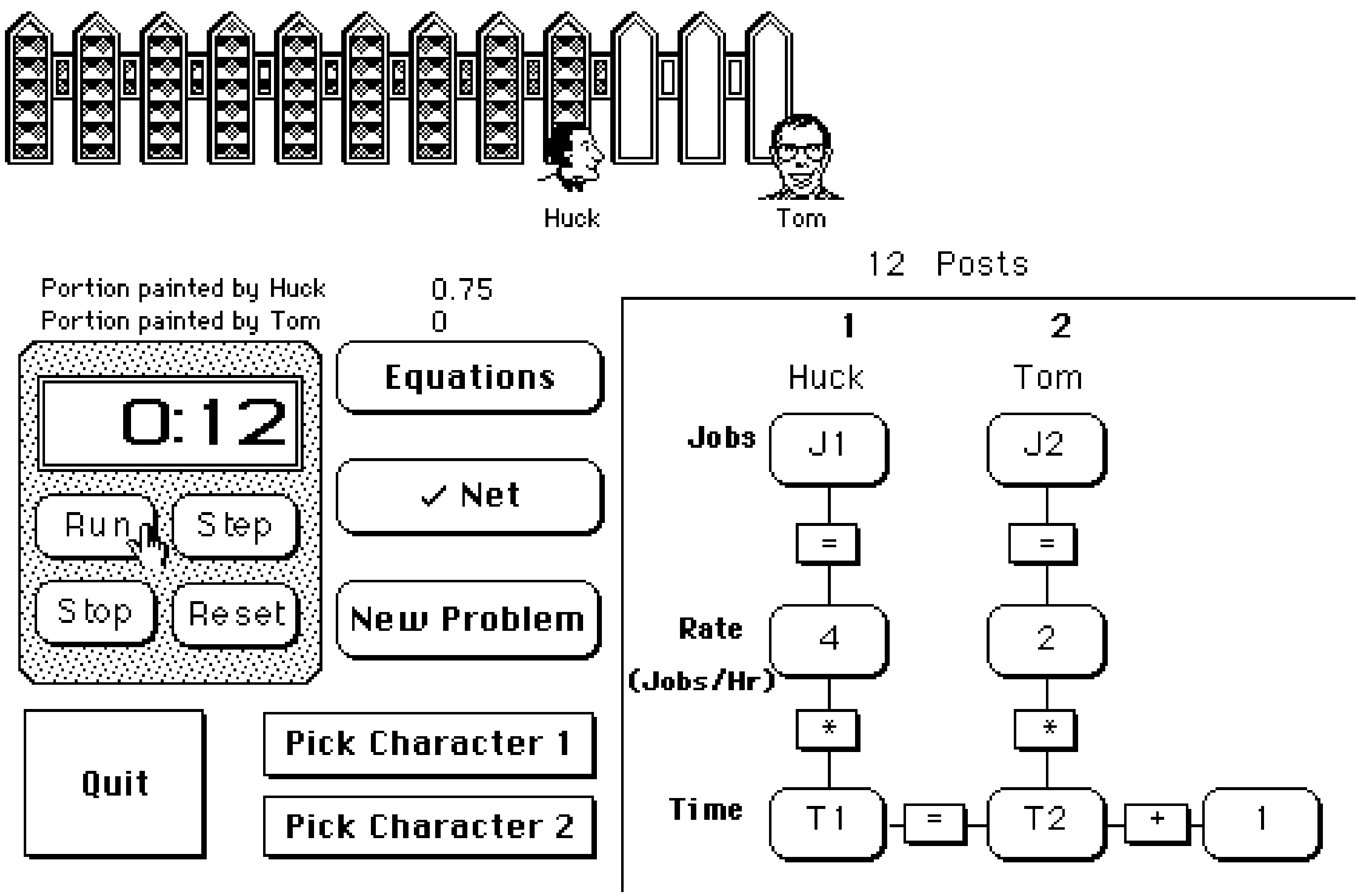


Figure 2. Having corrected the Time equation Subject 6 sees that the Rate values are too high. Huck takes four hours to paint the fence, but is 75% done after twelve minutes.

Figure 3 shows the current state of the solution and its manifestation after correcting (inverting) the work rates of the two characters. The animation proceeds nicely, but it continues indefinitely, leaving the two characters to repaint sections of the fence that have been already painted.

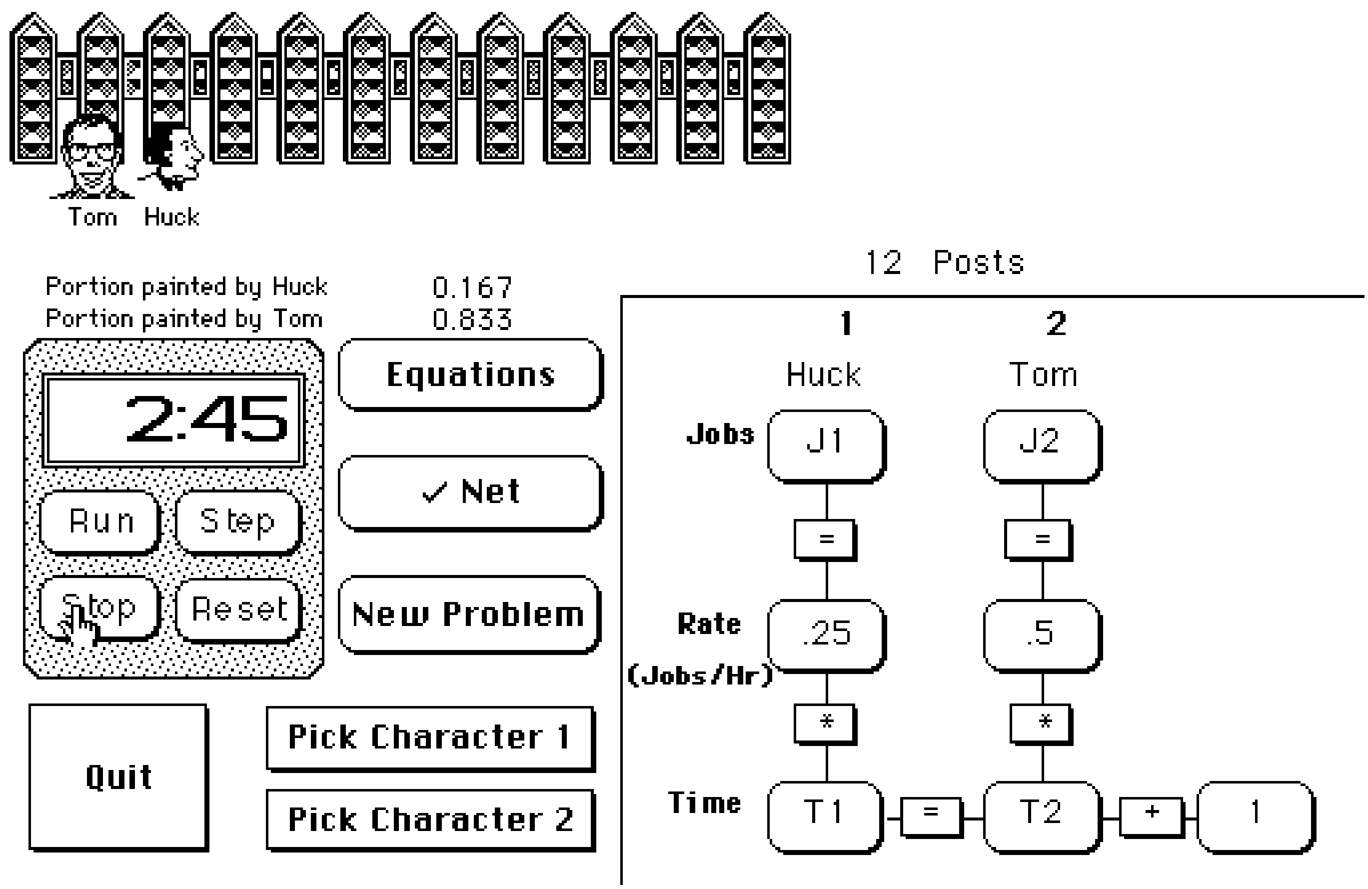


Figure 3. With no equation relating Job variables, J1 and J2 (the amount of work done), the characters continue to paint indefinitely.

