Information System Design
IT60105

Lecture 13

Statechart Diagrams

Lecture #13

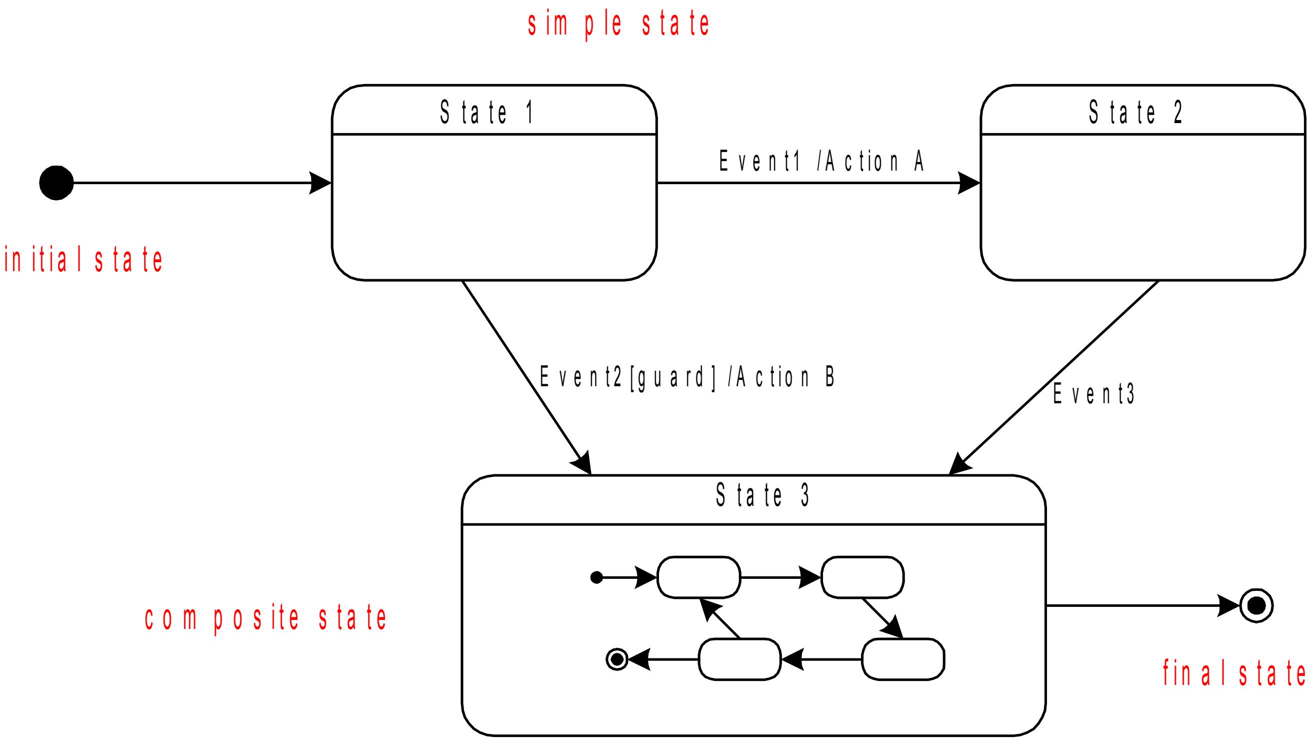
- What is a Statechart diagram?
- Basic components in a state-chart diagram and their notations
- Examples: *Process Order* in OLP system

What is a Statechart Diagram?

- A state-chart diagram is used to model the dynamic aspects of the system. The basic idea is same as the state machine in Finite Automata
- We can draw state diagrams for each object involved in the system
- An object may be in several states of its life time. When a message arrives to an object (event) it undergoes certain operations (action) or changes its state
- A state diagram shows how an object will react to the arrival of an event
  - Each reaction may be a sequence of actions, possibly accompanied by a transition from one named state to another
  - An event represents the receipt of a signal, or the effect of an operation call
  - An action represents the sending of a signal, or the call of an operation

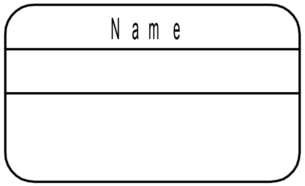
General Structure of a Statechart Diagram

