All you need to know about ITS (in 60 slides)

Created by Kurt VanLehn ©

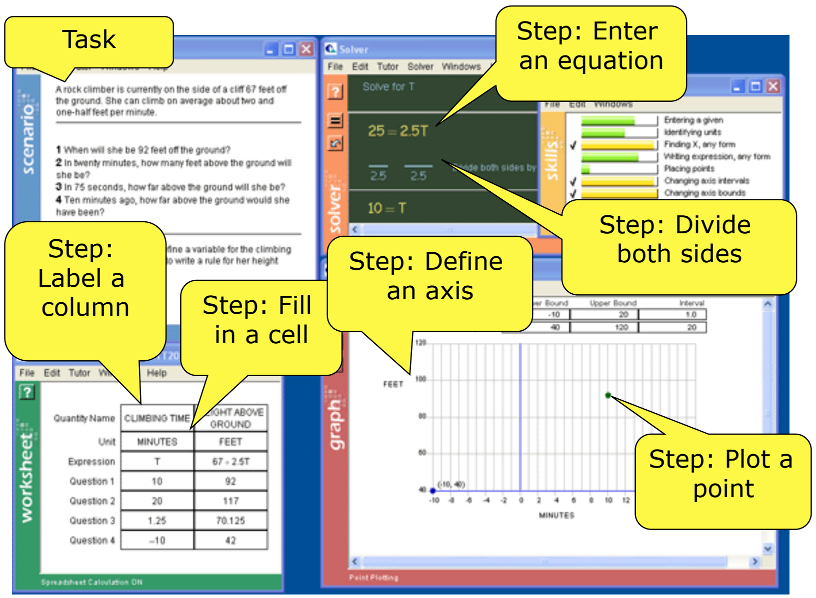
Presented by Vincent Aleven

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

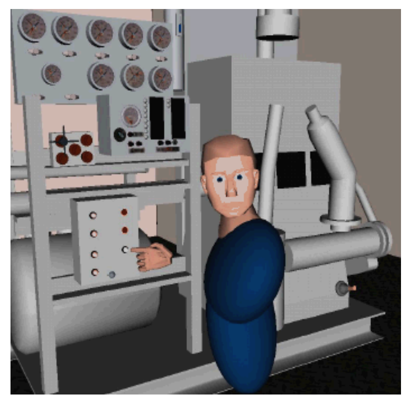
Some Key Terms

- **Task domain**: The information and skills being taught by the tutor
- **Task (or Problem)**: A multi-minute activity that can be skipped or interchanged with other tasks.
- **Step**: Completing a task takes many steps. Each is a user interface event.

Cognitive Algebra I Tutor (Carnegie Learning)

