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 → 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
  - right button up/erase
  - menu visible
  - Cursor Moved/Highlight item
**Activity**

- An activity is an operation that takes time to complete. Activities are associated with states.

```
State 1
do/ Activity 1
```

**Action**

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

```
State 1  event/action  State 2
```
General Notation

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

State 1 ➔ State 2

event(=attrib) [condition] / action

Attribute list ➔ Action list ➔ Condition list

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]
- select(item)
- [change<0]

- do/test item and compute change
- [change=0]
- [change>0]

- do/make change
- do/ dispense item
Example: Dispense Item

1. do/ move arm to correct row
2. arm ready
3. do/ move arm to correct col
4. arm ready
5. do/ push off shelf

Example: Select Item

1. do/reset
2. digit(n)
3. clear
4. enter
5. do/append digit
6. digit(n)
7. 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

![Diagram of a transmission system showing states and transitions between them. The diagram includes states like Forward, Neutral, Reverse, 1st, 2nd, and 3rd gears, with transitions for pushing the gear lever in different directions and shifting gears.]
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

Dynamic Model: Ignition

- Ignition state diagram
- Brake state diagram
- Transmission state diagram
- Accelerator state diagram

- Ignition
  - off
  - turn key [transmission in Neutral]
  - starting
  - turn key off
  - on
  - release key
**Dynamic Model: Transmission**

Transmission

- **Neutral**
  - push N
  - push F

- **Reverse**
  - push N

- **Forward**
  - 1st
    - downshift
    - stop
    - upshift
  - 2nd
    - downshift
    - upshift
  - 3rd

Contour

**Dynamic Model: Accelerator & Brake**

**Accelerator**

- off
  - press acc
  - on
  - rel acc

**Brake**

- off
  - press brake
  - on
  - rel brake
Concurrency

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

```
1

State of 1 is defined by state of 2 and of 3

2 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.
Concurrent Specify:

Superstate

- substate1
- substate2
- substate3
- substate4

event1

event2

Example:
Programmable Thermostat

Thermostat

- U.I.
- season switch
- fan switch
- furnace relay
- A/C relay
- fan relay
- Run

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.