Computer-Assisted Learning Environments

Andy Carle

Spring 2006

Outline

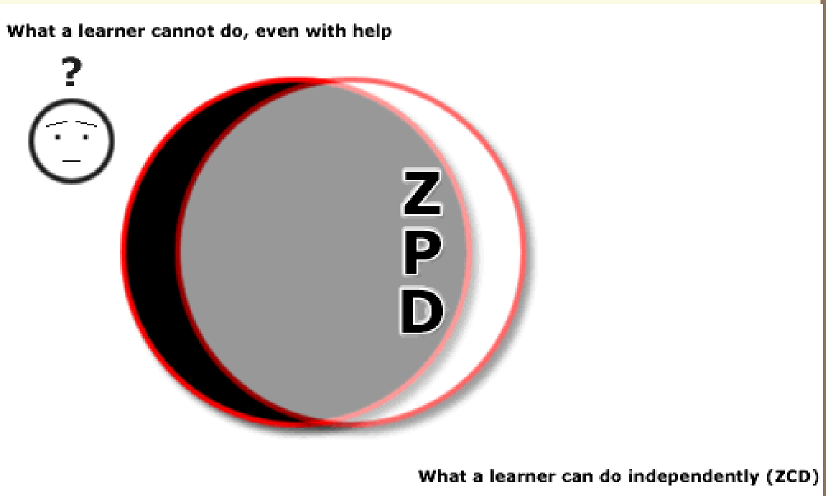
- Review of learning principles
  - Constructivism, Transfer, ZPD, Meta-cognition
- Constructivist Learning Systems:
  - Construction Toolkits
  - Collaborative learning
  - Meta-cognition
  - Inquiry-based environments
  - Agent-based Tutors
- Design Patterns for Education

Building Understanding

- Learning is a *process* of building new knowledge using existing knowledge.
- Knowledge is not acquired in the abstract, but constructed out of existing materials.
- Like any other human process, HCI researchers/practitioners seek to mediate learning via technology.

Learning and Experience

- Learning is most effective when it connects with the learner's ***real-world*** experiences.
- The knowledge that the learner already has from those experiences serves as a foundation for knew knowledge.
- In real societies, learners are helped by others.
- Zone of Proximal Development (ZPD): "zone" of concepts one can acquire with help.



Motivation and Abstraction

- Motivation encourages the user to visualize use of the new knowledge, and to try it out in new situations.
- Students are usually motivated when the knowledge can be applied directly.
- Abstract knowledge is packaged for portability. It's built with virtual objects and rules that can model many real situations.
- Our goal is students that are motivated to collect abstract knowledge and build general understanding

Metacognition

- Metacognition is the learner’s conscious awareness of their learning process.
- Strong learners carefully manage their learning.
- For instance, strong learners reading a textbook will pause regularly, check understanding, and go back to difficult passages.
- Weak learners tend to plough through the entire text, then realize they don’t understand and start again.

- We want to turn weak learners into strong learners.
  - Or, at least, make them act like strong learners.

Constructivist Learning Systems

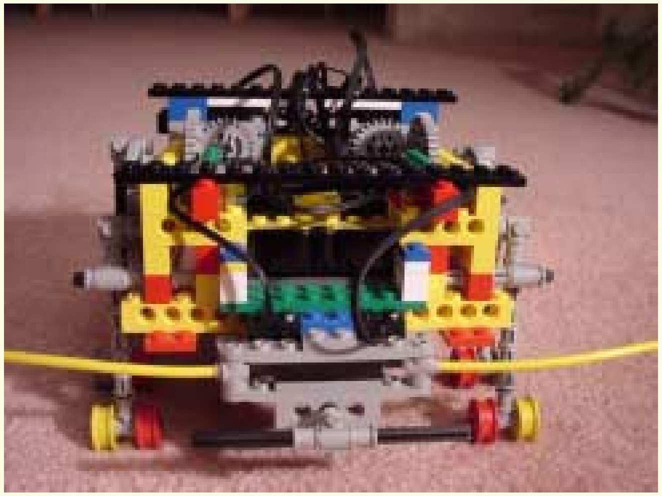
- Construction Kits
  - Logo, Microworlds, Boxer
- Group-learning Systems
  - CoVis, TVI, Livenotes
- Meta-Cognitive Systems
  - SMART, CSILE/Knowledge Forum
- Inquiry-Based Systems
  - ThinkerTools
- Automatic Tutors
  - Inquiry Island
- Integrated Learning Environments
  - WISE, UC-WISE

Logo

- The Logo project began in 1967 at MIT.
- Seymour Papert had studied with Piaget in Geneva. He arrived at MIT in the mid-60s.
- Logo often involved control of a physical robot called a turtle.
- The turtle was equipped with a pen that turned it into a simple plotter – ideal for drawing math. shapes or seeing the trace of a simulation.

Logo

- Early deployments of Logo in the 1970s happened in NYC and Dallas.
- In 1980, Papert published “Mindstorms” outlining a constructivist curriculum that leveraged Logo.
- Logo for Lego began in the mid-1980s under Mitch Resnick at MIT.



Logo

- The “Microworlds” programming environment was created by Logo’s founders in 1993. It made better use of GUI features in Macs and PCs than Logo.