- A statechart diagram typically look like this

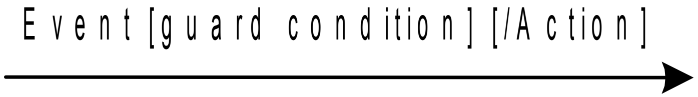


Basic Components in a Statechart Diagram

- Two basic elements are there
  - Rounded rectangle box representing the state



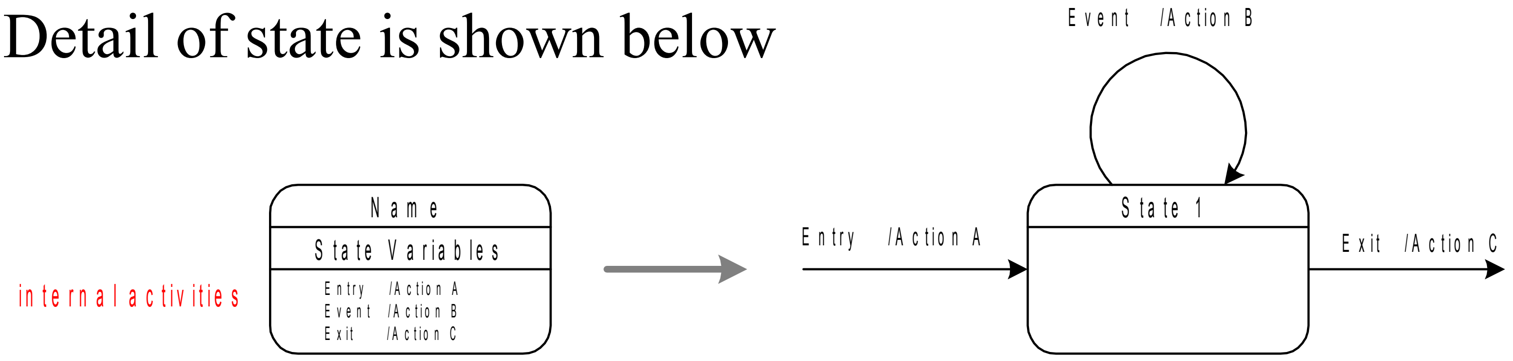
  - Labeled arrow indicating the transitions



    - Here, event is the message that is sent
    - guard condition is the Boolean expression of attribute values that allows a state transition only if the condition is true
    - An action is the behavior that occurs when the state transition occurs

