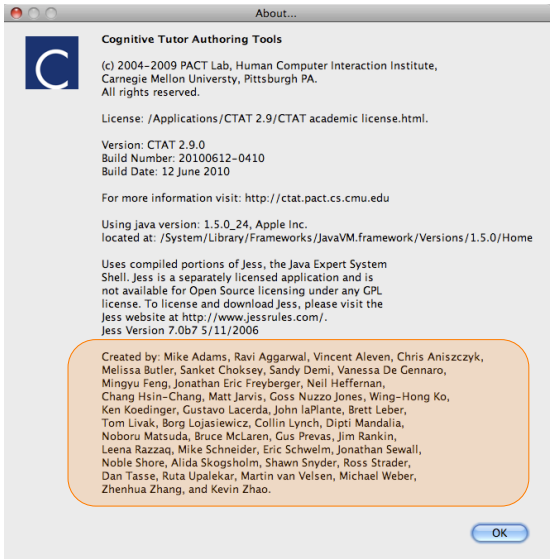
Building Intelligent Tutoring Systems with CTAT (the Cognitive Tutor Authoring Tools)

Vincent Aleven and the CTAT team

6th Annual PSLC LearnLab Summer School
Pittsburgh, July 12-16, 2010

CTAT CognitiveTutorAuthoringTools

LearnLab Pittsburgh Science of Learning Center



President Obama on Intelligent Tutoring Systems

"[W]e will devote more than three percent of our GDP to research and development. …. Just think what this will allow us to accomplish: solar cells as cheap as paint, and green buildings that produce all of the energy they consume; *learning software as effective as a personal tutor;* prosthetics so advanced that you could play the piano again; an expansion of the frontiers of human knowledge about ourselves and world the around us. We can do this."

http://my.barackobama.com/page/community/post/amyhamblin/gGxW3n

Senator Obama:

Barack Obama believes the nation can and must dramatically improve STEM education. As President, he will:

…

Integrate technology in the classroom so innovative learning technologies such as simulations, interactive games, and *intelligent tutoring systems* can assist in improving the quality of learning and instruction.

…

OBAMA BIDEN
WWW.BARACKOBAMA.COM

INVESTING IN AMERICA'S FUTURE
BARACK OBAMA AND JOE BIDEN'S PLAN FOR SCIENCE AND INNOVATION

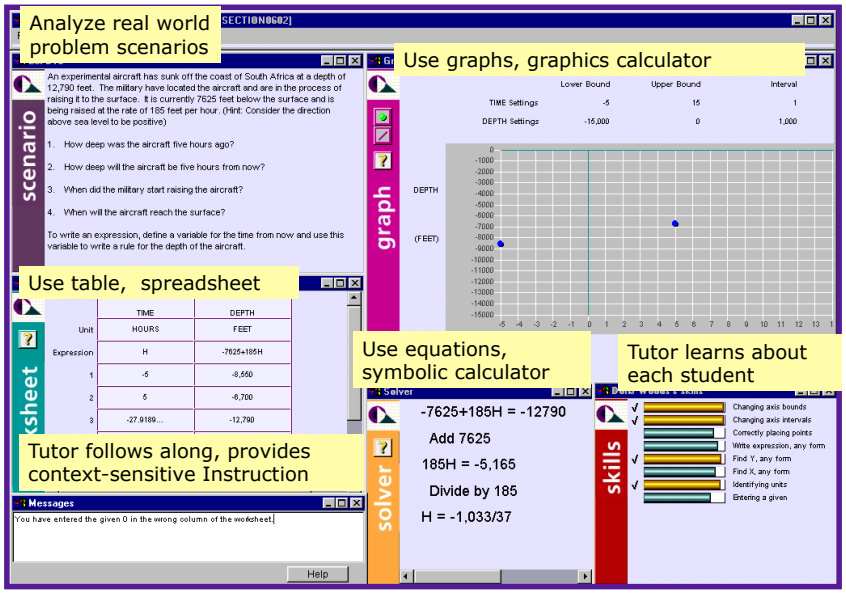
Overview

- What is "a tutor?"
  - What are essential characteristics of intelligent tutoring systems?
  - How do we know tutors help students learn more efectively?
- What can you do with CTAT?
  - Short movie of authoring with CTAT
  - Examples of projects that have used CTAT
- Planned activities in the CTAT track

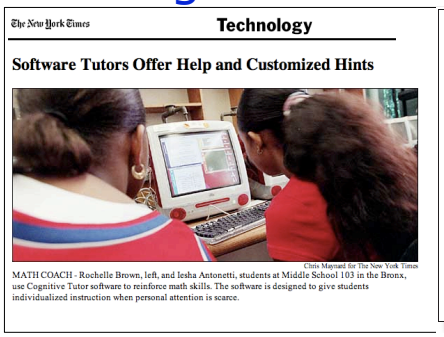
If you are not in the CTAT track, should you listen to this talk?

- CTAT relevant to most other tracks:
  - *In Vivo:* could do an *in vivo* experiment with CTAT-based tutors (happens all the time!)
  - Data Mining: many data sets in the Data Shop were generated using CTAT-built tutors
  - CSCL: Collaborative learning with intelligent tutors is an interesting and important research topic!

Algebra Cognitive Tutor



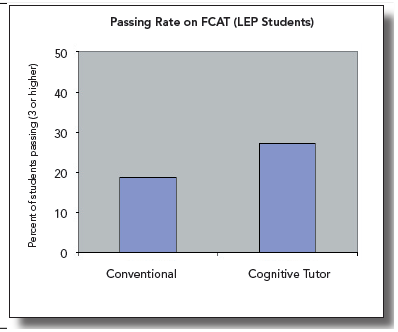
Cognitive Tutor math courses making a difference


The New York Times **Technology**

Software Tutors Offer Help and Customized Hints

Chris Maynard for The New York Times

MATH COACH - Rochelle Brown, left, and Iesha Antonetti, students at Middle School 103 in the Bronx, use Cognitive Tutor software to reinforce math skills. The software is designed to give students individualized instruction when personal attention is scarce.

