Classroom Integration of Intelligent Tutoring Systems for Algebra and Geometry

Steven Ritter, Carnegie Learning

Kenneth R. Koedinger, Carnegie Mellon University

William Hadley, Carnegie Learning

Albert T. Corbett, Carnegie Mellon University

Marsha Lilly, Carnegie Learning

DRAFT – DO NOT QUOTE

Introduction

For almost ten years, we have been producing software and curriculum focused on the teaching and learning of high school mathematics. Our products are called “Cognitive Tutors,” to reflect their basis in cognitive science research and their use of intelligent tutoring systems to guide student problem solving. Each of these products consists of software (called a “Cognitive Tutor”), print materials (equivalent to a textbook, homework assignments, teacher’s guide, etc.) and teacher training. Our intent is that each of these products provide teachers with all the materials and support they need to teach an entire mathematics course.

This approach represents a major shift in the way we view educational technology. Typically, such technology, although often paired with the main curriculum, is used as a supplement, either for students who need extra practice or for those that can benefit from enrichment. In the Cognitive Tutor approach, the software is an integral part of the curriculum. Students work with the software in a computer lab during their regular class period in the same way that students in a science course would have a lab component for some fraction of their regular classroom time.

Typically, students in the Cognitive Tutor courses spend 40% of their time in the computer lab. This represents a significant proportion of the students’ time in the course, and reflects our belief that students’ use of this software is essential to the success that we have seen with students. But the typical view of educational software as fundamentally supplemental suggests that, if the software were presented without the text and training, then it would be relegated to supplemental status.

The software automatically tracks the progress of student understanding over time and controls pacing through the software curriculum. This allows students entering the lab to begin work where they had left off the previous period and to proceed through the materials at their own pace. We believe that providing software that is capable of tracking student learning over the whole school year is necessary but not sufficient for the software to be viewed as a first-class component of the curriculum. It is the integration of the software into the fundamental course structure that give the software this status.

By providing the print materials used in the classroom, the software used in the lab, initial teacher training and ongoing support, we are able to have much more influence in the implementation than the developers of most educational technology. This is in keeping with Honey, Culp and Carigg's (2000) argument that, if technology is to have a significant impact on learning, the technology must be part of a broader systemic change. While we cannot claim to have influenced the educational environment to this degree, our broader influence on how mathematics is presented in the school may account for a large part of our success. In this chapter, we will explore how we came to believe in this novel approach to the implementation of educational technology in the classroom and why we believe it is a successful model for the deployment of educational technology.

Project History

The Cognitive Tutor's roots are in research conducted at Carnegie Mellon University. Earlier studies of intelligent tutoring systems based on Anderson's ACT-R theory (Anderson, 1993; Anderson and Lebiere, 1998) were used both as a way to collect data for basic cognitive science research and as a way to increase student achievement. ANGLE, an intelligent tutoring system for geometric proof and the Lisp tutor (which taught the Lisp programming language) had both proven to be successful in real-world use (Anderson, Corbett, Koedinger, & Pelletier, 1995; National Research Council, 2000).

Although these projects were successful in an educational and technological sense, it became evident that there were barriers to more widespread dissemination of the tutors. While ANGLE was effective in teaching geometric proof, the NCTM Principles and Standards (2000) de-emphasized geometric proof, so the tutor could no longer find a place as the foundation of a course following the standards.

Similarly, the programming tutors were limited in their broad applicability. The Lisp programming language is still popular in the artificial intelligence community, and the tutor is still used to teach the language at Carnegie Mellon University, but this is a limited audience. Although the programming tutor technology was extended to Pascal and Prolog in addition to Lisp, the extension of the tutors to these languages started to involve communities that were further removed from the researchers' direct interests (cognitive psychology and artificial intelligence) and thus it became more difficult to track trends in curricular focus without greater involvement with the community of people teaching more general programming languages. While the tutors were developed in close collaboration with teachers, one barrier to wider dissemination was that they did not take enough account of the broader national direction in which the interested communities were taking (away from proof in the case of Geometry and towards C++ and later Java in the case of programming).

