Using Technology to Support Constructive Learning and Problem Solving

Dr David Jonassen

http://www.coe.missouri.edu/~jonassen/
| Identify problem that students should learn how to solve | How do problems vary? Structuredness, complexity, dynamicity, domain specificity
Typology of problems |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Develop cognitive model of problem-solving processes required to solve it</td>
<td>Cognitive models for story, troubleshooting, case analysis, design problems</td>
</tr>
</tbody>
</table>
| Represent problem to learner in an appropriate manner | Problem Posing
Anchoring Problems in macrocontexts
Case-based Instruction |
| Provide tools and support systems for learners to represent the problem | Representing Semantic Organization
Representing Causal Reasoning
Causal Modeling
Influence Diagrams
Expert Systems
Modeling Dynamic Systems |
| Associate solutions with problems | Worked Examples
Case Libraries: Teaching with Stories
Supporting Problem Solving with Stories
Cognitive Flexibility Hypertexts |
| Support solution tryout with feedback | Simulations
Microworlds
Learning Objects
Simulations
Argumentation |
| Support reflection on processes | Peer Instruction/Thinking Aloud Pair
Problem Solving
Peer Instruction
Thinking Aloud Pair Problem Solving
Teachbacks and Abstracted Replays
Coding Protocols |
| Assess problem solving, not recall | Performance assessment with rubrics
Component conceptual skills
Structural knowledge
Argumentation |

(1) Identify fault symptoms
(2) Construct or access a model of the system or device (problem space)
   - Describe goal state, how do you know when system is functioning properly
   - Identify sub-system fault occurs in
(3) Diagnosis process
   - Examine faulty sub-system(s)
   - Examine experience; Reminding from previous cases
   - Rule out least likely hypotheses
   - Generate initial hypothesis and assumptions
   - Test hypotheses based on domain knowledge
   - Interpret results of test
   - Confirm or reject validity of hypotheses: If reject, generate new hypothesis
(4) Resolve fault
   - Select solution
   - Evaluate solution in terms of cost and effort or experience
   - Implement solution
   - Test solution(s): is goal state achieved?
(5) Add experience to personal case library of experiences
**Design Problem Solving**

1. Structuring the Problem Space
   - Modularity/decomposability: complexity of problems required decomposition into large number of modules
   - Incremental development of artifact: design ideas incrementally developed but rarely discarded and replaced
   - Control structure – Designers use a limited commitment mode control strategy
     - Designer must make and propagate commitments
   - Stopping rule and evaluation functions are personalized.
   - Construct and manipulate models: too costly to manipulate world so designers manipulate representations of world
     - Abstraction hierarchies

2. Expand problem space
   a. Articulate Design Goal.
     - Functionality goals
     - Performance goals (efficiency, cost, reliability)
     - Knowledge goals (gathering information needed to carry out design), and
     - Design goals to describe the route
   b. Specify Functional Requirements.
   c. Needs assessment and task redefinition: redefining the task is subjectively perceiving and defining the task
   d. Recall prior knowledge, including:
     - General schemas (world knowledge)
     - Procedural knowledge
     - Abstract conceptual knowledge (principles, laws (extractions from experience)
     - Domain-specific schemas
     - Analogies.


4. Problem Space Reduction
   a. Constraint Analysis (AKA parametric analysis: defining parameters of the product such as quantitative (size, weight, power, speed , strength), qualitative (ranked or scaled relative to other products- easier, more comfortable), and categorical (which of several alternative categories product belongs to – gasoline or electric, remote control or manual control).
     - Technical
     - Economic
     - Political
     - Cultural

5. Select solution.

6. Optimize solution

7. Construct design model or artifact

8. Evaluate solution