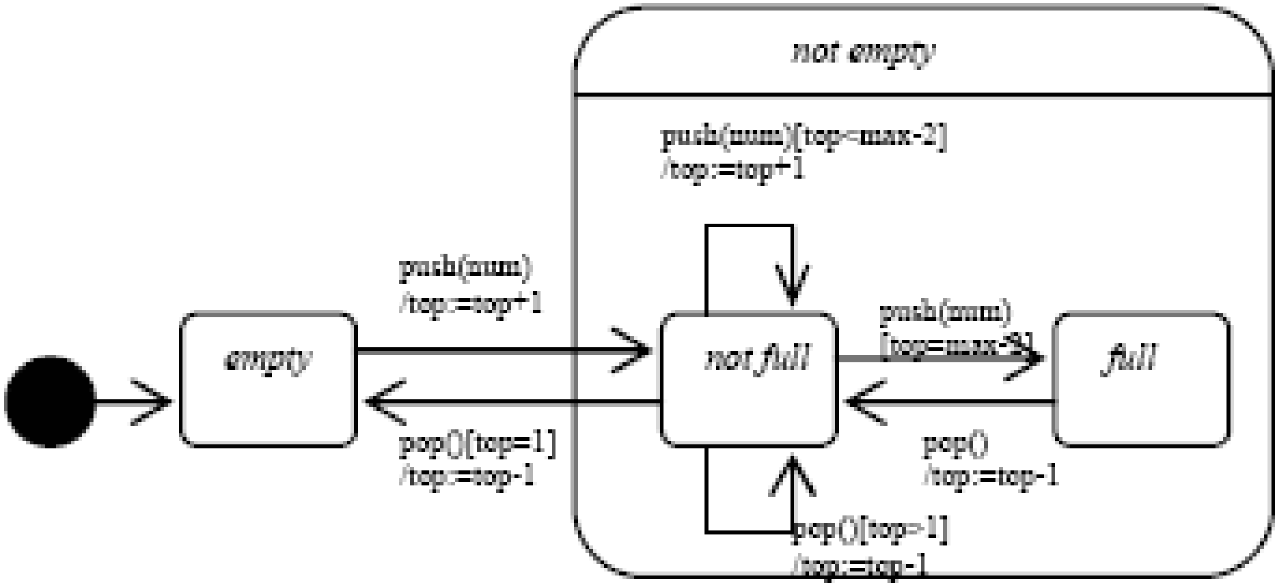
Detail State in a State-Chart Diagram

- Detail of state is shown below



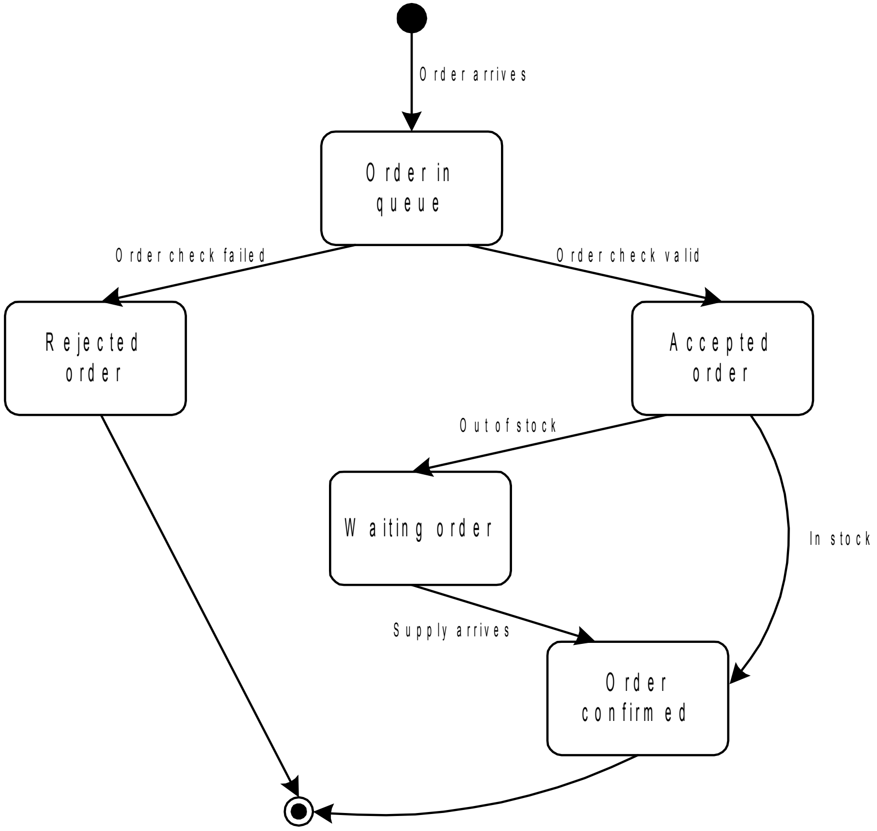
- Entry actions: that are common to every incoming transition
- Exit actions: that are common to every outgoing transition
- Self-transition: action within the state itself
- If there is no guard or if the guard is true, then for the event the actions will be followed and it can enter [exit] to [from] a state or remain in the same state