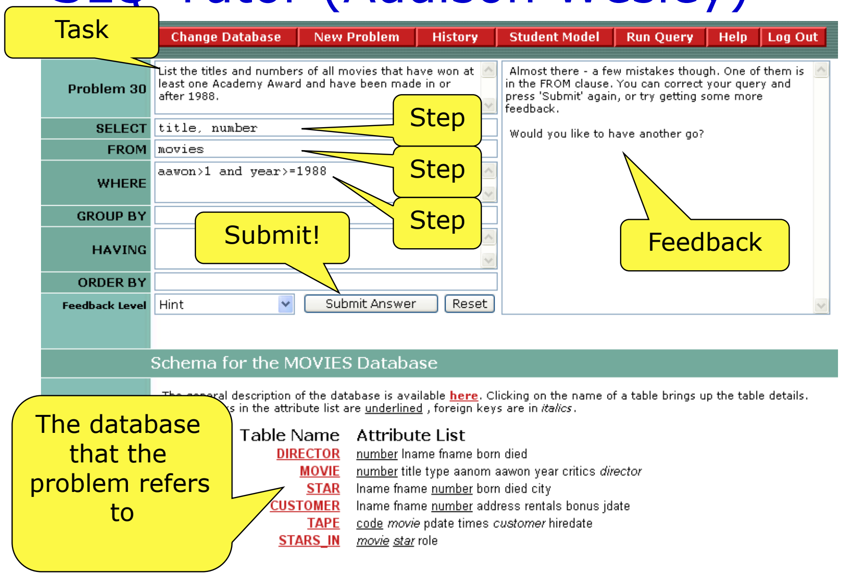


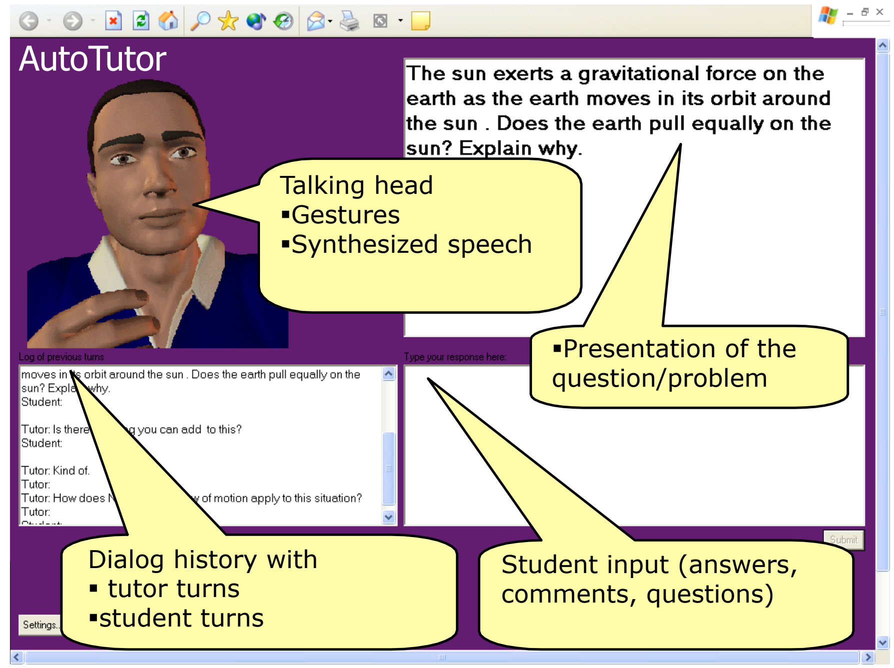
Steve coaches procedural skills

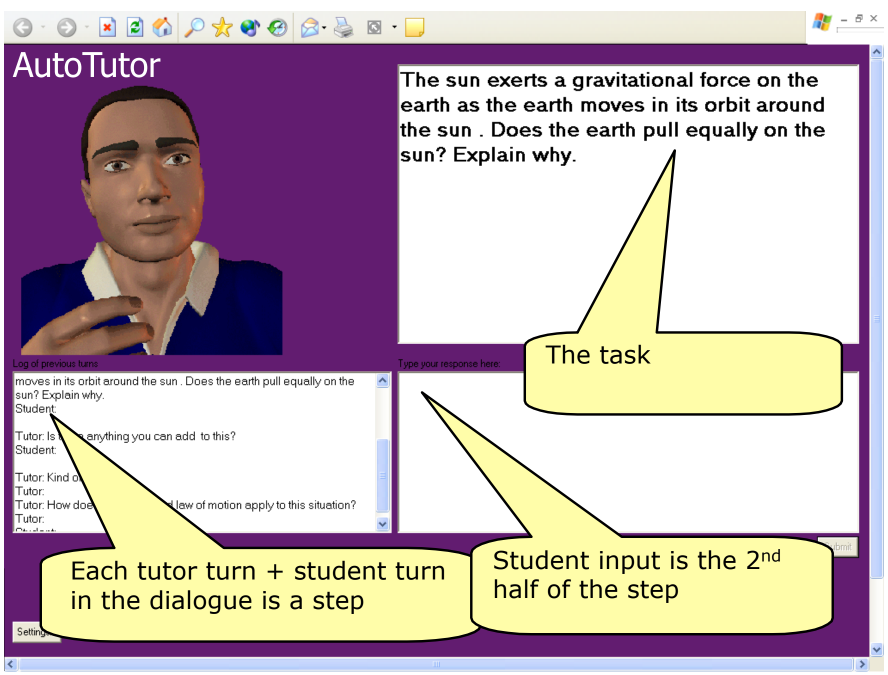


- Task, e.g.,
  - Start the air compressor
- Steps, e.g.,
  - Push a button
  - Check a dipstick
  - Read a dial

SLQ-Tutor (Addison Wesley)







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

Outer Loop

Inner Loop

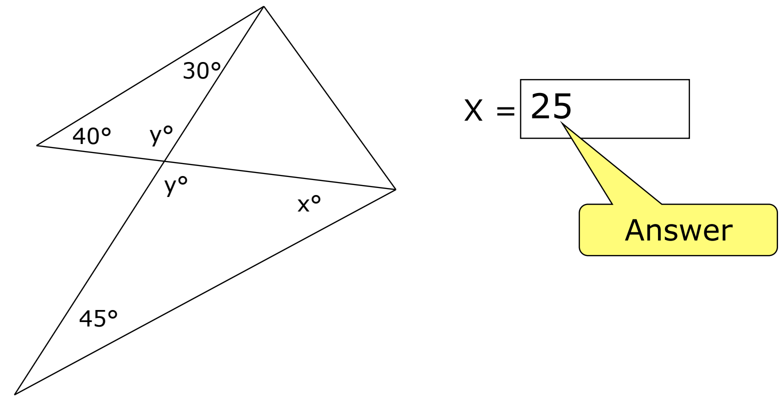
Terminology

Intelligent Tutoring Systems have “nested loops” (VanLehn, 2006)