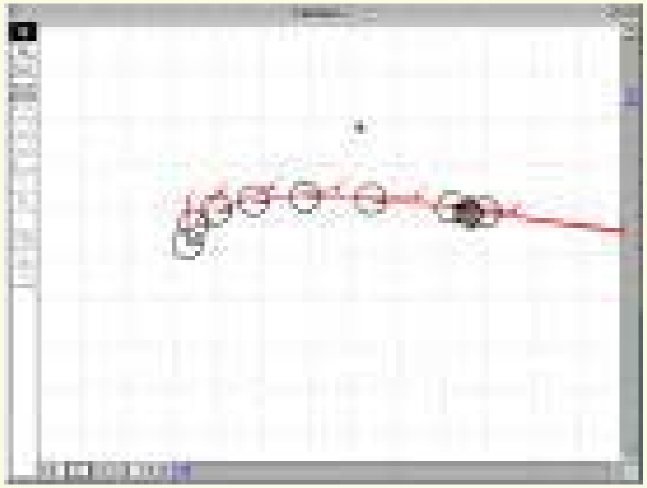
- In 1998, Lego introduced Mindstorms which had a Logo programming language with a visual “brick-based” interface.

Logo

- Logo was widely deployed in schools in the 1990s.
- Logo is primarily a programming environment, and assignments need to be programmed in Logo.
- Unfortunately, curricula were not always carefully planned, nor were teachers well-prepared to use the new technology.
- This led to a reaction against Logo from some educators in the US. It remains very strong overseas (e.g. England, South America).

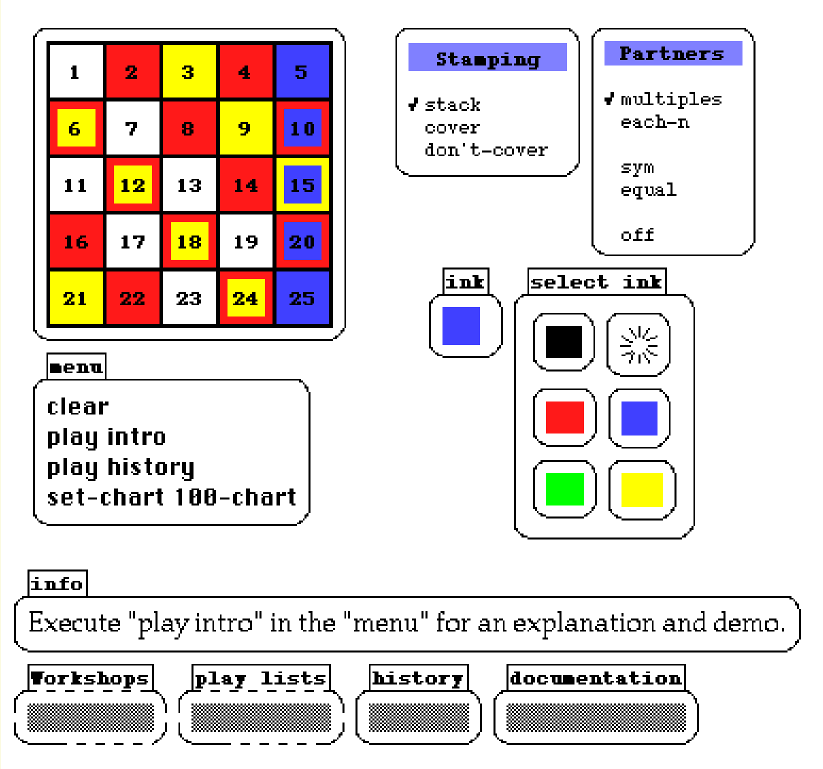
Uses of Logo

- Logo is designed to create “Microworlds” that students can explore.
- The Microworld allows exploration and is “safe,” like a sandbox.
- Children “discover” new principles by exploring a Microworld.
- e.g. they may repeat some physics experiments to learn one of Newton’s laws.



Boxer

- Boxer is a system developed at Berkeley by Andy diSessa (one of the creators of Logo).
- Boxer uses geometry (nested boxes) to represent nested procedure calls.
- It has a faster learning curve in most cases than pure Logo.



Strengths of Logo

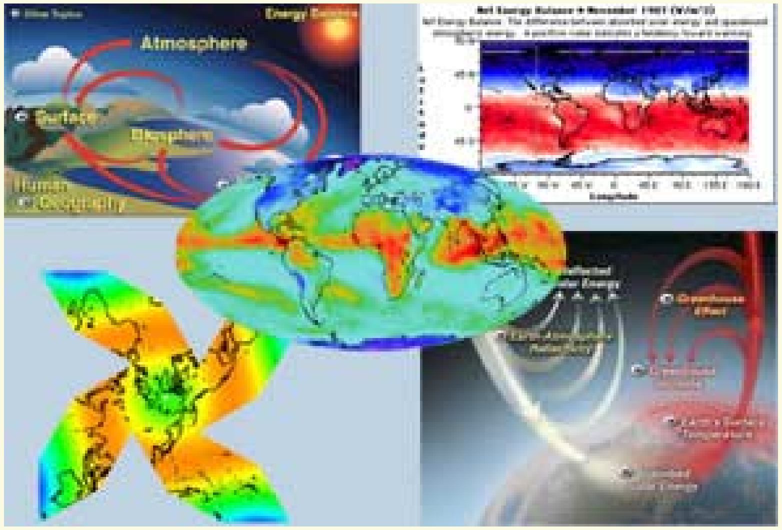
- Very versatile.
- Can create animations and simulations quickly.
- Avoids irrelevant detail.
- Tries to create “experiences” for students (from simulations).
- Provides immediate feedback – students can change parameters and see the results right away.
- Representations are rather abstract – which helps knowledge transfer.

Weaknesses of Logo

- Someone else has to program the simulations etc – their design may make the “principle” hard to discover. Usability becomes an issue.
- The “experience” with Logo/Mindstorms is not real-world, which can weaken motivation and learning.
- The “discovery” model de-emphasizes the role of peers and teachers.
- It does not address meta-cognition.

