Cognitive Tutors as Modeling Tool and Instructional Model

Dr. Kenneth R. Koedinger

Pittsburgh Advanced Cognitive Tutor (PACT) Center
Human-Computer Interaction Institute
School of Computer Science
Carnegie Mellon University

What gets measured, gets done.
If you don't measure results, you can't tell success from failure.
If you can't recognize failure, you can't correct it.
If you can't see success, you can't reward it.
If you can't see success, you can't learn from it.
From "Reinventing Government"
By David Osborne and Ted Gaebler

Koedinger, K. R. (2001). Cognitive tutors as modeling tool and instructional model. In Forbus, K. D. & Feltovich, P. J. (Eds.) Smart Machines in Education: The Coming Revolution in Educational Technology, (pp. 145-168). Menlo Park, CA: AAAI/MIT Press.

Introduction

Effective technologies for learning and doing mathematics should be based on sound cognitive theory, be empirically tested against alternatives, and be primarily addressed at mathematics as a modeling language. I illustrate these points in the context of an educational technology we call "Cognitive Tutors" (Anderson, Corbett, Koedinger, Pelletier, 1995). Cognitive Tutors are based in computer science research on artificial intelligence techniques and cognitive psychology research on the nature of human learning and performance. Cognitive Tutors have been created to help students learn in a variety of mathematics and computer programming domains and have been subject to laboratory and classroom evaluations that demonstrate the potential for dramatic learning gains from appropriate use of this technology.

This chapter will focus primarily on a Cognitive Tutor for algebra originally called the Pump Algebra Tutor or PAT (Koedinger, Anderson, Hadley, & Mark, 1997; Koedinger & Sueker, 1996). PAT is part of a complete algebra course that, in the 1998-1999 school year, had been disseminated by our university-based PACT Center[1] to thousands of students in some 70 schools across the country. This course, now called "Cognitive Tutor Algebra I", was designated by the US Department of Education as an exemplary mathematics curriculum in 1999. With the help of the Technology Transfer Office at Carnegie Mellon University, we formed a spin-off company "Carnegie Learning"[2] to market, support, and further develop Cognitive Tutors. In the 1999-2000 school year, Carnegie Learning had spread Cognitive Tutor Algebra I to over 150 schools. A few of these schools are high performing, resource rich suburban schools, but most of them are urban or rural schools, involve average teachers, and include a large number economically disadvantaged, minority or learning disabled students.

Before turning to a summary of Cognitive Tutors in general and Cognitive Tutor Algebra in particular, I want to make three general comments about:

- the role of empirical testing in the development of technology-enhanced learning innovations,
- the role of cognitive theory in guiding such development, and
- the role of mathematical modeling as an appropriate core focus for mathematics instruction.

---

[1] The Pittsburgh Advanced Cognitive Tutor (PACT) Center is codirected by Albert Corbett, Kenneth R. Koedinger, and John R. Anderson and is located in the Human-Computer Interaction Institute at Carnegie Mellon University. See http://act.psy.cmu.edu/ACT/tutor/tutoring.html.

[2] For more information on Carnegie Learning see the web site at http://carnegielearning.com.

Why empirical tests against alternatives?

Why is it important that we perform empirical tests of educational innovations in comparison with alternatives? If the intuitions and beliefs that guide the design of learning environments were fully informed and perfect, there would be no need for such experiments. Unfortunately, intuitions and beliefs about learning and instruction are limited and are not always accurate. One problem is that intuitions are based largely on conscious learning experiences, but a great fraction, perhaps the majority, of what we learn is at a level below our awareness. The grammar rules of our first natural language, English in my case, are an excellent example. We learn these rules, in the sense that they determine our behavior in language comprehension and production, well before we are consciously aware of them. To use an old twist of phrase, as early language learners we go from "not knowing we don't know" to "not knowing we know" without going through the intermediate states of conscious learning: "knowing we don't know" and "knowing we know". As we get older, of course, conscious learning processes play a greater role. However, it is a mistake to think conscious learning takes over. In fact, there is ample evidence from cognitive psychology research that our brains continue to engage in implicit learning processes (e.g., Berry & Dienes, 1993; Dienes & Perner, 1999).

Our intuitions about learning are biased by limited information -- overly influenced by our memories of our conscious learning experiences. We are subject to what I call *expert blindspot* -- as experts in a domain we are often poor judges of what is difficult and challenging for learners. Perhaps few would disagree about the importance of evaluating our educational innovations to better understand how they do or do not improve on current practice. Nevertheless, I think it is worth emphasizing the danger of being biased by expert blindspot and lured by our personal intuitions into assuming that our educational innovations and reforms will necessarily be for the better.

Why design systems based on cognitive theory?

Not every experiment can be run comparing alternative features of instruction and their interactions. Thus, we need a way to guide the generation of new instructional designs. Such a guide should help us prune design ideas not likely to enhance learning and inspire new ideas that will. Cognitive theory also provides a way to accumulate reasons for past successes and failures to inform future practices.

Why address math as a modeling language?

Although technology has mastered calculation of various kinds -- arithmetic, graphic, symbolic, logical -- humans are the only masters of translating problems into mathematics, building theories and producing communicative forms. Learning how to create mathematical models of problem situations is difficult but it is the key to mathematical success in our modern world.

An Example: Cognitive Tutors

The Cognitive Tutors technology is particularly suited (though certainly not exclusively) to facilitate these three goals: need for empirical testing, need for cognitive theory, and a focus on mathematics as a modeling language. Cognitive Tutors are based on the ACT theory of learning and performance (Anderson & Lebiere, 1998). ACT is a complex and broad "unified theory of cognition" (Newell, 1990). I highlight just a few key features that are particularly relevant to learning mathematics. The theory distinguishes between tacit performance knowledge, so-called "procedural knowledge" and static verbalizable knowledge, so-called "declarative knowledge". According to ACT, performance knowledge can only be learned *by doing*, not by listening or watching. In other words, it is induced from constructive experiences -- it cannot be directly placed in our heads. Such performance knowledge is represented in the notation of if-then production rules that associate internal goals and/or external perceptual cues with new internal goals and/or external actions. Here are three examples of English version of production rules:

IF	the goal is to prove two triangles congruent and the triangles share a side
THEN	check for other corresponding sides or angles that may congruent.
IF	the goal is to solve an equation in X
THEN	graph the left and right sides of the equation and find the intersection point(s).
IF	the goal is to find the value of quantity Q and Q divided by Num1 is Num2
THEN	find Q by multiplying Num1 and Num2.

It is important to note that the rules of mathematical thinking (which production rules are intended to represent) are **not** the same as the rules of mathematics (e.g., theorems, procedures, or algorithms) as they appear, for instance, in textbooks. Production rules represent people's tacit knowledge of when to chose particular mathematical rules as well as other tacit performance knowledge like plans and informal intuitions. Reading English versions of production rules like those above can be misleading because the rules are stated explicitly: however, production rules represent tacit or implicit knowledge. When we say people know a production rule we do not mean they can state it, as written above or otherwise, but only that it characterizes their behavior. In other words, a person is said to know a production when in the situation described by the if-part of the production, the person can perform the action described by the then-part[3].

The particular if-then notation of production rules is not as important as the features of human knowledge that production rules represent and the implications of these features for instruction. Production rules are *modular,* and this means that we can diagnose specific student weaknesses and focus instructional activities on improving these.

---

[3] Because a number of different production rules can be applicable in any particular situation, a person may have a production rule yet we do not see evidence of it in that situation because some other competing production "fires" instead (cf. Anderson & Lebiere, 1998).

