

Aspects of the UML

CA212 - Week 11
Dynamic Modelling
with
State Diagrams

Unified Modelling Language

Modelling Behaviour

Object Interactions
System Dynamics
Models

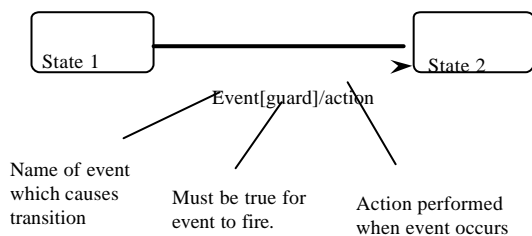
Describing Behaviour

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

State Diagram

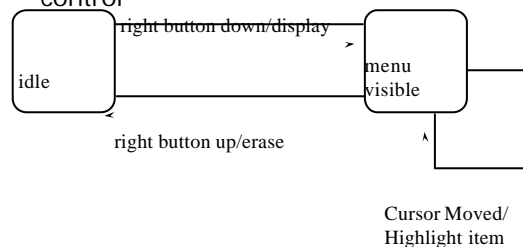
- Each Class may have an optional associated State Diagram.
- Developed by Harel.
- Incorporated into OO methods by Rumbaugh (OMT) and many others.

Notation



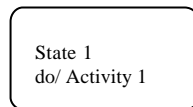
Example

- Pop-up menu control



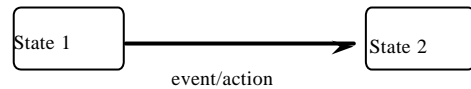
Activity

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



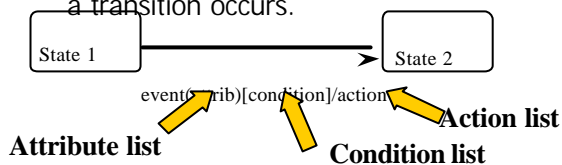
Action

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



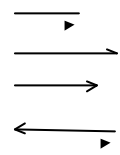
General Notation

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



Message Notation

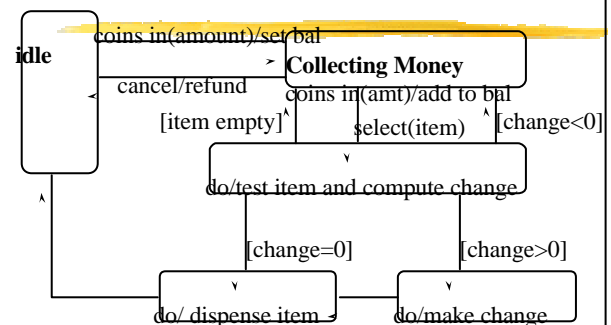
- Synchronous: blocking call.
- Asynchronous: non blocking call
- Simple: no details about communication.
- Synchronous with immediate return.