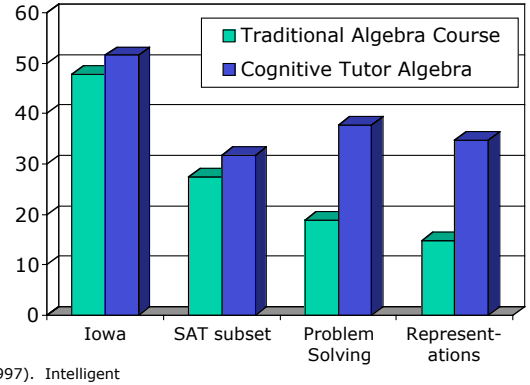


- Real-world impact of Cognitive Tutors
  - 10 of 14 full year evaluations are positive
  - Spin-off Carnegie Learning doing well
  - 500,000 students per year!

Replicated Field Studies

- Full year classroom experiments
- Replicated over 3 years in urban schools
- In Pittsburgh & Milwaukee
- Results:
  50-100% better on problem solving & representation use.

  15-25% better on standardized tests.



Koedinger, Anderson, Hadley, & Mark (1997). Intelligent tutoring goes to school in the big city. *International Journal of Artificial Intelligence in Education, 8*.

The nested loop of conventional teaching

For each chapter in curriculum

- Read chapter
- For each exercise, solve it
- Teacher gives feedback on all solutions at once
- Take a test on chapter

VanLehn, K. (2006). The behavior of tutoring systems. *International Journal of Artificial Intelligence in Education, 16*(3), 227-265.

The nested loops of Computer-Assisted Instruction (CAI)

For each chapter in curriculum

- Read chapter
- For each exercise
  - Attempt answer
  - Get feedback & hints on answer; try again
  - If mastery is reached, exit loop
- Take a test on chapter

VanLehn, K. (2006). The behavior of tutoring systems. *International Journal of Artificial Intelligence in Education, 16*(3), 227-265.

The nested loops of ITS

For each chapter in curriculum
- Read chapter
- For each exercise
  - For each step in solution
    - Student attempts step
    - Get feedback & hints on step; try again
  - If mastery is reached, exit loop
- Take a test on chapter

VanLehn, K. (2006). The behavior of tutoring systems. *International Journal of Artificial Intelligence in Education, 16*(3), 227-265.

Inner loop options: within-problem guidance offered by ITS

+	Minimal feedback on steps (classifies steps as correct, incorrect, or suboptimal)	
+		Immediate
+/–		Delayed (not built in, but some forms can be authored)
–		Demand
+	Error-specific feedback	
+	Hints on the next step	
+	Assessment of knowledge	
–	End-of-problem review of the solution	

VanLehn, K. (2006). The behavior of tutoring systems. *International Journal of Artificial Intelligence in Education, 16*(3), 227-265.

Aleven, V., McLaren, B. M., Sewall, J., & Koedinger, K. R. (2009). A new paradigm for intelligent tutoring systems: Example-tracing tutors. *International Journal of Artificial Intelligence in Education, 19*(2), 105-154.

–	CTAT supports it
(+)	CTAT will soon support it
+/–	CTAT supports a limited form of it
–	CTAT does not support it

Outer loop: problem selection options offered by ITS

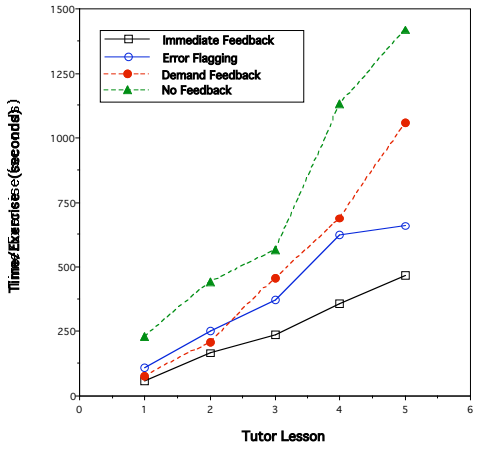
–	Student picks
+	Fixed sequence
(+)	Mastery learning
(+)	Macroadaptation

VanLehn, K. (2006). The behavior of tutoring systems. *International Journal of Artificial Intelligence in Education, 16*(3), 227-265.

Aleven, V., McLaren, B. M., Sewall, J., & Koedinger, K. R. (in press). Example-tracing tutors: A new paradigm for intelligent tutoring systems. *International Journal of Artificial Intelligence and Education.*

–	CTAT supports it
(+)	CTAT will soon support it
+/–	CTAT supports a limited form of it
–	CTAT does not support it

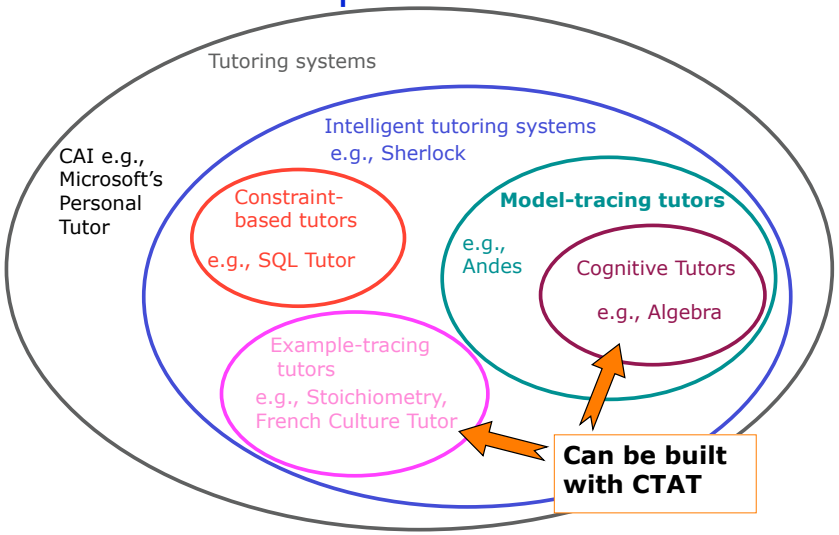
Feedback Studies in LISP Tutor (Corbett & Anderson, 1991)