Production rules are *context specific,* and this means that mathematics instruction cannot be effective if it disconnects mathematics from its contexts of use. Students need true problem solving experiences to learn the if-part of productions, the conditions for appropriate use of mathematical rules, as well as some occasional small exercises (which are still over-emphasized in many curricula) to introduce or reinforce the then-parts of productions, the mathematical rules themselves. Production rules are of *limited generality*. In other words, cognitive research (e.g., Singley & Anderson, 1989) has shown that the performance knowledge, though general (i.e., it applies in multiple contexts), tends to be fairly narrow in its applicability and tied to particular contexts of use and limited generalizations thereof (cf., Cheng & Holyoak, 1985). Thus, we must gauge our expectations about how far student learning will transfer and construct curricula that both encourage general encodings of mathematical ideas and also provide multiple examples and activities that apply these ideas in a variety of well-chosen contexts.

In applying the ACT theory to instruction, we have focused on the idea that human one-to-one assistance or tutoring is extremely effective in facilitating learning. Bloom (1984) showed that an individual human tutor can improve student learning by two standard deviations over classroom instruction. In other words, the average tutored student performs better than 98% of students receiving classroom instruction. This result provides a sort of "gold standard" for comparing the effectiveness of educational technologies. The results of meta-analyses of hundreds of studies of traditional computer-aided instruction (CAI) suggest that CAI leads, on average, to a significant 0.3 to 0.5 standard deviation improvement over non-computer-aided control classrooms (e.g., Kulik & Kulik, 1991). There are too few studies of multimedia and simulations at this point to provide a generic figure, though some of these studies indicate little effect, for instance, of animations (e.g., Pane, Corbett, & John, 1996) or of game-like simulations (e.g., Miller, Lehman, & Koedinger, 1999). In studies of our Cognitive Tutor technology we have shown Cognitive Tutors to yield about a one standard deviation effect (Anderson, Corbett, Koedinger, & Pelletier, 1995; Koedinger, Anderson, Hadley, & Mark, 1997).

To build a Cognitive Tutor, we create a cognitive model of student problem solving by writing production rules that characterize the variety of strategies and misconceptions students acquire. These productions are written in a modular fashion so that they can apply to a goal and context independent of what led to that goal. For simplicity of illustration, I provide an example from the domain of equation solving:

Strategy 1:	IF the goal is to solve $a(bx+c) = d$ THEN rewrite this as $bx + c = d/a$
Strategy 2:	IF the goal is to solve $a(bx+c) = d$ THEN rewrite this as $abx + ac = d$
Misconception:	IF the goal is to solve $a(bx+c) = d$ THEN rewrite this as $abx + c = d$

The first two productions illustrate alternative strategies for the same problem-solving goal. By representing alternative strategies for the same goal, the cognitive tutor can follow different students down different problem solving paths of the students' own choosing. The third "buggy" production represents a common misconception (cf., Matz, 1982). Buggy production rules allow the cognitive tutor to recognize such misconceptions and thus, provide appropriate assistance. The Cognitive Tutor makes use of the cognitive model to follow students through their individual approaches to a problem. It does so using a technique called "model tracing". Model tracing allows the Cognitive Tutor to provide students individualized assistance that is just-in-time and sensitive to the students' particular approach to a problem.

The cognitive model is also used to trace students' knowledge growth across problem-solving activities. The "knowledge tracing" technique is dynamically updating estimates of how well the student knows each production rule (Corbett & Anderson, 1995). These estimates are used to select problem-solving activities and to adjust pacing to adapt to individual student needs.

Cognitive Tutors have been subject to comparative evaluations in the lab and in classroom for more than 12 years (Anderson, Corbett, Koedinger, & Pelletier, 1995). A Cognitive Tutor for writing programs in the LISP computer language (Anderson, Conrad, & Corbett, 1989) was compared to a control condition in which students solved the same programming problems without the aid of the Cognitive Tutor. Students in the experimental group completed the problems in *1/3 the time* with better post-test performance than students in the control group. The LISP tutor allowed students to engage in productive problem solving search, but reduced unproductive floundering. Two different Cognitive Tutors for geometry proof design were used in classroom studies compared to control classes using a traditional geometry curriculum without the Cognitive Tutor. In both studies, students in the experimental classes scored *1 standard deviation better* than students in control classes (Koedinger & Anderson, 1993).

There were two important lessons from these studies. First, echoing results from experiments with LOGO (Lehrer, Randle, Sancillo, 1989; Klahr & Carver, 1988), we demonstrated that careful curriculum integration and teacher preparation were critical to our effectiveness results (Koedinger and Anderson, 1993). A second lesson came from a third party evaluator who studied changes in student motivation and classroom social processes as a consequence of the use of the Geometry Proof Tutor. Schofield, Evans-Rhodes, and Huber (1990) found the classroom evolved to be student centered with the teacher taking a greater facilitator role supporting students as-needed on the particular learning challenges each was experiencing. This point was repeated in a Math Teacher article by one of the participating teachers (Wertheimer, 1990). In that article, Wertheimer emphasized that because the Cognitive Tutor was effectively engaging students he was more free to provide individualized assistance to students who most needed it.

The Pump Algebra Tutor (PAT)

When we began to develop the Pump Algebra Tutor, two past experiences led us to take a different "client-centered" approach to development (Koedinger, et al., 1997). First, at the time of the geometry studies, the National Council of Teachers of Mathematics

Standards (1989) were coming out and suggesting a deemphasis on proof in high school geometry. Second, we had experienced the importance and difficulty of integrating the technology with the classroom and paper-based curriculum (Koedinger & Anderson, 1993). Thus, in the Pump Algebra Tutor project we designed the tutor and the curriculum hand-in-hand. A high school math teacher, Bill Hadley, and a curriculum supervisor, Diane Briars, had been working on a new algebra curriculum. Their goals were to make algebra accessible to more students, to help students make connections between algebra and the world outside of school, and to prepare students for the "world of work" as well as further academic study. We teamed up with Hadley and began evolving the curriculum and designing the tutor by sharing ideas from both research and practice. This client-centered process we used and continue to use is a form of "participant design" (cf., Beyer & Holtzblatt , 1998) whereby end-users, in this case teachers, fully participate on the design team.

Functional Models of Authentic Problem Situations

Table 1 shows the first part of a two-day performance assessment used in the Pump Algebra curriculum. Students are asked to analyze the financial costs and benefits of three alternative educational paths to a photography career. Unlike traditional algebra story problems that ask for a particular numerical answer, activities in Pump Algebra and PAT ask students to produce an analysis and models of that analysis in multiple mathematical representations including tables, graphs, equations, and words.

The assessment activity in Table 1 illustrates a number of key features of the course that are typical of both the text curriculum and computer tutor. Students are expected to read realistic problem contexts: "Your friend has decided he is very interested in a career as a photographer. You look up Photographer in the Pennsylvania Career Guide and find out that there are three different paths to becoming a photographer." They must compare alternatives:

High School Diploma: Start photographing right away and make $1115 per month.

Technical School: Study for 18 months, pay $18,000, and make $1925 per month.

College: Study for 4 years, pay $50,000, and make $2754 per month.

Students are asked to "use algebra" to make this analysis and, like most activities in the curriculum, they build tabular and graphical models of these alternatives as well as symbolic equations like "$1115x = 1925(x - 18) - 18{,}000$". Students use these models to find break even points (e.g., how many months before Technical School pays off), for instance, by finding points of intersection in a graph or by solving equations. In other words, the course emphasizes the use of multiple representations and strategies to provide students with both different perspectives on mathematical understanding and a variety of tools for problem solving. In the classroom component of the course, writing is emphasized and students are asked to make recommendations based on the mathematical models they create: "use this analysis to write a letter to your friend explaining clearly the advantages and disadvantages of each option. You also want to make a recommendation to him as to what he should do!"

