Jim Waldo's assertion:
When you test a program, you are testing state transitions.
You pick up a sword, after which you hold the sword.
You drop something, after which it stays where it was.
...
This is true in general:
Functions take one state (e.g., a state of arguments) and transform it to another (a value of output).
Cost of testing is relative to the complexity of state transformations.
Input states: equivalence partitions.
(Output states don't drive complexity)
Jim Waldo's claim:
The transition back to a baseline (saved) state is all that has to work.
This means that:
The finer-grain state transitions don't have to be tested, except to insure that the baseline doesn't get corrupted.
=> it's a lot cheaper to test: less cases.
=> you only have to test the data model.
=> testing is a lot cheaper.
Important and central points:
BPMN expresses the manager's point of view for business process. Contains a global view of process.
BPEL allows things to become services. Translating from BPMN to BPEL isn't automatic: details must be added. There's a "semantic wall" between the level of detail in BPMN and the level of detail in BPEL (required for AXIS).
Assignment 6 issues
    Cache walls
    Principle of locality of reference
    How to test performance.
    Controlling memory access patterns.