Example: Stack Machine



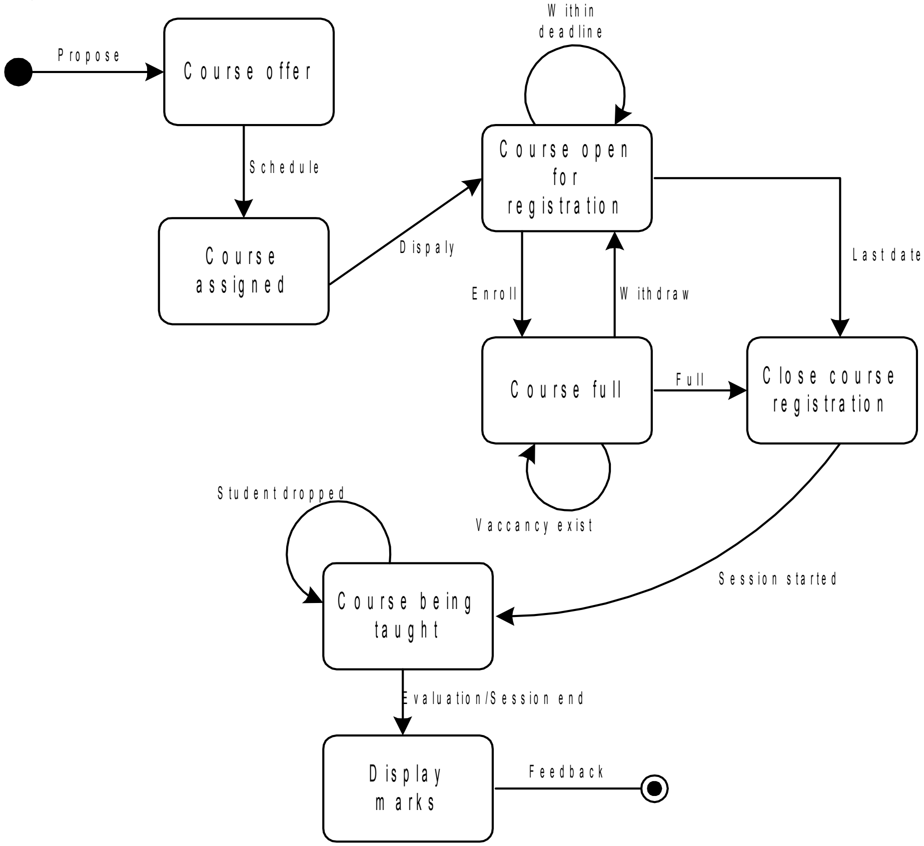
Example: Statechart Diagram

- In the “Process Order” use case, the object that is being manipulated is, namely, ORDER. So, a state diagram can be drawn to model how the object changes its state in this particular use case



Statechart Diagram: Another Example

- Suppose, a COURSE is an object in a usual "Course Registration" use case (consider the SEIIT). Following is a state diagram to model the behavioral view of the object COURSE