- Outer loop over tasks
  - Task selection
- Inner loop over steps
  - Within-problem guidance (hints, feedback, end-of-problem reflection)

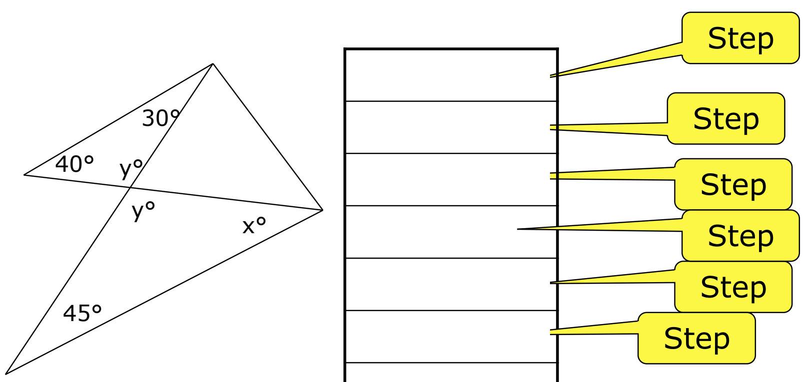
Tutoring systems with outer loop only:

Called computer aided instruction (CAI), CBT…



What is the value of x?

Tutoring systems with both inner & outer loops: Called intelligent tutoring system (ITS)

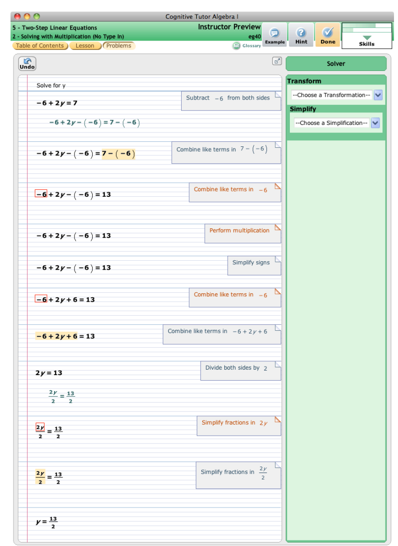


What is the value of x?

Inner loop

Step-by-step guidance

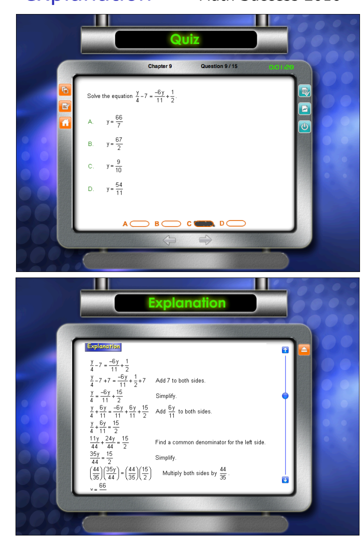
Cognitive Tutor Algebra



No inner loop

Multiple choice, end-of-quizz explanation

Math Success 2010



"Systems that lack an inner loop are generally called Computer-Aided Instruction (CAI), Computer-Based Training (CBT) or Web-Based Homework (WBH). Systems that do have an inner loop are called Intelligent Tutoring Systems (ITS)."

VanLehn, 2006 (p. 233)

Outline: Design issues for ITS

- Outer loop
  - **How to select the next task**
- Inner loop

Next

Outer Loop: Common Task Selection Techniques

1. Student selects task from a menu
2. Prescribed task sequence
3. Mastery learning
4. Macro-adaptation

1. Student selects task from menu

- Easy way of giving instructor control over task assignment
  - Different instructors assign different tasks
  - Can assign different students different tasks
  - Speed up / slow down
- Students can explore, do extra work, etc.
- Do students make good task selection decisions?

2. Follow a prescribed task sequence

- Example
  - Study chapter
  - Study 3 examples
  - Solve 6 problems with help from system
  - Solve 3 problems without help
  - Study next chapter
  - Etc.
- Student must complete the work on one chapter before moving to the next
- No individualization

3. Mastery Learning

- For each chapter ***C*** do:
  - Study text and examples of chapter ***C***
  - Do each required task of ***C***
  - Until mastery is reached, do another task of ***C***
- Different students do different numbers of tasks, but all achieve mastery eventually
  - Self-paced learning
- Requires assessing student's competence
  - Inner loop does the assessment, or
  - Tests

4. Macro-adaptation

- Same control structure as Mastery Learning
  - But based on more fine-grained assessment of the student
  - Results in fewer tasks required to reach mastery
- Basic idea: choose a task that requires just a few unmastered knowledge components
  - Tasks are described by the knowledge components that they address
  - System assesses the student's mastery of individual knowledge components
  - System chooses a task that matches the student's needs

In the Cognitive Tutor literature, the term "Cognitive Mastery Learning" is used for the notion that VanLehn calls "Macro-adaptation."

