A RAND NOTE

Algebraic Thinking Tools: Supports for Modeling Situations and Solving Problems in Kids' Worlds

David McArthur, Matthew Lewis,
Tor Ormseth, Abby Robyn,
Cathy Stasz, Don Voreck

July 1989

RAND

The work described in this report was sponsored by the National Science Foundation under Grant No. MDR-8751515.

This Note contains an offprint of RAND research originally published in a journal or book. The text is reproduced here, with permission of the original publisher.

Published by The RAND Corporation
1700 Main Street, P.O. Box 2138, Santa Monica, CA 90406-2138

A RAND NOTE

N-2974-NSF

Algebraic Thinking Tools: Supports for Modeling Situations and Solving Problems in Kids' Worlds

David McArthur, Matthew Lewis,
Tor Ormseth, Abby Robyn,
Cathy Stasz, Don Voreck

July 1989

Prepared for
The National Science Foundation

RAND

Preface

For the past three years, the Algebra Tutor Project has been developing and field testing an intelligent computer tutor for basic algebra. This Note, originally published in the January/February 1989 issue of *Technology and Learning* (Vol. 3, No. 1), discusses different versions of the tutor.

This research was supported by a grant from the National Science Foundation's Directorate for Science and Engineering Education (Application of Advanced Technologies Program).

Additional reports on the Algebra Tutor Project, listed below, can be obtained from The RAND Corporation's Publications Department:

D. McArthur, C. Stasz, and J. Y. Hotta, *Learning Problem Solving Skills in Algebra*, The RAND Corporation, N-2595-NSF, May 1987.

D. McArthur, C. Burdorf, T. Ormseth, A. Robyn, and C. Stasz, *Multiple Representations of Mathematical Reasoning*, The RAND Corporation, N-2758-NSF/RC, May 1988.

D. McArthur, C. Stasz, J. Hotta, O. Peter, and C. Burdorf, *Skill-Oriented Task Sequencing in an Intelligent Tutor for Basic Algebra*, The RAND Corporation, N-2966-NSF, June 1989.

Summary

This Note describes several versions of an intelligent tutor for basic algebra that we have been developing at RAND over the past three years. The versions of the tutor are built around several "core" components, including an algebra expert system and a student modeling component that can make inferences about misconceptions underlying students' errors. The different versions help students learn distinct kinds of mathematical reasoning skills. We first discuss the equation-solving tutor, which focuses on the acquisition of relatively "low-level" symbol manipulation skills. Then we review the model-building tutor, a recent version that helps students acquire important mathematical reasoning skills that are not part of most algebra curricula, including the ability to formulate a mathematical model of real-world situations and to test mathematical hypotheses. The Note concludes with a discussion of the implications of the tutors for curriculum change. We note how the pedagogical approach implicit in our tutor diverges from most classroom teaching principles. While traditional classrooms often engage in *isolated* and *bottom-up* practice, we advocate *embedded* and *global-before-local* practice.

Over the last three years we have been building a collection of intelligent tutoring systems and tools to help students learn basic algebra. The various versions of the algebra tutor have been tested in the lab and in the classroom as part of algebra courses at a local high school. Common to all versions of the tutor is a set of modules constructed using ideas from artificial intelligence and expert-systems technology, which give the different versions much of their human-like intelligence. In particular, common to each of the versions are: An *algebra expert system* that solves and can explain its solutions to problems in basic symbolic algebra using reasoning steps that students can understand; a *student modeling* component that can make inferences about the misconceptions underlying many of the students' overt errors; and a *task sequencing* component that can decide which concepts the student should learn next on the basis of inferences about students' past performance in the student model. The task sequencing component then generates problems that embody those concepts. (See McArthur, Stasz, and Hotta (1987) for more information on the algebra tutor versions. McArthur, Stasz, Hotta, Peter, & Burdorf (1988) contains a detailed discussion of task sequencing.)