This is because an important formal constraint has been omitted from the solution; one which relates the amount of work done by Character 1, **J1**, to the work done by Character 2, **J2**. Subject 6 decides they must be related. He interprets the phrase "complete the job together," as meaning that both characters paint the same fence. Analytically, Subject 6 represents "complete the job together," by equating the amount of work performed by each character, **J1** and **J2**. Subject 6 enters,

$$J1 = J2$$

The tutor reflects this algebraic constraint by stopping the animation when the amount of work performed by the two characters is equal. This event is shown in Figure 4.

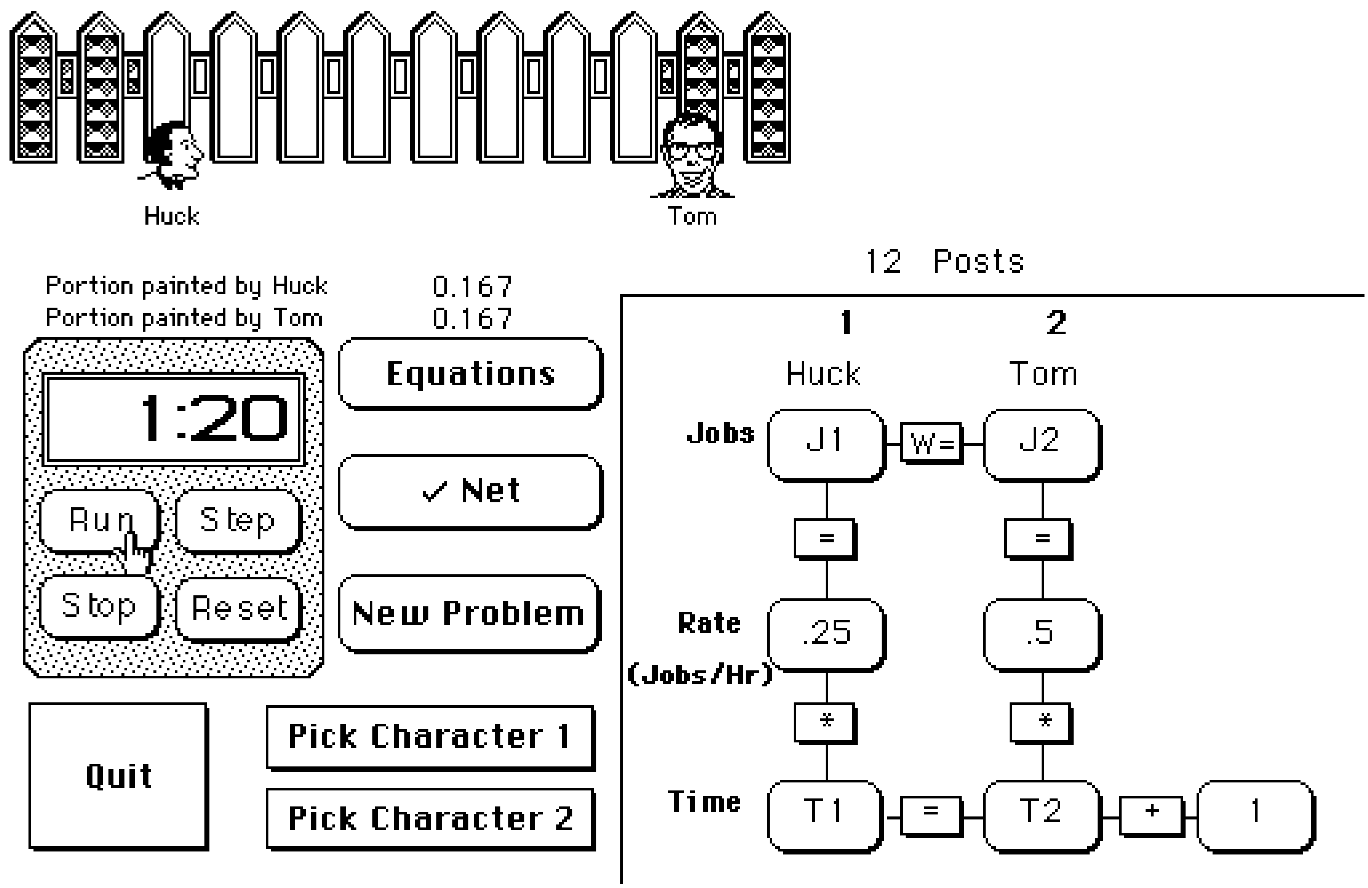


Figure 4. Given the constraint J1=J2, the animation stops when the amount of work performed by the two characters is equal. The proper constraint (see text in Figure 1) is J1+J2=1.

So far this interaction has pointed out how students resolve flaws in their formal specifications of the solution. But, interactions with ANIMATE can also help students identify problems in their analysis of a situation. Concluding music indicates when the animation is done running, but it is apparent in Figure 4 that the fence is not yet completed.The current solution specifies a situation different than the one stated. To Subject 6, the misspecification is subtle: If both characters do the same job, the two should be equated. The meters showing the "proportion of work done" for each character help in diagnosing the error since the two numbers do not add up to one complete fence. When this is finally noticed, it becomes clear that **J1** and **J2** must sum to one, a single fence.

From these examples one can see how this technologically-simple learning environment helps students to construct expressions, and identify and rectify errors in their solutions for word problems. The situation-based feedback helps students to see the implications of the equations. The students are then encouraged to reason about how the mathematics can model the intended situation. That subjects can do this form of problem solving has been shown. Next the learning that occurs from this form of instruction is addressed.

<u>Early Empirical Results with ANIMATE</u>. Engaging in this form of reasoning has some important, lasting benefits for students learning to solve word algebra problems. A brief review of the results of a pretest-posttest, control group design show some of the strengths and limitations of this type of learning environment (Nathan, 1991; Nathan et al., in press).

Ninety-six subjects were randomly assigned to one of five treatments: the Animation group (with 24 subjects), Stopping-Condition group (with 14 subjects), Network group (with 14 subjects), Situation-Only group (with 13 subjects), and the Equation or control group (with 31 subjects). Those in the Network, Stopping-Condition, Situation-Only, and Animation groups used different learning environments during the training which ran on Apple™ Macintosh computers, distributed one to each subject.

The several learning environments varied in functionality. ANIMATE 1.0 was used by the Animation treatment group. It is similar to the tutor shown in Figures 1 through 4 above, but it is limited to Motion problems solved by the equations Distance = Rate X Time (Figure 5).

Motion Problem

A train travels west at fifty-five miles per hour. Another train leaves on a parallel course one hour later and travels west at eighty miles per hour. How far will the second train travel when it overtakes the first train?