An additional barrier had to do with guiding the proper implementation. The implementations of these earlier tutors were closely watched and guided by the researchers. As the number of implementations increased, the researchers could not be involved as closely. Perhaps more importantly, they could not be as closely involved in the culture governing the conditions under which the tutors would be used. As the tutors moved to environments that were further removed from the culture of higher education, it became essential that they be developed in collaboration with people who had more experience with the diverse cultures in which they would be implemented. To address high school mathematics in a way that would make sense in high school mathematics classrooms, it was clear that high school mathematics teachers needed to be involved in all stages of the development of the tutors. It was also clear that the developers needed to be aware not only of the pilot teachers' needs and concerns, but of the general direction in which the field was moving and of the concerns of different members of the community.

While dissemination considerations provided some of the need for collaboration between researchers and teachers, it is clear that the exchange of ideas between researchers and practitioners led to novel approaches and understandings that produced products that were of higher quality, in addition to being more appropriate for broad dissemination.

Cognitive Tutor Description

Software

The fundamental approach in the software is to present a student with a problem and with a set of tools that are appropriate to solving that problem. This approach parallels the view of tasks presented in Hiebert, et. al, 1997. As the student works through his or her problem solution, the system "watches" the steps that the student is performing. If the student's actions indicate that the student is confused or harboring a misconception, then the system will intervene and present a message to get the student back on track. Otherwise, the student can proceed as if the tutor did not exist. The student can also ask for a hint at any step in the problem-solving process, and the system will respond with suggestions that are specific to the student's solution path and to the particular problem that the student is solving.

Figure 1 shows a screen shot from a typical problem in the Algebra I Cognitive Tutor. This problem is presented in Unit 2 of 31 units in the curriculum and would typically be encountered by a student at the end of the first month of class. In later sections of the curriculum, new tools are introduced, and the functionality of some tools changes.

Text materials and classroom activities

As in the software, the classroom materials ask students to consider relationships between different mathematical representations (text, tables, equations and graphs). This curricular focus is not unique and is similar to that taken in several new texts, including Concepts in Algebra (Fey and Heid, 1999).

As the example text pages shown in Figure 2 illustrate, the student text is unusual, in that it contains relatively little declarative instruction. Most of the declarative instruction takes place in a problem-solving context. As in the software, students are expected to develop both conceptual knowledge and procedural skills through interacting with real-world problem situations. The tasks incorporated into the main student text ask students to reflect on the mathematics they are performing to solve problems and require students to articulate their mathematical knowledge and problem-solving methods in full sentences.

Much of students' class time is spent in small-group problem solving. The text materials include assignments, which can be used for this purpose. As shown in Figure 3, some of these assignments are similar to the kinds of activities that students perform in the software, while others relate to topics or approaches not addressed in the software. In addition to assignments that ask students to articulate their findings in text, many involve class presentations.

Training and Support

With its emphasis on group activities, problem solving and class presentations, the Cognitive Tutor course represents a major shift in pedagogy for most current mathematics teachers (Stigler and Hiebert, 1999). Early on in the research project , it became evident that, if this curriculum was to proceed beyond a closely-watched research project, we would need to build the course in a way that, though it required different behaviors and pedagogical approaches from teachers than a traditional textbook, did not impose such a burden of implementation on teachers that only the best could succeed with the approach.

We believe that we have succeeded in creating such a course, in part because we recognize that teachers face a hurdle in understanding how they fit in to a course that seems so different from what they have previously done in class. To help overcome this hurdle, new teachers attend a four-day training class. Most teachers go in to this class thinking that they will primarily be learning how to use the software. However, once they work with students in the computer lab, they realize that their students have no problem entering values in text fields, re-arranging windows and dragging-and-dropping points on the graph. Students are technologically savvy. Students' difficulties with the software involve understanding the mathematics required to complete the problems, not the mechanics of manipulating the software to reflect their understanding. Once in the computer lab, teachers spend relatively little time talking to students about the software itself and much more time talking to individual students about mathematics.

During training, time is devoted to helping teachers understand how best to help students who are struggling with the mathematics and how to best implement the pedagogical approach and objectives of the curriculum. A particular concern relates to the integration of software and the student text and classroom activities. After all, students proceed at their own pace through the software, while the classroom necessarily proceeds at the same pace for all students. How, teachers ask, can they deal with students who, in the software, are ahead of or behind the classroom activities?

