Wisdom is not the product of schooling but the lifelong attempt to acquire it.
- Albert Einstein

Learner-Centered Design: Beyond “Gift-Wrapping”

Gerhard Fischer and Leysia Palen
Spring Semester 1999

March 29, 1999
Computational Support for Learning

- Intelligent Tutoring Systems
  - Contextualized Tutoring
  - Self-Directed Learning
- Domain-Oriented Design Environments
  - Learning on Demand
  - End-User Modifiability
- Interactive Learning Environments
  - Learner Support
  - Learner Control
Taxonomy of Intelligent Support Systems

- Type of Relationship:
  - tutor
  - advisor
  - critic
  - assistant
  - consultant
  - agent

- Type of Activity:
  - teach
  - guide
  - visualize
  - explain
  - constrain
  - criticize
  - argue
Learning on Demand *between* Tutoring Systems and Open-Ended Learning Environments

- **tutoring systems:**
  - *strength:* teach basic concepts and skills of a problem domain
  - *weaknesses:*
    * cannot closely match the concrete problem solving situations of users
    * learner must relate training to real-world problem situations

- **open learning environments:**
  - *strength:* controlled by the learner
  - *weaknesses:*
    * provide limited support in helping learners detect mistakes or overcome breakdowns
    * misconceptions may accumulate into chains
    * learners get trapped on suboptimal plateaus because they fail to discover the relevant knowledge

- **learning environment for learning-by-doing / learning on demand --->**
  challenge: to combine the problem-solving experience and motivation of discovery learning with the effective guidance of tutorial interactions
How the West Was Won
—
An Environment to Support “Guided Discovery Learning”


• a game (designed for Plato Project) to give students drill and practice in arithmetic

• a coaching system — issues explored:
  - turn mistakes into constructive episodes (the “Skiing Paper”)
  - when to interrupt
  - what to say once after an interruption
  - diagnostic modeling based on a differential model
  - coaching by issues and example
Coaching Principles Based on Pedagogical Strategies

- **Principle 1:** Before giving advice, be sure the Issue used is one in which the student is weak.

- **Principle 2:** When illustrating an Issue, only use an Example (an alternative move) in which the result or outcome of that move is dramatically superior to the move made by the student.

- **Principle 3:** After giving the student advice, permit him to incorporate the Issue immediately by allowing him to repeat his turn.

- **Principle 4:** If a student is about to lose interrupt and tutor him only with moves that will keep him from losing.

- **Principle 5:** Do not tutor on two consecutive moves, no matter what.

- **Principle 6:** Do not tutor before the student has a chance to discover the game for himself.
Coaching Principles — Continued

• **Principle 7:** Do not provide only criticism when the Tutor breaks in! If the student makes an exceptional move, identify why it is good and congratulate him.

• **Principle 8:** After giving advice to the student, offer him a chance to retake his turn, but do not force him to.

• **Principle 9:** Always have the Computer Expert play an optimal game.

• **Principle 10:** If the student asks for help, provide several levels of hints.

• **Principle 11:** If the student is losing consistently, adjust the level of play.

• **Principle 12:** If the student makes a potentially careless error, be forgiving. But provide explicit commentary in case it was not just careless.
Tutoring versus Critiquing

• **Tutoring:**
  - this is what I think you should do
  - problem of coverage is important
  - user has little control -- learning paths are determined by the tutor
  - a sequence of increasing complex microworlds can be constructed a priori

• **Critiquing:**
  - this is what I think of what you have done
  - users must be competent in the subject domain being critiqued
  - learning on demand / contextualized tutoring
  - critic system needs to infer to which microworld the user belongs
Critiquing Strategies

- control the presentation component of a critic
  - interrupt user’s work; separate window; flag
  - disable critics (in case where one disagrees)

- what aspects should be critiqued
  - educational critiques
  - performance critics

- negative versus positive critics

- how and when to intervene ---> intrusiveness
  - active critics: immediately, after one action, after a semantic unit
  - passive critics: critiquing is initiated by the users

- ground critiquing strategies in a user model (in addition to a task model)
Motivation and Associated Learning Strategies

• critiquing lets learners see for themselves the usefulness of new knowledge for actual problem situations; users are informed
  - when they are getting into trouble
  - when they are missing important information
  - when they come up with suboptimal solutions

• most of our critic rules state what one may not do; this makes for greater freedom of choice than if the rules were prescriptive
  - “You must not do X!” leaves open a whole range of possibilities in terms of what one may in fact do
  - “You must do X!” reduces the range of possibilities to the scope of X itself

• unasked-for help breeds incompetence and is often seen as an intrusion
Goal Acquisition and Product Analysis

- **domain knowledge**
  - generic — e.g., Lisp programs, kitchens, networks, voice dialog systems
  - what is a domain (residential versus commercial kitchens, kitchens for disabled persons)

- **goal knowledge**
  - specific about an individual product
  - inferred from partial specifications and partial constructions

- **differential critiquing:**
  - system generates its own solution
  - compares it with the user’s solution and points out the difference

- **analytical critiquing:**
  - system checks the product with respect to predefined features and effects