The beginning of the course starts with simpler situations than the Photographer problem, but such activities usually include many of these features: reading a problem situation, constructing multiple representations, comparing alternatives, and writing an answer. A major goal is to aid students in developing successively more sophisticated models of quantitative relationships using multiple representations each with different costs and benefits (cf., Koedinger & Anderson, 1998; Tabachneck, Koedinger, & Nathan, 1994).

Table 1. An Assessment from Pump Algebra that Illustrates the Problem Solving and Mathematical Modeling Objectives of the Curriculum

My Life as a Photographer
Final - Day One

Your friend has decided he is very interested in a career as a photographer. You look up Photographer in the Pennsylvania Career Guide and find out that there are three different paths to becoming a photographer. You can enter the field upon graduating from high school, go to a technical school or attend college. You research these options and find out the following:

Option 1: High School Diploma. He can become a photographer with only a high school diploma. The average salary for these photographers is $1115 per month.

Option 2: Technical School. Completing high school <u>and</u> attending a technical school for an eighteen-month (1 and 1/2 years) course in photography is a second option. This technical program costs about $18,000 and the average salary for those completing the course is $1925 per month.

Option 3: College. Completing high school <u>and</u> attending a four-year college program in photography is the third option. The average cost for a four-year program is $50,000 and a graduate can expect to earn about $2745 per month.

You want to make a complete comparison of these three options for your friend. Since you are an excellent algebra student you want to use algebra to make this analysis, and then use this analysis to write a letter to your friend explaining clearly the advantages and disadvantages of each option. You also want to make a recommendation to him as to what he should do!

Learning to Model with Algebraic Symbols: The Inductive Support Strategy

To reach the goal of creating improved instructional supports to help students learn to be successful on assessments like the Photographer Career problem in Table 1, we began to research issues of mathematical modeling and the underlying competencies required. In particular, we focused on studying what students know and do not know about symbolic modeling. Skills for symbolic modeling are important because they are not currently automated, they are the entry point to using today's powerful calculation tools (e.g., graphic and symbolic calculators, spreadsheets, programming), and they are particularly difficult skills for students to acquire. We began to experiment with different approaches to helping students learn to model with algebraic symbols (Koedinger & Anderson, 1998).

Table 2 shows a problem from a popular algebra textbook (Forester, 1984). Hadley had been using a similar problem format, but put a particular emphasis on problem contexts that would be more authentic to students and that contained real data. Forester intended this problem format to illustrate the nature of an algebraic variable as truly varying, in contrast to traditional algebra word problems (e.g., leave out questions 1-3 in Table 2) in which there is an unknown constant, but no variable. As I began to observe and analyze student thinking on such problems, I formed the simple hypothesis that having students answer the concrete "result-unknown" questions 2 and 3 before answering the symbolization question 1 might facilitate student learning, particularly of the symbolization process.

Table 2. Textbook problem with different question types.

Drane & Route Plumbing Co. charges \$42 per hour plus \$35 for the service call.	
1. Create a variable for the number of hours the company works. Then, write an expression for the number of dollars you must pay them.	Symbolization Question
2. How much you would pay for a 3 hour service call? 3. What will the bill be for 4.5 hours?	Result-Unknown Questions
4. Find the number of hours worked when you know the bill came out to \$140.	Start-Unknown Question

This hypothesis followed from my prior cognitive science research on the importance of inductive experiences in the evolution of geometry knowledge (Koedinger & Anderson, 1990). It also followed from observations of students who could successfully solve concrete result-unknown questions, like 2 and 3, but could not produce the corresponding algebraic sentence to answer question 1 (Koedinger & Anderson, 1998). Such students already had effective performance knowledge (or production rules) for comprehending the English problem statement, for extracting the relevant quantities and quantitative relations, and for a producing a numerical answer. However, production rules for "writing algebra", that is, for taking a problem understanding and expressing it in algebraic symbols, were either weak or missing (cf., Heffernan & Koedinger, 1997; 1998). One such production rule is illustrated below:

> If the goal is express a quantity Q1 in algebraic symbols
> and Q1 is result of combining Q2 and Q3 with operator Op
> and the expression for Q2 is Expr2
> and the expression for Q3 is Expr3
> Then set a goal to write: Expr2 Op Expr3
> set a goal to check for correct order of operations

This production rule characterizes *tacit* performance knowledge for composing algebraic "embedded clauses" for a quantity, like 42h + 35 for the total bill (Q1), from knowledge of simple clauses, like 42h for the hourly charge (Q2) and 35 for the service charge (Q3).

Why should learners solve result-unknown questions (questions 2 and 3 in Table 2) before attempting to symbolize (question 1)?[4][5] It is easier for students to step through the arithmetic operations in a problem with concrete numbers than to write the corresponding algebraic sentence. As they do so, declarative memory traces are stored that characterize the problem's quantitative structure (e.g., 42 times 3 is 126 and 126 plus 35 is 161). These traces are analogous to the structure of the algebraic sentence students need to produce (e.g., 42h and 42h+35). In attempting this more difficult step of symbolizing, students' brains perform analogical problem solving processes (Anderson & Lebiere, 1998) that make use of these concrete traces to guide the writing of an algebraic sentence. As a side effect of these processes the brain induces new production rules, like the one illustrated above, for writing algebraic sentences.

I call this learning strategy of making use of concrete modes of thinking (numeric instances in this case) to help induce abstract modes of thinking (writing algebraic sentences) the *inductive support strategy*.

We used an early version of PAT to implement an experimental learning study in which we compared a control "Textbook" condition to an experimental "Inductive Support" condition. In the Textbook condition students solved problems in the Forester textbook format. In the Inductive Support condition, students solved the same problems but with the questions rearranged so that the concrete result-unknowns (#2 and #3) appeared before the symbolization question (#1). Figure 1 shows the results of that study. Students in the Inductive Support group learned significantly more from pre-test to post-test than students in the Textbook group (Koedinger & Anderson, 1998). This inductive support effect was successfully replicated by Gluck (1999), who also collected eye movement data that provides an interesting and more direct window into the changes in the thinking process that result from inductive support.

---

[4] The following explanation makes use of aspects of the ACT-R theory, in particular, the declarative and procedural memories and the analogical learning process that operates between them (Anderson & Lebiere, 1998).

[5] The original "PUMP" Algebra curriculum stood for Pittsburgh Urban Mathematics Project and expressed the goal of making algebra a "pump" not a "filter" to further academic and workplace success.