Corbett, A., McLaughlin, M., & Scarpinatto, C. K. (2000). Modeling student knowledge: Cognitive Tutors in high school and college. *User Modeling and User-Adapted Interaction, 10*(2), 81-108.

Student model

Student model = persistent data on the student

  - Lousy name, but traditional
- Which tasks have been done
  - Performance data for these tasks
- Students' knowledge at this moment (i.e., competence so far)
- Learning styles, preferences, etc.
  - E.g., visual vs. verbal
- Demographics (e.g., SAT, College Major)

How to represent student's knowledge? Depends on the task selection method!

- Macro-adaptive task selection
  - Need mastery of each knowledge component
- Mastery learning
  - Which chapter is being studied now
  - Competence (a single number) on this chapter
- Choose next task on the list
  - Student model has just a pointer into the list
- Student chooses the next task
  - No student model of knowledge needed

- Other uses of student model: tailoring hints, feedback; personalization of cover stories; politeness strategies

Outline: Design issues in ITS

- Outer loop
  - How to select the next task
- Inner loop
  - **Minimal feedback on a step** Next
  - Hints on the next step
  - Error-specific feedback on an incorrect step
  - Assessment of student's knowledge.
  - Review of the student's solution

Inner loop offers within-problem (step-by-step) guidance

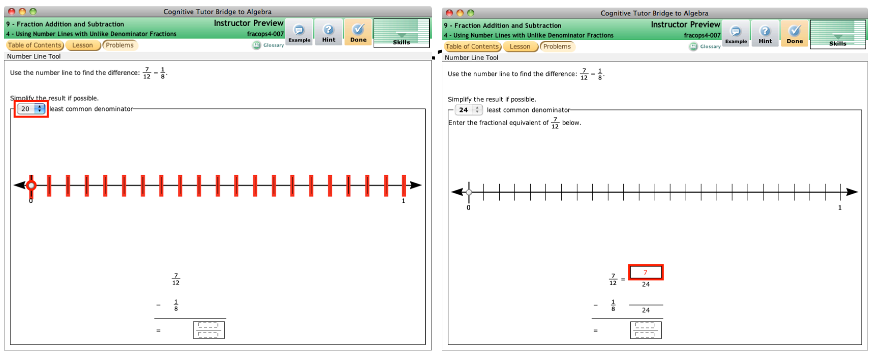
- Designed to help students learn: Scaffolding / guidance
- Common features
  - Minimal feedback (correct/incorrect) on a step
  - Error-specific feedback on an incorrect step
  - Hints on the next step
  - Assessment of student's knowledge
  - Review of the student's solution

- Warning: Any tutor feature can be misused!
  - Often called "gaming the system"

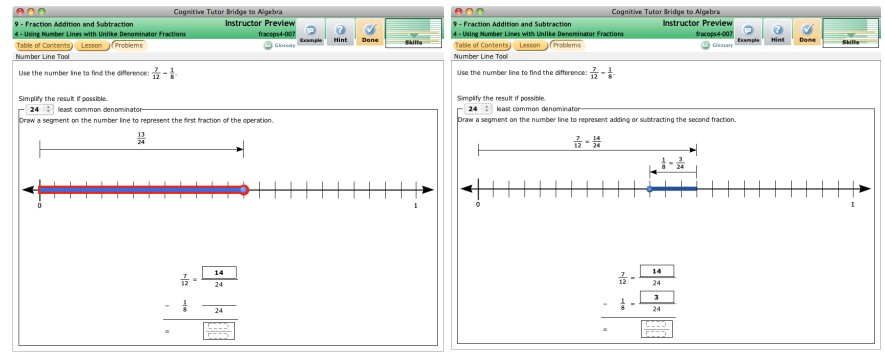
Minimal feedback: How?

- Andes: Color the step red or green
- Steve: Shake head "yes" or "no"
- AutoTutor: Start tutor's turn with
  - "No…"
  - "Well..."
  - "Okay…"
  - "Yeah…" or
  - "Great!"

Minimal Feedback Examples



More Than Minimal



Student's number line actions are marked right or wrong; additional info about answer is displayed above the number line

Minimal feedback: What counts as incorrect?

- Andes: Should it have Yellow steps for correct but useless equations?
- Sherlock: Better to waste time or waste parts?
  - "Instead of replacing circuit board G1X2, some experts would have tested it. Here's how."
- Pyrenees: Skipping steps is incorrect for novices but allowed for intermediates.