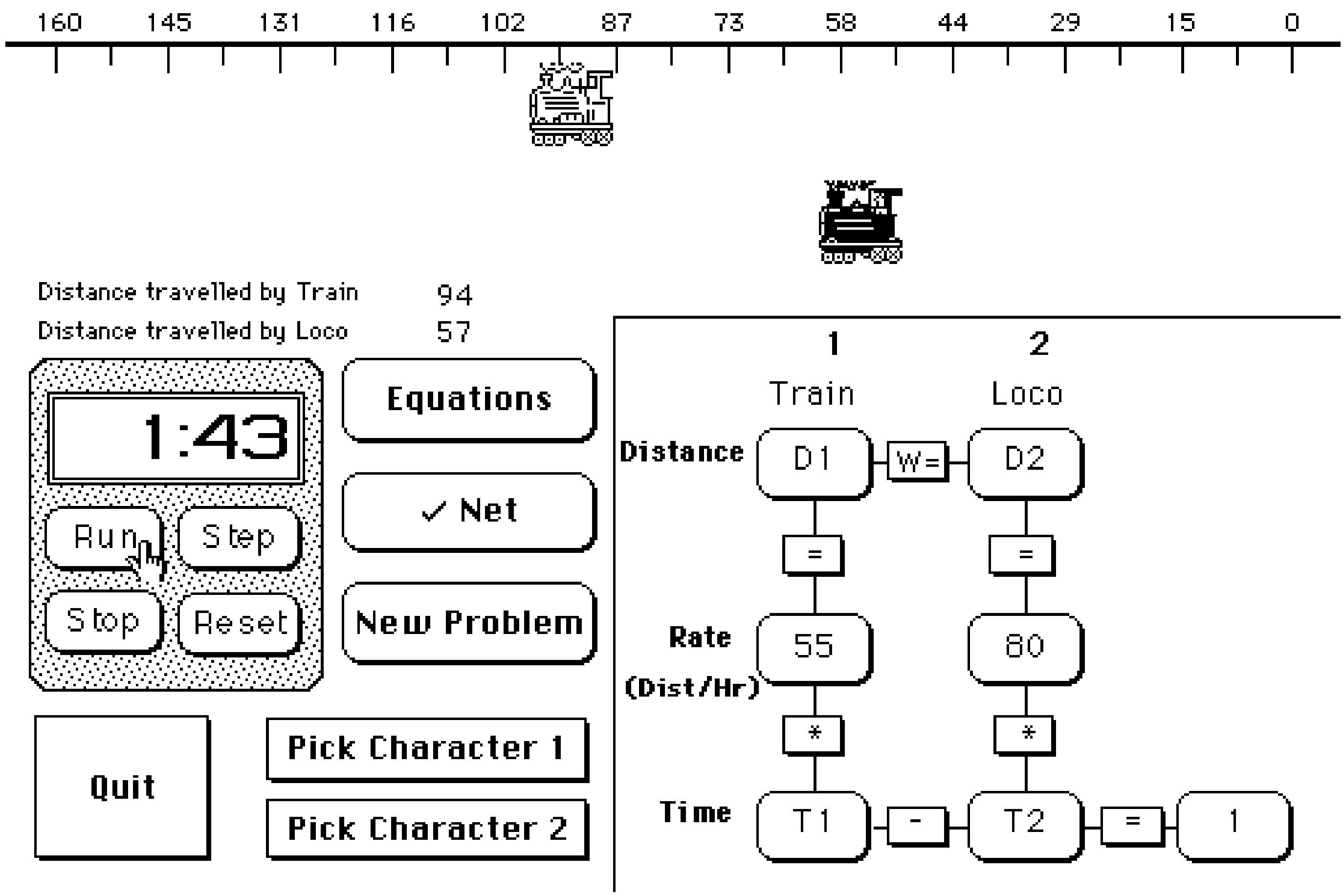


Figure 5. Motion problem text and a screen dump from ANIMATE (in progress).

The Stopping-Condition group used a tutor closest in functionality to ANIMATE 1.0. Subjects set up an equation network and a situation, just as Animation subjects did, but were not able to "run" an animation in order to receive situation-based feedback on their equations. Their only feedback was for the algebraic self-consistency of the network, which was available to all computer users. The Network-Only tutor was identical to the Stopping-Condition tutor except students set up no situation to accompany their equations. The Situation-Only tutor was similar to the ANIMATE 1.0 tutor, but this was not driven by an algebraic formalism. Subjects set up and ran the situation-based animation where the starting and ending times and speeds of the characters were specified as part of each character's profile. Subjects using this tutor were able to direct the same animation behavior as those using the ANIMATE 1.0 system, but they were not encouraged to tie this behavior to mathematical principles. The Equation group used no computer tutor and performed all training tasks using paper and pencil. This group controlled for effects of reviewing algebra and a repeated exposure to the experimental tasks.

The Network-Only and Stopping-Condition tutors are *non-animation tutors*, while ANIMATE 1.0 and the Situation-Only tutor are *animation tutors*. ANIMATE 1.0, the Stopping-Condition tutor, and the Situation-Only tutor are (differing levels of) *situation-based tutors*. The four "tutor" conditions -- Animation, Stopping-Condition, Situation-Only, and Network -- served to tease apart effects due to different aspects of the full tutor. The differences are shown in Table 1.

TABLE 1

Comparative functionality of the tutors used in Nathan (1991)

Tutor	Network formalism	Link situation to network	Set up and run animation
ANIMATE 1.0	•	•	•
Stopping-Condition	•	•	
Network-Only	•		
Situation-Only			•

Error Analyses. As mentioned earlier, most pretest errors were due to subjects' poor translation and inferencing abilities. After training, those subjects who used ANIMATE produced the least number of these inference-based errors in the posttest. Subjects using the other situation-based tutors -- Stopping-Condition and Situation-Only -- were next in line, while frequency of these errors for Network and equation users actually increased.

The ability to make inferences from a text is said to tap into the theoretical construct of a person's *situation model* (e.g., Kintsch, 1986). This knowledge structure represents the "deep structure," or gist of a text, and links text information to the reader's long term knowledge base. By accessing this, subjects activate associated knowledge which helps them elaborate and draw the inferences needed to make the story more coherent and the solution more complete. Using any of the situation-based tutors facilitates these important memory and reasoning processes, but ANIMATE, which explicitly links the mathematics to the situation, helps most of all.

Problem-solving Performances. Test performance and improvement is shown graphically in Figure 6, where Raw score is plotted against time of test for the five experimental conditions. A highly significant main effect of time of test (pretest versus posttest) was found, indicating that, overall, subjects improved from pretest to posttest, $\underline{F}(1,91)=80.25$, $\underline{p}<.0001$, MS=10.97. A significant interaction of time of test with treatment (the between-subjects variable) was found, $\underline{F}(4,91)=5.56$, $\underline{p}<.001$, MS=.76. This indicates that the improvement is reliably different for the five forms of training. A Duncan's Multiple-Range post hoc test (Duncan, 1955) comparing test improvement for each treatment shows that ANIMATE 1.0 users improved significantly more than subjects in all of the other groups (MS=.14, α=5%). No other reliable group differences were found. At posttest time the Animation group emerged with the highest mean performance at the 1% level (MS=.31). Training task performance results were in line with the posttest results.

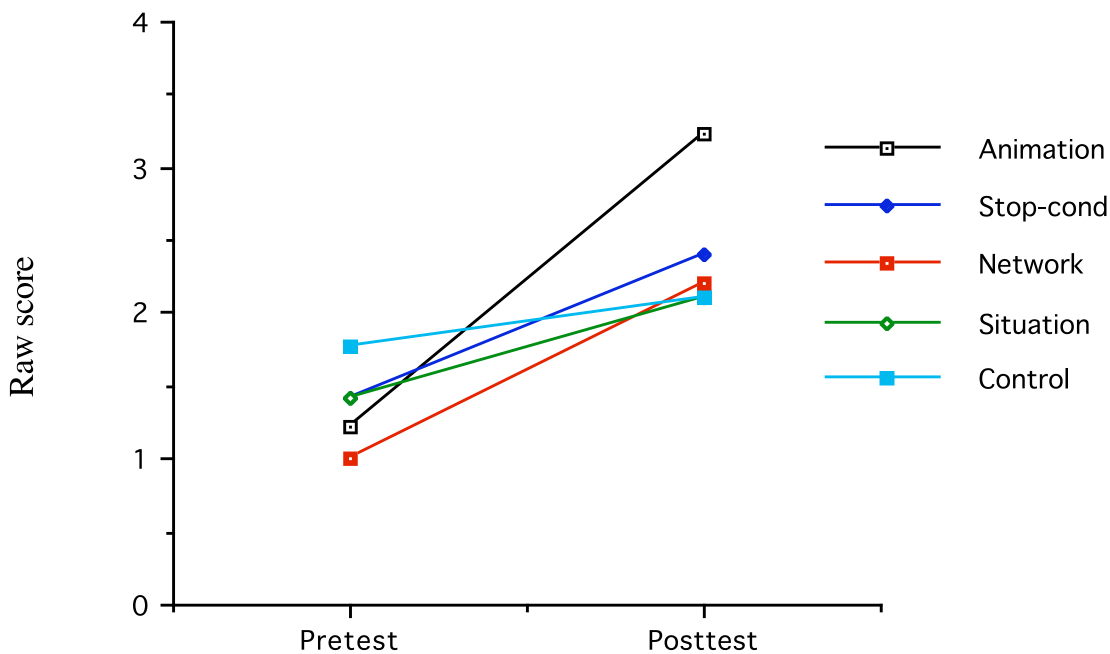


Figure 6. Pretest and posttest scores for all treatments. The Animation group used ANIMATE 1.0. A score of 4 is perfect performance.