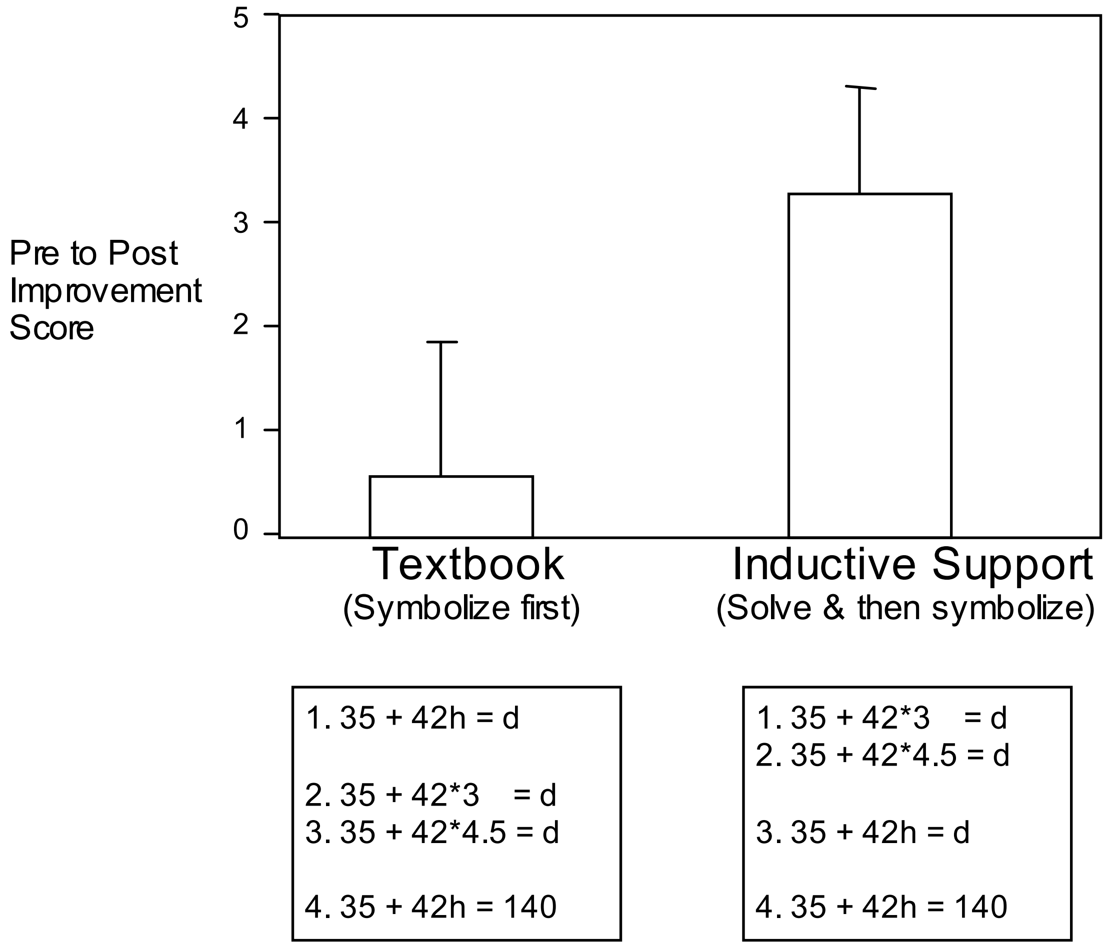

Figure 1. Improving algebraic modeling by bridging off students' existing knowledge.

A crucial point to emphasize here is that students not only need to learn mathematical concepts, but they must also develop a mastery or fluency with mathematical modeling languages such as algebraic symbols, programming languages, statistics notation, and dynamic geometry tools. This experiment illustrates one effective way to assist students in developing mathematical language fluency. The inductive support strategy suggests that we create instructional activities that help students bridge from existing, more concrete modes of thinking (e.g., situations and numbers) and more familiar languages (e.g., English) to more abstract and powerful modes of thinking and languages (e.g., algebra and algebraic symbols). The inductive support strategy is a demonstrated realization of "progressive formalization" (Bransford, Brown, & Cocking, 1999, p. 125).

Description of PAT: A Cognitive Tutor for Practical Algebra

As part of the development of the Pump Algebra Tutor (PAT), Pittsburgh teachers wrote problem situations, like the Photography problem discussed above, intended to be personally or culturally relevant to students. Some problem situations are of potential general interest (e.g., the decline of the condor population), while others are somewhat more specific to Pittsburgh 9th graders (e.g., making money shoveling snow or declining population in Pittsburgh after the demise of the steel mills). These problems were added to PAT using an intelligent problem authoring system in which teachers type the problem description and enter an example solution (Ritter, Anderson, Cytrynowitz, & Medvedeva, 1998). This authoring system has an incomplete and imperfect model for reading English text but can make reasonable guesses about how quantities and relations in the entered solution map to phrases in the text. The author can then correct or edit these guesses. The connections formed between elements of the problem and elements of the solution are the basis for the automated feedback and hints the tutor can provide as needed.

Figure 2. In the Pump Algebra Tutor (PAT), students create tabular (lower-left), graphical (upper-right), and symbolic (lower-center) models of problem situations (upper-left) with as-needed assistance (middle-right) and dynamic assessment (lower-right) from the Cognitive Tutor.

Students begin work on PAT problem situations by reading a description of the situation and a number of questions about it. They investigate the situation by representing it in tables, graphs, and symbols and by using these representations to answer the questions. Helping students understand and use multiple representations is a major focus of PAT (cf., Hall, Kilber, Wenger, & Truxaw, 1989; Janvier, 1987; Koedinger & Tabachneck, 1994; Tabachneck, Koedinger, & Nathan, 1994).

In Figure 2, the PAT screen shows a student's partial solution for a problem. This problem appears in the later stages of the curriculum after students have acquired some expertise with constructing and using graphs and tables for single linear equations. The top-left corner of the tutor screen provides a description of the problem situation. This problem involves comparison of costs between two rental companies, Hertz and Avis, that charge different rates for renting large trucks. Students investigate the problem situation using multiple representations and computer-based tools, including a spreadsheet, grapher, and symbolic calculator (in Figure 2 these are the Worksheet, Grapher and Equation Solver windows, respectively).

Students construct the Worksheet (lower-left of Figure 2) by identifying the relevant quantities in the situation, labeling the quantities (at the top of the columns), identifying appropriate units (first row), answering the questions in the problem description (numbered rows), and entering algebraic expressions (in the “Formula” row). The Formula row is at the bottom of the table in early lessons to facilitate use of the Inductive Support strategy, but moves to the top in later lessons. Like a spreadsheet, in later lessons the formula automatically generates a dependent (y) variable value when a value of the independent variable (x) has been entered.

Students construct the graph of the problem situation (upper-right of Figure 2) by labeling axes, setting appropriate bounds and scale, graphing the lines, and identifying the point of intersection. The Equation Solver (lower-center) can be used at any time to help fill in the spreadsheet or identify points of intersection in the graph. The student can use these representations to reason about real-world concerns, such as deciding when it becomes better to rent from one company rather than another.

The steps in table, graph, and equation construction and use emphasize multiple perspectives on quantities and their relationships through the description to these in multiple forms. Understanding of quantity emerges when students actively engage in making connections between, and practicing correct use of, these multiple representational forms (cf., Hall, Kilber, Wenger, & Truxaw, 1989; Janvier, 1987; Koedinger & Tabachneck, 1994; Tabachneck, Koedinger, & Nathan, 1994). The PAT interface representations and problem structure encourage these connections, and the PAT tutor component assists students, as needed, in appropriate use of these representations. By supporting students in thinking about quantity labels, PAT assists students in making connections to more concrete situational and verbal knowledge. By thinking about units, students are making connections to more abstract verbal knowledge. By thinking about numerical relationships, students make connections to arithmetic knowledge and, as we have shown in the Inductive Support studies (Koedinger & Anderson, 1998; Gluck, 1999), this thinking process facilitates the difficult developmental transition to fluent use of algebraic symbols. In addition to the aspects of PAT’s Worksheet noted above, students are also assisted in graph and equation construction and use. The problem-solving activities in the PAT curriculum are structured and ordered to engage students in increasingly sophisticated uses of and translations among a variety of representations. The goal is for students to achieve “representational fluency” through developmentally appropriate connections and sufficient practice.

At all times the tutor is monitoring student activities and providing feedback and assistance as needed. This provision of timely feedback is one way in which Cognitive Tutor’s individualize instruction. Like good human tutors, PAT's tutorial interactions with students are brief and focused on individual students' particular learning needs as they arise in the context of problem solving. When a student is having trouble, PAT does not immediately provide correct answers or even detailed advice. Instead, PAT tries to maximize student opportunities to discover or reinforce appropriate concepts or skills on their own. PAT provides two kinds of assistance: just-in-time feedback on problem solving steps taken and on-demand hints on next steps needed.