Of course, this situation is not unique to the Cognitive Tutor course. In every class, there are some students whose understanding lags behind the pace of the class and others who perceive the class as moving too slowly. But the Cognitive Tutor class' emphasis on problem solving and small group activities allows students to naturally approach class activities in a way that is consistent with their own level of understanding. More advanced students may be more articulate and may use more formal methods in their problem solving, and these abilities may be shared with other students in group problem solving and presentations.

Our approach in the training follows the pedagogical approach taken in the course. Teachers learn the pedagogy and goals of the course by solving some of the same problems that students would solve in the class. They discuss issues like software and text integration as a way of understanding the issues that may arise in the classroom.

Networking among users is critical. With the availability of the web and on-site user group sessions, teachers are able to communicate and connect.

Effectiveness

A strong foundation in cognitive theory gives products like the Cognitive Tutors a good chance at success, but continual testing and refinement with real students is the ultimate measure of success. Through the years, the Cognitive Tutors have been subjected to many controlled evaluations (Carnegie Learning, 2002; Koedinger, Anderson, Hadley and Mark, 1997; Anderson, Corbett, Koedinger and Pelletier, 1995).

Evaluating the Cognitive Tutors presents many practical and methodological problems. A Cognitive Tutor course differs from a traditional mathematics course on several dimensions. It uses software as part of its primary instruction, it emphasizes small-group classroom activities and it focuses on active learning. In addition, the Cognitive Tutor course's emphasis on mathematical problem solving, communication and translation between multiple representations differs somewhat from the goals of most traditional mathematics courses.

Any of these differences could become the focus of an evaluation wishing to determine the independent effect on a particular aspect of the course. Instead, our evaluations have taken the course as a whole (text, software and training) as a single manipulation. This reflects our belief that each aspect of the course reinforces the others, so the whole is greater than the sum of the parts. To address different curricular emphases, evaluations have used multiple measures of effectiveness, including standardized exams, problem-solving exams, and attitude surveys.

A particularly interesting study was conducted in the Moore, OK Independent School District (Carnegie Learning, 2002b). In this study, students at five schools used either the

Cognitive Tutor course or a traditional textbook. Six teachers in three schools agreed to teach both Cognitive Tutor and traditional sections of the course. Students were randomly assigned to condition, which provides the strongest evidence that differences between the courses are really due to the curriculum (Shadish, Cook and Campbell, 2002). Within-teacher controls ensure that the results are not due to differences in teacher ability or style. Outcome measures included performance on the ETS Algebra I End-of-Course exam, course grades and measures of student confidence and perceived usefulness of mathematics (this survey was based on one developed by Fennema and Sheman, 1976).

The results, shown in Table 1, show students in the Cognitive Tutor classes significantly outperforming the control students on all measures.

Measure	Textbook	Cognitive Tutor
ETS Algebra I EOC Exam	15.4 (5.8)	16.9 (5.5)
Semester 1 Grade	2.70 (1.20)	2.94 (0.10)
Semester 2 Grade	2.36 (1.30)	2.67 (1.19)
Attitude: Confidence	3.4 (1.0)	3.6 (0.9)
Attitude: Usefulness	3.6 (0.9)	4.0 (0.8)

Table 1: Means (and standard deviations) from the Moore, OK study. All results are statistically significant, p<.05, except "attitude: confidence", p<.08. All data reflect scores from students whose teachers simultaneously taught one or more Cognitive Tutor *and* one or more textbook courses. Students were randomly assigned to condition.

The Cognitive Model Approach

A fundamental belief behind our approach is that a detailed and accurate model of student thinking allows us to better select tasks for students, understand individual students' strategies and provide feedback appropriate to individual student errors. Although the close relationship between our cognitive modeling efforts and the ACT-R theory is unique, the general idea that instruction benefits from knowledge about student thinking is not new.

Cognitively Guided Instruction (Carpenter, Fennema, Franke, Levi and Empson, 1999), for example, emphasizes understanding students' current knowledge and problem-solving strategies. CGI stresses that teachers can make better decisions when their instruction is informed by knowledge about student strategies and concepts in the domain of concern.