The group that experienced training with ANIMATE exhibited a superior overall posttest score, with no advantage at pretest, and superior test improvement. The different learning environments were designed to ensure that each experimental group received incrementally less support in the problem-solving process. These experimental controls show that the improvement of ANIMATE 1.0 users cannot be solely attributed to algebra practice, use of a computer tutor, or use of the situation-based method with no explicit link to the mathematical formalism. It appears that coupling the mathematical expressions to a concrete depiction of the situation is necessary to give ANIMATE 1.0 users a measurable boost over the other subjects.

What ANIMATE Users Actually Learn

Results with ANIMATE 1.0 suggest that providing a situation-based interpretation of algebraic expressions helps students solve problems. This gives a very gross account of the learning that takes place. It does not really concern itself with *how* students actually *use* ANIMATE, or the extent to which animation-based feedback helps students understand the algebraic concepts presented. What is the *form* of the learning and how does it produce these reliable gains in problem-solving performance? From the above experiment it is clear that ANIMATE users construct with the greatest accuracy certain inference-based relations. Do ANIMATE users simply make less errors? Are their errors more benign? To address these questions one needs to identify the aspects of the system that are actually used by students and the specific mechanisms underlying these performance improvements. A process-level description of students' use of ANIMATE during problem-solving may begin to address these questions and provide us with a more complete description of the learning process that takes place from interactions with the tutor. This description is the goal of the following experiment.

The Expanded Functionality of ANIMATE 2.0. The second generation of the ANIMATE learning environment supports word algebra problem solving in multiple scenarios (Figure 7) and includes Motion (Figure 5), Work (Figures 1 through 4) and Investment problems (Figure 8).

Process-level Description of ANIMATE Use. ANIMATE 2.0 was used by seven students to solve Motion, Work, and Investment word algebra problems during a pretest, training tasks and a posttest. This experiment did not employ a control group design. Subjects were run individually in two experimental sessions broken up by a two-day delay. They produced concurrent Think Aloud protocols of their problem-solving processes in a manner consistent with Ericsson & Simon (1984). In addition to the verbal reports, computer protocols were collected by the enhanced tutoring environment for every mouse and keyboard entry, through use of the ProtoTymer™ stack (Miller & Stone, 1989). Process-level descriptions were made by combining these protocols to obtain detailed descriptions of subjects' specific interactions with the ANIMATE 2.0 system.

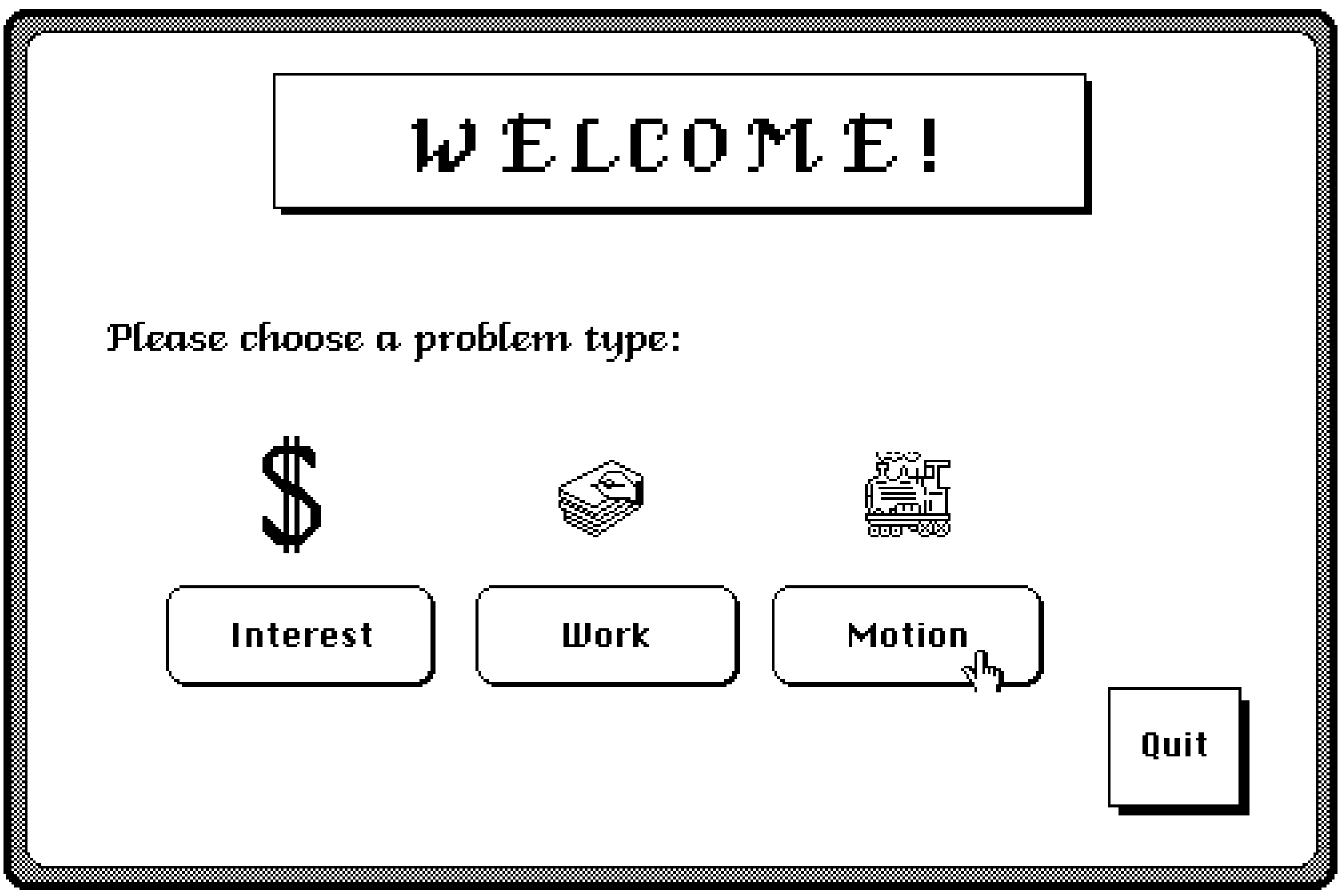


Figure 7. The start-up screen of ANIMATE 2.0.

On Day 1, each subject took a pretest. The tutor was then introduced as part of a general algebra review. Subjects next solved a set of training tasks using the tutor, with minimal intervention from the experimenter. Upon completion, subjects were dismissed. Each subject arrived approximately 48 hours later. With no review, the subject set about solving a new set of training tasks comparable to those presented on Day 1 (the order of the tasks was counterbalanced between subjects). Upon completion of these tasks, the tutor was removed and subjects solved a series of familiar and novel (i.e., transfer) problems as a final posttest.