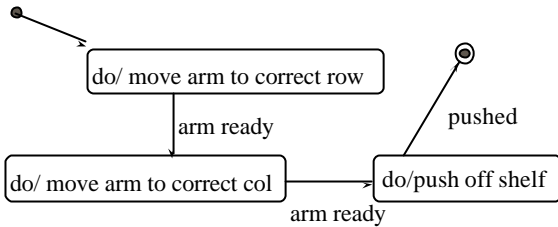
Nesting State Diagrams

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

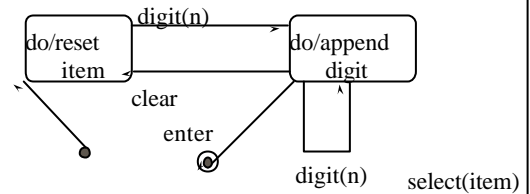
Example: Vending Machine



Example: Dispense Item



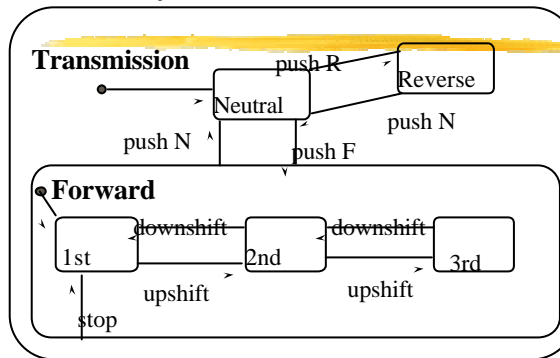
Example: Select Item



Generalisation of States

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

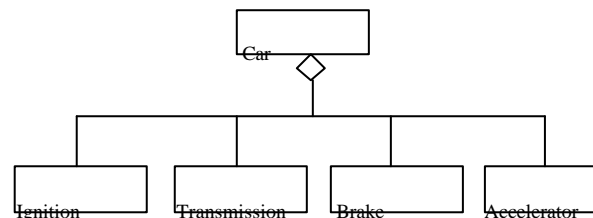
Example: Transmission



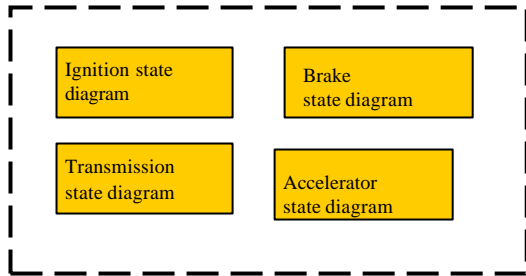
Example: Generalisation

- 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**.

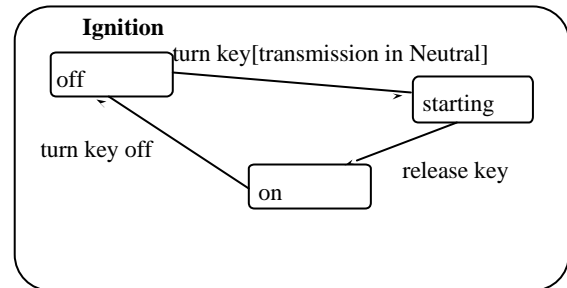
Example: Object Model



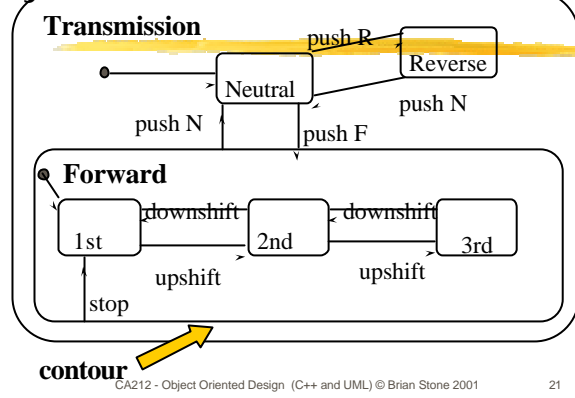
Dynamic Model



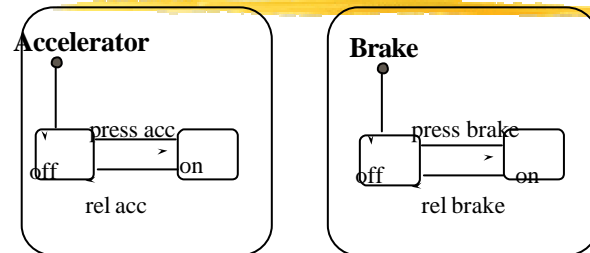
Dynamic Model: Ignition



Dynamic Model: Transmission

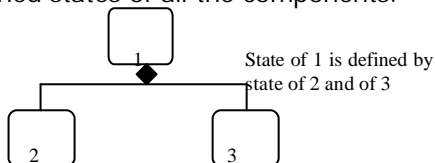


Dynamic Model: Accelerator & Brake



Concurrency

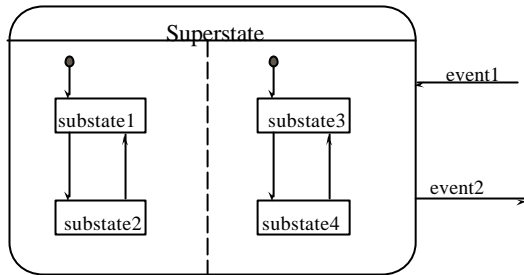
- Aggregation concurrency: The aggregate state corresponds to the combined states of all the components.



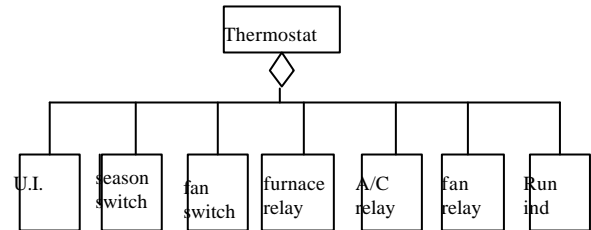
Concurrency (cont..)

- 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..)



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.