Like human tutors, PAT's feedback on student errors avoids direct verbal statements like "wrong" and instead uses non-verbal cues. PAT cannot raise its eyebrows, but it does provide non-verbal feedback by "flagging" errors using a font highlight such as outline text or color. As Schofield et al. (1990), Wertheimer (1990), and many of our current teachers have observed, students do not feel the same social stigma when making errors on the computer as they do when making errors in class. Commonly occurring slips or misconceptions are recognized by "buggy" production rules and specific advice can be provided that may explain what is wrong with the current step or hint toward an appropriate correct step. Examples of student errors that are recognized by PAT's buggy productions include leaving out the initial value (intercept) in a formula, leaving out parentheses or otherwise violating order of operations, confusing the dependent and independent variable in a table, graph or formula, and many others.

The provision of timely feedback is a critical feature of Cognitive Tutors that leads to substantial cognitive and motivational benefits. In a parametric study with the LISP tutor, Corbett and Anderson (1991) compared levels of timing of feedback. Students receiving immediate feedback (after each problem-solving step) learned significantly faster than students receiving delayed feedback (at the end of the problem).. In addition to cognitive benefits, there also appear to be motivational benefits of timely feedback. Much like the motivational attraction of video games, students know right away that they are making progress and get satisfaction from this success at a challenging task.

In addition to timely feedback, a second way PAT individualizes instruction is by providing context-sensitive hints. Using the model tracing mechanism described above, PAT is always following each student's particular approach to a problem. At any point in constructing a solution, a student can request a hint and PAT will provide one that is sensitive to what the student has done up to that point. The tutor chooses a hint message by using the production system to identify the set of possible next strategic decisions and ultimate external actions. It chooses among these based on the student's current focus of activity, what tool and interface objects the student has selected, the overall status of the student's solution, and internal knowledge of relative utility of alternative strategies. Successive levels of assistance are provided in order to maximize the students' opportunities to construct or generate knowledge on their own. This approach to learning through assisted performance bears close resemblance to cognitive apprenticeship (Collins, Brown & Newman, 1989) and procedural facilitation (Vygotsky, 1978).

The "Message" window in Figure 2 shows the result of a student help request. At any stage in problem solving, the tutor can provide assistance on whatever interface element the student has selected (e.g., a cell in the Worksheet, a point in the Grapher, an equation in the Equation Solver). Based on the tutor's representation of alternative strategies and solution contingencies, the tutor may suggest an alternative selection in some situations.

For instance, it will recommend that students complete a row in the Worksheet (an x-y pair) before graphing the corresponding point in the Grapher.

In the situation shown in Figure 2, the student has selected the Worksheet cell for question 4 in the column she labeled “MILES DRIVEN”. Question 4 reads “If we have budgeted a total of $1000 to rent this truck, how many miles can we drive it if we rent it from Avis?". To process the hint request, the tutor’s cognitive model is run to see what different production rule paths are potentially relevant at this point in the problem. Possibilities include the following:

1. In the Grapher, graph one or the other of the lines corresponding to either of expressions in the FORMULA row of the Worksheet.
2. In the Worksheet, enter the given value $1000 in the “COST OF RENTING FROM AVIS” column in row 4.
3. In the Equation Solver, enter an equation, like “0.13x + 585 = 1000”, that can be solved to find the MILES DRIVEN.
4. In the Worksheet, enter the value or an expression for computing the value (e.g., “(1000 – 585) / .13”) of MILES DRIVEN in the selected cell.

In tracing student problem solving, the tutor is flexible and would follow the student no matter which of these paths she pursued. In this case, though, the student has requested a hint and the tutor must decide which of these paths to hint toward. This decision is constrained by both flow and pedagogical concerns. To preserve the flow, if possible, a path is chosen that is relevant to the window in which the student is working. Thus, path #1 is not chosen since it is relevant to the Grapher not the Worksheet. Also to preserve flow, if possible, a path is chosen that is relevant to the currently selected interface object, in this example, the cell in column 1, row 4. Thus, path #2 is not chosen since it is relevant to the cell in column 3, row 4. Besides flow concerns, hint selection is also driven by content-specific pedagogical concerns (see Shulman, 1987 for discussions of the importance of pedagogical content knowledge). Both paths #3 and #4 are relevant to the student’s selected cell. Path #3 is an equation solving strategy, which is the typical textbook approach to such problems. Path #4 recommends an informal “unwind” strategy and is chosen for pedagogical-content reasons described below. In general, the flow constraints on hint selection are implemented within the Cognitive Tutor architecture and apply in all tutors, whereas the pedagogical constraints are implemented in content-specific production rules and apply to specific situations within a tutor.

To the surprise of many teachers (Nathan, Koedinger, & Tabachneck, 2000), students are able to solve problems like question 4 without an equation (Koedinger & MacLaren, 1997). Students use informal strategies, like guess-and-test and unwind, that they can perform more effectively and efficiently than equation solving on such problems (Koedinger & Alibali, 1999; Nhouyvanisvong, 1999). Thus, for problems like question 4, PAT hints toward the informal unwind strategy because it better builds off students’ prior verbal knowledge of informal strategies. PAT also includes problems, like question 5, that thwart the unwind strategy and thus motivate the need for and practice the use of equation solving. I now provide examples of hint sequences associated with questions 4 and 5.

Hints along a chosen path begin with general assistance, and the tutor provides incrementally more specific assistance only as demanded by the student. In this example, the initial hint focuses the student's attention on the fact that the usual given and goal variables are reversed and therefore the need to "unwind" the arithmetic procedure: "To find the distance driven, instead of the cost of renting from avis, unwind your calculation. Do the reverse of what you would normally do." If the student asks for more help, the tutor provides a more specific hint toward how to unwind: "To find the distance driven, instead of the cost of renting from avis, take the value you are given for the cost of renting from avis, first subtract numbers that you would normally add and then divide by numbers you would normally multiply." This hint provides a general strategy for how to unwind expressions like the one in this problem. If student requests it, the next and final hint provides the specific expression to which the unwind strategy should be applied: "To calculate the distance driven, try unwinding as a way of solving the expression 1000 equals 0.13 times the distance driven plus 585." Sometimes the final hint in a sequence can be quite specific, for instance, "Subtract 585 from 1000 and then divide by 0.13" or even "Type 3192.3". Such detailed hints are like the examples provided in textbooks illustrating a new idea. The difference is that, in PAT, these examples come in the process of problem solving when students are better able to understand and make use of the example. Also, in such cases PAT will give students later opportunities to perform this skill on their own.

In the case above, the last hint is still fairly general. In some contexts, like this one, teachers have asked to have the hint stay relatively vague so that they are aware and can intervene if a student gets really stuck. This case is also an example of how our basic cognitive research, in this case on students' invented informal strategies (Koedinger & MacLaren, 1997; Nathan, Koedinger, & Tabachneck, 2000), has influenced teacher practices and the content of our teacher professional development. The vocabulary and associated strategy of "unwinding" an arithmetic procedure to find an unknown is emphasized in our teacher professional development workshops. However, like students, teachers also learn by doing, and thus it helps to have this idea reinforced as teachers observe the tutor and interactions with students in the computer lab.

Although the hint messages recommend the unwind strategy for questions like 3 and 4, the students are free to choose whichever path they like. The Equation Solver window (lower-center in Figure 2) illustrates how the equation solving strategy (path #3) can be performed to answer question 3. The student enters her own equation, "850 = 0.21X + 525" and solves it by indicating standard algebraic manipulations. As is the case for all tools in PAT, students can receive feedback and hints in the Equation Solver that are sensitive to their chosen strategy and current state of the solution. If the student were to ask for hints prior to the second step shown in Figure 2, the successive hints would be the following:

- What can you do to both sides to get x by itself?
- To change 0.21x to x, divide by 0.21.

- Divide both sides by 0.21.

In contrast, a second student might have started differently, by dividing both sides of the equation by 0.21 rather than subtracting 525 from both sides. If requested, a hint on the next step in this case would follow through on this student's chosen path[6] and be different from the hint sequence shown above.

