Object Behaviours
UML Statecharts

Software Requirements and Design
CITS 4401
Lecture 5

Based on Ch 5 Lecture Notes by Bruegge & Dutoit

Finite state machine

Ex 1. Interaction with my PC (1)
Statechart Diagrams
- Graphs whose nodes are **states** and whose directed arcs are transitions labelled by **events**.
- States capture conditions which hold for a period of time
  - e.g. light is on, light is off
- Events **change** the state (except internal)
  - e.g. turning the light on, turning the light off
- Statechart diagrams represent behaviour from the perspective of a single object only
  - An object model with a set of objects must be represented by a set of state diagrams

UML Statechart Diagram Notation
- UML notation based on statecharts by Harel in 1987
- Added are a few object-oriented modifications
- A UML statechart diagram (or UML state machine) can be mapped into a finite state machine (FSM)

**State**
- An abstraction of the attributes of a class
  - State is the aggregation of several attributes of a class
- Basically an equivalence class of all those attribute values and links that do not need to be distinguished as far as the control structure of the system is concerned
  - Example: State of a user interface screen
    - logged in, logged out, active, idle
    - active is an abstraction of all the user's logged in activity
- State has duration
Event

- Something that happens at a point in time (e.g. button press, mouse click)
- Triggers a transition
  - Internal transition (no state change)
  - External transition (change to different state)
- May result in an action being executed

Example 2: vending machine

Another example

States of the Incident object of FRIEND

Exercise

Draw a state diagram for the book class:

- "A book can be borrowed from the library on a short term or long term loan and then must be returned. A long term loan can only be taken if there is no one on the waiting list. A short term loan can be extended to a long term loan"
Problem Statement: Direction Control for a Toy Car

- Power is turned on
  - Car moves forward and car headlight shines
- Power is turned off
  - Car stops and headlight goes out.
- Power is turned on
  - Headlight shines
- Power is turned off
  - Headlight goes out.
- Power is turned on
  - Car runs backward with its headlight shining.

Practical Tips for Dynamic Modeling

- Construct dynamic models only for classes with significant dynamic behavior
  - Avoid “analysis paralysis”
- Consider only relevant attributes
  - Use abstraction if necessary
- Look at the granularity of the application when deciding on actions and activities
- Reduce notational clutter

Exercise

- Consider the operation of the CITS4401 discussion page: https://secure.csse.uwa.edu.au/run/help4401

Model the dynamic behavior of a controller object for someone reading and posting to the forum.