We specifically designed the different versions of the tutor to support students' learning of distinct kinds of mathematical thinking skills. Our designs are based on several principles:

- *Articulation.* We begin with an analysis that uncovers or articulates the different kinds of skills involved in a particular type of mathematical expertise. Often articulation uncovers several kinds of tacit knowledge that even teachers are not aware students need to learn.
- *Reification.* We then design software tools that are specifically designed to *reify* each of these skills (Collins, Brown, & Newman, 1988). Reification means making visible and explicit a skill or activity that is often hidden and implicit -- as mathematical reasoning often is.
- *Coaching and supports.* The software tools we construct range from *coaches* that actively tutor the student in the skills and reasoning that are the topic of the particular tutor version, to more passive *supports.* Supports do not coach the students but rather provide information and tools that afford them opportunities to learn effectively by themselves.
- *Role sharing.* Because we articulate several different kinds of skills that contribute to a particular kind of mathematical expertise, a given problem will require the completion of several activities. The principle of role sharing says that students generally will *not* accomplish all of these activities themselves. Instead, the tutor will share problem solving with the student by accomplishing a specified subset of activities.

In the following sections we briefly show how different versions of our algebra tutor exemplify these principles while supporting the learning of mathematical thinking skills that are not currently part of most beginning algebra curricula in the United States. In addition, we also discuss how the tutor versions are distinct in pedagogy as well as content. They embed assumptions about learning and teaching that are substantially different from those implicit in traditional didactic theories.

Algebra Tutor Versions

The different versions of our tutor are most easily distinguished in terms of the "level" of the mathematical thinking skills they attempt to help students learn. The earliest versions of the tutor focused mainly on relatively "low-level" symbol manipulation skills, in contrast to higher-order thinking or problem-solving skills. (See Collins, Brown, & Newman (1988), and Schoenfeld (1985) for more discussion on the "continuum" of cognitive skills in mathematics).

The Equation Solving Tutor

The equation solving tutor was one of the the first algebra tutor versions we developed. As the name implies, it is mainly intended to help students learn skills involved in solving simple symbolic equations. The student sees the tutor as a collection of windows and menus, shown in Figure 1. The menus on the left allow the tutor and student to converse about reasoning and problem solving. To the right of the menus, on the bottom, is the "work window," where the student creates each new line in his or her solution. New lines or reasoning steps can be created by selecting commands from menus (as in Figure 1), typing in algebraic expressions, or writing them on an electronic tablet. To the right of the work window is the "comment window" where the tutor sends textual feedback to the student.

The large window in the upper right is the "display window," where the student's reasoning is recorded and queried. Problem solving is represented here as a *reasoning tree*. Many of the menu items to the left are used to manipulate the "nodes" in this tree. For example, "Explain Your Step" permits the student to point at parts of the reasoning tree done by the tutor and obtain justifications for the tutor's reasoning (see Figure 1). Similarly, "Help Next Step" allows the student to obtain several levels of hints from the tutor. Using these options, the student can obtain important coaching and learning supports.

Like AlgebraLand (Collins & Brown, 1987) and the Algebra Workbench (Richards & Feurzeig, 1988), the tutor displays the student's work as a solution tree, thus reifying the student's reasoning process by showing connections between steps. Each branch in the tree represents an alternate solution, or line of attack, on the problem. Hence a tree representation allows easy comparison of different solutions, both the student's and tutor's. Menu items like "Move Box" permit the tree to be exploited effectively: By selecting this item the student can move the site of activity in the problem solving from the current expression to a previous expression or an expression on a different path. They can then manipulate this expression, question its justification, or ask for help as to a possible next step.

The most important feature of this version is that it articulates and reifies different levels of problem-solving skills that are necessary to solve symbolic algebra problems. We distinguish three levels of skill here: *goals* -- the abstract problem solving goals that the student should adopt (the goals are shown in the menu on the upper-left corner of Figure 1), *operations* -- the mathematical operations that implement the selected goal (the operations are shown in the menu on the upper-right corner of Figure 1), and finally *manipulation* -- the application of an operation to create a new equation. It is our experience that classroom teachers are rarely aware that symbol manipulation expertise comprises at least these three distinct skills.