Collaborative Software

- CoVis (Northwestern, SRI) was a system for collaborative visualization of data for science learning, primarily in geo-science, 1994-…
- Students work online with each other, and with remote experts.
- They take virtual field trips, or work with shared simulations.



CoVis

- CoVis included a “Mentor database” of volunteer experts that teachers could tap to talk about advanced topics.
- It also included a collaboration notebook. The notebook included typed links to guide the student through their inquiry process.
- Video-conferencing and screen-sharing were used to facilitate remote collaboration.

TVI

- TVI (Tutored Video Instruction) was invented by James Gibbons, a Stanford EE Prof, in 1972.
- Students view a recorded lecture in small groups (5-7) with a Tutor. They can pause, replay, and talk over the video.
- The method works with a live student group, but also with a distributed group, as per the figure at right.

DTVI

- Sun Microsystems conducted a large study of distributed TVI in 1999.
- More than 1100 students participated.

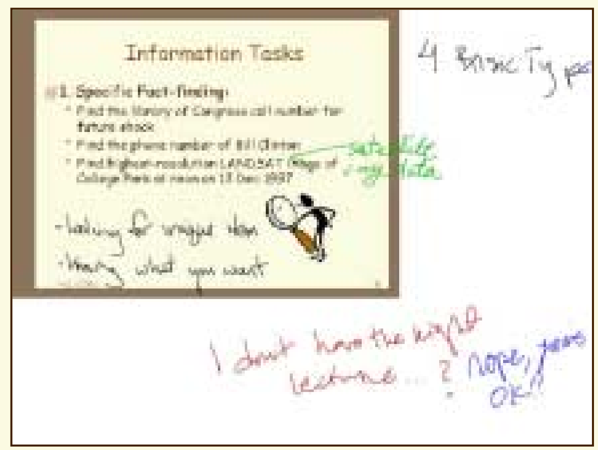
- The study showed significant improvements in learning for TVI students, compared to students in the live lecture (about 0.3 sdev).

DTVI

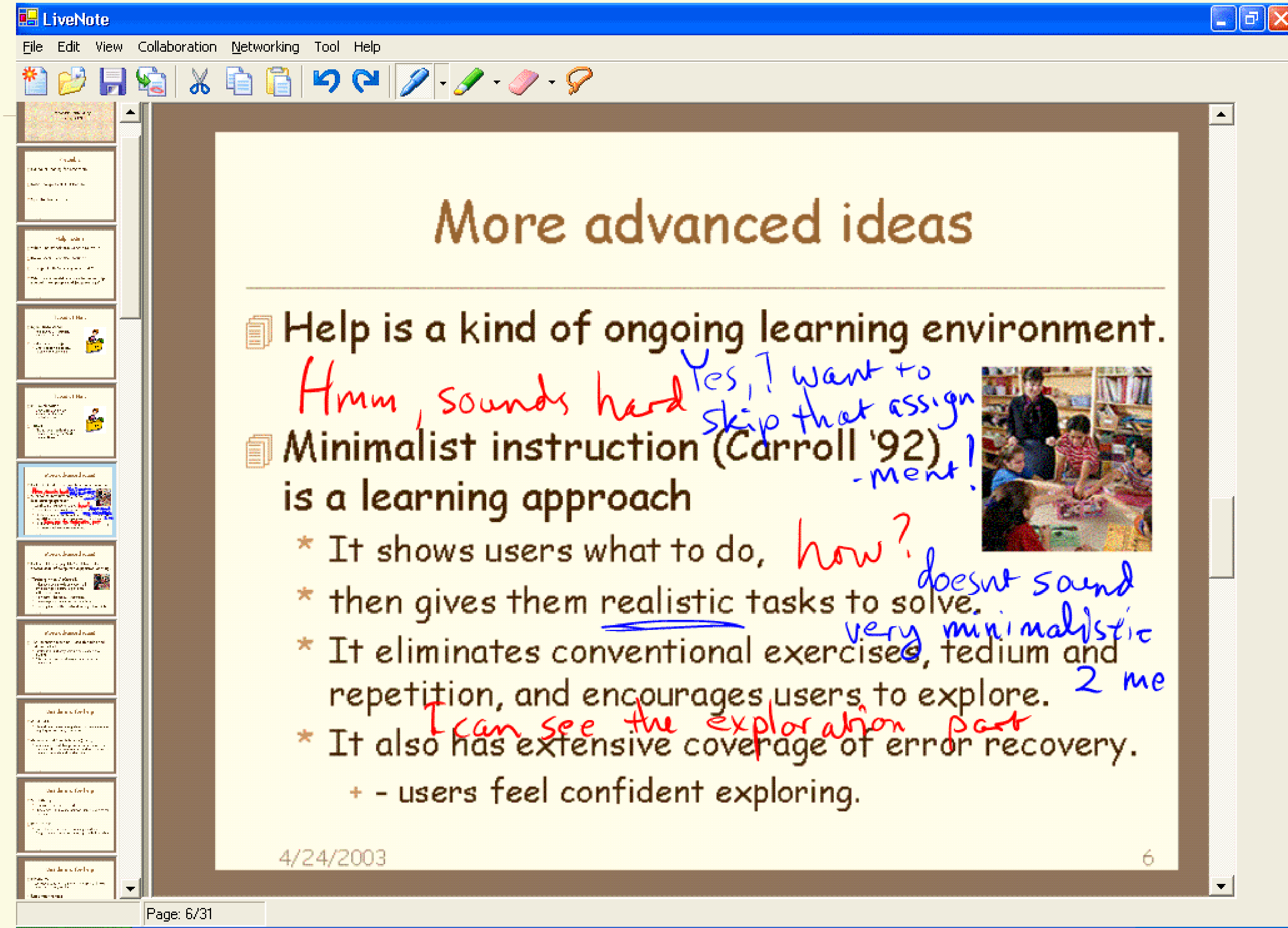
- The DTVI study produced a wealth of interesting results:
- Active participation was high (more than 50% of students participated in > 50% of discussions).
- Amount of discussion in the group correlated with outcomes (exam scores).
- Salience of discussion did not *significantly* correlate with outcome (*any* conversation is helpful??).