Time to Complete Programming Problems in LISP Tutor

Immediate Feedback Vs Student-Controlled Feedback

Kinds of Computer Tutors



ACT-R: A Cognitive Theory of Learning and Performance

- Big theory … key tenets:
  - Learning by doing, not by listening or watching
  - Production rules represent performance knowledge:

These units are:		Instruction implications:
• modular	→	isolate skills, concepts, strategies
• context specific	→	address "when" as well as "how"

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

Cognitive Tutor Technology:

Use ACT-R theory to individualize instruction

- Cognitive Model: A system that can solve problems in the various ways students can

Strategy 1: IF the goal is to solve a(b*x*+c) = d
THEN rewrite this as ab*x* + ac = d

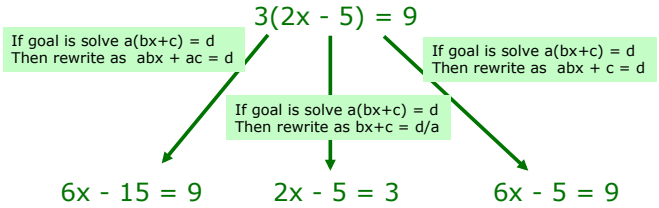
Strategy 2: IF the goal is to solve a(b*x*+c) = d
THEN rewrite this as b*x* + c = d/a

Misconception: IF the goal is to solve a(b*x*+c) = d
THEN rewrite this as ab*x* + c = d

Cognitive Tutor Technology:

Use ACT-R theory to individualize instruction

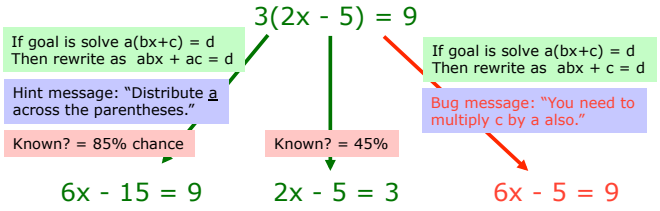
- Cognitive Model: A system that can solve problems in the various ways students can



- Model Tracing: Follows student through their individual approach to a problem -> context-sensitive instruction

Cognitive Tutor Technology:
Use ACT-R theory to individualize instruction

- Cognitive Model: A system that can solve problems in the various ways students can



- Model Tracing: Follows student through their individual approach to a problem -> context-sensitive instruction
- Knowledge Tracing: Assesses student's knowledge growth -> individualized activity selection and pacing

CTAT motivation: Make tutor development easier and faster!

- Cognitive Tutors:
  - Large student learning gains as a result of detailed cognitive modeling
  - ~200 dev hours per hour of instruction (Koedinger et al., 1997)
  - Requires PhD level cog scientists and AI programmers
- Development costs of instructional technology are, in general, quite high
  - E.g., ~300 dev hours per hour of instruction for Computer Aided Instruction (Murray, 1999)

Murray, T. (1999). Authoring Intelligent Tutoring Systems: An Analysis of the state of the art. *The International Journal of Artificial Intelligence in Education, 10*, 98-129.

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

CTAT goal: broaden the group of targeted authors

- Instructional technology developers
- Instructors (e.g., computer-savvy college professors)
- Researchers interested in intelligent tutoring systems
- Learning sciences researchers using computer-based tutors as platforms for research

How to reduce the authoring cost?

- No programming!
  - Drag & drop interface construction
  - Programming by demonstration
- Human-Computer Interaction methods
  - *Use-driven design:* summer schools, courses, consulting agreements with users, own use
  - User studies, informal & formal comparison studies
- Exploit existing tools
  - Off-the shelf tools: Netbeans, Flash, Excel
- Component-based architecture & standard inter-process communication protocols

Tutors supported by CTAT

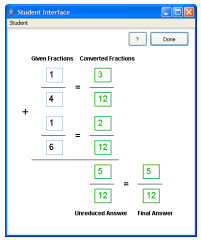
- Cognitive Tutors
  - Difficult to build; for programmers
  - Uses rule-based cognitive model to guide students
  - General for a class of problems
- Example-Tracing Tutors
  - Novel ITS technology
  - Much easier to build; for non-programmers
  - Use generalized examples to guide students
  - Programming by demonstration
  - One problem (or so) at a time

Building an example-tracing tutor in 5 easy steps …

- CTAT basics only!
  - Drag-and-drop techniques
  - Programming by demonstration
- Fraction addition example:

  1/4 + 1/6 = 3/12 + 2/12 = 5/12

  1/4 + 1/6 = 6/24 + 4/24 = 10/24 = 5/12

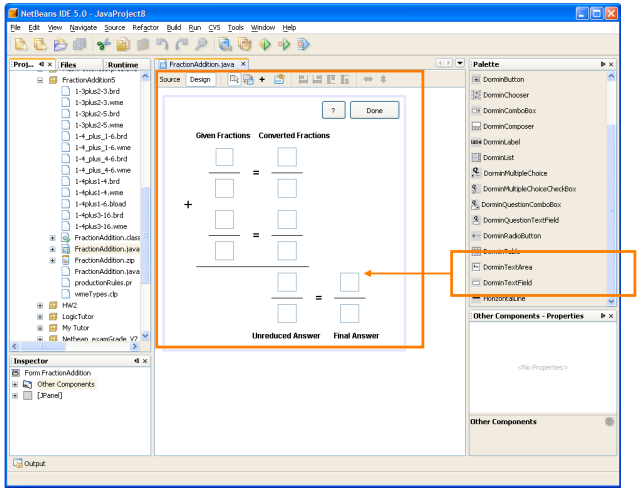


Building an example-tracing tutor

1. Decide on educational objectives
2. Cognitive Task Analysis
3. Design and create a user interface for the tutor
4. Demonstrate correct and incorrect behavior (i.e., create a behavior graph)
   - Alternative strategies, anticipated errors
5. Generalize and annotate the behavior graph
   - Add formulas, ordering constraints
   - Add hints and error messages
   - Label steps with knowledge components
6. Test the tutor
7. (Optional) Use template-based Mass Production to create (near)-isomorphic behavior graphs
8. Deliver on the web - import problem set into LMS (TutorShop)

