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

Created by Kurt VanLehn ©

Presented by Vincent Aleven


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

Steve coaches procedural skills

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

\[ x = 25 \]

What is the value of x?

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

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

Outro 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

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)

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

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
  - 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

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

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

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!

Feedback Studies in LISP Tutor

![Graph](image)


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 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.

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
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

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

40 + 30 + y = 180
y = 250

Oops! Check your arithmetic.

What is the value of x?

(2) Error-specific feedback becomes more specific

40 + 30 + y = 180
y = 250

You seem to have made a sign error.

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

40 + 30 + y = 180
y = 250

Try taking a smaller step.
(4) Next-step hints become more specific

40° 30° 45°
40° y° x°
45°

What is the value of x?

40° 30° 45°
40° y° x°
45°

Try doing just one arithmetic operation per step.

OK

40+30+y=180

y = 250

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

40° 30° 45°
40° y° x°
45°

What is the value of x?

Enter 70+y=180, and keep going from there.

OK

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


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

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
- 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 decisions (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?


- 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

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

The End