Question 5 asks the student to find out the crossover point where the cost of one option (Hertz in this case) catches up with the cost of another (Avis). Alternative strategies for finding this point include equation solving or using the Grapher tool (upper right in Figure 2) to graph the lines to find the intersection. In this case if requested, PAT hints toward equation solving: "Given that the expression for the cost of renting from hertz and the cost of renting from avis are equal, write an equation and solve it to find the distance driven." If needed, the student is further hinted toward setting up and solving the equation "0.21x + 525 = 0.13x + 585".

By keeping students engaged in successful problem solving, PAT's feedback and hint messages reduce student frustration and provide for a valuable sense of accomplishment. In addition to these functions of model tracing, PAT provides learning support through *knowledge tracing*. Results of knowledge tracing are shown to student and teacher in the Skillometer window (labeled "Lesson Ten" in the bottom right of Figure 2). By monitoring a student's acquisition of problem solving skills through knowledge tracing, the tutor can identify individual areas of difficulty (Corbett, Anderson, & O'Brien, 1995) and present problems targeting specific skills that the student has not yet mastered. For example, a student who was skilled in writing equations with positive slopes and intercepts, but had difficulty with negative slope equations would be assigned problems involving negative slopes. Knowledge tracing can also be used for "self-pacing", that is, the promotion of students through sections of the curriculum based on their mastery of the skills in that section.

The activities in PAT are organized hierarchically so that related problem situations that draw on a core set of skills are organized into "sections" and then sections that use the same set of notations, tools, and broader concepts are organized into "lessons". For instance, the PAT curriculum used in schools in the 1997-98 school year included 22 lessons, each of which contained about 4 sections on average, and each section contained about 5 required problems and about 5 additional problems. Initially, students explore common situations involving positive quantities, mostly whole numbers and some simple fractions and decimals, and represent these situations mathematically in tables, expressions and graphs. As the year progresses more complex situations are analyzed, involving negative quantities, and four-quadrant graphing is introduced. Similarly, as situations increase in complexity, more sophisticated equation solving and graphing techniques are introduced to enable students to better find solutions. Systems of linear equations and quadratics are developed through the introduction of situations in which

---

[6] This hint would recommend to either distribute or to subtract 2500 (i.e., 525/.21) from both sides depending on the option settings in the Equation Solver, which would determine whether the second step is displayed as "850/0.21 = (0.21x + 525)/0.21" or "4047.619 = x + 2500".

they naturally occur, for example, modeling and comparing the price structures of two rival companies that make custom T-shirts. Modeling vertical motion and area situations provide contexts for introducing and using quadratic functions.

Unlike the short "two minute problems" of most math software and traditional classroom instruction (Schoenfeld, 1989), the PAT curriculum includes mini-projects, like the one in Figure 2, which may last 20-30 minutes. Shorter practice exercises (e.g., equation solving exercises) are also interspersed to zoom in on particularly difficult and important skills. Over the weeks of the course, students alternate between playing the larger game of mathematical problem solving and engaging in decontextualized practice of the more difficult component skills, much like: play tennis, practice backhand, play tennis, practice serve, play tennis …. The emphasis is on using project activities first to motivate the need for particular kinds of practice and then, after practice, put them back to use in context.

Classroom Context of PAT Use

The majority of schools using PAT also use the Pump curriculum and text materials. The typical procedure is to spend 2 days a week in the computer lab using PAT and 3 days a week in the regular classroom. In the classroom, learning is active, student-centered, and focused primarily on learning by doing. Teachers spend less time in whole-group lecture and more time in individual and cooperative problem solving and learning. Teachers are often playing a facilitator role, but also lead whole-group discussions to highlight student discoveries or to introduce new concepts or procedures that, ideally, respond to student needs that have emerged from prior activities.

In the classroom, students often work together in collaborative groups to solve problems similar to those presented by the tutor. Teams construct their solutions by making tables, expressions, equations, and graphs that they then use to answer questions and make interpretations and predictions. Teachers play a key role in helping students to make connections between the computer tools and paper and pencil techniques and to see how the general concepts and skills for representation construction and interpretation are the same on paper and on the computer. Literacy is stressed by requiring students to answer all questions in complete sentences, to write reports and to give presentations of their findings to their peers.

The Pump curriculum uses alternative forms of assessment including performance tasks, long term projects, student portfolios, and journal writing. From the first day all answers must be written in complete sentences to be accepted. At the end of each quarter students are given a performance assessment, like the excerpt shown in Table 1. At the end of each semester teachers grade these assessments in a group scoring conference. In the span of an intense afternoon, teachers inspect student solutions, construct a scoring rubric, and double-grade all the student papers. Because all teachers score papers from every other teacher's class as well as their own, they come to have a better understanding of the objectives of the curriculum, what students know and do not know, and in what ways other teachers' students may differ.

Replicated Field Study Results

An important, sometimes hard-learned, lesson of classroom use of educational technology is that to be effective in improving student learning, educational technology must be closely integrated with curriculum goals and other learning resources such as texts and teacher practices. Research with intelligent tutors (Koedinger & Anderson, 1993) and other educational technologies, like LOGO (Lehrer, Randle, Sancilio, 1989; Klahr & Carver, 1988), has demonstrated the importance of curriculum integration and teacher support. We have emphasized these contextual factors throughout the development of PAT. The benefits of cognitive tutors, as with individualized just-in-time assistance in the context of rich problem solving activities, can be reduced or masked if the social context of classroom use is not addressed (Koedinger & Anderson, 1993). However, if such factors are addressed, use of cognitive tutors in the classroom can have dramatic impact on student learning and achievement. We have demonstrated this impact in experimental field studies in city schools in Pittsburgh and Milwaukee, replicated over 3 difference school years. The assessments used in these field studies targeted both 1) higher order conceptual achievement as measured by performance assessments of problem solving and representation use and 2) basic skills achievement as measured by standardized test items, for instance, from the math SAT. In comparison with traditional algebra classes at the same and similar schools, we have found that students using PAT and the Pump curriculum perform 15-25% better than control classes on standardized test items and 50-100% better on problem solving & representation use (Koedinger, Anderson, Hadley, & Mark, 1997; Corbett, Koedinger, & Anderson, 1999).

Following the observations of Schofield, Evans-Rhodes, & Huber (1990) and Wertheimer (1990), we have also observed the impact of the use of PAT on changes in classroom social and motivational processes (Corbett, Koedinger, & Anderson, 1999). Visitors to our classrooms often comment on how engaged students are. PAT may enhance student motivation for a number of different reasons. First, authentic problem situations make mathematics more interesting, sensible, or relevant. Second, students on the average would rather be doing than listening, and the incremental achievement and feedback within PAT problems provide a video-game-like appeal. Third, the safety net provided by the tutor reduces potential for frustration and provides assistance on errors without social stigma. Finally, the longer-term achievement of mastering the mathematics is empowering.

In the computer lab, teachers are glad to essentially have a teacher's aid for every student and thus be freed to be facilitators and provide more one-on-one instruction with individual students. This experience is eye opening for many teachers who may see new aspects of student thinking and feel the advantages of greater student-centered learning by doing.

Cognitive Tutors as Teacher Change Agents

How and why does the use of Cognitive Tutors facilitate the spread of effective teaching principles and practices and the institution of curriculum reform? Is there something special about Cognitive Tutors that makes such spread more likely than from alternative educational technologies like books, traditional CAI, simulations, or representational tools? Unlike other educational technologies, Cognitive Tutors have a running model of student thinking and of adaptive student-centered instruction. Thus, the system provides an active "living example" of research-based principles and practices. Much like teachers use textbooks to guide their teaching practices, teachers often borrow from Cognitive Tutor problems, representational tools, feedback and hint strategies and incorporate them in their teaching practices. However, there are crucial differences between the instructional model provided by textbooks and that provided by Cognitive Tutors. Whereas examples of instruction in textbooks are static and non-interactive, examples of instruction in Cognitive Tutors are dynamic and can be observed in live interaction with students.