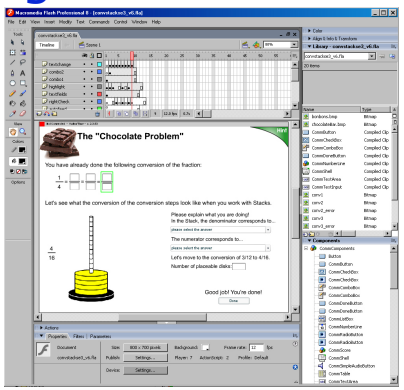
Movie Showing How an Example-Tracing Tutor is built

1. Create student interface with GUI builder

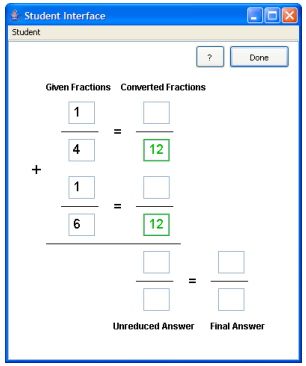


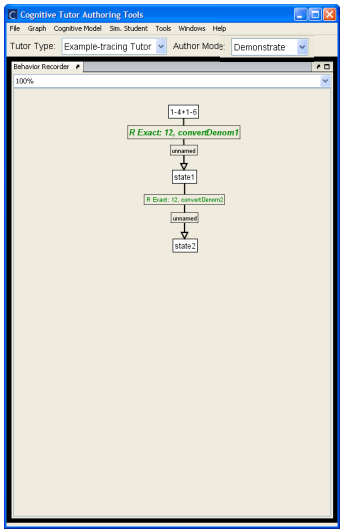
NetBeans IDE

1.a. Alternative way of building interfaces: Flash

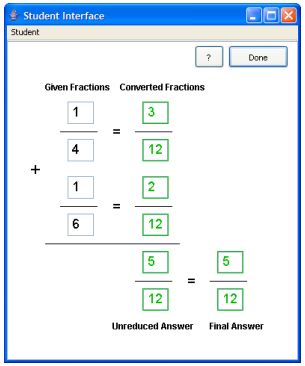


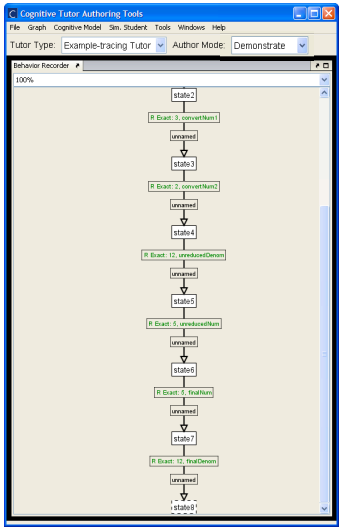
2. Demonstrate problem-solving behavior



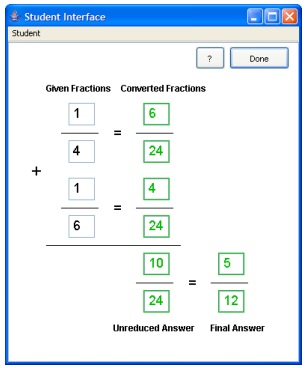


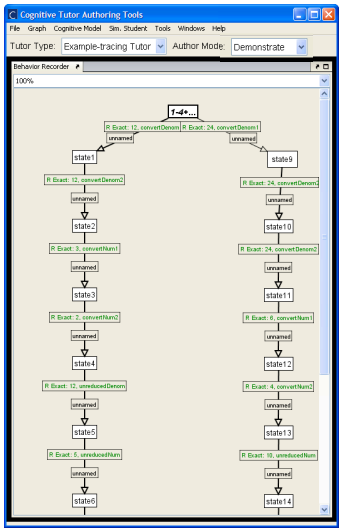
2. Demonstrate problem-solving behavior



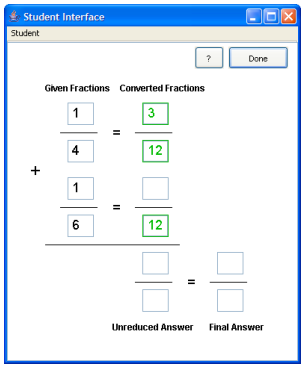


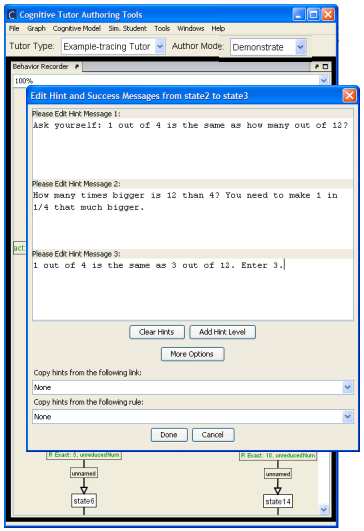
2. Demonstrate problem-solving behavior



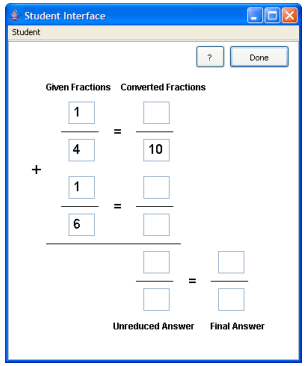


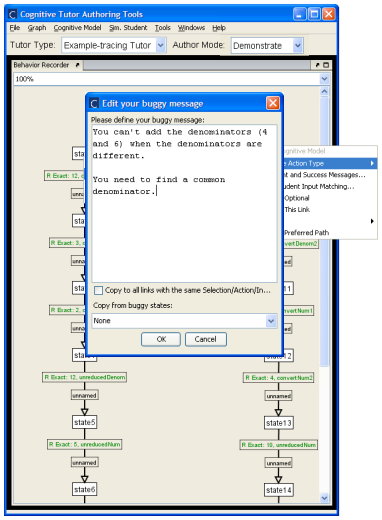
3. Annotate graph: hint messages



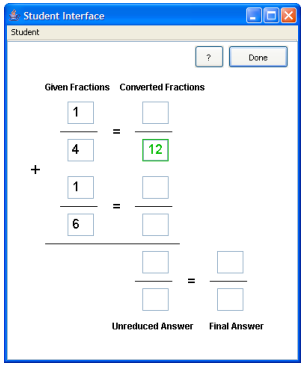