Unlike CGI, however, our approach results in a specific curriculum that embodies our knowledge about student problem solving in a particular domain. Our goal is to encourage students to solve problems using strategies that they have already found to be successful. The curriculum then builds to more difficult problems of the same type, where students realize that more sophisticated (and, often, formal mathematical) methods are more effective.

Initially, most students are able to solve problems using informal reasoning. As shown in Figure 1, the tutor interface supports and encourages this type of reasoning. Students are encouraged to complete specific instances before worrying about the formalization of relationships in algebraic terms.

While some students may still solve the problem using formal methods (they're free to fill in the expression row before the specific instance rows and to solve equations on paper) but there's no incentive for them to do so. In fact, with simple problems like this, even advanced students are faster and more accurate using informal methods than formal ones.

As the curriculum develops, students encounter problems that are superficially similar, but that involve large decimal numbers for both slope and intercept. The system then puts the expression row at the top of the worksheet and includes the equation solving tool. By this time, many students have evolved a semi-formal subtract and divide unwinding strategy. The system's hints support students executing that strategy, but, if a student is not able to execute the strategy, the system will recommend using equation solving. That is, if a student is unable to answer this question and has not yet entered the algebraic expression for the amount of money saved, then the system will direct the student to work on that part of the problem. Once this formal expression is developed, if the student still is not able to answer the question, the system will direct the student to solve the equation $112.42+48.43m=499.86$. If the student uses the system's equation solving tool, the system will give the student support in completing that part of the problem, as well.

One could say that cognitive modeling in the Cognitive Tutor classroom has two different types of effects. First, the cognitive model is an active part of the software that students use as an integral part of their classroom experience. The model thus directly affects the tasks, feedback and instructional material that a student sees in the classroom.

Second, the process of building a cognitive model teaches us about student thinking in the domain of interest. This process includes conducting basic research aimed at understanding student thinking and describing the task in sufficient detail that a computer model can be built to simulate student behavior. Through this process of research and modeling, we learn more about what makes particular tasks in a domain difficult, and this knowledge affects the composition, ordering and presentation of tasks in the curriculum (both software and text). This type of effect might be considered “indirect,” in that it results from the process of cognitive modeling, rather than the presence of an active cognitive model.

Developing the cognitive model

The effectiveness of our courses depends on how well we understand student thinking about mathematics and how well we have represented that thinking in our cognitive models. We use a mixture of formal and informal techniques to develop cognitive models. An important emphasis in the use of these methods is to maintain frequent contact with real teachers and real students. This leads to an overall development process where one cognitive modeling technique suggests a particular approach which is then tested in the classroom, refined and then, perhaps, enhanced by results from another modeling technique. This development methodology fits well with Sabelli and Dede’s (1999) description of “implementation research.” Some of the more important techniques are difficulty factors assessment, formative experimentation, focused testing and skill refinement

Conclusion

The Cognitive Tutor programs represent a unique approach to the use of technology in the classroom. As an implementation strategy, the integration of text and software as a standard part of in-class activity prevents the software from being marginalized as supplemental. A more fundamental form of integration results from the research strategy of basing both the software and text in cognitive modeling. Cognitive modeling can be thought of as a way of understanding student thinking and learning in a domain. The particular form of cognitive modeling that we use, with its basis in ACT theory, allows us to include an active cognitive model to interact with students and to collect data that lets us continually improve the model itself.

This approach has proven to be successful at improving student understanding of mathematics, as measured both by standardized test scores and by students' ability to solve mathematical problems.

References

Aleven, V., Koedinger, K. R., Sinclair, H. C., & Snyder, J. (1998). Combatting shallow learning in a tutor for geometry problem solving. In Goettl, B. P., Halff, H. M., Redfield, C. L., & Shute, V. J. (Eds.) *Intelligent Tutoring Systems, Proceedings of the Fourth International Conference*. (pp. 364-373). Lecture Notes in Computer Science, 1452. Berlin: Springer-Verlag.

Anderson, J. R. (1993). *Rules of the Mind.* Hillsdale, N. J.: Erlbaum.

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. & Lebière, C. (1998). *The atomic components of thought*. Mahwah, NJ: Erlbaum.