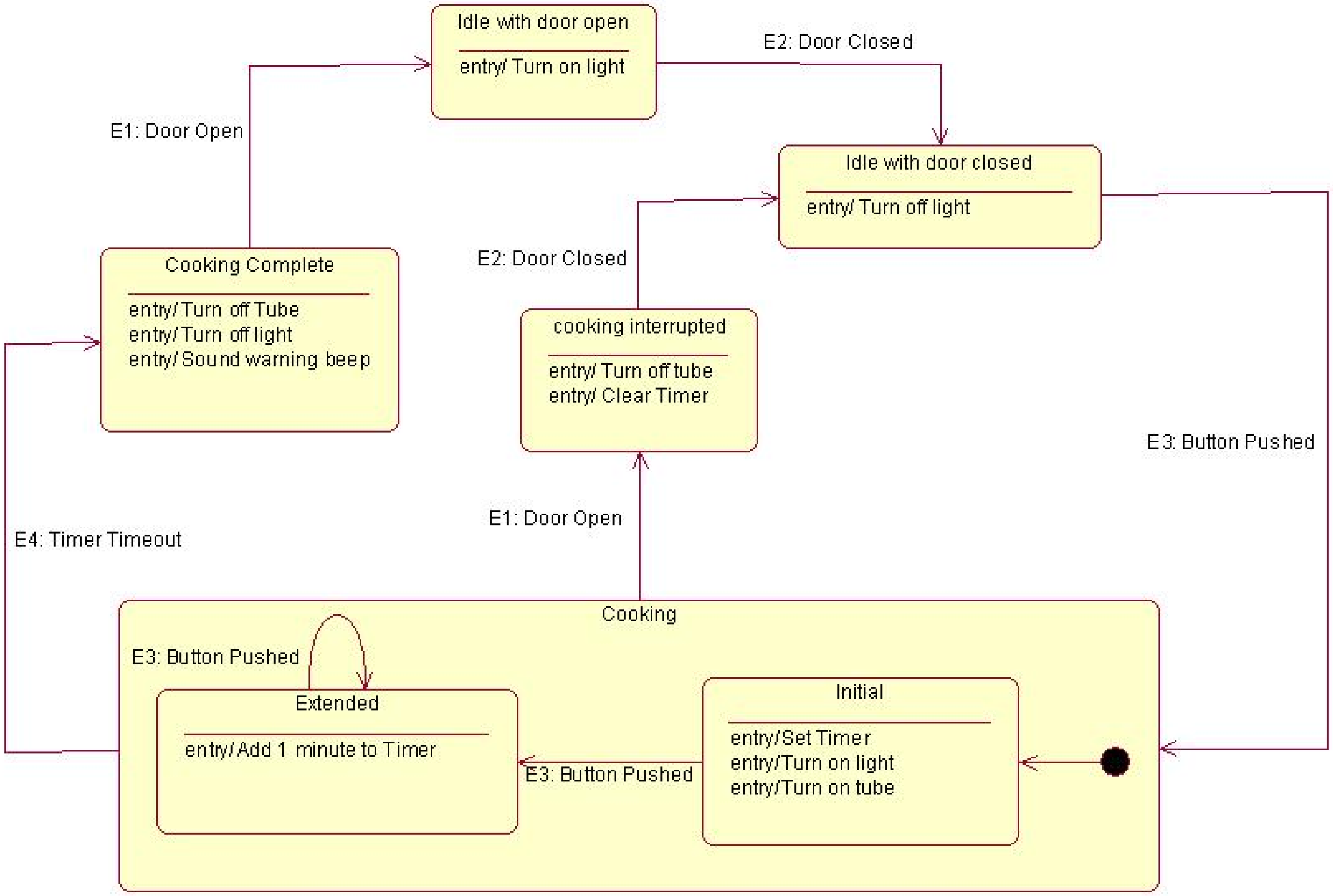
Advantages of Statechart Diagram

- A statechart diagram is used to model how the state of an object changes in its life time

- Statechart diagrams are good describing how the behavior of an object change across several use case executions

- However, if we are interested in modeling some behavior that involves the several objects collaborating with each other, the statechart diagram is not appropriate