Consistent with the principle of reification, each of these different layers of decision-making is made explicit. The student must make a visible decision at each of these levels. In addition to breaking out the different levels of decision-making, the tutor is also designed so that each of these levels of reasoning can be either done by the student or the tutor, thus permitting role sharing. In the environment shown in Figure 1, the student decides goals or operations but the tutor executes the chosen operation or goal and creates a new equation.

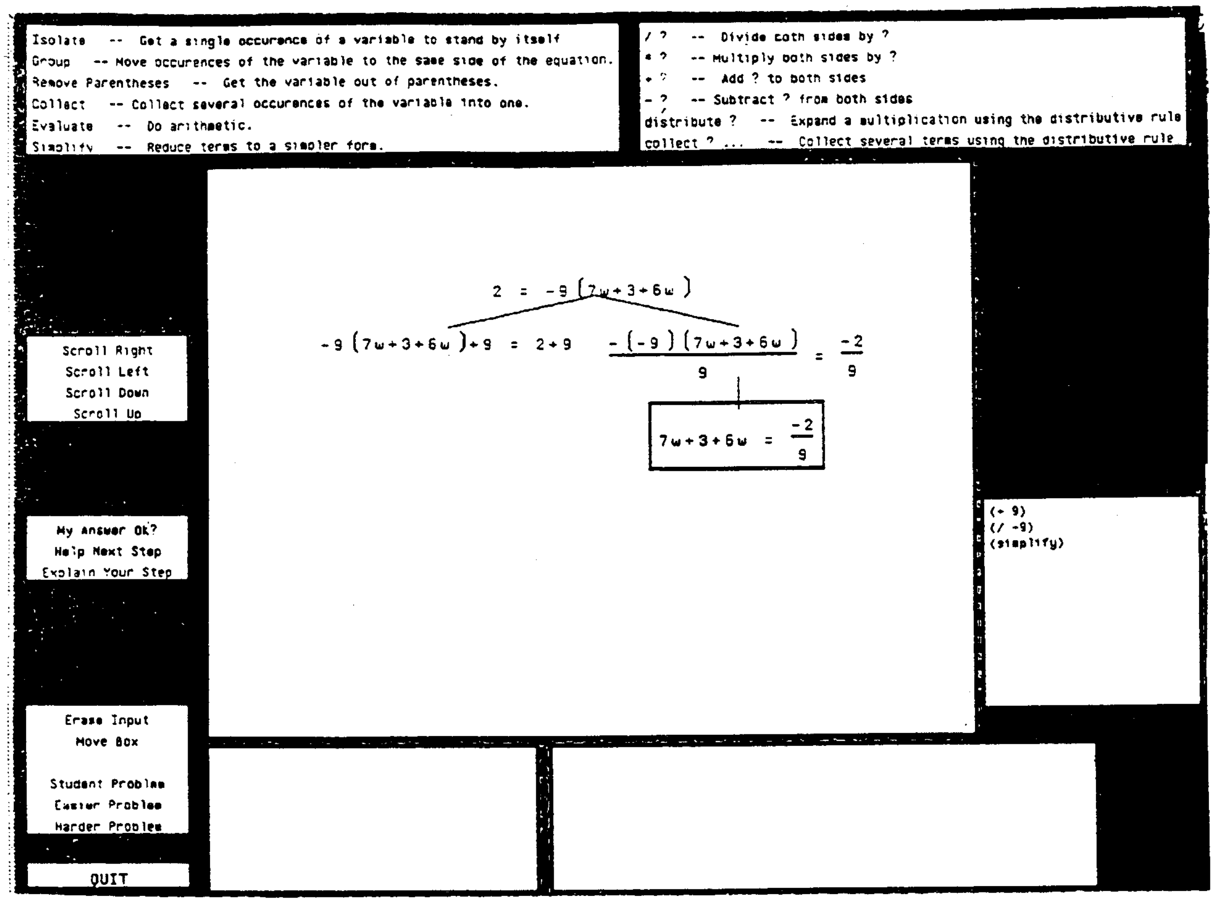

Figure 1. Interface of an equation solving tutor.

Model Building Tutor

More recent algebra tutor versions focus on higher-level mathematical thinking skills. One motivation for recent versions is the observation that symbol manipulation skills are no longer as important as they once were for students to learn. Why bother to teach a skill, so the argument goes, when it can be done faster and more accurately by a machine? For example, should students memorize multiplication tables when cheap calculators are readily available?