By serving as a teacher's aid for each student in the classroom, Cognitive Tutors free teachers to observe individual student thinking more often and more closely and to reflect on their instructional practices in this context. Student responses in such close interactions provide teachers with immediate and detailed feedback on the effectiveness of the tutor's or their own practices. Teachers can thus adjust their practices accordingly (as well as give feedback, as they often do, on how to improve the Cognitive Tutor).

We had the opportunity to observe PAT-inspired changes in curricula and teacher practices over multiple semesters through a Department of Education, FIPSE project in which we adapted PAT for use in college-level developmental math courses (Koedinger & Sueker, 1996). These PAT-inspired changes were accompanied by significant quantitative improvements in student learning. The general methodology we employed, a multi-semester "design experiment" (Brown, 1992; Collins, 1992), involved an iterative process of course design, qualitative and quantitative observation and evaluation, and course redesign.

PAT was initially used at the University of Pittsburgh in the fall semester of 1995 as an add-on to a traditional "Intermediate Algebra" course. As shown in the left of Figure 3, this supplementary use led to significant improvement in students' problem solving abilities over a traditional course without PAT (45% vs. 30% average correct as measured using a rubric scoring scheme of an assessment much like the one in Table 1).

In the spring of 1996, University of Pittsburgh instructors used the PAT authoring tool to create new problems better adapted to the particular needs and interests of their students. More interestingly, changes were not limited to the software. Instructors began to use PAT problems in their regular classes and began to experiment with more student-centered learn by doing outside of the computer lab. The consequence of these new practices was an increase in end-of-course achievement beyond that found in the experimental classes in the fall (65% vs. 45%).

Over the next two semesters, these practices evolved, and PAT use became better integrated with regular classroom instruction, resulting in further modest increases in student learning over past semesters (68%, 71%). In preparation for the Spring of 1997, the University of Pittsburgh, led by Lora Shapiro, made a decision to fully reform their Intermediate Algebra course to more generally target "quantitative literacy". Lecturing was deemphasized in favor of more “workshop” time in which students worked on projects in collaborative groups. The typical college level approach consisting of a lot of instructor-centered lecture time and a little student-centered recitation time, was replaced with a lot of student-centered work and a little whole-group reflection and targeted lectures. The consequence of these changes, in the first semester of their implementation, was another increase in end-of-course student performance (79%).

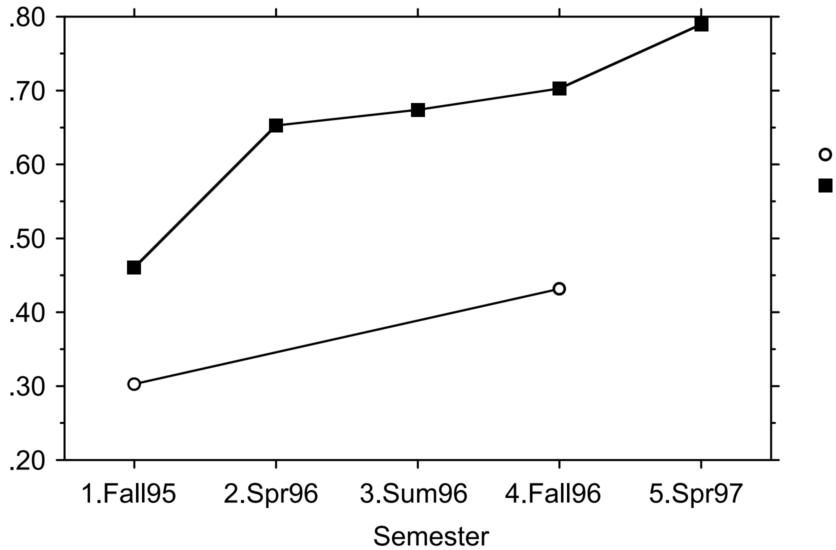

Figure 3. Increasing PAT-inspired reforms lead to increasing scores across five semesters.

Conclusions

Effective technologies for learning and doing mathematics should be based on sound *cognitive theory*, be *empirically tested against alternatives*, and be primarily addressed at *mathematics as a modeling language*. I have argued for and illustrated these points in the context of Cognitive Tutors generally and in particular, the Pump Algebra Tutor (PAT). PAT is based on the ACT theory of cognition and a production rule model of student problem solving and mathematical modeling.

PAT has been subject to empirical tests both in the laboratory and in the field. In an early laboratory study with PAT, we contrasted a research-inspired “inductive support” strategy with an existing textbook strategy. This study demonstrated improved student learning, particularly of difficult symbolic modeling skills.

In field studies of the use of PAT and the Pump curriculum, we demonstrated that the combination of the two leads to dramatic increases in student learning on both standardized test items (15-25% better than control classes) and on new standards-oriented assessments of problem solving and representation use (50-100% better than control classes). The focus of PAT and the Pump curriculum is on developing student competence in creating mathematical models of problem situations rather than on answers to isolated questions. By developing mathematical modeling skills, students can construct a deeper understanding of problem situations such that multiple, unanticipated questions can be addressed and answered. Better mathematical understanding and learning result from such multi-representational approaches.

Cognitive Tutors, like PAT, have the potential not only to dramatically increase student achievement, but also to serve a professional development function for teachers. Because of the underlying cognitive model and associated pedagogical strategies, Cognitive Tutors can provide a living example of effective instructional practices. Teachers working in the computer lab have more time to observe student performance on thought-revealing problems and to observe learn-by-doing instruction in action. In this way, Cognitive Tutors can carry research-based practices into the classroom and serve as change agents for professional development.

In addition to the Cognitive Tutor Algebra course, our PACT Center has developed Cognitive Tutor courses for high school Geometry (e.g., Aleven, Koedinger, & Cross, 1999) and Algebra II (e.g., Corbett, McLaughlin, & Scarpinatto, in press). These three courses are being marketed by our spin-off company, Carnegie Learning. Recently, Carnegie Learning has funded a three-year PACT Center project for research and development of Cognitive Tutor courses for middle school mathematics. Through the combined efforts of the PACT Center and Carnegie Learning, Cognitive Tutors are beginning to reshape the mathematics classroom, the way teachers teach, and what and how students learn.

Acknowledgements

This chapter reviews research that was supported by multiple sources. For support of research, development, and evaluation of the Algebra Cognitive Tutor, thanks to the National Science Foundation, the Department of Education's Fund for the Improvement of Post-Secondary Education (FIPSE), the DARPA Computer Aided Educational and Training Initiative, and the DoDEA Presidential Technology Initiative. For support of related cognitive science research on early algebra, thanks to the McDonnell Foundation's Cognitive Studies for Educational Practice.

References

Aleven, V., Koedinger, K. R., & Cross, K. (1999). Tutoring answer-explanation fosters learning with understanding. In Lajoie, S. P. & Vivet, M. (Eds.) *Artificial Intelligence in Education, Open Learning Environments: New Computational Technologies to Support Learning, Exploration, and Collaboration, Proceedings of AIED-99*, (pp. 199-206). Amsterdam: IOS Press.

Anderson, J.R., Conrad, F.G. & Corbett, A.T. (1989). Skill acquisition and the LISP Tutor. *Cognitive Science,* 13, 467-505.

Anderson, J. R., Corbett, A. T., Koedinger, K. R., & Pelletier, R. (1995). Cognitive tutors: Lessons learned. *The Journal of the Learning Sciences*, *4 (2)*, 167-207.