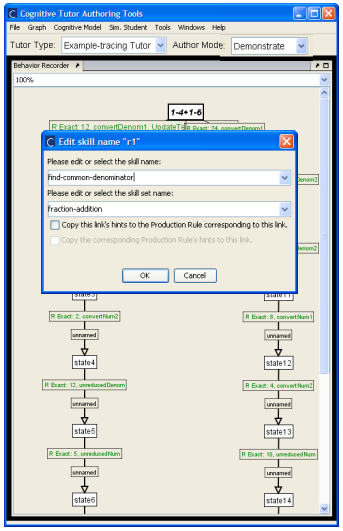
3. Annotate graph: incorrect step, feedback



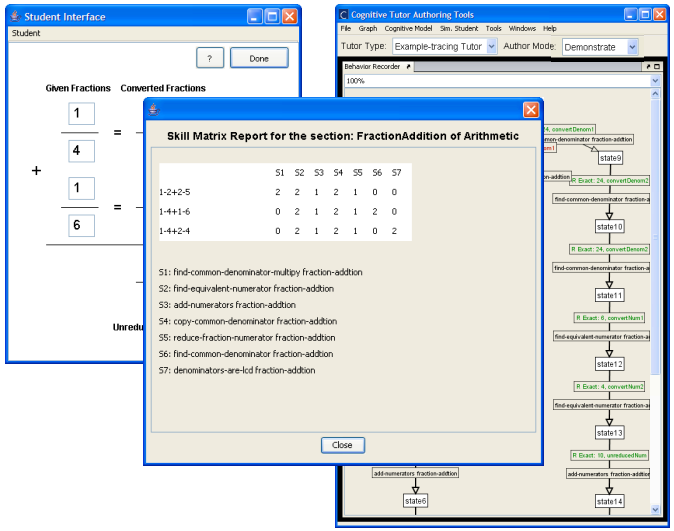


3. Annotate graph with knowledge components





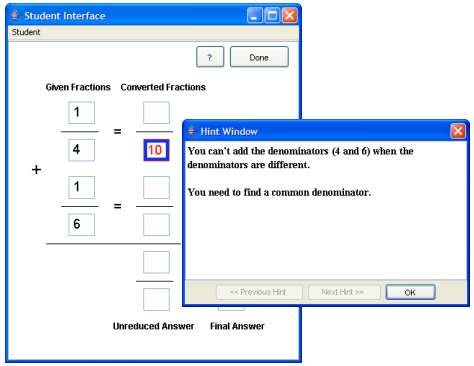
3. View knowledge component matrix



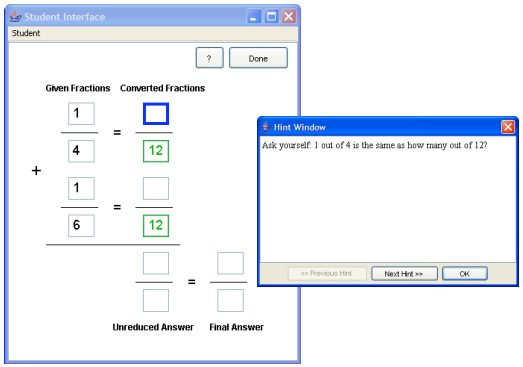
4. Generalize

- Ask if you'd like to hear more about this

5. Test the tutor



5. Test the tutor



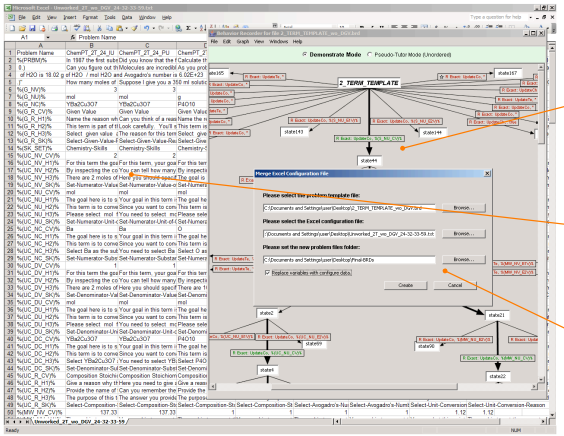
Example-tracing algorithm

- Basic idea: To complete a problem, student must complete one path through the graph
- Example tracer *flexibly* compares student solution steps against a graph
  - Keeps track of which paths are consistent with student steps so far
  - Can maintain *multiple parallel interpretations* of student behavior
  - Accepts student actions as correct when they are consistent with prior actions