Livenotes

- TVI requires a small-group environment (small tutoring rooms).
- **Livenotes** attempts to recreate the small-group experience in a large lecture classroom.
- Students work in small *virtual* groups, sharing a common workspace with wireless Tablet-PCs.
- The workspace overlays PowerPoint lecture slides, so that note-taking and conversation are integrated.



Livenotes Interface



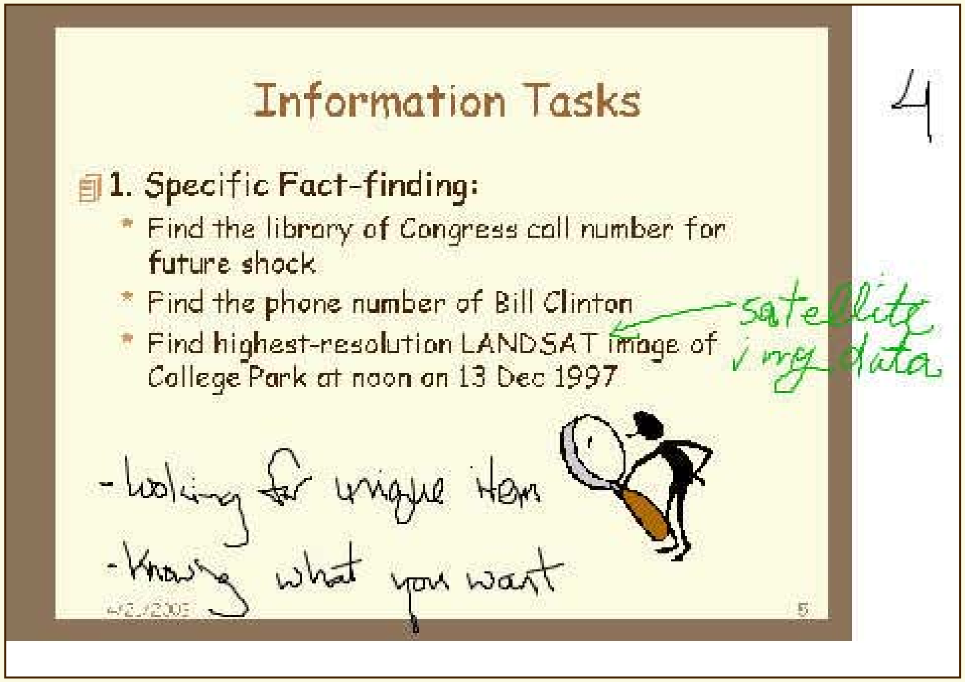
Livenotes Findings

- The dialog between students happens spontaneously in graduate courses – where student discussion is common anyway.
- It was much less common in undergraduate courses.
- Students have different models of the lecture – something to be “captured” vs. some that is collaboratively created.

Livenotes Findings

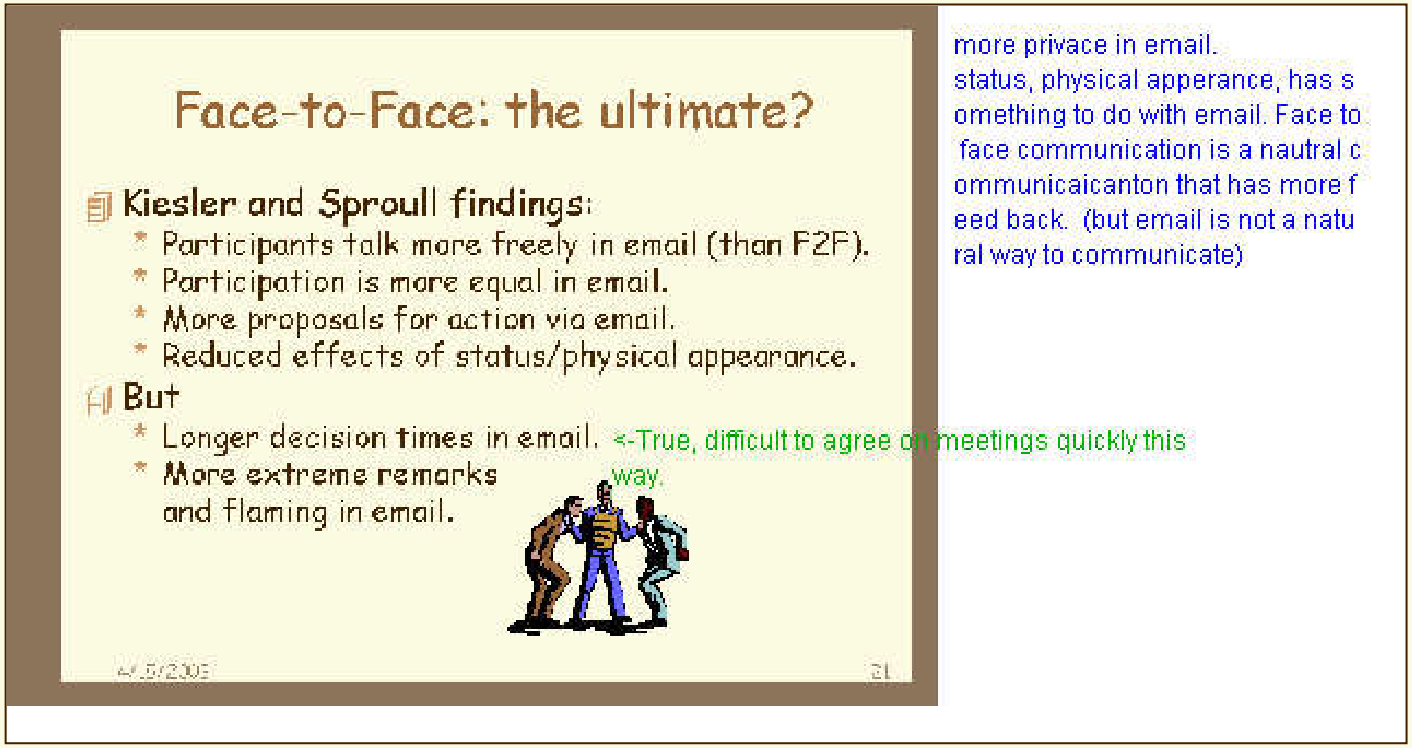
- But what was very common in undergraduate transcripts was student “dialog” with the PowerPoint slides:

- Students often add their own bullets.



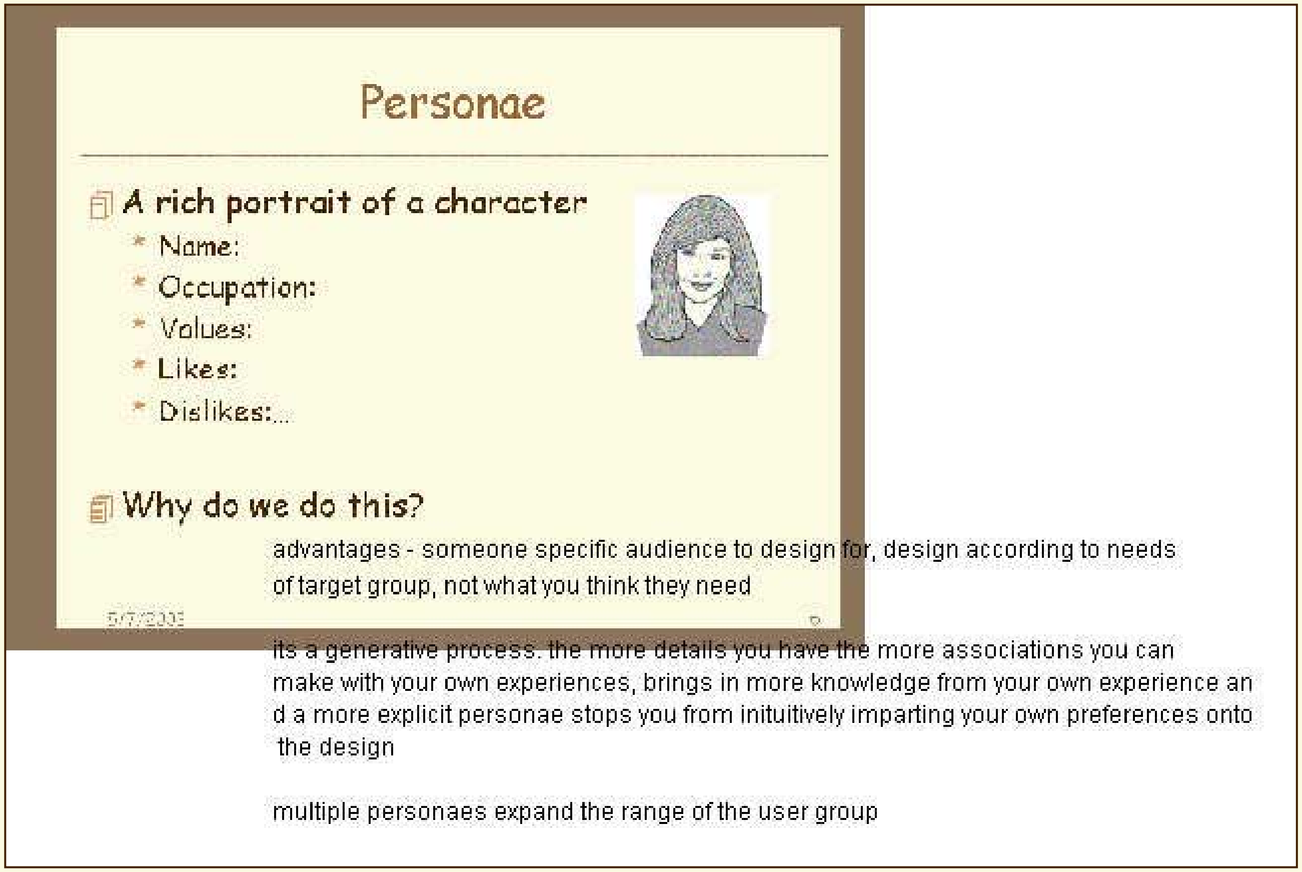
Livenotes Findings

- Reinforcing/rejecting a bullet:



Livenotes Findings

- Answering a question in a bullet:



Collaborative Systems

- Given what you know about learning, list some advantages and disadvantages of the 3 systems (CoVis, TVI/DTVI, Livenotes).
- What collaborative class features have you experienced in school?

Meta-Cognitive Systems

- The SMART project (Vanderbilt, 1994-) gave students science activities with meta-cognitive scaffolds.
- Students choose appropriate instruments to test their hypothesis – requiring them to understand the kind of information an instrument can give.
- The case study was an environmental science course called the "Stones River Mystery".

Meta-Cognitive Systems

- The SMART lab required students to justify their choices – it encouraged them to reflect after their decisions, and hopefully while they are making them.

