

## Just Enough UML

15-413: Introduction to Software Engineering

Jonathan Aldrich



## Announcements



- Course time last week
  - Mostly reasonable (8-11 hours)
  - A few people spent a LOT of time (15-20 hours)
    - If you spent more than 3 hours/person on the prototype you can "charge" this to your project time
    - If you're ahead on project hours (target 6/person/week) you can reduce them this week
  - Project hours include client meetings, team interaction, XP planning & reporting, as well as pair programming
- A few people spent very little time (2-3 hours)
  - Offloading onto teammates is OK for one week (with their permission!) but you need to make it up

7 October 2005

## Announcements



- Next homework out tonight
  - Modeling, due next week
- Iteration 1 reports and Iteration 2 plans due next week
- Meetings scheduled (Wean 8212)
  - 11:30 Group 6
  - 3:30
  - 4:00
  - 4:30 Group 3
  - 5:00 Group 2
  - 5:30 Group 5
  - 6:00 Group 4
- We will ask to see CVS, unit tests, and your code, so make this accessible by telnet/ssh/FTP

7 October 2005

## Outline



- Class Diagrams
- Use Case Diagrams
- Sequence Diagrams
- Statechart Diagrams

7 October 2005

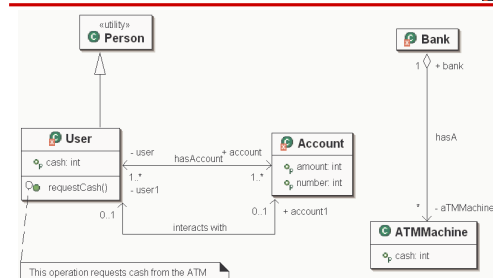
## Class Diagrams



- Used to model OO design
  - Classes
  - Attributes & Methods
  - Associations: relationships among classes
    - Aggregation
    - Multiplicities
  - Inheritance / subtyping
- Can also be used to capture model of information in requirements domain
  - What attributes does each concept have?
  - How are domain concepts related?

7 October 2005

## Bank System



7 October 2005

## Use Cases

- Describe a scenario of using system
  - Name
  - Description
  - Pre/Post-condition
  - Normal flow
  - Alternative/Exceptional flow
- The diagram is **almost** useless
  - At least shows actors in each use case
    - Suggest areas where you should study external domains

7 October 2005

## ATM Use Cases



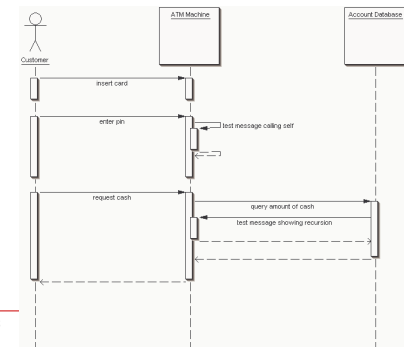
7 October 2005

## Sequence Diagrams

- Show order of interactions in scenario
  - Actors and components
  - Message sends, nested messages
- Useful to pin down and easily visualize temporal properties of scenario

7 October 2005

## ATM Sequence Diagram



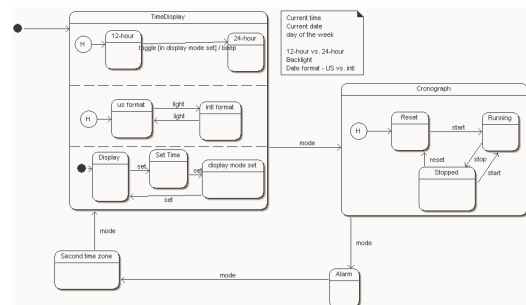
7 October 2005

## Statechart Diagrams

- Shows how system reacts to events
  - States
  - Transitions triggered by events
  - Initial and final states
    - Maybe no final states!
  - Nested state machines
- Especially useful for reactive control systems
- Vs. sequence diagram
  - Exhaustive coverage vs. one trace
  - Events vs. calls

7 October 2005

## Watch Statechart



7 October 2005

## Statechart Analysis



- Simulation
  - How does the machine run given these inputs?
- Model checking
  - Can the machine get into bad states?
  - From the start state, explore all possible transitions to another states; repeat for each of these states

7 October 2005

## More to explore



- Statecharts
  - Guard conditions & actions
- Object diagram
  - Like a class diagram, but a snapshot of objects in the heap
- Collaboration diagram
  - Like sequence diagram, but shows structure instead of timing
- Activity diagram
  - Shows flow of data and control in system
- Component diagram
  - Shows system organization & dependencies
- Deployment diagram
  - Shows how components map to processing nodes

7 October 2005

## EclipseUML Tool



- Free academic license
- CMU key will be on Blackboard

7 October 2005