Creating example-tracing tutors using the Cognitive Tutor Authoring Tools

Vincent Aleven

7th Annual PSLC Summer School

Pittsburgh, July 25 - 29, 2011

CTAT CognitiveTutorAuthoringTools

LearnLab Pittsburgh Science of Learning Center

General remarks

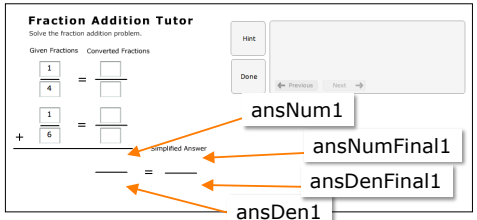
- CTAT is still being developed – we welcome your feedback
- CTAT is available for free for research purposes
- If you'd like to try CTAT in a research project, we can provide various kinds of help as you get started
- Likewise, the DataShop team welcomes and supports external users

Part 1: Working with CTAT's basic features

- Interface building
  - "Making thinking visible"
  - Drag-and-drop interface building
    - Illustrated in Flash, but can also be done in Java
- Editable behavior graph
  - Create single-path (lcm) graph through programming by demonstration
  - Relaxing ordering constraints
  - Multiple paths (product)
  - Demonstrate error(s) and mark them in graph
  - Add hints and error messages
  - Graph editing

Hands on: Edit interface
(using the Flash IDE)

- In the Flash IDE, open file fractionAddition-incomplete.fla
- Make sure you have the component palette (Windows > Components)
- Add four text areas to the canvas for the sum fraction and the final answer fraction
- Name the components (Windows > Properties)
- Embellish the interface?



Hands on: Create behavior graph
Example: 1416-lcd-prod.brd

- Start "CTAT for Flash"
- Run interface (Control > Test Movie); use the one you made, or use fractionAddition-noskills.fla
- Create a behavior graph for a new fraction addition problem
- Enter start state (Graph > Create Start State)
- Demonstrate one way of solving the problem
- Test tutor – switch to Test Tutor mode, enter correct & incorrect steps, ask for hints
- Set top-level group to "unordered" (r-click on "Top Level" in the Group Editor)
- Test tutor (explore how the change affects tutor behavior)
- Switch to Demonstrate mode
- Demonstrate a 2nd strategy (different denominator)
- Test tutor
- Add some hints
- Demonstrate some errors, add error messages

Part 2: Using CTAT's advanced features

- Dynamic interfaces
  - Tutor takes over some of the "grunt work"
  - Reveal interface components at opportune moments during problem solving
- Greater flexibility in matching student behavior:
  - Fine-grained ordering constraints
  - Formulas
- Tracking students' skill growth over time
  - Labeling steps with knowledge components
  - Adding a "skill meter" to the tutor interface
- Presenting tutor problems as (partially) worked examples

Hands on: Dynamic Interfaces
Example: 1416-lcd-prod-tpa.brd

- Make the tutor copy the common denominator; i.e., make tutor perform actions on secDenConv and ansDen1
- Open "Edit Student Input Matching" on the relevant links
- Set Actor to "Any"
- Do this in both branches
- Extra credit: make tutor copy the denominator regardless of whether the student enters the first or second one first
- Hint: need to have separate paths for 1st denom – 2nd denom and 2nd denom – 1st denom, but these paths can join

Hands-on: Dynamic interfaces challenge problem

(not illustrated in example file – ask us for help)

- Make the tutor "reveal" the components as they are needed
  - Initially, only the components for the converted fractions are visible
  - After converting, the components for the unreduced sum fraction become visible
  - After entering the sum fraction, the components for the reduced sum become visible
- Insert 2 blank links immediately below the links for converting the fractions
- Edit these links (Edit Student Input Matching) so they make components visible: set selection to component name (ansNum1, ansDenom1) action to SetVisible, and input to true (for Demonstrated Value and for Matcher Input); and set Actor to Tutor
- Need to hide components in the start state

Greater flexibility in matching student behavior

1416-lcd-prod-tpa-groups.brd

- Add ordering constraints to your graph (e.g., so that student must convert first, then add, then reduce)
- Do this within each branch
- Top-level must be ordered
- Lower-level groups must be unordered

1416-lcd-any-common-mult-tpa-groups.brd

- Make the tutor accept any common multiple of the two denominators
- You need to consider the order in which steps may be executed; change groups so first action is denominator
- In Edit Student Input Matching dialogue, select "Formula Match" for input

Example formula (for converted denominator)

```
and (not(lessThan(input, firstDenGiven*secDenGiven)),
     equals(mod(input,firstDenGiven),0),
     equals(mod(input,secDenGiven),0))
```


Hands on: Adding skills and knowledge tracing

1416-lcd-any-common-mult-tpa-groups-skills.brd
FractionAddition-finished.fla

- Set up tutor to track skill mastery across problems
- Need to update interface: add a CommSkillBar
- Make space by enlarging the canvas and the CommShell
- To enable the tutor to map problem-solving steps to knowledge components, must attach knowledge component labels to links in the graph
- Turn on display of skill labels (Graph > Show Skill Labels)
- Add labels to Links (click on skill label > Edit Skill Name)
- Skills should show up in Skil Meter and be updated by tutor depending on correctness of student problem solving