6. Dealing with problem isomorphs and near-isomorphs: Mass Production

- Goal: avoid duplicating behavior graph structure across or within problems
- Would like to re-use behavior graph for
  1/4 + 1/6 = 3/12 + 2/12 = 5/12
  1/4 + 1/6 = 6/24 + 4/24 = 10/24 = 5/12
- When creating behavior graph for *isomorphic problems:*
  1/6 + 3/8 = 4/24 + 9/24 = 13/24
  1/6 + 3/8 = 8/48 + 18/48= 26/48 = 13/24
- And when creating behavior graph for *near-isomorphic problems*:
  1/6 + 1/10 = 5/30 + 3/30 = 8/30 = 4/15
  1/6 + 1/10 = 10/60 + 6/60 = 16/60 = 4/15

Mass Production: template-based tutor authoring to generate (near-)isomorphic behavior graphs



1. Turn Behavior Graph into template (insert variables)

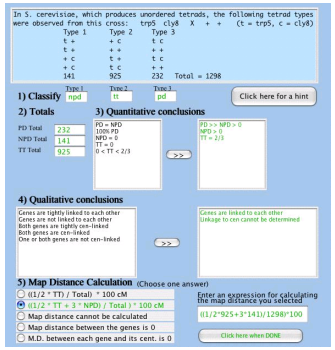
2. Fill in spreadsheet with problem-specific info; provide variable values per problem

3. Merge spreadsheet values into template

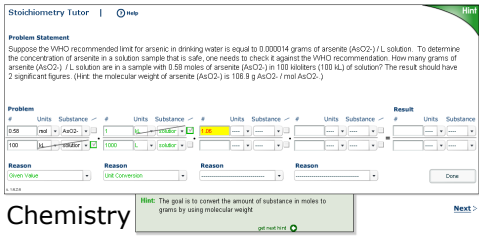
Vote-with-your-feet evidence of CTAT's utility

- Over 400 people have used CTAT in summer schools, courses, workshops, research, and tutor development projects
- In the past two years
  - CTAT was downloaded 4,300 times
  - the CTAT website drew over 1.5 million hits from over 100,000 unique visitors
  - URL: http://ctat.pact.cs.cmu.edu

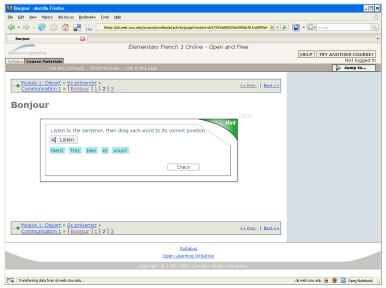
Some CTAT tutors used in online courses and research



Genetics



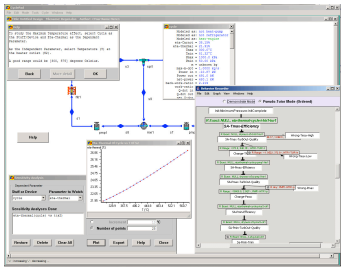
Chemistry

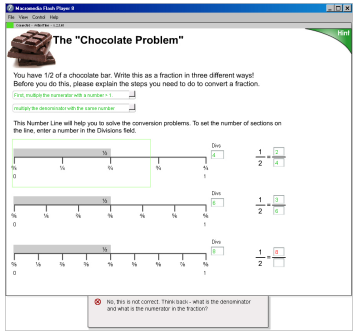


French

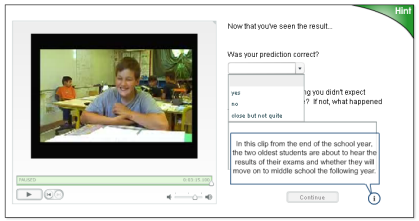
Some CTAT tutors used in research

Thermo-dynamics





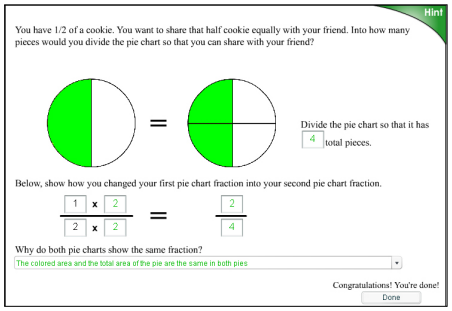
Elementary Math

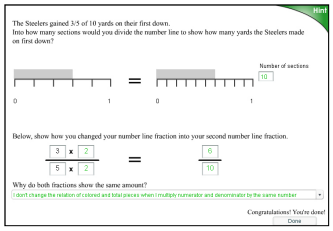


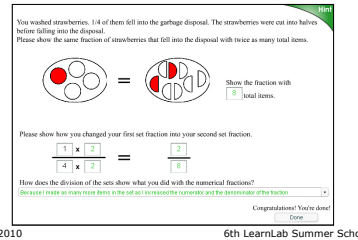
French (intercultural competence)

Blocked vs interleaved practice with multiple representations

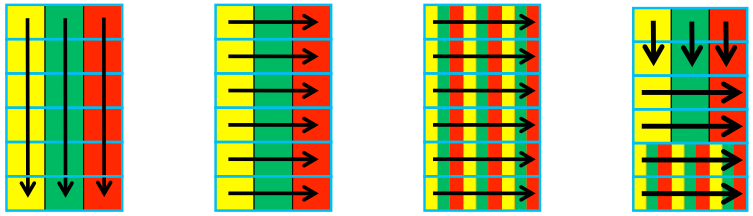
Martina Rau, Nikol Rummel, Vincent Aleven







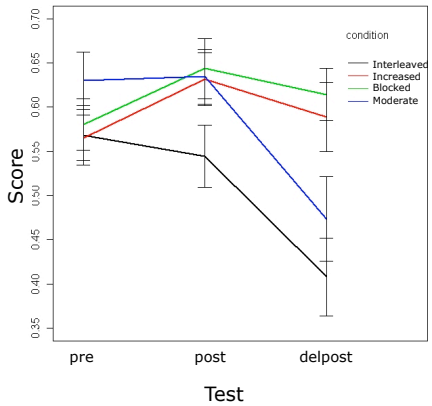
Experimental Design



- Colors indicate graphical representations of fractions
- The rows are problem types
- The cells are individual problems
- The arrows indicate the order in which students went through the problems

