What can CTAT do for you?
Overview of the CTAT track

Vincent Aleven and the CTAT team
4th Annual PSLC LearnLab Summer School
Pittsburgh,
July 7-11, 2008

Overview

• What is “a tutor?”
  - How is it different from other forms of computer-based instruction?
  - How do we know tutors work?
• 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:
  - TuTalk: TuTalk dialogue tutor agent and CTAT intelligent tutor would make an awesome duo!
  - In Vivo: could do an in vivo experiment with CTAT-based tutors [happens all the time!]
  - Data Shop: many data sets in the Data Shop were generated using CTAT-built tutors
  - TagHelper: Hmm... think-aloud data of how people learn with tutors may bolster a theoretical explanation for differences in robust learning caused by different tutor versions used in an in vivo experiment
• Track hopping is allowed!
  - E.g., if in the In Vivo track, could attend CTAT sessions

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

Skills

- Problem solving
- Strategy selection
- Conceptual reasoning
- Strategic reasoning
- Algorithmic reasoning
A simple example: Fraction Addition

• Interact with the fraction addition tutor on the CTAT website ...

ITSs v. Other Instruction

• First, compare Conventional Teaching and Computer-Aided Instruction (CAI)

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 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

Computer Aided Instruction (CAI)

- Underlying technology: Hypertext & Behaviorism
- Pedagogy: Didactic feedback on answers.
- Example:

  Solve $2+2x=12$
  - $x=5$
  - $x=3$
  - $x=7$

  Multiplication has a higher precedence than addition, so $2+2x$ is the same as $2+(2x)$, not $(2+2)x$. Try again.

  Excellent!

Simple CAI Example

- Tutor: Solve $2+2x=12$
- Student: $4x=12$
- Tutor: Not quite. Try again.
- Student: $2x=12-2$
- Tutor: Good!

Intelligent Tutoring Systems (ITS)

- Underlying technology: Artificial intelligence & cognitive psy
- Pedagogy: Assistance on problem solving steps, not just final answers.
- Example:

  - Tutor: Solve $2+2x=12$
  - Student: $4x=12$
  - Tutor: Not quite. Try again.
  - Student: Clicks on “hint” button
  - Tutor: Think about operator precedence.
  - Student: $2x=12-2$
  - Tutor: Good!

The nested loops of 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)

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

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

**Kinds of Computer Tutors**

- **Intelligent tutoring systems**
  - e.g., Sherlock

- **Model-tracing tutors**
  - e.g., Andes

- **Constraint-based tutors**
  - e.g., SQL Tutor

- **Example-tracing tutors**
  - e.g., Stoichiometry, French Culture Tutor

Can be built with CTAT

**Cognitive Tutors: Real-world success of Intelligent Tutoring Systems technology**

- **Cognitive Tutor Courses**
  - Goal: Improve quantitative literacy & fluency
  - Computer-based one-to-one tutoring
    - Based on computational models of student thinking
    - Based on the concept of Model Tracing

- **Used by many students in many schools**
  - Algebra Cognitive Tutor: In over 2,000 schools in the USA, 300,000 students per year
  - Geometry Cognitive Tutor: In 350 schools
    - Most widely used intelligent tutoring system?

- University created a company, Carnegie Learning, to disseminate

**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.


**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

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

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

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

3(2x - 5) = 9

6x - 15 = 9
2x - 5 = 3
6x - 5 = 9

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

CTAT motivation: Make tutor development easier and faster!

• Development costs of instructional technology are high
  - E.g., ~300 dev hours per hour of instruction for Computer Aided Instruction (Murray, 1999)

• 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

• Solution: Easy to use Cognitive Tutor Authoring Tools (CTAT) - makes authoring possible for non-programmers and 4 to 8 times more cost effective


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)

CTAT supports two types of tutors

- Cognitive Tutors and Example-Tracing Tutors
- Behavior is similar though not identical
  - Cognitive Tutors handle more sophisticated dependencies between steps, and more subtle ordering constraints
- Technology and development are very different
  - Example-Tracing Tutors: built through “programming by demonstration”
  - Cognitive Tutors: built by AI programming (much harder)


Movie Showing How an Example-Tracing Tutor is built
Authoring an Example-Tracing Tutor

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

Step 2: Demonstrate Behavior
- Demonstrate correct, alternative correct, and incorrect solutions

Step 3: Generalize
- Specify how demonstrated behavior could vary within given problem
  - allowed order of steps
  - allowed variants for a given step

Step 4: Annotate the Graph
- Annotate solutions steps in the resulting “behavior graph” with:
  - hint messages,
  - error messages,
  - labels for concepts or skills associated with actions

Test and Iterate on Steps 1-4 ...

Step 5: Publish Tutor to Web

Vote-with-your-feet Evidence of CTAT’s Utility

- Over 350 people have used CTAT
- CTAT tutors in regular use at CMU and elsewhere
- CTAT-built tutors were used in 20 research studies in real educational settings
- CTAT is freely available on the web
- 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.

http://ctat.pact.cs.cmu.edu

CTAT Architecture
Chinese: tone study
Wang, Perfetti, Liu, et al.

Stoichiometry study
McLaren et al.

Behavior Graph for Stoichiometry Tutor

Geometry self-assessment tutor
Roll et al.

French culture
Ogan et al.

Help seeking
Aleven, McLaren, Roll, et al.

French
Jones et al.

Chinese: listen and read
Mei, Haney, et al.
Division Tutor (Stefan King)

Fractions tutor (Martina Rau)

Middle-School Math Tutor Website

Self-explanation of incorrect examples
Julie Booth et al.
Another example - Thermodynamics

Simulator hooked up with CTAT

CTAT provides tutoring for analyzing power plant efficiency

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

Thank you!