Anderson, J. R., & Lebiere, C. (1998). *The Atomic Components of Thought.* Hillsdale, NJ: Erlbaum.

Berry, D. C. & Dienes, Z. (1993). *Implicit Learning: Theoretical and Empirical Issues*. Hove, UK: Erlbaum.

Beyer & Holtzblatt (1998). *Contextual Design: Defining Customer-Centered Systems*. San Francisco, CA: Morgan Kaufman.

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*, 3-16.

Bransford, J. D., Brown, A. L, & Cocking, R. R. (1999). *How People Learn: Brain, Mind, Experience, and School*. Committee on Developments in the Science of Learning Commission on Behavioral and Social Sciences and Education. National Research Council. Washington, D.C.: National Academy Press.

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

Cheng, P. W., & Holyoak, K. J. (1985). Pragmatic reasoning schemas. *Cognitive Psychology*, *17*, 391-416.

Collins, A., Brown, J. S., & Newman, S.E. (1989). Cognitive apprenticeship: Teaching the crafts of reading, writing, and mathematics. In L. B. Resnick (Ed.), *Knowing, learning, and instruction: Essays in honor of Robert Glaser.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Collins, A. (1992). Toward a design science of education. In E. Scanlon & T. O'Shea (Eds.), *New Directions in Educational Technology*, (pp. 15-22). Springer-Verlag: Berlin.

Corbett, A. T. & Anderson, J. R. (1991). Feedback control and learning to program with the CMU LISP tutor. Paper presented at the annual meeting of the American Educational Research Association, Chicago, IL.

Corbett, A.T. & Anderson, J.R. (1995). Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*, 4, 253-278.

Corbett, A. T., Anderson, J. R. & O'Brien, A. T. (1995). Student modeling in the ACT programming tutor. In P. Nichols, S. Chipman, & R. Brennan (Eds.), *Cognitively diagnostic assessment* (pp. 19-41). Hillsdale, NJ: Erlbaum.

Corbett, A. T., McLaughlin, M. and Scarpinatto, K. C. (in press). Modeling student knowledge: Cognitive Tutors in High School and College. *User Modeling and User-Adapted Interaction*.

Corbett, A., Koedinger, K. R., & Anderson, J. R. (1999). Intelligent computer tutors: Out of the research lab and into the classroom. Paper prepared for the annual meeting of the *American Educational Research Association*, Montreal, Canada.

Dienes, Z. & Perner, J. (1999). A theory of implicit and explicit knowledge. *Behavioral and Brain Sciences, 22*, (5).

Forester, P.A. (1984). *Algebra I: Expressions, Equations, and Applications*. Menlo Park, CA: Addison-Wesley.

Gluck, K. (1999). Eye movements and algebra tutoring. Doctoral dissertation. Psychology Department, Carnegie Melon University.

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

Heffernan, N. & Koedinger, K. R. (1997). The composition effect in symbolizing: The role of symbol production vs. text comprehension. In *Proceedings of the Seventeenth Annual Conference of the Cognitive Science Society.* Hillsdale, NJ: Erlbaum.

Heffernan, N. & Koedinger, K. R. (1998). A developmental model for algebra symbolization: The results of a difficulty factors assessment. In *Proceedings of the Twentieth Annual Conference of the Cognitive Science Society*, (pp. 484-489). Hillsdale,NJ: Erlbaum.

Janvier, C. (1987). *Problems of representation in the teaching and learning of mathematics*. Hillsdale: Lawrence Erlbaum Associates.

Klahr, D., & Carver, S. M. (1988). Cognitive objectives in a LOGO debugging curriculum: Instruction, learning, and transfer. *Cognitive Psychology, 20*, 362-404.

Koedinger, K. R. & Alibali, M. W. (1999). A developmental model of algebra problem solving: Trade-offs between grounded and abstract representations. Paper prepared for the annual meeting of the *American Educational Research Association*, Montreal, Canada.

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

Koedinger, K. R. & Anderson, J. R. (1993). Effective use of intelligent software in high school math classrooms. In Proceedings of the *World Conference on Artificial Intelligence in Education*, (pp. 241-248). Charlottesville, VA: Association for the Advancement of Computing in Education.

Koedinger, K. R., & Anderson, J. R. (1998). Illustrating principled design: The early evolution of a cognitive tutor for algebra symbolization. In *Interactive Learning Environments*, 5, 161-180.

Koedinger, K. R., Anderson, J. R., Hadley, W. H., & Mark, M. A. (1997). Intelligent tutoring goes to school in the big city. *International Journal of Artificial Intelligence in Education, 8*, 30-43.

Koedinger, K. R., & MacLaren, B. A. (1997). Implicit strategies and errors in an improved model of early algebra problem solving. In *Proceedings of the Seventeenth Annual Conference of the Cognitive Science Society.* Hillsdale, NJ: Erlbaum.

Koedinger, K. R., & Sueker, E. L. F. (1996). PAT goes to college: Evaluating a cognitive tutor for developmental mathematics. In *Proceedings of the Second International Conference on the Learning Sciences*, (pp. 180-187). Charlottesville, VA: Association for the Advancement of Computing in Education.

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

Kulik, C. C. & Kulik, J. A. (1991). Effectiveness of computer-based instruction: An updated analysis. *Computers in Human Behavior, Vol. 7*, pp. 75-95.

Lehrer, R., Randle, L., & Sancilio, L. (1989). Learning preproof geometry with LOGO. *Cognition and Instruction*, *6*, 159-184.

Matz, M. (1982). Towards a process model for high school algebra errors. In D. Sleeman and J. S. Brown (Eds.) *Intelligent tutoring systems.* New York: Academic Press.

Miller, C. S., Lehman, J. F., & Koedinger, K. R. (1999). Goals and learning in microworlds. *Cognitive Science, 23*, (3), 305-336.

Nathan, M. J., Koedinger, K.R., & Tabachneck, H. J. M. (2000). Teachers' and researchers' beliefs of early algebra development. *Journal of Mathematics Education Research*.

National Council of Teachers of Mathematics (1989). *Curriculum and Evaluation Standards for School Mathematics*. Reston, VA: The Council.

Newell, A. (1990). Unified Theories of Cognition. Cambridge, MA: Harvard University Press.

Nhouyvanisong, A. (1999). Enhancing mathematical competence and understanding: Using open-ended problems and informal strategies. Doctoral Dissertation. Psychology Department, Carnegie Mellon University.

Pane, J.F., Corbett, A.T., and John, B.E. (1996). Assessing Dynamics in Computer-Based Instruction. In *Proceedings of ACM CHI'96 Conference on Human Factors in Computing Systems*. Vancouver, pp. 197-204.

Ritter, S., Anderson, J. R., Cytrynowitz, M., & Medvedeva, O. (1998). Authoring content in the PAT algebra tutor. *Journal of Interactive Media in Education, 2*.

Schoenfeld, A. (1989). Explorations of students' mathematical beliefs and behavior, *Journal for Research in Mathematics Education*, *20:1*, 338-355.

Shulman, L. (1987). Knowledge and teaching: Foundations of the new reform. *Harvard Educational Review, 57*, 1-22.

Singley, M. K., & Anderson, J. R. (1989). *Transfer of Cognitive Skill*. Hillsdale, NJ: Erlbaum.

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 Science Computer Review, 8:1*, 24-41.

Tabachneck, H. J. M., Koedinger, K. R., & Nathan, M. J. (1994). Toward a theoretical account of strategy use and sense-making in mathematics problem solving. In *Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society*, (pp. 836-841). Hillsdale, NJ: Erlbaum.

Vygotsky, L. S. (1978). *Mind in Society*. Cambridge, MA: Harvard University Press

Wertheimer, R. (1990). The geometry proof tutor: An "intelligent" computer-based tutor in the classroom. *Mathematics Teacher*, April, 308-317.