What is CTAT and why would you want to use it?

Overview of the CTAT track

Vincent Aleven and the CTAT team

5th Annual PSLC LearnLab Summer School
Pittsburgh,
July 7-13, 2009

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.

Overview

- What is “a tutor?”
  - What are essential characteristics of intelligent tutoring systems?
  - How do we know tutors help students learn more effectively?
- 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!
- Track hopping is allowed!
  - E.g., if in the In Vivo track, could attend CTAT sessions

What is an intelligent tutoring system?

- Tutor provides step-by-step support for practice of complex cognitive skill:
  - Interface makes reasoning steps (in given problem type) explicit
  - Correctness feedback
  - Next-step hints
  - Individualized problem selection based on detailed assessment of each student’s skill
Algebra Cognitive Tutor

- Analyze real world problem scenarios
- Use graphs, graphics calculator
- Use table, spreadsheet
- Use equations, symbolic calculator
- Tutor learns about each student
- Tutor follows along, provides context-sensitive instruction

Cognitive Tutor math courses making a difference

- 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.

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


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


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


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

Kinds of Computer Tutors

- **Intelligent tutoring systems**: e.g., Sherlock
- **Constraint-based tutors**: e.g., SQL Tutor
- **Example-tracing tutors**: e.g., Stoichiometry, French Culture Tutor
- **Model-tracing tutors**: e.g., Andes

Can be built with CTAT
**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: modular context specific
    Instruction implications:
    - isolate skills, concepts, strategies
    - address "when" as well as "how"


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

- Cognitive Model: A system that can solve problems in the various ways students can solve them.
  1. **Strategy 1:** IF the goal is to solve $a(bx+c) = d$ THEN rewrite this as $abx + ac = d$
  2. **Strategy 2:** IF the goal is to solve $a(bx+c) = d$ THEN rewrite this as $bx + c = d/a$
  3. **Misconception:** IF the goal is to solve $a(bx+c) = d$ THEN rewrite this as $abx + c = d$

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

- Cognitive Model: A system that can solve problems in the various ways students can solve them.
  1. **Strategy 1:** IF the goal is to solve $a(bx+c) = d$ THEN rewrite this as $abx + ac = d$
  2. **Strategy 2:** IF the goal is to solve $a(bx+c) = d$ THEN rewrite this as $bx + c = d/a$
  3. **Misconception:** IF the goal is to solve $a(bx+c) = d$ THEN rewrite this as $abx + c = d$

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

- 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)
- The International Journal of Artificial Intelligence in Education, 10, 98-129.

CTAT goal: broaden the group of targeted authors

- Instructional technology developers (e.g., instructional media dept. at university, or developers of on-line courses)
- Researchers interested in intelligent tutoring systems
- Instructors (e.g., computer-savvy college professors)
- Learning sciences researchers interested in using computer-based tutors in their experiments
  - Within the PSLC, CTAT-based tutors are often used in *in vivo* experiments

How to reduce the authoring cost?

- Less programming, more automation
  - Drag & drop interface construction
  - Demonstration-based programming
  - New: Machine learning and data mining
- Human-Computer Interaction (HCI) methods
  - User studies, summer schools, informal & formal comparison studies
- Exploit tools already in use
  - Component-based architecture with "standard" tool-tutor protocol
  - Off-the-shelf tools and languages (e.g., Netbeans, Eclipse, Flash, Jess)

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:
  \[ \frac{1}{4} + \frac{1}{6} = \frac{3}{12} + \frac{2}{12} = \frac{5}{12} \]
  \[ \frac{1}{4} + \frac{1}{6} = \frac{6}{24} + \frac{4}{24} = \frac{10}{24} = \frac{5}{12} \]

Authoring an Example-Tracing Tutor

1. **Create a User Interface**
   - Create the graphical user interface (GUI) used by the student

2. **Demonstrate Behavior**
   - Demonstrate correct, alternative correct, and incorrect solutions

3. **Annotate the Graph**
   - Annotate solutions steps in the resulting "behavior graph" with:
     - hint messages,
     - error messages,
     - labels for concepts or skills associated with actions

4. **Generalize**
   - Specify how demonstrated behavior could vary within given problem
     - allowed order of steps
     - allowed variants for a given step (formulas, ranges, regexps)

Test and Iterate on Steps 1-4 ...