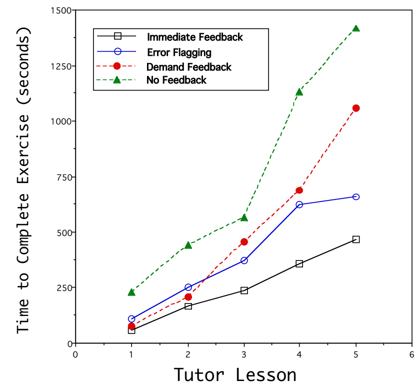
What if the step cannot be classified as correct or incorrect?

- Cognitive tutors and example-tracing tutors: If students step can't be recognized, it's *incorrect*
- SQL tutor: If student's step can't be recognized, it's *correct*

Minimal feedback: When?

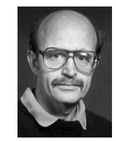
- Fixed policies
  - Immediate – as soon as step is entered
  - Demand – wait until student asks
  - Delayed – wait until problem is submitted
    - Common with real-time skills, e.g., air combat
- Adaptive
  - Fading – changes policy as competence increases
  - Decision theoretic – Decide at each error whether feedback maximizes expected utility

Feedback Studies in LISP Tutor



Time to Complete Programming Problems in LISP Tutor

Immediate Feedback Vs Student-Controlled Feedback



Corbett, A. T., & Anderson, J. R. (2001). Locus of feedback control in computer-based tutoring: Impact on learning rate, achievement and attitudes. In J. Jacko, A. Sears, M. Beaudouin-Lafon, M. & R. Jacob (Eds.), *Proceedings of ACM CHI'2001 Conference on Human Factors in Computing Systems* (pp. 245-252). New York: ACM Press.

Other results on feedback strategies

- Immediate feedback based on a novice-like model of desired performance more effective than intermediate feedback based on an expert-like model (Mathan & Koedinger, 2005)
  - Difference: experts catch their errors even before they get feedback from the environment, whereas (outstanding) novices catch them as soon as they get feedback from the environment
- Delayed feedback in a simple Lisp tutor better than immediate feedback (Schooler & Anderson, 1990)
- More!

Outline: Design issues in ITS

- Outer loop
  - How to select the next task
  - How to obtain a large set of tasks
- Inner loop
  - Minimal feedback on a step
  - **Hints on the next step**
  - Error-specific feedback on an incorrect step
  - Assessment of student's knowledge.
  - Review of the student's solution

Next

Next step hints: How?

- Hint sequences
  - Forward button prints next hint
  - Back button
  - Done button lets student go back to entering steps
- Start in middle of sequence (contingent tutoring)
- Dialogues (usually menu-based) e.g.,
  - Andes: What quantity is the problem seeking?
  - Student: Acceleration of the car at T2
  - Andes: Good. What physics principle will you use?
  - Student: Newton's second law.
  - Andes: Good. You've already done the first step of applying Newton's law. The second step is…

Next step hints: When? Some fixed policies are…

- Andes:
  - Give hint only when the student asks
- Cognitive Tutors:
  - Give hint only when the student asks
  - When students has failed several times and has not yet asked for a hint, give them one
- Steve:
  - Before even attempting the step, if the step is new or student failed miserably last time, then give them a hint.

Next step hints: Which step to hint?

- A correct step, and
- A step not yet done by student, and
- A step that the instructor would prefer, and maybe
- A step that is consistent with the student's plan
  - Requires plan recognition (can be hard AI problem)
  - Pedagogical benefit unknown
    - Would make a good in vivo experiment

Hints in CTAT example-tracing tutors

- How?
  - Click hint, navigate through hint levels starting at level 1
- When?
  - When student asks
- Which step?
  - Step has to be on viable solution path
  - Must not have been completed
  - Has to observe ordering constraints
  - If previous student action was error, give hint on the error step (unless out of order)
  - Otherwise, prefer hint on current selection
  - Otherwise, prefer hint on preferred path out of current state

Learner Control: Good or Bad?

Influential view:

- ITS as 'coach' paradigm: The system should intervene as little as possible (Burton & Brown, 1982).
- Many systems provide on-request help.

However:

- Balancing use of help and errors in important (Wood & Wood, 1999).
- Higher ability students are better help seekers (Wood & Wood, 1999).
- Learner control may mean that those who need help the most, may be the least likely to receive it in time.

Wood, H., & Wood, D. (1999). Help seeking, learning and contingent tutoring. *Computers & Education, 33(2/3), 153-169.*

Studies showing positive relations between help or help seeking and learning

- In a simple tutor for propositional logic, when on-demand hints were available, the students attempted and completed more problems and had improved learning outcomes (Stamper, et al., 2011)
- Help use relates positively with learning in a large data set from a tutor that helps elementary students learn to read (Beck et al., 2008)
- Time spent with bottom-out hint relates positively to learning – use of hints to create examples and self-explain them? (Shih et al., 2008)
- Immediate feedback on help seeking (by "Help Tutor" nested within Cognitive Tutor) leads to lasting improvement in help-seeking behavior with tutor (Roll et al., 2011)
- More!