<u>Group Performance Improvement</u>. The individual subjects all experienced improvement from pretest to posttest, although the magnitude of this differed widely. This improvement is attributable to both the review of algebra and the practice gained while working with the ANIMATE 2.0 environment. These results are shown in Table 2 as error rates (1- the proportion correct). Average error rates for the tests and training tasks are depicted graphically in Figure 9.

TABLE 2

Error rates for ANIMATE 2.0 users

Subject	Pretest	Day 1 Training	Day 2 Training	Posttest (Overall)	Posttest (Familiar)	Posttest (Novel)
1	.13	.08	.08	0	0	0
2	.3	0	0	.05	.08	0
3	.56	.17	0	.4	.5	.25
4	.3	.08	.08	.22	.25	.17
5	.94	.17	.08	.45	.33	.63
6	.75	0	0	.2	.25	.13
7	.5	0	0	.15	.25	0
Average	.55	.07	.04	.21	.24	.17

Investment Problem

Suppose seven hundred and fifty dollars is invested at an interest rate of five percent, to be compounded annually. What amount will be in the account at the end of the second year?

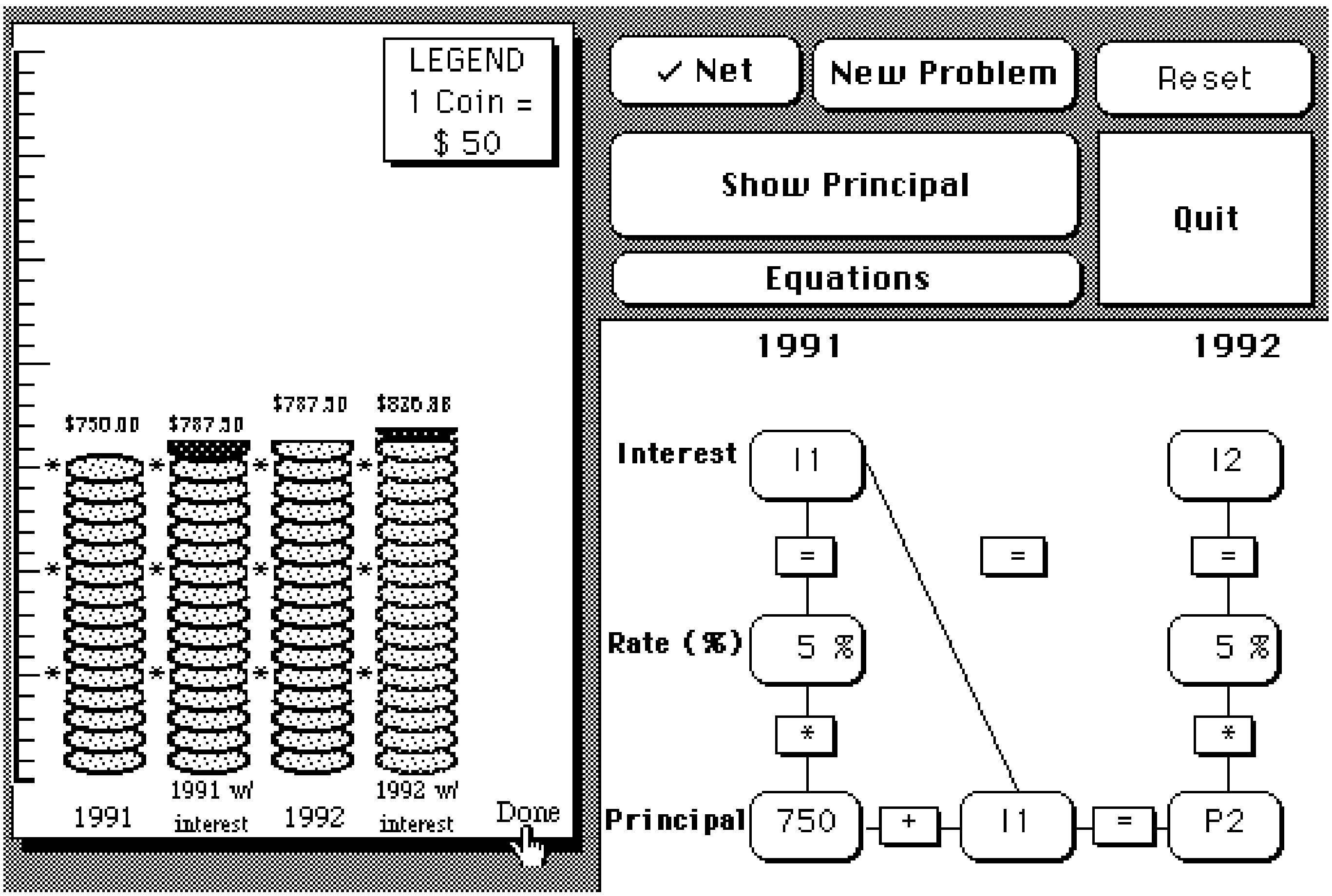


Figure 8. Solving an Investment problem with ANIMATE 2.0.

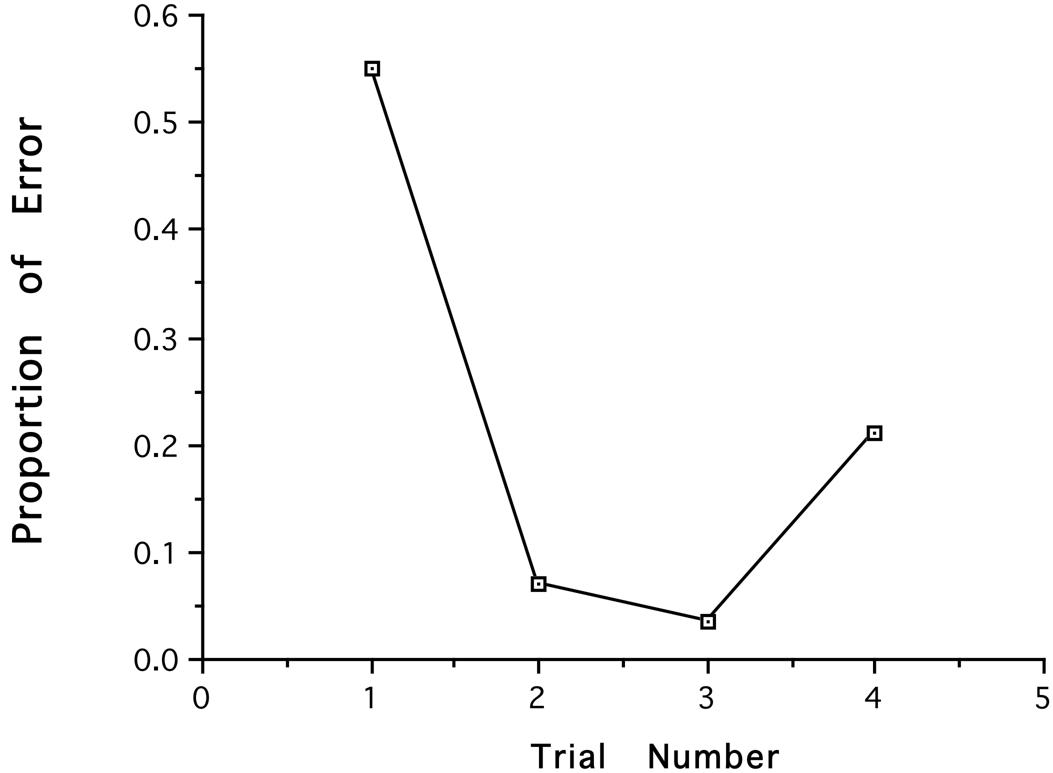


Figure 9. Average proportion of errors for ANIMATE 2.0 users as a function trial. Trial 1 represents mean pretest performance, Trial 2 is the training performance on Day 1, Trial 3 is training on Day 2, and Trial 4 is the mean performance for all posttest problems.