Carpenter, T. P., Fennema, E., Franke, M. L., Levi, L. and Empson, S. B. (1999). Children's Mathematics: Cognitively Guided Instruction. Portsmouth, N. H.: Heinemann.

Corbett, A.T. (2001). Cognitive computer tutors: Solving the two-sigma problem. User Modeling: Proceedings of the Eighth International Conference, UM 2001, 137-147.

Fennema, E. & Sherman, J.A. (1976). Fennema-Sherman mathematics attitude scales. Instruments designed to measure attitudes toward the learning of mathematics by males and females. *JSAS Catalog of Selected Documents of Psychology*, 6(31). (Ms. No. 1225)

Fey, J.T. and Heid, M.K. (1999). *Concepts in Algebra: a technological approach*. Chicago: Everyday Learning Corporation.

Gluck, K. (1999). Eye movements and algebra tutoring. Unpublished doctoral dissertation. Carnegie Mellon University.

Heibert, J., Carpenter, T. R, Fennema, E., Fuson, K. C., Wearne, D., Murray, H. Oliver, A. and Human, P. *Making Sense: Teaching and learning mathematics with understanding*. Portsmouth, N.H.: Heinemann.

Honey, Culp and Carrig (1999). [need full ref] www.ed.gov/Technology/TechConf/1999/whitepapers/paper1.html.

Koedinger, K. R., & Anderson, J. R. (1998). Illustrating principled design: The early evolution of a cognitive tutor for algebra symbolization. *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.

National Council of Teachers of Mathematics (2000). *Curriculum and Evaluation Standards for School Mathematics*. Reston, VA: The Council.

National Research Council (2000). Technology to support learning. In *How People Learn: Brain, mind, experience and school*. (pp. 206-230). Washington: National Academy Press.

Ritter, S. and Anderson, J. R. (1995). Calculation and strategy in the equation solving tutor. In J.D. Moore & J.F. Lehman (Eds.),*Proceedings of the Seventeenth Annual Conference of the Cognitive Science Society*. (pp. 413-418). Hillsdale, NJ: Erlbaum.

Sabelli, N. and Dede, C. (1999). [need full ref] www.virtual.gmu.edu/SS_research/cdpapers/index.htm

Shadish, W. R., Cook, T. D. and Campbell, D. T. (2002). *Experimental and quasi-experimental designs for generalized causal inference*. Boston: Houghton Mifflin.

Singley, M. K. and Anderson, J. R. (1989). *The transfer of cognitive skill*. Cambridge, MA: Harvard University Press.

Stigler, J. W. and Hiebert, J. (1999). *The Teaching Gap*. New York: The Free Press.