The skills we have recently focused on fall under the general classification of *model building*. By model building skills we mean the ability to see how formal mathematical objects can relate to the real world; for example, how equations can provide powerful descriptions, or models, of the interactions among real-world objects or properties. Consistent with the four principles outlined above, we can articulate model building into several related skills. They include:

- *Formulation*, in a given real-world situation, involves determining which objects or properties should be represented as mathematical objects. For example, in modeling the finances of a car wash to raise money to send the marching band to a distant competition one might decide that properties like the number of cars and the amount you have to spend on supplies should be variables in an equation that can be used to predict profit or loss.
- *Data gathering and representation* represents a collection of related skills for collecting and examining information about values of properties selected during formulation. The intent of these activities is to understand empirical patterns of co-occurrence among variables. For example, in the car wash situation, one might record tables of values and graphs showing how profit appears to covary with number of cars washed.
- *Translation* involves positing a mathematical representation that captures the observed patterns of interrelations of the formulated situational properties. For example, in the car wash situation, translation may involve postulating an equation that relates the variable for profit to variables representing the amount spent on supplies, the fee charged per car, and so on. This skill is often referred to as hypothesis generation.
- *Inferencing or prediction* involves answering questions about the situation at issue, either by manipulating mathematical representations that act as models for the situation, or using less formal techniques that manipulate data representations. For example, given a problem in which one must infer the required duration of exercise to achieve a given weight loss one can use an equational model for the situation or perhaps consult graphs and tables of values. This skill is related to hypothesis testing.

Our interest in developing computer tools to help students learn model building skills stems not only from a personal belief in their importance but also from NCTM curriculum reform efforts (Thompson, 1988), and other related research (e.g., Fey, 1984, Usiskin, 1985).

Figure 2 shows the interface of a tutor version that provides coaching and passive supports for various model building skills. In this version, the situation to be modeled is described verbally in the "situation window" at the upper-left. Below the situation description is a "word graph" where the student has enumerated (formulated) a list of variables that relate the dependent variable, weight loss, to various independent factors. Using the word graph, the student can input values for selected variables. The tutor then computes the appropriate values for other variables, based on an underlying equation that relates the variables. The student's task, at this point, is to gather sufficient data to infer the underlying equation or model.