Results: Representational knowledge

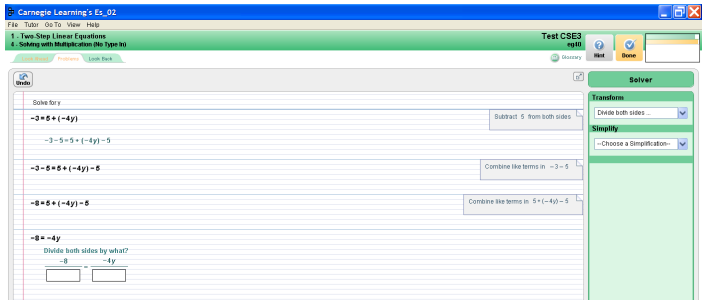
- Interaction effect for test*condition, $F(6, 418) = 5.09$ ($p < .01$)
  - Blocked and increased > interleaved at immediate post-test
  - Blocked and increased > moderate and interleaved at the delayed post-test



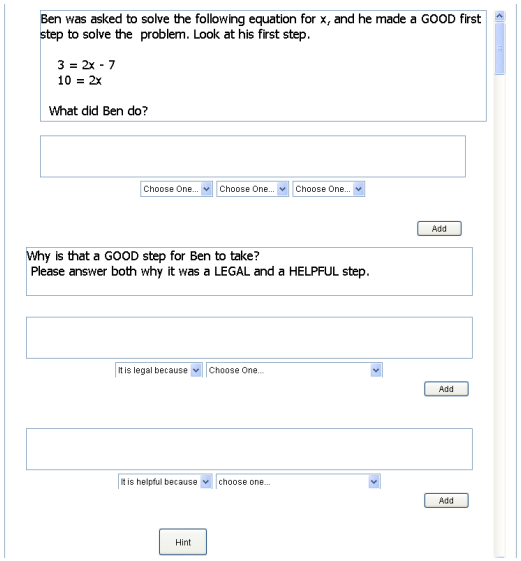
Using Self-Explanation to Improve Algebra Learning

Julie Booth, Ken Koedinger

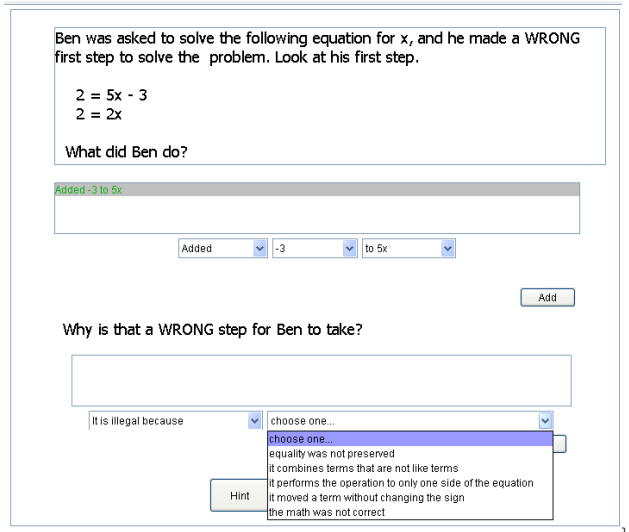
- Does the combination of worked examples and self-explanation improve learning in real-world algebra I classrooms?



Correct worked examples (built with CTAT)



Incorrect worked examples (built with CTAT)



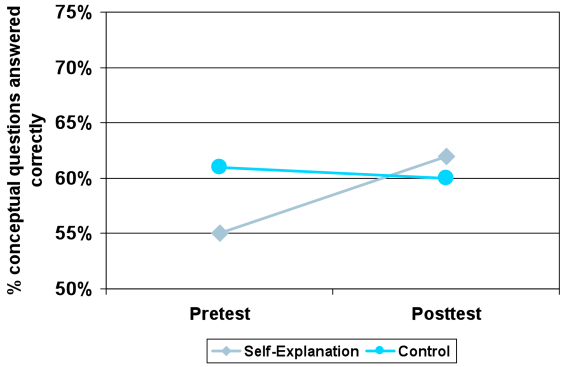
Method: Design

Self-Explanation of Correct Examples

Self-Explanation of Incorrect Examples		**No**	**Yes**
	No	Control	Typical
	Yes	Corrective	Typical + Corrective (half of each)

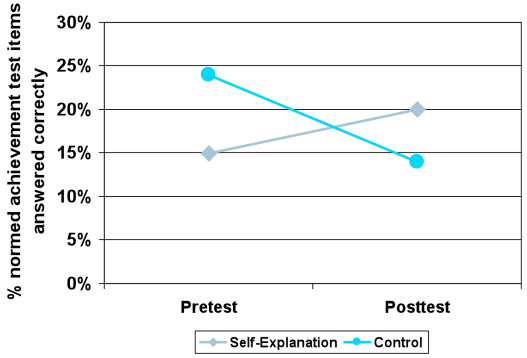
- All treatment groups receive the same number of self-explanation exercises
- Control for time between groups (self-explanation groups get less procedural practice)

Results: Conceptual knowledge



Results: Standardized test items

- Self-explain group improves more ($p < .05$)



Cost estimates from large-scale development efforts

- Historic estimate: it takes 200-300 hours to create 1 hour of ITS instruction, by skilled AI programmers (Anderson, 1991; Koedinger et al., 1997; Murray, 2003; Woolf & Cunningham, 1987)
- Project-level comparisons:
  - + Realism, all phases of tutor development
  - – High variability in terms of developer experience, outcomes (type and complexity of tutors), within-project economy-of-scale
  - – Many arbitrary choices in deriving estimates
  - – Can be difficult to track
  - – Can be difficult to separate tool development and tutor development