Figures

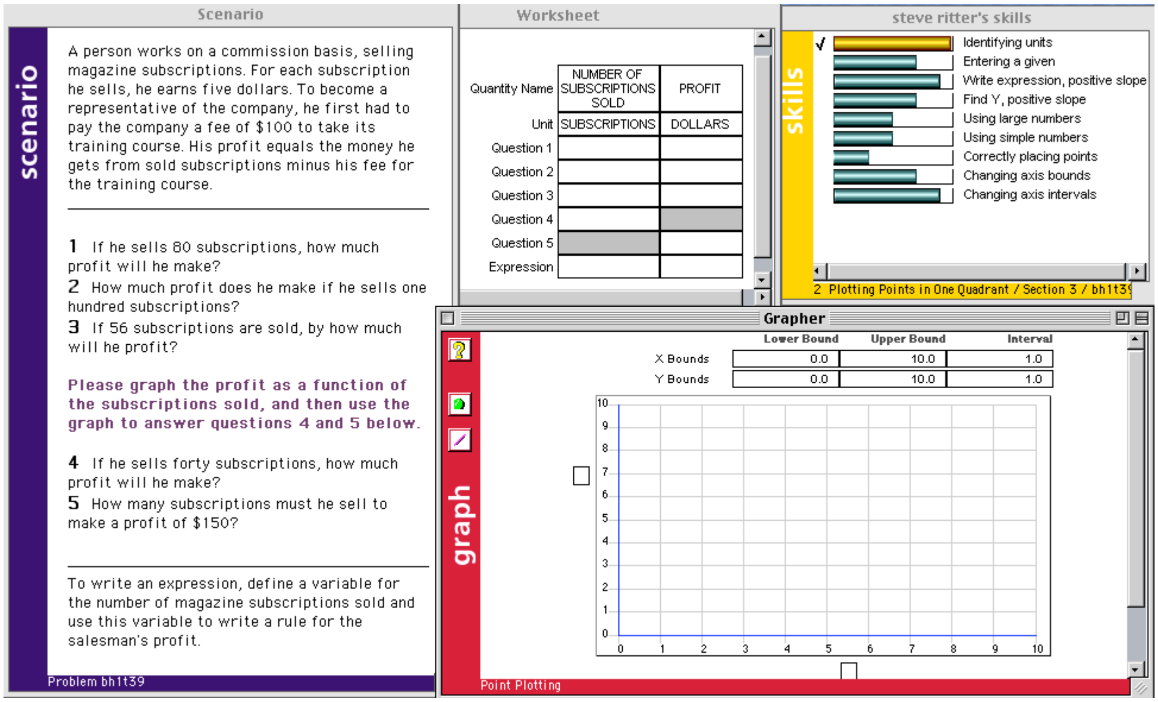

Figure 1: Screen shot from Cognitive Tutor Algebra I software

Task 4

In this task you use *function notation* to express the relationship between side length and bottom width.

To emphasize that the dependent variable of a function depends on the independent variable, and that a function is a process of taking input values and generating output values, mathematicians invented a special notation called *function notation* or *"f of x" notation.*

Definition:

Function notation replaces the dependent variable y with an expression like $f(x)$. This expression is read "*f of x*." It means that f is a function of the variable x.

Note: The expression $f(x)$ does not mean that f is multiplied by x. The letter f as it is used in this notation is *not* a variable; it is merely a name for a function.

$f(x)$ is simply an

$y = f(x) =$ [some

1. Write the algeb notation. Use th the variable x to

2. Write a sentenc used in problem

This notation is also input and output v

Example: $y = B(x) = 8.5 - 2x$

a. Find $B(0.5)$.

The question being asked is "What is the bottom width when the side length is 0.5 inches?" In general terms, the question is "What is the output of the function when the input is 0.5?"

To find the answer, replace x in the function with the value 0.5, and perform the arithmetic:

$$B(0.5) = 8.5 - 2(0.5) = 8.5 - 1 = 7.5$$

Answer: The bottom width is 7.5 inches when the side length is 0.5 inches.

b. Find x when $B(x) = 6.5$.

The question being asked is "What is the side length when the bottom width is 6.5 inches?" In general terms, the question is "What is the input of the function when the output is 6.5?"

To find the answer, replace $B(x)$ with 6.5, and solve the resulting equation for x:

$$6.5 = 8.5 - 2x$$
$$6.5 - 8.5 = -2x$$
$$-2 = -2x$$
$$\frac{-2}{-2} = x$$
$$x = 1$$

Answer: The side length is 1 inch when the bottom width is 6.5 inches.

3. $B(x) = 8.5 - 2x$

In each of the following, state what question is being asked and find the numerical answer.

a. Find $B(1.75)$.

Figure 2: Two pages from the Cognitive Tutor Algebra I text

Assignment 7

Name: ____________________

Teams in a League Problem

You have been asked to help schedule the games in your younger sister's soccer league. There are 8 teams in the league. You have been asked to determine the total number of games that must be played so that *every team will play every other team exactly once.*

Someone suggests that to find the answer, you first try to figure out the total number of games for a league that has three teams, then four teams, and so on.

You must make a written presentation to the league officers. The report should include a description of how you arrived at your answer. Write the re

Assignment 14

Name: ____________________

PROBLEM SITUATION

You are going on vacation by airplane. You will be traveling at an aver speed of 325 miles per hour.

Make a table showing how far you will travel in various time periods fr to 20 hours. Use this information to construct a line graph and find the algebraic formula.

Labels	Time in the Air	Distance Traveled
Units		
Formula		
	1	
	2	
	5	
	11	
	15	
	18	
	20	

Figure 3: Assignments from student text