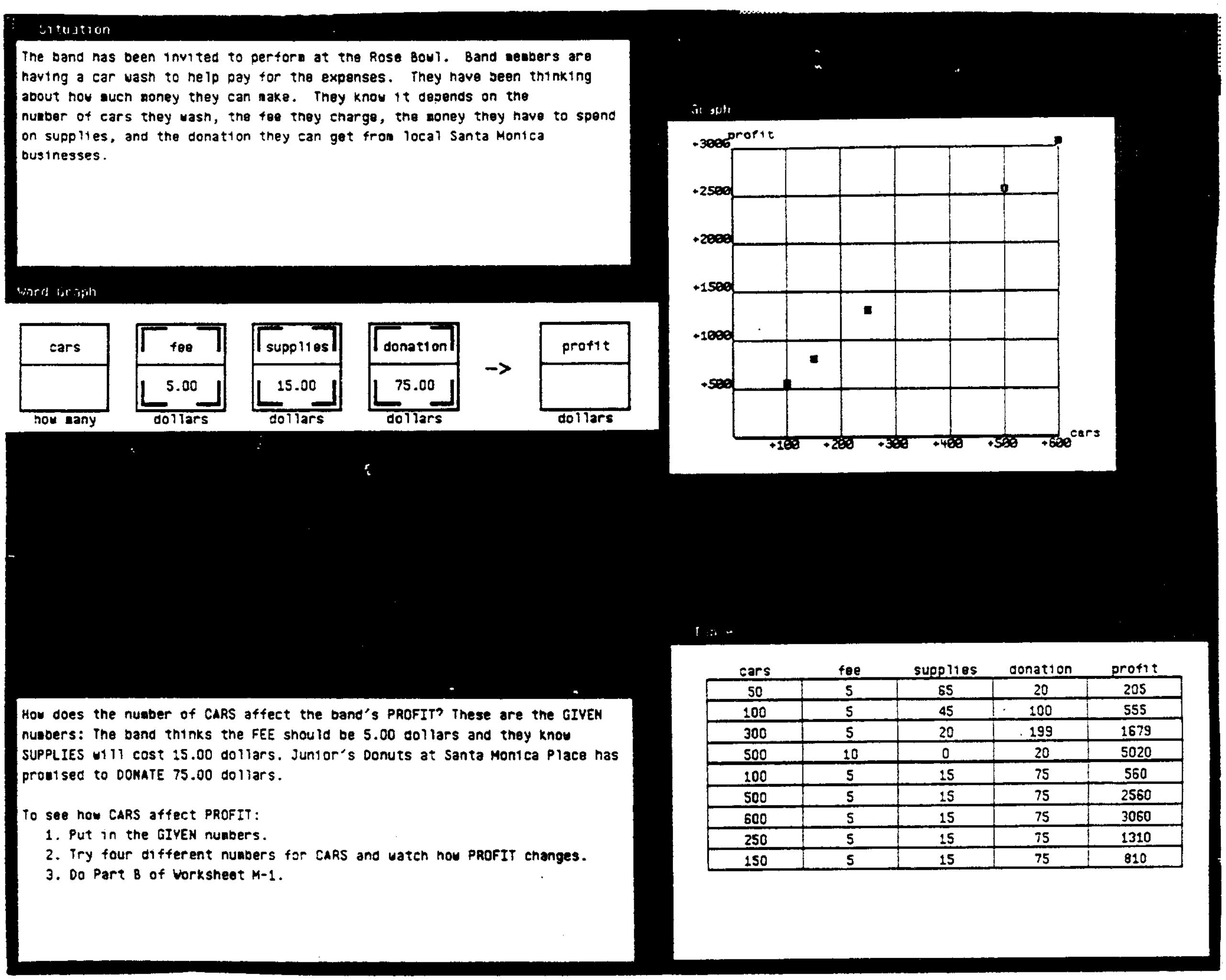

Figure 2. Interface of a model building tutor.

The environment provide several tools to help the student gather and organize data. These tools include tables of values (lower-right), and cartesian graphs (upper-right). In addition, the student is guided in his or her data gathering by specific situations (lower-left). After students have a chance to freely explore varying values for variables, the specific situations generally constrain the student to explore covariation of pairs of variables. Here, the number of cars washed and the profit can can vary while the fee per car, cost of supplies, and amount donated by business stay the same. The data gathered in this "constrained exploration" permits the student to observe a positive linear relationship between time and loss.

After such controlled explorations the student is usually in a position to specify the operators (*, /, +, and -) that relate the variables in the word graph, thus achieving translation. Equations are first represented as "word equations," which look like word graphs with operators connecting the "boxes" that represent variables. Next, symbolic equations are computed from word equations by substituting letters for variable boxes (see Figure 3). Subsequently, students are given a series of problems about the topic, e.g., "The band sets the fee at $4.75, supplies are $15.00 and the donation comes to $75.00. The band will need to make 500.00 dollars to pay for the Rose Bowl performance. How many cars does the band need to wash?" The students can then use their equational models to generate answers. It is at this point that the equation-solving tutorial software comes into play, coaching the student through the manipulations required to make the desired inference.

The most important feature of this version of the tutor is that it reifies each of the model building activities mentioned earlier. We have already noted the various tools designed for data gathering and representation. In addition, formulation and translation activities are each associated with their own graphic representations (word graphs and word equations, respectively). As with the equation solving tutor, the model building version also permits role sharing. Each activity -- including not only model building (formulation, data gathering, translation, etc.) but also the equation-solving activities entailed in solving specific problems (goals, operations, and manipulation) -- can be done either by student or tutor. Indeed, we define "role profiles" that, for any particular situation, stipulates which roles will be done by which agent.

Pedagogical and Content Considerations

The various algebra tutoring versions we have described depart from the curricula found today in U.S. schools. These departures from how beginning algebra is traditionally taught include differences in both content and pedagogy.