Several things are apparent from these data. There is a dramatic decline in the proportion of errors made by subjects once they use ANIMATE. This improvement was maintained across the two-day delay (from Trial 2 to Trial 3). And there is a rise in the error rate when the learning environment is removed. Even with this rise, group performance improved significantly from 45% correct at pretest to 79% correct (a 21% error rate) at posttest, $\underline{t}(6)=5.36$, $\underline{p}<.002$, MS=.5.

ANIMATE as a problem-solving aid. The posttest not only robbed students of their problem-solving aid, it also included two novel problems, problems which had algebraic concepts which were not reviewed or practiced. Thus, a possible cause for the rise of errors at posttest time is that the test was simply more difficult than the training tasks. The last two columns of Table 2 show the posttest data reorganized. Familiar posttest problems -- Work, Motion, and Investment problems -- which received review and practice during the experiment, were as difficult as novel problems (the differences are not significant). Thus, it is the removal of the learning environment that leads to the slight rise in error rate at posttest time.

Impossible problems: Evidence of different approaches to problem-solving. Subjects also addressed two problems describing physically impossible situations which could nonetheless be modelled algebraically. These problems helped to further reveal the underlying processes which drove students' reasoning and problem solving. The first was the Mixture problem shown below.

> **Mixture Problem**
>
> A grocer mixes peanuts priced at one dollar and sixty-five cents per pound with almonds priced at two dollars and ten cents per pound. She wants thirty pounds of the mixture to be worth one dollar and fifty-three cents per pound. How many pounds of each must the grocer include in the mixture?

The second was a Coin problem taken from Paige and Simon (1966).

> **Coin Problem**
>
> The number of quarters Ruth has is seven times the number of dimes she has. The value of the dimes exceeds the value of the quarters by two dollars and fifty cents. How many has she of each coin?

Table 3 shows how subjects performed when they either solved an impossible problem as though it were an actual problem, or identified it as describing a physically unrealizable situation. A percentage score in Rows 1 and 2 indicates the score a subject received for his or her solution attempt at solving, respectively, the Mixture and Coinage problems. Presence of the **imp** symbol, indicates that the problem was identified as "impossible" by a subject . Mean performance on these problems for each subject is provided (Row 3) along with the count of the number of times impossible problems were detected (Row 4). Performance on the impossible problems is compared with each subject's posttest performance (Row 5).

TABLE 3

Mean performance on "impossible problems" and overall on posttest performance

Subject No.	1	2	3	4	5	6	7
Mixture performance	100%	100%	100%	**imp***	50%	**imp***	**imp***
Coinage performance	100%	100%	50%	67%	25%	75%	**imp***
Mean equation score	100%	100%	75%	67%	38%	75%	N/A
"Impossible" count	0	0	0	1	0	1	2
Posttest Performance	100%	95%	65%	78%	50%	77%	85%

* The **imp** symbol is used to reflect the fact that a subject labelled a problem as describing an impossible situation.

Examination of the verbal protocols of subjects shows that those students who did not detect the contradictory nature of the Mixture and Coinage problems used what Paige and Simon (1966) have termed a "direct" approach to problem solving. They did not reflect on "the bigger picture" of the situation as described. Rather, they delved right into the problem-solving process, set up the network and equations on paper and cranked out a solution. There were two occasions where subjects (Subjects 3 and 5) detected and corrected situational inconsistencies within their equations; and many subjects made appropriate inferences and properly formulated them as algebraic expressions. Thus, there was much reasoning with respect to the problem situation. But, for these subjects, this reasoning did not uncover the illegitimacy of the problem situation.

Of those subjects who did recognize the impossible nature of the Mixture problem, two subjects (Subjects 4 and 6) used approaches which could be called highly situation- or auxiliary-based. Subject 4 carefully considered the values presented and did not bother to set up any formal relations. Subject 6 set up the network formalism only after reflecting on the problem situation and determining that it was unrealistic. Only Subject 7 combined an equation-based and a situation-based approach as well as considering the specific values employed; and only this subject labelled both the Mixture and Coin problems as impossible.

The detection of irregularities in these problems was made by subjects who performed at the median level (Figure 10). These students also improved most in their test scores (Figure 11).

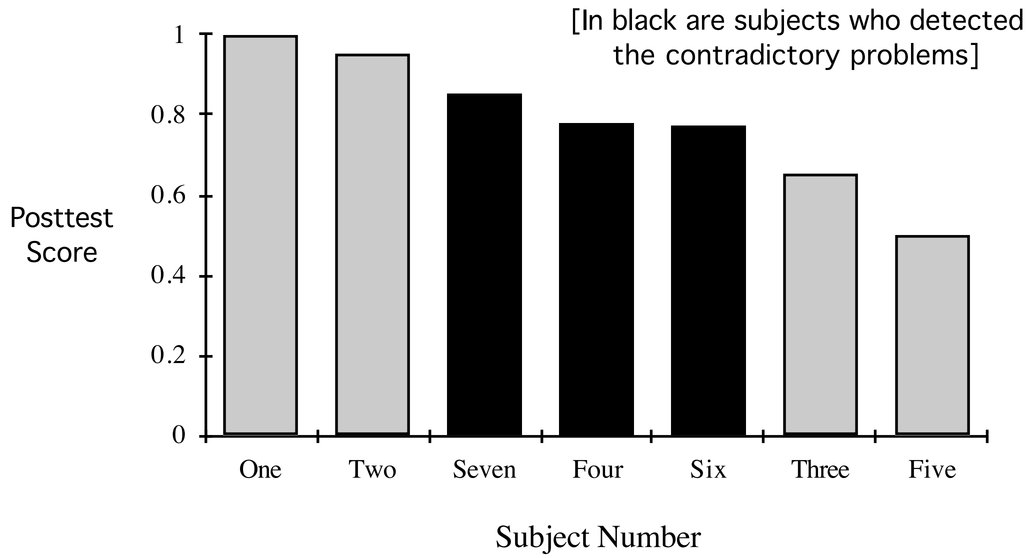


Figure 10. By ordering the subjects by posttest score it is apparent that median ranking subjects were the ones who detected the "impossible problems."

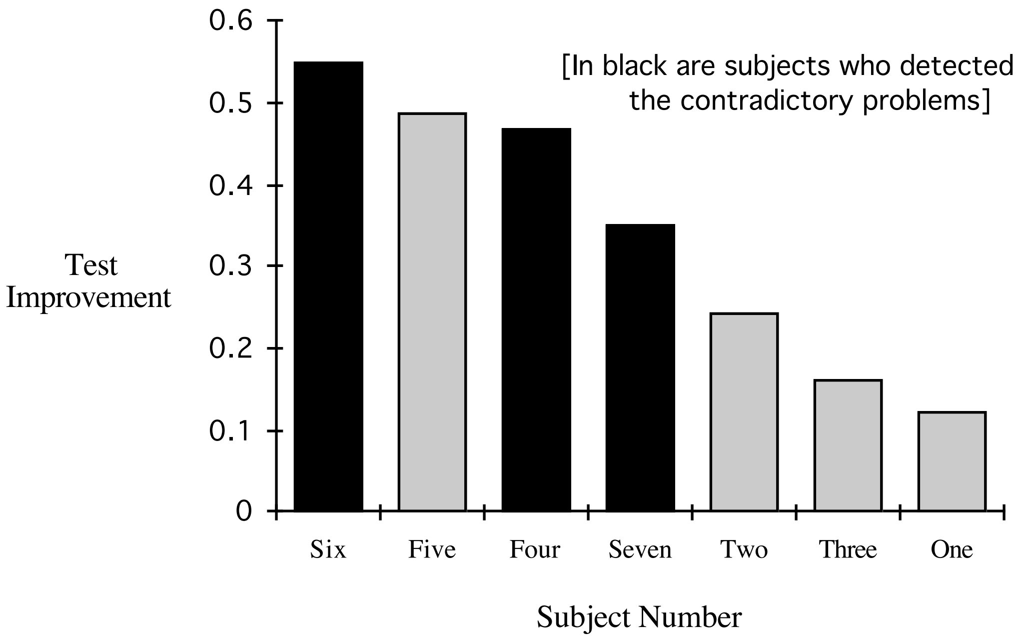


Figure 11. By ordering the subjects by test score improvement it is apparent that those subjects who improved most were most likely to detect the "impossible problems."

