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

Vincent Aleven and the CTAT team
8th Annual LearnLab Summer School
Pittsburgh, August 6-10, 2012

Overview

- What is “a tutor?”
  - What are essential characteristics of intelligent tutoring systems?
- Use of CTAT be used to author tutors?
  - Motivation
  - Basic approaches
  - Short movie of authoring with CTAT
  - Examples of projects that have used CTAT
- Evidence of authoring efficiency with CTAT

If you are not in the CTAT track, why might this talk still be of interest?

- Intelligent Tutoring Systems are an effective and increasingly important educational technology
  - Ask President Obama!
- CTAT relevant to most other tracks:
  - In Vivo: could do an in vivo experiment using CTAT-based tutors as research platform (happens all the time!)
  - EDM/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
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

What is an “Intelligent Tutoring System” (ITS)?

- An advanced learning technology
  - Supports “learning by doing” with step-by-step guidance
  - Supports “tutored problem solving”
- Uses artificial intelligence techniques to
  - Provide human tutor-like behavior
  - Be more flexible, diagnostic & adaptive
- Components of an ITS:
  - Interface or problem solving environment, domain knowledge, student model, pedagogical (tutoring) knowledge

Cognitive Tutor Algebra

- Analyze real world problem scenarios
- Tracked by knowledge tracing
- Use graphs, graphics calculator
- Use table, spreadsheet
- Use equations, symbolic calculator
- Model tracing to provide context-sensitive Instruction

Cognitive Tutor Geometry
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


Inner loop options: within-problem guidance offered by ITS

| + | Minimal feedback on steps (classifies steps as correct, incorrect, or suboptimal) |
| + | Immediate |
| + | Delayed (two kinds) |
| + | 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

<table>
<thead>
<tr>
<th>(+)</th>
<th>Student picks</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>Fixed sequence</td>
</tr>
<tr>
<td>−</td>
<td>Mastery learning</td>
</tr>
<tr>
<td>+</td>
<td>Macro-adaptation (called “Mastery learning” in CTAT and Cognitive Tutor research)</td>
</tr>
</tbody>
</table>


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

![Graph showing time to complete programming problems in LISP Tutor](image)

- Immediate Feedback
- Error Flagging
- Demand Feedback
- No Feedback

- Time to Complete Problems in LISP Tutor
- Immediate Feedback vs Student-Controlled Feedback

### Kinds of Computer Tutors

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

**Can be built with CTAT**

### 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)

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: Eclipse, 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

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)
Some CTAT tutors used in research

Thermo-dynamics

Elementary Math

French (intercultural competence)

Mathtutor: free web-based tutors for middle-school math

Vincent Aleven, Bruce McLaren

Mathtutor: http://mathtutor.web.cmu.edu

In vivo study: Blocked vs interleaved practice with multiple representations

Martina Rau, Nikol Rummel, Vincent Aleven

In vivo study: Correct and incorrect worked examples in Algebra learning

Julie Booth, Ken Koedinger

Study Design

Self-Explanation of Correct Examples

Self-Explanation of Incorrect Examples

CTAT tutors interleaved with Carnegie Learning Cognitive Tutor

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
Some tutors for you to look at

- French culture
  - http://www.andrew.cmu.edu/user/aeo/tutor/ibrahim-exp.swf
  - http://www.andrew.cmu.edu/user/aeo/tutor/ibrahim-controlV2.swf

- Stoichiometry

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 – i.e., occur on a solution path that all accepted prior actions are on

Example: Use of formulas

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

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

Pennies: \( =200-100*\text{link7.input} \)

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

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 500 CTAT users in summer schools, courses, workshops, research, and tutor development projects
  - Domains: mathematics, chemistry, genetics, French culture, Chinese, ESL, thermodynamics
  - At least 44 research studies built tutors and deployed in real educational settings

- In the past two years
  - CTAT was downloaded 6,600 times
  - the CTAT website drew over 2.9M hits from 164k unique visitors
  - URL: http://ctat.pact.cs.cmu.edu

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>Development 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>3</td>
<td>268</td>
<td>16 mins each for 2 conditions</td>
<td>~120 hrs</td>
<td>240:1</td>
</tr>
<tr>
<td>Using Elaborated Explanations to Support Geometry Learning</td>
<td>Geometry</td>
<td>1</td>
<td>90</td>
<td>30 mins</td>
<td>~2 months</td>
<td>720:1</td>
</tr>
<tr>
<td>The Self-Assessment Tutor</td>
<td>Geometry - Angles, Quadrilaterals</td>
<td>1</td>
<td>67</td>
<td>45 mins</td>
<td>~9 weeks</td>
<td>540:1</td>
</tr>
<tr>
<td>Enhancing Learning Through Worked Examples with Interactive Graphics</td>
<td>Algebra - Equation Models of Problem Situations</td>
<td>1</td>
<td>60-120</td>
<td>~3 hrs</td>
<td>~260 hrs</td>
<td>87:1</td>
</tr>
<tr>
<td>Fluency and Sense Making in Elementary Math Learning</td>
<td>4th-Grade Math - Whole-number division</td>
<td>1</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>
</tr>
<tr>
<td>The Fractions Tutor</td>
<td>6th-Grade Math - Fraction Conversion, Fraction Addition</td>
<td>1</td>
<td>132</td>
<td>2.5 hours each for 4 conditions</td>
<td>12 weeks</td>
<td>48:1</td>
</tr>
<tr>
<td>Effect of Personalization and Worked Examples in the Solving of Stoichiometry</td>
<td>Chemistry Stoichiometry</td>
<td>4</td>
<td>223</td>
<td>12 hrs</td>
<td>1016 hrs</td>
<td>85:1</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

That's all for now!