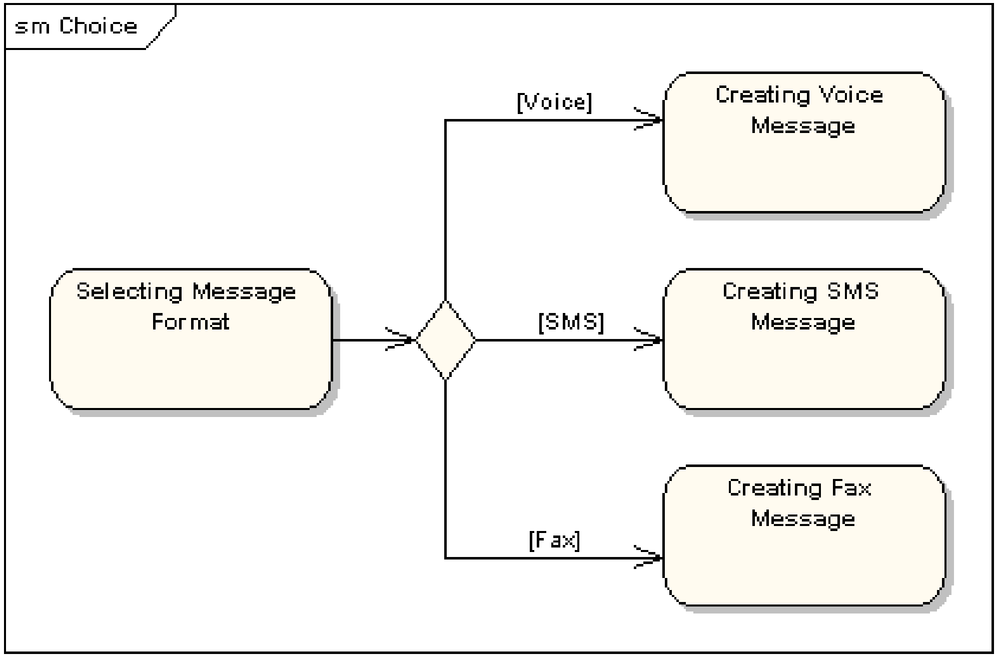
Example: Microwave Oven



More features in Statechart Diagram

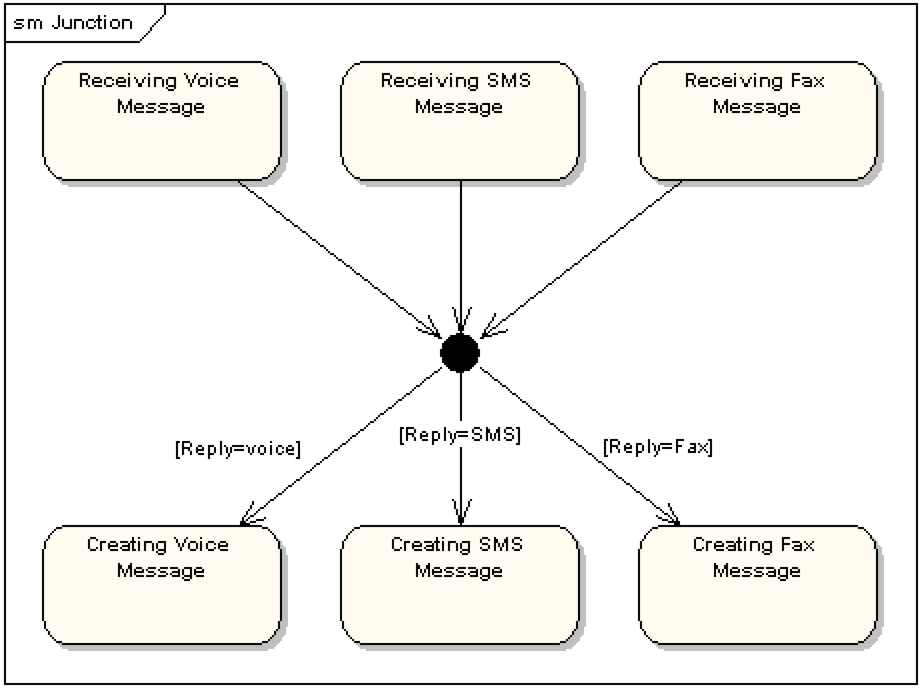
Choice Pseudo-State

- A choice pseudo-state is shown as a diamond with one transition arriving and two or more transitions leaving. The following diagram shows that whichever state is arrived at after the choice pseudo-state is dependent on the message format selected during execution of the previous state