One interpretation of these findings is that subjects highly adept at problem solving -- those with posttest scores at or above 95% correct performance (i.e., **A** students, Subjects 1 and 2) -- were so confident of their solution-generating abilities that they saw no need and perhaps had no opportunity to step back and reflect on the problem at hand. Their problem-solving skills may be so honed and automatic that inspecting or interrupting them would require major effort (Ericsson & Simon, 1984; also see Shiffrin and Schneider, 1977). Consequently, these "top" students simply executed their well-practiced solutions. (A review of Table 2 reveals that the two subjects in the highest posttest group started out that way at pretest.) The poorest students -- those with posttest scores at or below the 65% level (**D** students, Subjects 3 and 5) -- approach problems with the intention of "getting through" them as best they can. Indeed, the scores for their mathematical solutions to these problems were below the mean.

The median group of subjects has general competency in problem solving, yet they do not have their problem-solving procedures in the automatic state of the highest group. They still find it necessary to reflect upon a problem before posing a solution.

Impossible problems show that it may not always be possible to make competent problem solvers out of poor ones. It also demonstrates that certain forms of situation-based reasoning have their limitations. The high posttest subjects, for example, demonstrated some situation-based reasoning *within the problem-solving context*, but this did not alert them to the absurdity of the larger situation described. Recognition that the problem met the necessary conditions for an algebraic solution seemed most critical. However, it was shown that situation-based reasoning with the "big picture" in mind, as demonstrated by median subjects, can be advantageous.

<u>Changes in how time is spent</u>. Total solution times did not differ greatly on average from Day 1 to Day 2. However, subjects allocated their total solution time somewhat differently over the two days. Subjects using ANIMATE 2.0 performed a variety of subtasks in order to produce an adequate formal specification of a word problem, and a situation-based verification. Six areas

were specifically identified on which subjects spent time. These are: Setting up the algebraic Network; setting up the animation; debugging the Network; running the animation (i.e., "checking" a solution attempt); reading the problem; and (for Motion problems only) specifying the stopping event for an animation. The relative amounts of time subjects spent on each of these tasks are shown in Figure 12 for Day 1 and Figure 13 for Day 2.

From training session 1 (Day 1) to session 2 (Day 2) subjects changed their problem-solving behaviors and spent a greater proportion of their time reviewing the problem texts ("Reading") and correcting initial solution attempts ("Debugging"). *Debugging* is taken here to mean alteration of a part of the solution network following execution of the animation (i.e., feedback). Setting up the animation is inherently fast, as is establishing the stopping event for the animation[1]. It could be argued that the time spent running the animation might also be included in the debugging category, as subjects used it to test their solutions and diagnose problems. Clearly, much of the reasoning that leads to error detection and correction occurs during animation execution. However, the more conservative stance is taken where running is considered separately.

Reading is not the major temporal focus of subjects. The texts are only a few lines long; and, though they may be far more dense propositionally than regular stories of this length (cf. Nathan, 1988), their format is familiar or quickly learned. Subjects seem to know where to go in the texts for information, with little serial scanning apparent in the protocols.

---

[1] To establish the stop event subjects identified the algebraic relation which stopped the animation when it was true. The time allocation for establishing this is inappropriately weighted in Figures 12 and 13 since set up times for stop events of Work and Investment problems are necessarily zero. When one considers just Motion problems, stop events are found to play a larger part in the solution process although this step still occupies a smaller proportion of the solution time than setting up the network, running the animation, and debugging the network.

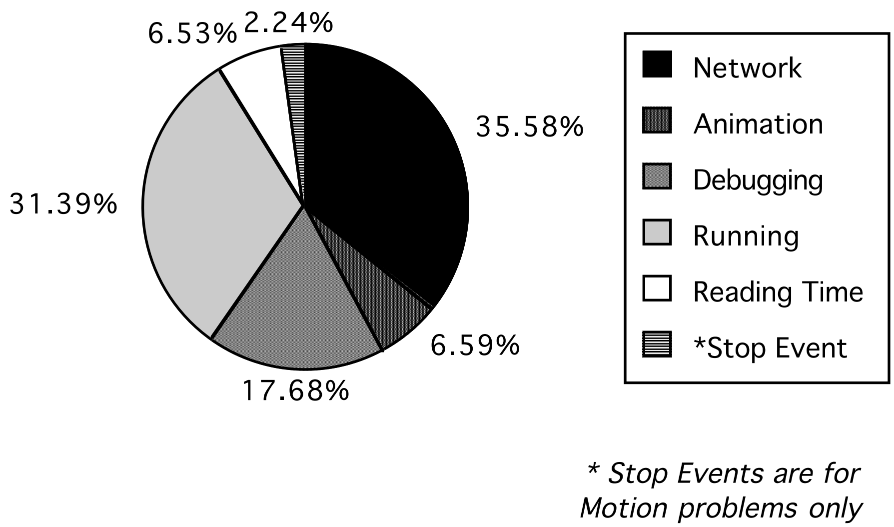


Figure 12. Proportion of time all subjects spend solving problems with ANIMATE 2.0 on Day 1

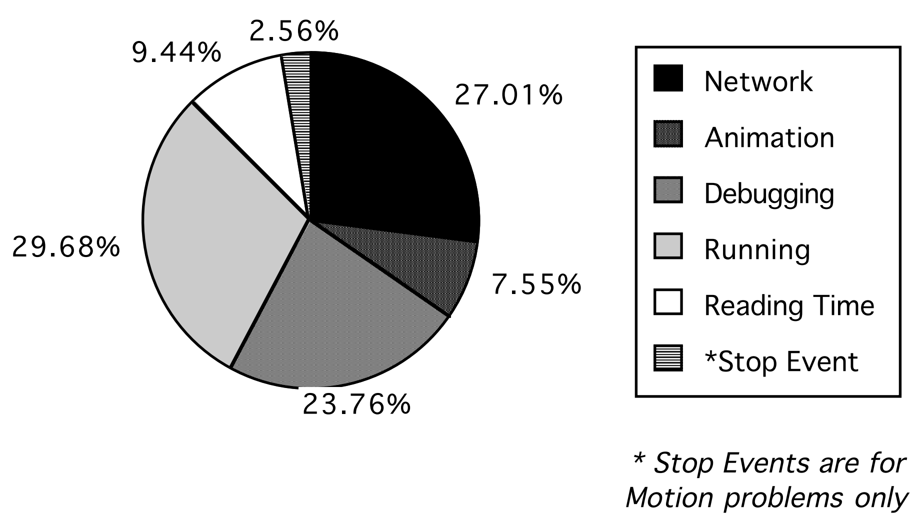


Figure 13. Proportion of time all subjects spend solving problems with ANIMATE 2.0 on Day 2

Effectiveness of the Animation as Feedback. In addition to the way subjects allocate their time when performing the above subtasks, it is valuable to note the frequency of two events: Reading the problem statement; and running the equation-driven animation. These serve as the main sources of outside information during a problem-solving session. Furthermore, the subject's goal while working within this learning environment is to match these two situational descriptions.

Data for the average number of runs and the average number of times subjects read each passage (Table 4) shows that subjects revisited the texts more often on Day 2. This may indicate the need to establish a closer link to the original scenario. Subjects conducted virtually the same number of runs from Day 1 to Day 2. Yet they apparently gleened more from each run over time. On Day 2 they spent more time debugging their solutions (Figure 13) and achieved a higher level of performance (Table 2 and Figure 7).