In terms of *content*, three differences exist. First, in teaching topics that are part of the traditional curriculum (such as equation solving), we attempt to articulate levels of reasoning that are usually left tacit. Second, we also focus on topics (such as model building) that are not part of current curricula. Finally, we are attempting to connect topics that are part of the traditional curriculum with our newer topics. For example, the model building version of the tutor actually uses the equation solving software as a component. Equation solving is therefore not ignored, but neither is it taught as a topic in isolation. It is presented as an inferential tool; a means to an end and not an end in itself. In other words, it is *situated* in natural contexts of use (Brown, Collins, & Duguid, 1988).

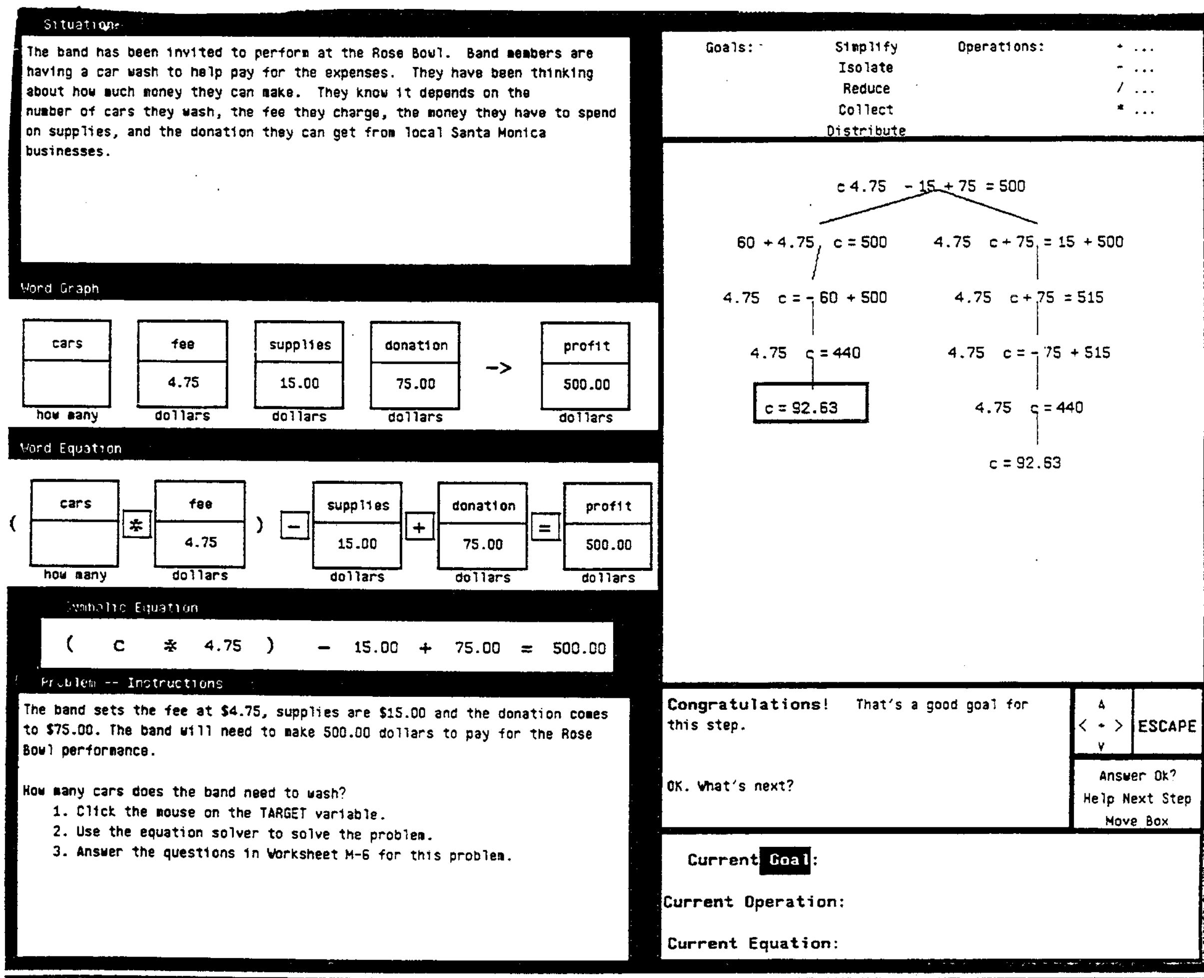

Figure 3. Interface of the combined model building and equation solving tutors.