Students often use help (seemingly?) ineffectively

References on hints and help seeking

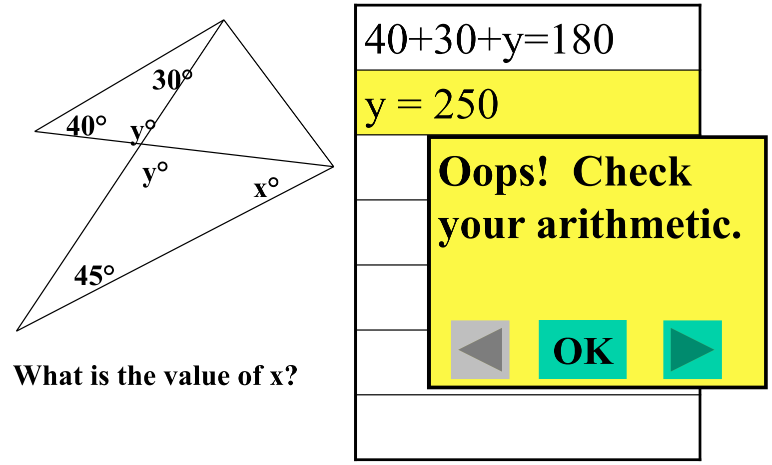
- Aleven, V., & Koedinger, K. R. (2000). Limitations of student control: Do students know when they need help? In G. Gauthier, C. Frasson, & K. VanLehn (Eds.), *Proceedings of the 5th international conference on intelligent tutoring systems, ITS 2000* (pp. 292-303). Berlin: Springer.
- Aleven, V., Stahl, E., Schworm, S., Fischer, F., & Wallace, R.M. (2003). Help seeking and help design in interactive learning environments. Review of Educational Research, 73(3), 277-320.
- Beck, J. E., Chang, K., Mostow, J., & Corbett, A. T. (2008). Does help help? Introducing the Bayesian evaluation and assessment methodology. In B. P. Woolf, E. Aimeur, R. Nkambou, & S. Lajoie (Eds.), Proceedings of the 9th international conference on intelligent tutoring systems, ITS 2008 (pp. 383-94). Berlin: Springer.
- Roll, I., Aleven, V., McLaren, B. M., & Koedinger, K. R. (2011). Improving students' help-seeking skills using metacognitive feedback in an intelligent tutoring system. *Learning and Instruction, 21*(2), 267-280.
- Shih, B., Koedinger, K. R., & Scheines, R. (2008). A response time model for bottom-out hints as worked examples. In R. S. J. d. Baker, T. Barnes, & J. Beck (Eds.), Educational Data Mining 2008: 1st International Conference on Educational Data Mining, Proceedings (pp. 117-26). Montreal, Canada.
- Stamper, J., Eagle, M., Barnes, T., & Croy, M. (2011). Experimental evaluation of automatic hint generation for a logic tutor. In J. Kay, S. Bull, S. & G. Biswas (Eds.), *Proceeding of the 15th International Conference on Artificial Intelligence in Education (AIED2011)* (pp. 345-352). Berlin: Springer Verlag.

Outline: Design issues in ITS

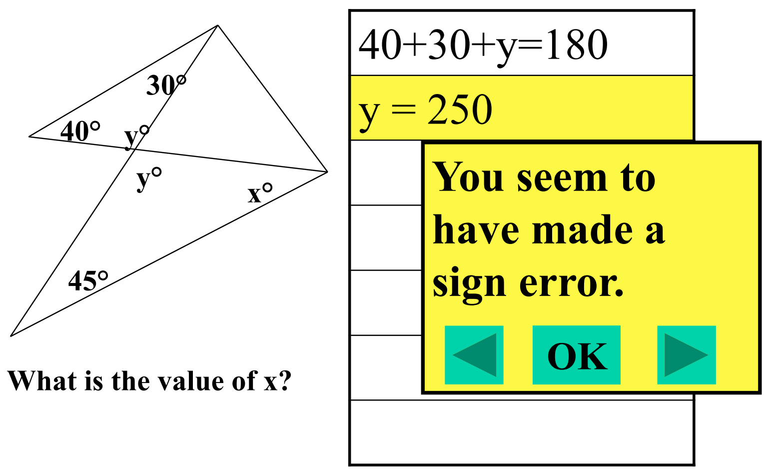
- Outer loop
  - How to select the next task
- Inner loop
  - Minimal feedback on a step
  - Hints on the next step
  - **Error-specific feedback on an incorrect step**
  - Assessment of student's knowledge.
  - Review of the student's solution