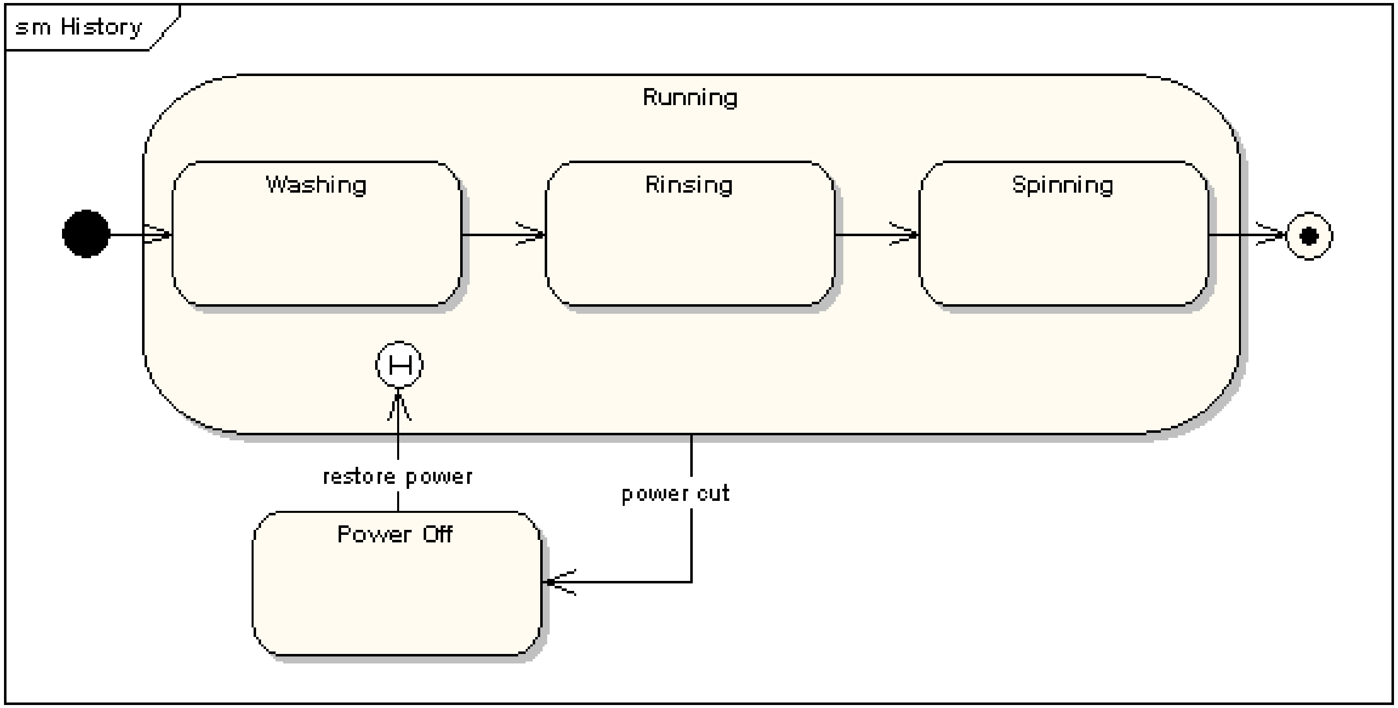
Choice Junction-State

- Junction pseudo-states are used to chain together multiple transitions. A single junction can have one or more incoming and one or more outgoing transitions and a guard can be applied to each transition. Junctions are semantic-free; a junction which splits an incoming transition into multiple outgoing transitions realizes a static conditional branch as opposed to a choice pseudo-state which realizes a dynamic conditional branch



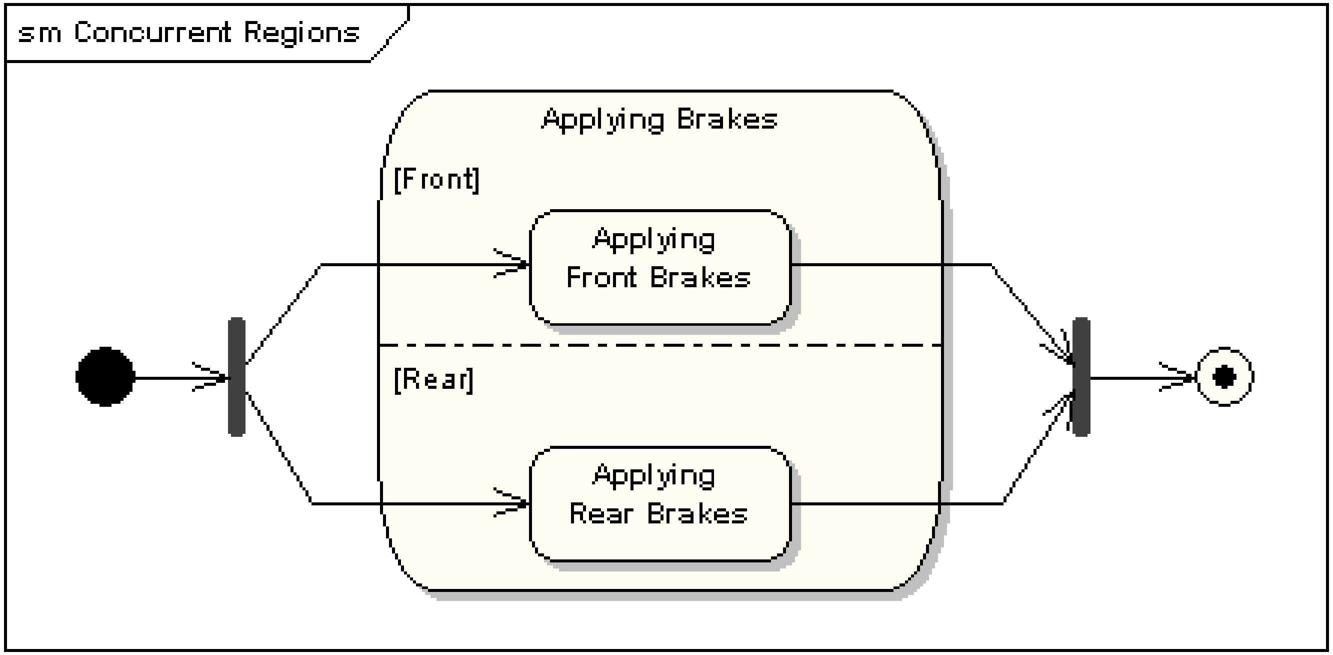
History State

- A History State is used to remember the previous state of a state machine when it was interrupted. The following diagram illustrates the use of history states. The example is a state machine belonging to a washing machine



Concurrent State

- A state may be divided into regions containing sub-states that exist and execute concurrently. The example below shows that within the state "Applying Brakes", the front and rear brakes will be operating simultaneously and independently. Notice the use of fork and join pseudo-states rather than choice and merge pseudo-states. These symbols are used to synchronize the concurrent threads


sm Concurrent Regions
Applying Brakes
[Front]
Applying Front Brakes
[Rear]
Applying Rear Brakes