Perhaps deeper than issues of content are *pedagogical* differences between our tutors and traditional classroom didactic practice. Our approach to helping students learn mathematical thinking skills differs fundamentally from most classroom teaching principles. More importantly, some of our basic views on how students learn appear contrary to prevailing assumptions. A simplified characterization of traditional classroom learning and pedagogical tenets might include:

1. *Intensive practice*. Individual skills must be practiced intensively if they are to be mastered. For example, if a student is to learn equation solving, each of the skills involving deciding on goals, operations, use of axioms like the distributive rule, and so on, must be exercised many times. This is a principle of learning that appears well-founded on many decades of research in cognitive psychology and education.
2. *Focused practice*. Students cannot learn more that one or two new skills at the same time. For example, students should not learn how to factor quadratics and solve linear inequalities at the same time. This also appears to be a well-founded principle of learning.
3. *Isolated practice*. New skills should be practiced in a relatively "pure" fashion, divorced from related skills that are not familiar and also from well-learned skills. For example, when learning how to solve simple equations, problems should isolate those skills and not elicit well-learned arithmetic skills (e.g., the problem shouldn't have lots of "big numbers"). Although not "well-founded," one might regard this as a principle of learning or of "cognitive economy": Skills will be mastered faster if distractions are minimized.
4. *Bottom-up practice*. New skills should be learned in a sequence that reflects a certain "logical" prerequisite ordering. For example, to solve an algebra problem you need to have mastered arithmetic and to do model building and make inferences using models, you need to understand the algebra of such models. Again, although not "well-founded" in research, this tenent is prevalent in most current instructional design.

As obvious as these tenets might appear, we wish to argue that while 1 and 2 are correct, 3 and 4 are not. We believe that isolated practice and bottom-up practice are often counterproductive to useful learning and that they are not logically necessary but rather only pragmatically expedient. This stems from the fact that, given the way classroom learning is now structured, it is unrealistic for teachers to consider interesting alternatives.

An emerging view of learning that differs radically from traditional didactic theory is the model of *situated cognition* voiced by Lave (1988) and extended by Brown and Collins (Collins, Brown, & Newman, 1988; Brown, Collins, & Duguid, 1988). This view of situated learning suggests that individual skills only acquire their meaning when used in their natural context; that is, when they are used in typical situations and in conjunction with the other skills with which they usually interact. Such a perspective argues that isolated practice will result in skills that might be mastered locally but that will not transfer to the important situations in which they are naturally used. In response, a theory of situated cognition might suggest:

3**Embedded practice*. New skills should not be practiced in isolation. They must be learned in the context of realistic problems that elicit this new skill and all the other skills that typically are needed to solve the problems.

Similarly, situated cognition argues that sequencing skill acquisition in a bottom-up curriculum hides the meaning of individual skills by presenting them in an order that may be expedient but which obscures their natural contexts of use. For example, it might be argued that teaching equation-solving skills before skills involving their use as inferential tools imparts meaningless symbol manipulation abilities. Consequently, these symbol manipulation skills might have to be substantially relearned when students are finally shown how they are used to model real-world situations. As a contrary tenet, a theory of situated cognition might suggest:

4*Global-before-local practice. Teach more global skills that allow the student to form a conceptual framework for a topic before teaching the skills for accomplishing local operations. (See Brown, Collins, & Duguid (1988) for further discussion of this principle and related issues.)