1. Create student interface with GUI builder
   - **NetBeans IDE**
1.a. Alternative way of building interfaces: Flash

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

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
  \[
  \frac{1}{4} + \frac{1}{6} = \frac{3}{12} + \frac{2}{12} = \frac{5}{12}
  \]
  \[
  \frac{1}{4} + \frac{1}{6} = \frac{6}{24} + \frac{4}{24} = \frac{10}{24} = \frac{5}{12}
  \]
- When creating behavior graph for *isomorphic* problems:
  \[
  \frac{1}{6} + \frac{3}{8} = \frac{4}{24} + \frac{9}{24} = \frac{13}{24}
  \]
  \[
  \frac{1}{6} + \frac{3}{8} = \frac{8}{48} + \frac{18}{48} = \frac{26}{48} = \frac{13}{24}
  \]
- And when creating behavior graph for *near-isomorphic problems*:
  \[
  \frac{1}{6} + \frac{1}{10} = \frac{5}{30} + \frac{3}{30} = \frac{8}{30} = \frac{4}{15}
  \]
  \[
  \frac{1}{6} + \frac{1}{10} = \frac{10}{60} + \frac{6}{60} = \frac{16}{60} = \frac{4}{15}
  \]

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)

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


Outer loop: problem selection options offered by ITS

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


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

<table>
<thead>
<tr>
<th>Project Title</th>
<th>Domain</th>
<th>Studies</th>
<th>Students</th>
<th>Instructional Time</th>
<th>Development Time</th>
<th>Time Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Improving Skill at Solving Equations through Better Encoding of Algebraic Concepts</td>
<td>Middle and High School Math - Algebra</td>
<td>256</td>
<td>16 mins each for 2 conditions</td>
<td>~120 hrs</td>
<td>240:1</td>
<td></td>
</tr>
<tr>
<td>Using Elaborated Explanations to Support Geometry Learning</td>
<td>Geometry</td>
<td>90</td>
<td>30 mins</td>
<td>~2 months</td>
<td>720:1</td>
<td></td>
</tr>
<tr>
<td>The Self-Assessment Tutor</td>
<td>Geometry - Angles, Quadrilaterals</td>
<td>67</td>
<td>45 mins</td>
<td>~9 weeks</td>
<td>540:1</td>
<td></td>
</tr>
<tr>
<td>Enhancing Learning Through Worked Examples with Interactive Graphics</td>
<td>Algebra - Equation Models of Problem Situations</td>
<td>60-120</td>
<td>~3 hrs</td>
<td>~260 hrs</td>
<td>87:1</td>
<td></td>
</tr>
<tr>
<td>Flexibility and Sense Making in Elementary Math Learning</td>
<td>6th Grade Math - Whole-number division</td>
<td>~35</td>
<td>2.5 hrs each for 2 conditions plus 1 hr of assessment</td>
<td>~4 months</td>
<td>107:1</td>
<td></td>
</tr>
<tr>
<td>The Fractions Tutor</td>
<td>6th Grade Math - Fraction Conversion, Fraction Addition</td>
<td>132</td>
<td>2.5 hours each for 4 conditions</td>
<td>12 weeks</td>
<td>48:1</td>
<td></td>
</tr>
<tr>
<td>Effect of Personalization and Worked Examples in the Solving of Stoichiometry</td>
<td>Chemistry - Stoichiometry</td>
<td>223</td>
<td>12 hrs</td>
<td>1016 hrs</td>
<td>85:1</td>
<td></td>
</tr>
</tbody>
</table>

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
  – Number of “how to” lectures
  – In your project you could decide to focus mainly on example-tracing tutors
  – (Then again, this is your chance to get some mentoring as you build a Cognitive Tutor)
• If you are not in the CTAT track, but interested in learning to build tutors in limited time, it is best to focus on example-tracing tutors

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.

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

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

<table>
<thead>
<tr>
<th>Unit</th>
<th>$10</th>
<th>$1</th>
<th>$0.10</th>
<th>$0.01</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Dollars: \[ \text{memberOf(input,0,1,2)} \]

Pennies: \[ \text{memberOf(input,0,100,200)} \]

Pennies: \[ 200 - 100 \times \text{link7.input} \]

Dollars: \[ \text{round}(2 - \text{link18.input}/100) \]