Aspects of the UML

CA212 - Week 11
Dynamic Modelling with
State Diagrams

Unified Modelling Language
Modelling Behaviour

Object Interactions

System Dynamics

Models
Describing Behaviour

- Dynamic Modelling
  - **UML State Diagram**
  - **UML** Sequence Diagram
  - **UML** Collaboration Diagram
  - **UML** Activity Diagram
State Diagram

- Each Class may have an optional associated State Diagram.
- Developed by Harel.
- Incorporated into OO methods by Rumbaugh (OMT) and many others.
Notation

State 1  \rightarrow  State 2

Event[guard]/action

Name of event which causes transition

Must be true for event to fire.

Action performed when event occurs
Example

- Pop-up menu control

idle

right button down/display

menu visible

right button up/erase

Cursor Moved/
Highlight item
An activity is an operation that takes time to complete. Activities are associated with states.

Diagram:

State 1
do/ Activity 1
Action

- An action is an *instantaneous* operation associated with an event.
- Semantics of *instantaneous* is ambiguous.
General Notation

- Event may have optional attributes associated (event supplies data)
- Optional guard conditions (pre-conditions) must be satisfied before a transition occurs.

![Diagram of state transition with event, attributes, conditions, and actions]
Message Notation

- Synchronous: blocking call.
- Asynchronous: non blocking call
- Simple: no details about communication.
- Synchronous with immediate return.
Nesting State Diagrams

- State diagrams for an Object may be nested, allowing the control mechanism to be viewed at different levels.
Example: Vending Machine

- **idle**
  - **collecting money**
    - coins in(amount)/set bal
  - **cancel/refund**
    - **[item empty]**
      - **collecting money**
        - coins in(amt)/add to bal
  - **select(item)**
    - **[change<0]**
      - **do/test item and compute change**
        - **[change=0]**
          - **do/ dispense item**
        - **[change>0]**
          - **do/make change**
Example: Dispense Item

do/ move arm to correct row
arm ready

do/ move arm to correct col

arm ready

do/push off shelf
pushed
Example: Select Item

do/reset
item

digit(n)  do/append
digit

clear  enter

select(item)
Generalisation of States

- Groups of substates with common transitions can be combined into a single superstate, and inherit transitions from the superstate.
Example: Transmission

Transmission

Neutral

Reverse

push R

push N

push F

Forward

1st

2nd

3rd

downshift
downshift

upshift upshift

stop
Example: Generalisation

- **Forward** is an abstract state.
- Selecting **N** in any forward gear will cause a transition to **Neutral**.
- Selecting **Stop** in any forward gear will cause a transition to **First**.
Example: Object Model

- Car
  - Ignition
  - Transmission
  - Brake
  - Accelerator
Dynamic Model

- Ignition state diagram
- Brake state diagram
- Transmission state diagram
- Accelerator state diagram
Dynamic Model: Ignition

Ignition

- off
- on
- starting
- release key

turn key [transmission in Neutral]

turn key off
Dynamic Model: Transmission

Transmission

Neutral

Reverse

push N

push R

push F

Forward

1st

downshift

2nd

downshift

3rd

upshift

upshift

stop

contour
Dynamic Model: Accelerator & Brake

Accelerator

- press acc
- rel acc

Brake

- press brake
- rel brake
Concurrency

Aggregation concurrency: The aggregate state corresponds to the combined states of all the components.

State of 1 is defined by state of 2 and of 3
Concurrency (cont..)

- Concurrency within an Object:-
  Concurrency within the state of a single Object arises when an object can be partitioned into subsets of attributes or links, each of which has its own state diagram.
Concurrency (cont..)

Superstate

substate1

substate2

substate3

substate4

event1

event2
Example: Programmable Thermostat

Example of Aggregation Concurrency
Links to the Class Diagram

- Keep SD as simple as possible.
- Events, actions, activities **must** each map directly to functions on the UML Class Diagram.
- A “dictionary” of all functions and data is maintained for consistency across all diagrams and models.
Class and State Diagrams

- If a function appears on a State Diagram, then it must appear on a corresponding Class Diagram, otherwise there is no rigor.

- CASE tools like Rational Rose help support this rigor by assisting modeller with lists of operations and generating reports of orphan operations (not on Class diagram).
OO Method Adaptations

- ROOM
- Octopus
- INSYDE’s OMT*
  - http://www.compapp.dcu.ie/~bstone/research
- Catalysis
  - http://www.iconcomp.com
- Rational’s Process (Objectory)
  - http://www.rational.com
UML Tools

- At present there are two main UML tool vendors...
  - Rational: the Rose CASE tool
    - http://www.rational.com
  - Object Team: the Cayenne CASE tool.
    - http://www.objectteam.com
Graded Exercise

- This is the final exercise. Well worth doing!!!
- An ATM case-study is defined on the public directory.
  - Develop a Class Diagram for the ATM
  - Develop a State Diagram for performing a Query on Account.
  - Use Rational Rose.
- Hints: Make “Transaction” a class. “Query” is a type-of “Transaction” (inherited from).
- Partial Telecomms example available on public directory for reference.