TABLE 4

Average number of times per problem subjects ran an animation
and read a problem text during training

Exposure	# Runs/problem/subject	# Reads/problem/subject
Day 1 training	2.5	2
Day 2 training	2.6	2.52

The animation as a source of feedback for students' formal expressions proved to be useful in problem solving. The data in Column 1 of Table 5 show the frequency with which subjects used the animated feedback to make corrections. Debugging was exercised equally on Days 1 and 2.

Subjects' higher performances were not accompanied by drastically fewer errors in their initial formulations of a solution. What then lead to the observable improvements? The data in Column 2 of Table 5 suggest an interesting possibility. These data show the frequency with which subjects corrected their solution attempts at each trial without use of animation-based feedback from the learning environment. These are times when subjects' changes were motivated by their own reasoning -- so-called "self-directed debugging." The increase shows that over time subjects took a greater responsibility in assessing and correcting their own errors. Perhaps this is because subjects *could*, as their understanding of algebraic expressions grew.

A startling and valuable finding is that self-directed debugging continued to be exercised at posttest time when the tutor was absent, indicating that subjects learned to debug the erroneous expressions they generated. This learned behavior would naturally support performance improvement from pretest to posttest. The rise in error rates from Day 2 training to the posttest can be explained by noting that without use of the feedback, there were no *additional* improvement from animation-driven corrections. Thus performance slipped.

TABLE 5

Frequency of corrections to algebraic errors made with and without the use of animation-based feedback

	Number of animation-driven corrections ("Debugging")	Number of student-motivated corrections ("Self-directed debugging")
Pretest (no tutor)	N/A	0
Day 1 training (w/ tutor)	22	3
Day 2 training (w/ tutor)	21	10
Posttest (no tutor)	N/A	10

Limitations and Successes. ANIMATE 2.0 represents a further attempt at crystalizing an instructional method for word algebra problem comprehension.The learning environment has shown itself to be an effective problem-solving aid, and its benefits for some are long-lasting. In its current form, the preferred audience for the tutoring system is not the expert problem solver. This population does not need help and, as was evident with the impossible problems, they are unlikely to take on this problem solving method. Yet, all ANIMATE subjects showed steep improvement, even with a two-day hiatus.

Conclusions

Performance tasks, control group learning environments, and verbal and computer protocols all help to identify the mechanisms which underlie changes in ANIMATE users' problem-solving behaviors. The experimental controls of the first study show that the greatest gains in performance inference-making occur when students are exposed to the ANIMATE learning environment which links situational and mathematical aspects of a problem. Reasoning about the referent situation as an effective way to solve these notorious problems. Protocols from the second study suggest that ANIMATE users tend to establish a closer link between the problem-solving process and the problem statement over time. Students do not learn to generate error-free solutions, but continue to make errors. These errors can be new learning opportunities since their correction can reinforce the target concepts (cf. Johnson, 1990). That students became proficient at correcting errors during training is flattering to the design of the learning environment. The equation-driven animation as a form of feedback proved to be highly effective. Students also routinely corrected expressions during training without the aid of the animation. Furthermore, they continued to do so at posttest time with no tutor present. This finding shows that some important conceptual learning about the problem-solving process took place among these students.

The changes in the problem-solving processes of ANIMATE users lead to problem-solving behaviors reminiscent of experts. Skilled behavior in a variety of domains is often characterized by a substantial number of errors early in the process, which expert practitioners have learned to correct. In computer-based text editing, Card, Moran, and Newell (1983) show that skilled secretaries spent about one-quarter of their task time making and correcting errors. Professional mathematicians also make mistakes while performing routine manipulations in algebra (Lewis, 1981). Flawlessness is not what makes them experts. They know their flaws can be detected upon reexamination. In writing compositions, experts tend to write their initial ideas down, knowing they have not stated them perfectly the first time, and that they can later revise them (Scardamalia, Bereiter, & Steinbach, 1984). Novices, in contrast, tend to make a single pass over the information. This dichotomy is similar to the behavior of our students. Early on, students write expressions with little reflection or alteration of these initial attempts. From exposure to the ANIMATE environment, students develop methods for assessing their solutions without external feedback. Mathematics problem solving becomes a process of formulation and revision.

The results are promising, but by no means do they suggest a miracle for mathematics education. They indicate that the development of global comprehension skills -- adherence to the original problem statement, consideration of the situation, interpretation of expressions and assessment of their validity -- are valuable to teach. And they are learnable in a setting whereby otherwise abstract expressions are grounded in situations relevant to the problem-solving process; and generation, testing and alteration of them is encouraged. In essence, these students teach themselves concepts in mathematics and approaches for mathematics problem solving in a technologically simple learning environment. The system may be unintelligent by ITS standards, but it communicates vital knowledge to the student. Necessarily, when problem solving with ANIMATE, the responsibility for learning, goal-setting, and diagnosis is placed on the student. It has been shown that students can take the initiative and valuable learning occurs in the process.

ACKNOWLEDGEMENTS

The author extends his warmest appreciation to Dr. Walter Kintsch for his advice and support throughout this research, to Violetta Cavalli-Sforza and Elana Joram for comments on an earlier draft, and to Dr. Lauren Resnick. The author is currently a James S. McDonnell Postdoctoral Fellow at the University of Pittsburgh, JSMF Grant No. 91-18.

REFERENCES

Anderson, J. R. (1988). The expert module. In M.C. Polson and J.J. Richardson (Eds.) *Foundations of intelligent tutoring systems*. Hillsdale, NJ: Erlbaum.

Card, S. K., Moran, T. P., & Newell, A. (1983). *The psychology of human-computer interaction*. Hillsdale, NJ: 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.

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

DeCorte, E., Verschaffel, L., & DeWin, L. (1985). The influence of rewording verbal problems on children's problem representation and solutions. *Journal of Educational Psychology, 77,* 460-470.

Duncan, D. B. (1955). Multiple range and multiple F tests. *Biometrics, 11,* 1-41.

Ericsson, K. A., & Simon, H. A. (1984). Protocol analysis: Verbal reports as data. Cambridge, MA: MIT

Hall, R., (1990). Exploring the episodic structure of algebra story problem solving. Unpublished doctoral dissertation. University of California, Irvine, CA.

Johnson, W. L. (1990). Understanding and debugging novice programs. *Artificial Intelligence, 42*, 51-97.

Kintsch, W. (1986). Learning from text. *Cognition and Instruction, 3*.

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

Lewis, C. H. (1981). Skill in algebra. In J. R. Anderson (Ed.), *Cognitive skills and their acquisition,* pp. 85-110. Hillsdale, NJ: Erlbaum.

Miller, D. P., & Stone, A. C. (1989). ProtoTymer: Human performance instrumentation for Hypercard prototyping. *Proceedings of the Human Factors Society 33rd Annual Meeting*.

Nathan, M. J. (1988). Recall of stories and story problems. Unpublished Masters Thesis. University of Colorado.

Nathan, M. J. (1990). Empowering the student: Prospects for an unintelligent tutor for word algebra problem solving. *Proceedings of CHI '90*. (Seattle). ACM. 407-414.

Nathan (1991). A theory of word algebra problem comprehension and its implications for the design of computer-based learning environments. Unpublished doctoral dissertation, University of Colorado. (University Microfilms Vol. 52, Issue 11, No. 9206623).

Nathan, M.J., Kintsch, W., & Young, E. (in press). A theory of word algebra problem comprehension and its implications for the design of learning environments. *Cognition and Instruction*.

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

Resnick, L. B. (1989). *Knowing, learning and instruction*. Hillsdale, NJ: Erlbaum.

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

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

Schneider, W., & Shiffrin, R. M. (1977). Controlled and automatic human information processing: I. Detection, search, and attention. *Psychological Review, 84*, 1-66.

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