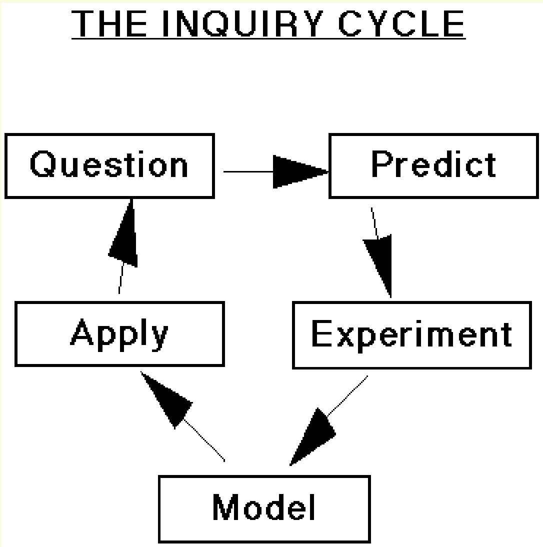
- It also included several tools for collaboration between students. Explaining, asking questions, and reaching joint conclusions help improve meta-cognition.

Inquiry-Based Systems

- A development of Piaget based on similarities between child learning and the scientific method.
- In this approach, learners construct explicit theories of how things behave, and then test them through experiment.
- The "ThinkerTools" system (White 1993) realized this approach for "force and motion" studies.

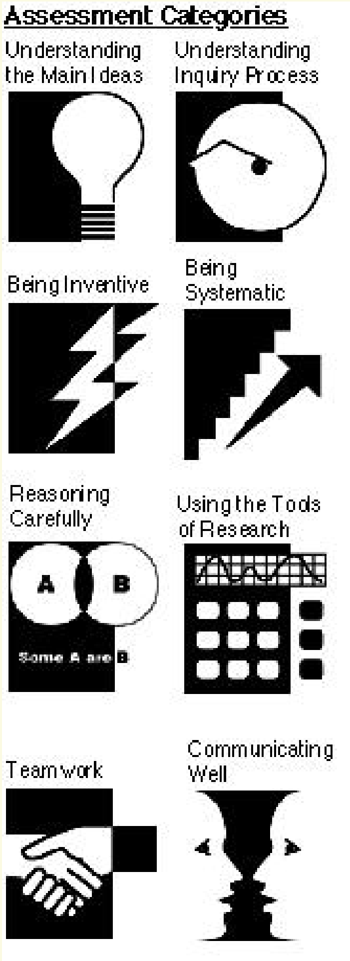
ThinkerTools

- ThinkerTools uses an explicit inquiry cycle, shown below.
- Students are scaffolded through the cycle by carefully-designed exercises.



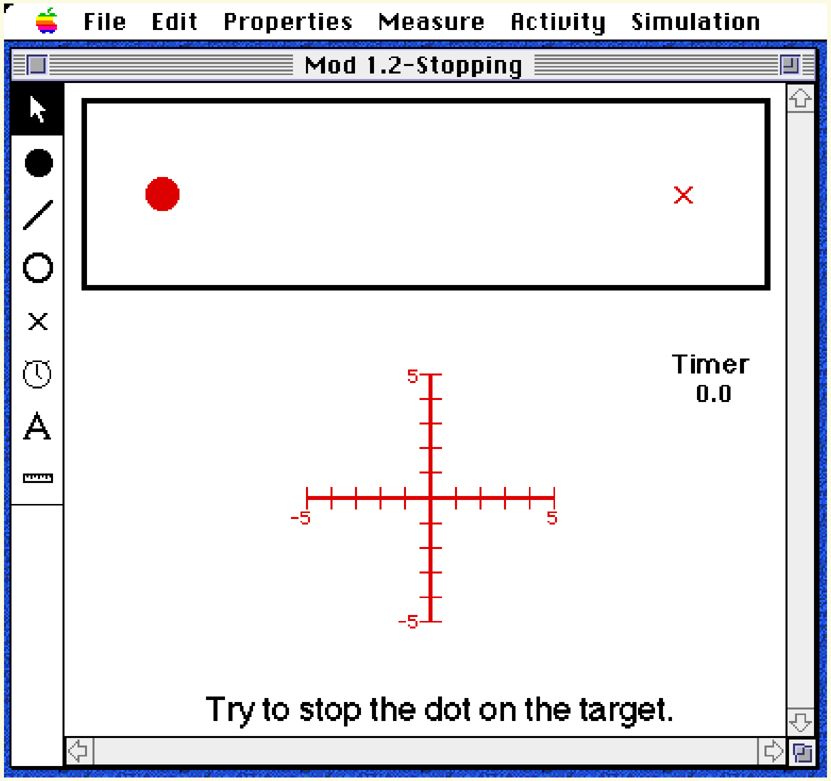
ThinkerTools

- ThinkerTools uses "reflective assessment" to help students gauge their own performance and identify weaknesses.