If one adopts the situated view of learning embodied in 3* and 4* the challenge then becomes designing learning environments that permit such principles to be implemented. From the perspective of traditional didactic theory tenets 3 and 4 appear not only reasonable, but impossible to circumvent even if one wished to. In tutoring algebra for example, how can we expect students who haven't mastered the basic skills of arithmetic to learn equation-solving skills? How can we teach students about the use of equations as models if they haven't already mastered equation solving? To add to the difficulty, since we have now articulated skills like equation solving and model building into several subskills, how can we expect students to get intensive, focused practice on any *single* skill if *every* skill required to solve a realistic problem must be executed?

Brown and Collins have begun to discuss new kinds of learning environments that respond to these challenges (Collins, Brown, & Newman, 1988; Brown, Collins, & Duguid, 1988). The algebra tutors and tools discussed above represent our first attempt to provide such different learning environments. Note that by articulating and reifying skills and supporting shared problem-solving roles, we allow the student to obtain focused, embedded practice. For example, our "role profiles" can be set so the tutor does all the manipulation. This allows the student to focus exclusively on deciding which equation-solving goals exist at any moment in problem solving. The student exercises only one or two skills at a time, but the problems do not have to be crafted to only elicit these skills. The problems can be as realistically complex as desired, since the tutor will supply the other skills, as needed.

The same principles of articulation, reification, and role sharing enable global-before-local practice. For example, because the actions of setting goals, choosing operations, and doing manipulations can be selectively done for the student, we have tested an environment in which the skills of manipulation are done for the student. Hence, they can can (and do) acquire algebra proficiency without mastering the symbolic manipulation skills. On a larger scale, we can also permit students to explore model building aspects of mathematics before mastering equation-solving skills. In some cases, we permit students using the model-building version of the tutor to explore situations and answer specific questions before they have received any training on symbol manipulation. To generate answers they either use skills such as approximation (using a graph) or table look-up (using a table of values); alternatively the tutor plays the role of the equation solver. In both cases, the students see the value of equation solving in a realistic context of use.

In summary, along with Brown and Collins, we feel one key to providing powerful situated learning environments is *collaboration*. When the many skills required to solve complex realistic problems are articulated they can become different roles that multiple agents share in cooperatively building a solution. We believe that one of the most potentially powerful roles the computer might play in education is that of a collaborating agent who shares problem solving with the student.

Acknowledgments

This research is being funded by the National Science Foundation (Applications of Advanced Technologies Program). Views or conclusions expressed herein do not necessarily represent the policies or opinions of the sponsor.

References

Brown, J. S., Collins, A., & Duguid, P. Cognitive apprenticeship, situated cognition and social interaction. BBN Tech. Report 6886, June, 1988.

Collins, A., & Brown, J. S. (1987). The computer as a tool for learning through reflection. In H. Mandl and A. Lesgold (Eds.), *Learning issues for intelligent tutoring systems*. New York: Springer-Verlag.

Collins, A., Brown, J. S. & Newman S. E. (1988). Cognitive apprenticeship: Teaching the craft of reading, writing, and mathematics. In L.B. Resnick (Ed.), *Cognition and Instruction: Issues and Agendas*. Hillsdale NJ: Lawrence Erlbaum.

Fey, J. T. (1984). *Computing and Mathematics: The Impact on Secondary School Curricula*, Report on the National Council of Teachers of Mathematics Conference.

Lave, J. (1988). *Cognition in Practice*, Cambridge, Cambridge University Press.

McArthur, D., Stasz, C., & Hotta, J. (1987). Learning problem-solving skills in algebra. *The Journal of Educational Technology Systems, 15*(3), 303-324.

McArthur, D., Stasz, C., Hotta, J., Peter, O., & Burdorf, C. (in press). Skill-oriented task sequencing for basic algebra. *Instructional Science*.

Richards, J., & Feurzeig, W. (1988). Intelligent Tools for Algebra. *Technology and Learning, 2*(3), 1-5.

Schoenfeld, A. H. (1985). *Mathematical Problem Solving*. New York: Academic Press.

Thompson, C. S., & Rathmell, E. C. (1988). NCTM's standard for school mathematics, K-12. *Mathematics Teacher, 81*(5), 348-351.

Usiskin, Z. (1985). We need another revolution in secondary school mathematics. In *The Secondary School Mathematics Curriculum*, National Council of Teachers of Mathematics.

RAND/N-2974-NSF