Next

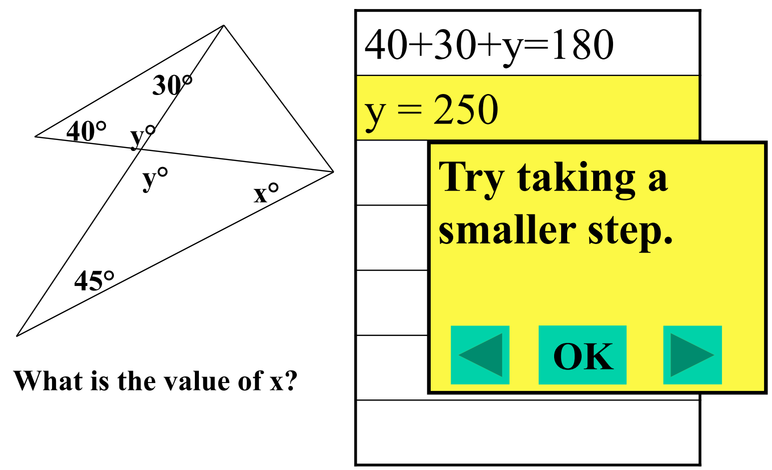
Error specific feedback: How? (1) Signal error, explain what's wrong and perhaps why



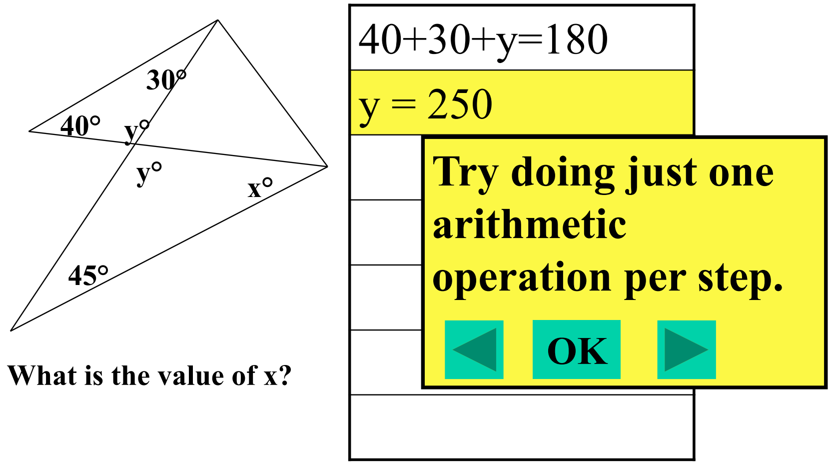
(2) Error-specific feedback becomes more specific



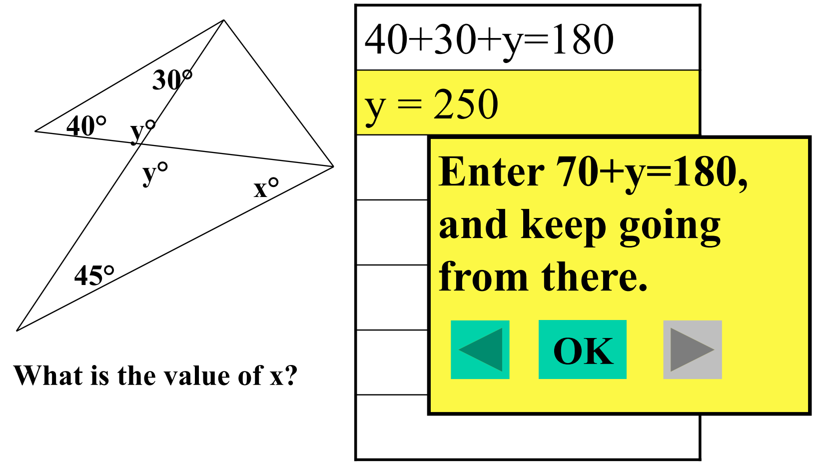
(3) Messages segue from error specific feedback to next-step hinting



(4) Next-step hints become more specific



(5) A *bottom-out hint* is the last hint, which tells the student what to enter.



Error specific feedback: How to recognize errors?

- Many error recognizers must be constructed by…
  - Hand-authored from student errors
  - Collected from DataShop!
  - Induced from student errors via machine learning
  - Theory-based error generators
- Only worth doing if error-specific feedback is more effective than minimal feedback & next-step hints alone
  - Are errors corrected during the error-specific hints in a hint sequence?
  - Do errors corrected then re-occur less? Self-corrected more?
  - Another good in vivo experiment

Results on feedback content

- In geometry proof tutoring, reminding students of their current goals supported both performance and learning more effectively than pointing out rule conditions that are not satisfied (McKendree, 1990)
- More abstract messages support performance better for students with high cognitive development, more concrete messages for students of lower cognitive development (Arroyo et al., 2000)
- Prompted self-explanation after errors that elicits rule-problem mapping more effective than correctness feedback only (Butcher & Aleven, 2010; under review)
- More!