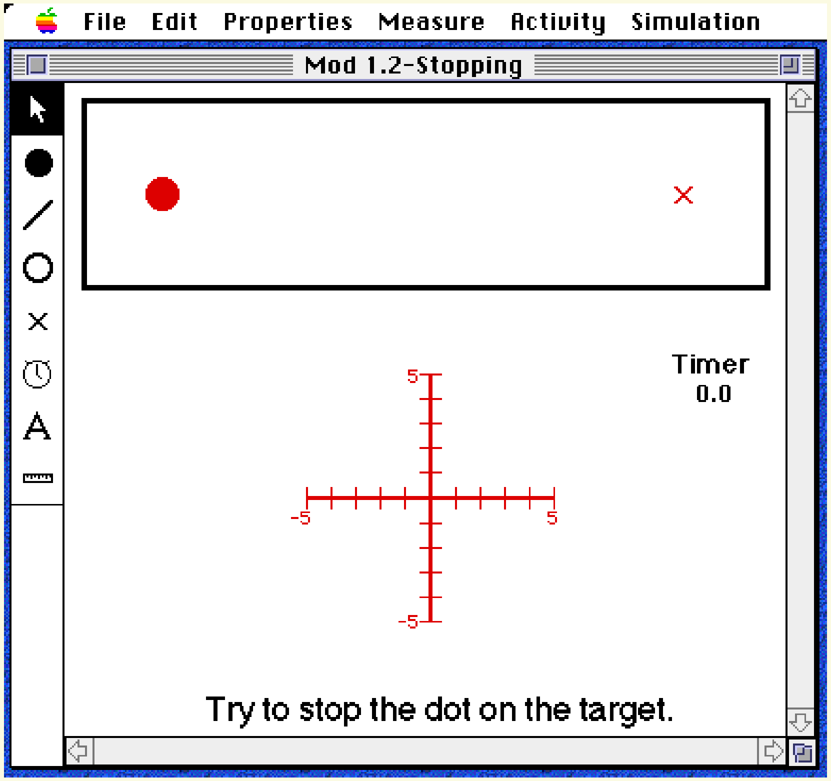
ThinkerTools

- The tools include simulation (for doing experiments) and analysis, for interpreting the results.



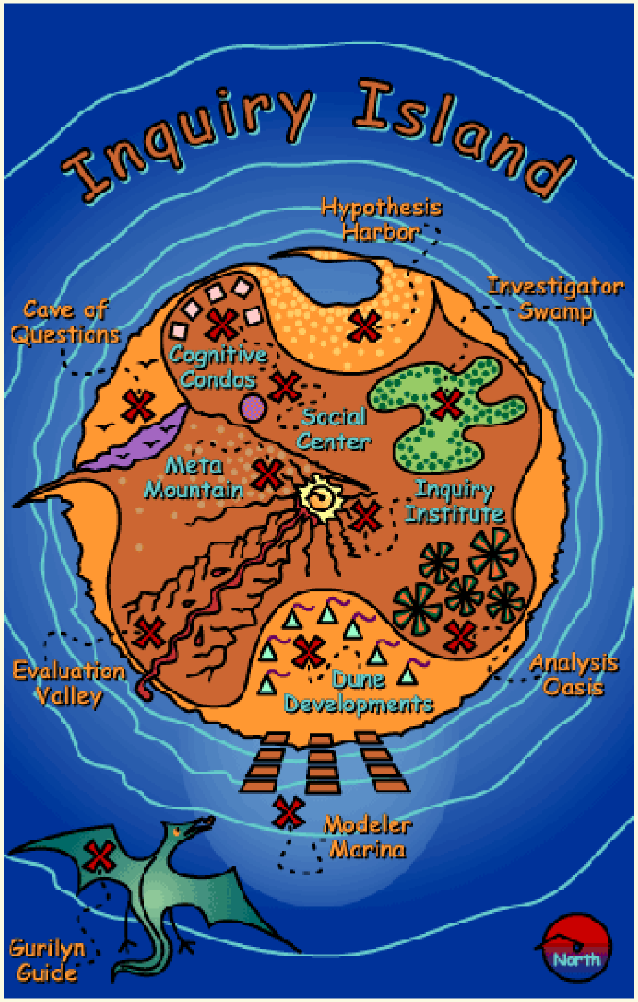
ThinkerTools

- Students can modify the “laws of motion” in the system to see the results (e.g. F=a/m instead of ma).



Agents: Inquiry Island

- An evolution of the ThinkerTools project.
- Inquiry Island includes a notebook, which structures students inquiry, and personified (software agent) advisers.



Inquiry Island

- Task advisers:
  - Hypothesizer, investigator
- General purpose advisers:
  - Inventor, collaborator, planner
- System development advisers:
  - Modifier, Improver

- Inquiry Island allows students to extend the inquiry scaffold using the last set of agents.

Integrated Learning Environments

- Web-Based Inquiry Science Environment (WISE)
  - UC Berkeley TELS group
  - Middle School ~ High School science classes
- UC-WISE
  - TELS group + CS Division
  - UC Berkeley & Merced lower division CS courses
- Sakai
  - Multiple institutions
  - Called bSpace in the UC system

UC-WISE – Question

- What components of UC-WISE are similar to the systems we've considered thus far?

- What components are noticeably different?

UC-WISE Features

- Learning Management System
  - Cohesive collection of lessons, tasks, assignments, assessments, and related info
- Collaborative Tools
  - Brainstorms, discussion forums, collaborative reviews
- Inquiry-Based Tools
  - Web-Scheme, Eclipse exercises
- Meta-Cognitive Tools
  - Quick quizzes, “extra brain,” peer assessment

Question

- How portable (across different courses) are these systems (SMART, ThinkerTools, Inquiry Island) and their content (UCB CS3)?

Design Patterns for Education

- Recall Lecture 15:
  - Design patterns for architecture & software
  - Communicate design problems and solutions
  - Not too general, not too specific
    - Use a solution "a million times over, without ever doing it the same way twice."