Development time estimates

Project Title	Domain	Studies	Students	Instructional Time	Development Time	Time Ratio
Improving Skill at Solving Equations through Better Encoding of Algebraic Concepts	Middle and High School Math - Algebra	3	268	16 mins each for 2 conditions	~120 hrs	240:1
Using Elaborated Explanations to Support Geometry Learning	Geometry	1	90	30 mins	~2 months	720:1
The Self-Assessment Tutor	Geometry - Angles, Quadrilaterals	1	67	45 mins	~9 weeks	540:1
Enhancing Learning Through Worked Examples with Interactive Graphics	Algebra - Equation Models of Problem Situations	1	60-120	~3 hrs	~260 hrs	87:1
Fluency and Sense Making in Elementary Math Learning	4th-Grade Math - Whole-number division	1	~35	2.5 hrs each for 2 conditions plus 1 hr of assessment	~4 months	107:1
The Fractions Tutor	6th-Grade Math - Fraction Conversion, Fraction Addition	1	132	2.5 hours each for 4 conditions	12 weeks	48:1
Effect of Personalization and Worked Examples in the Solving of Stoichiometry	Chemistry Stoichiometry	4	223	12 hrs	1016 hrs	85:1

Discussion of cost-effectiveness

- All tutors were used in actual classrooms
- Small projects worse than historical estimates (1:200 to 1:300)
- Large projects (> 3 hrs.) 3-4 times better (1:50 to 1:100)
- Factor in that programmers cost 1.5-2 times as much as non-programmer developers: total savings 4-8 times

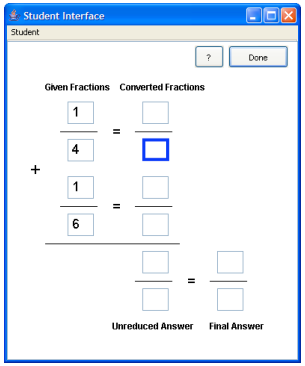
- Caveats: Rough estimates, historic estimates based on larger projects

During the summer school

- The CTAT track will cover development of Cognitive Tutors *and* Example-Tracing Tutors
  - – Lecture about grounding of Cognitive Tutor technology in ACT-R
  - – Number of "how to" lectures about cognitive modeling and model tracing
  - – Hands-on activities focused on building tutors
  - – Project

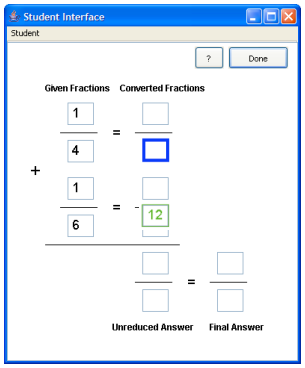
That's all for now!

Multiple solution paths enable context-sensitive hints



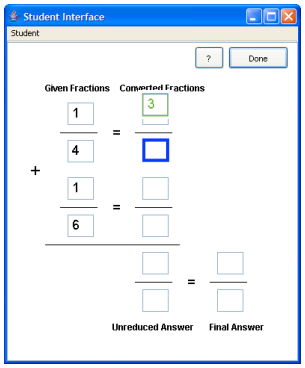
- You need to convert the fractions to a common denominator.
- You need to find a number that is a multiple of 4 and a multiple of 6.
- The smallest number that is a multiple of 4 and a multiple of 6 is 12.

Multiple solution paths enable context-sensitive hints



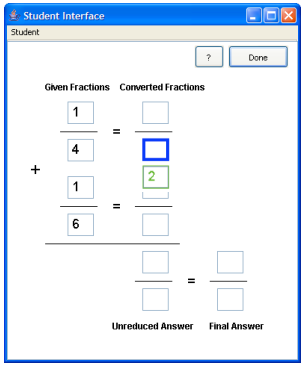
- You need to convert both fractions to the same denominator.
- Please enter '12' in the highlighted field.

Multiple solution paths enable context-sensitive hints



- 1 goes into 4 the same as 3 goes into what number?
- You multiplied by 3 to go from 1 to 3. You need to multiply 4 by the same number.
- Please enter '12' in the highlighted field.

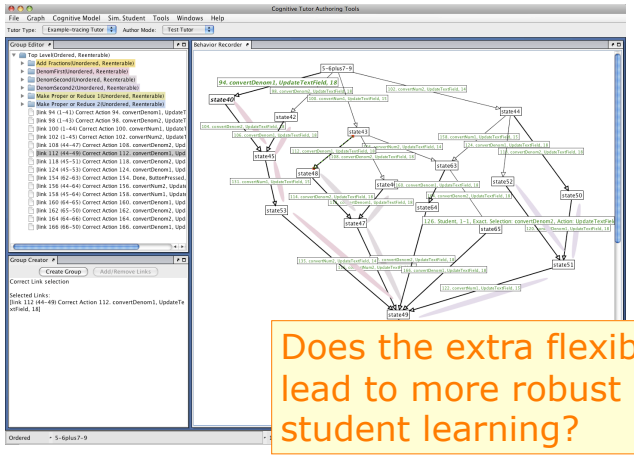
Multiple solution paths enable context-sensitive hints



Would not give a hint for the first converted denominator.

Would give hints for the second denominator first.

To realize this hinting flexibility, need more elaborate behavior graph

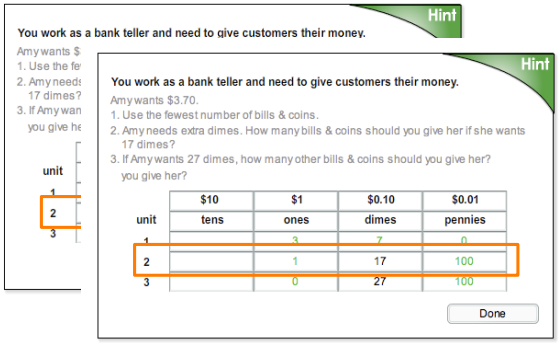


Does the extra flexibility lead to more robust student learning?

Multiple solution strategies by "formulas"

- Excel-like formulas express how steps depend on each other
- A form of end user programming

Example: Use of formulas



- Enumeration of paths: 6 paths for question 2

Example: Use of formulas