References on feedback

- Arroyo, I., Beck, J. E., Woolf, B. P., Beal, C. R., & Schultz, K. (2000). Macro-adapting Animalwatch to gender and cognitive differences with respect to hint interactivity and symbolism. In In G. Gauthier, C. Frasson, and K. VanLehn (Ed.), *Proceedings of the 5th International Conference on Intelligent Tutoring Systems, ITS 2000* (pp. 574-583). Berlin: Springer.
- Butcher, K. R., & Aleven, V. (2010). Learning during intelligent tutoring: When do integrated visual-verbal representations improve student outcomes? In S. Ohlsson & R. Catrambone (Eds.), Proceedings of the 32nd *Annual Meeting of the Cognitive Science Society* (pp. 2888 – 2893). Austin, TX: Cognitive Science Society.
- Mathan, S. A., & Koedinger, K. R. (2005). Fostering the intelligent novice: Learning from errors with metacognitive tutoring. *Educational Psychologist, 40(4), 257-265.*
- McKendree, J. (1990). Effective feedback content for tutoring complex skills. *Human Computer Interaction, 5,* 381-413.
- Schooler, L. J. & Anderson, J. R. (1990). The disruptive potential of immediate feedback. *In Proceedings of the Twelfth Annual Conference of the Cognitive Science Society,* 702-708, Cambridge, MA.

Outline: Design issues in ITS

- Outer loop
  - How to select the next task
- Inner loop
  - Minimal feedback on a step
  - Hints on the next step
  - Error-specific feedback on an incorrect step
  - **Assessment of student's knowledge.**
  - Review of the student's solution

Next

More Terminology

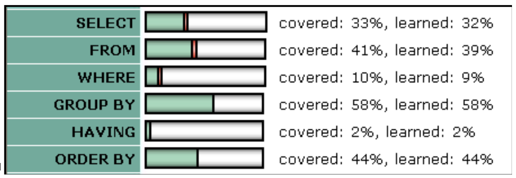
- An ***assessment*** judges a student
  - How competent?
  - What knowledge?
- An ***evaluation*** judges a tutoring system
  - Formative – what needs to be fixed?
  - Summative – how effective compared to control?
  - Parametric – why is it effective?

ITS assessments: Who needs them?

- Tutoring system (maybe)
  - Used by some (not all) outer loop task selectors
  - Used by adaptive feedback & hinting
- Students (definitely)
- Teachers (but rarely for grading)
  - Can't defend what they didn't do themselves.
- "The testing establishment" (not yet)

Assessments: Granularity

- Large grained assessments
  - Single number
  - Based on evidence from a whole course
  - Good for making big decision (e.g., pass/fail)
- Fine grained assessments
  - One number per knowledge component
  - Based on learning events for only that component
  - Good for making small decisions
    - Next task
    - Next hint



Assessments: How?

- Corbett, A. T., & Anderson, J. R. (1995). Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction, 4(4), 253-278.*
- The field of Educational Data Mining has produced many new methods and variations in recent years

Outline: Design issues in ITS

- Outer loop
  - How to select the next task
- Inner loop
  - Minimal feedback on a step
  - Hints on the next step
  - Error-specific feedback on an incorrect step
  - Assessment of student's knowledge.
  - **Review of the student's solution**

Next

Why review solutions?

- Don't want to interrupt the task
  - Real-time skills
  - Immediate feedback and hints have been faded out
  - So students must detect and correct own errors
  - Little headroom for conceptual learning during problem solving?
- Hindsight is often clearer
  - Can see effects of bad strategic choices
- Students more likely to focus on learning, instead of focusing on finishing the task

The review can be a long discussion: Who controls it?

- Tutor led
  - Katz's extension to Andes: Tutor leads a dialogue
- Student led
  - Display a list of the student's steps
  - Display minimal feedback (color) on each one
  - Student can click on any step for a discussion of it
    - Typically clicks on an incorrect step

SQL Tutor exemplifies mixed control of end-of-problem review

Student selects level of review:

1. The number of errors.
2. Minimal feedback (color) on the tutor's favorite error
3. Error-specific feedback on the tutor's favorite error.
4. More detailed error-specific feedback on the tutor's favorite error.
5. All errors: A list containing a description of every error.
6. Partial solution: The correct version of the clause where the tutor's favorite error appeared.
7. Complete solution: The ideal solution to the problem.

Outline: Design issues in ITS

- Outer loop
  - How to select the next task
- Inner loop
  - Minimal feedback on a step
  - Hints on the next step
  - Error-specific feedback on an incorrect step
  - Assessment of student's knowledge.
  - Review of the student's solution

All you need to know …

The End