- This concept can be applied to education!
  - Pedagogical Patterns

Pedagogical Patterns Project

- “Attempt to capture expert knowledge of the practice of teaching and learning in a portable, salient format.”
- http://www.pedagogicalpatterns.org/

- E.g. “Expand the Known World”

“Expand the Known World”

- Context:
  - * You have a new concept to introduce. Your students have some related knowledge and experience.
- Forces/Key Problem:
  - * A student's learning will be deeper if they associate a new concept to their existing knowledge and experience.
- Solution:
  - * Therefore introduce the concept by explicitly linking it to experiences that you know the students have already…
- Additional Information:
  - * Time consuming, works well with Larger than Life, etc…

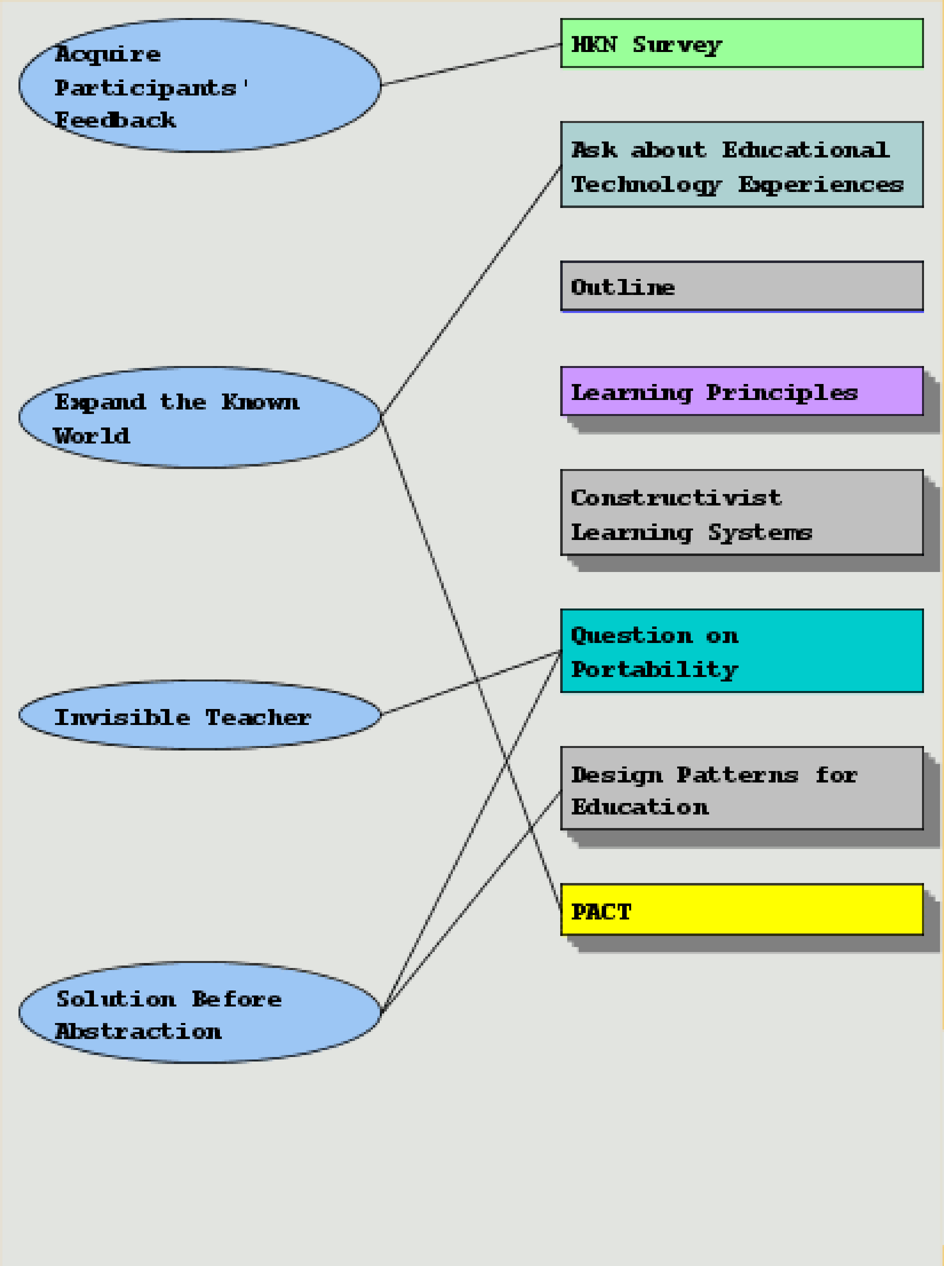
Problems in Practice

- Pedagogical patterns have a tendency to be too abstract to be useful.
  - Difficult to apply to a new context
- Pattern-informed environments rarely reveal clues about the underlying patterns to the untrained observer
- Collaboration between content experts and pedagogical specialists is rare
  - Individuals that can fill both roles are even more scarce.

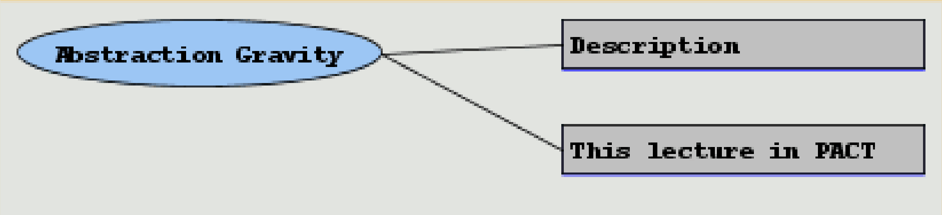
Pattern Annotated Course Tool

- Research project intended to bridge the gap between pedagogical patterns in theory and in practice
- Visual editor in which expert course designers can create representations of their own courses, complete with references to pedagogical patterns
- Novice instructors can see patterns instantiated in a context that they can relate to directly

CAL Environments



Details



Summary

- We reviewed some learning principles from lec 19.
- We gave some systems that roughly track the frontier of learning technology:
  - Construction toolkits
  - Collaborative systems
  - Meta-cognitive scaffolding systems
  - Inquiry systems
  - Agent-based tutoring systems
  - Integrated learning environments
- We considered the application of design patterns to pedagogy and a tool to facilitate this process