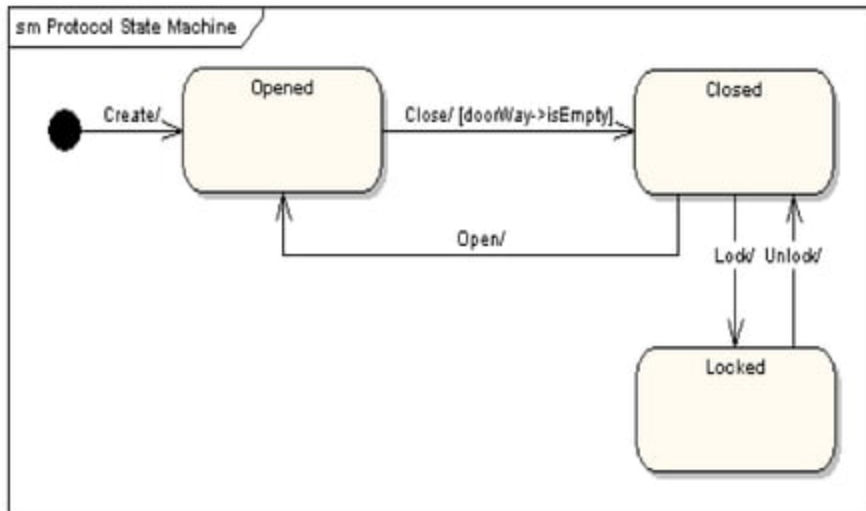


UML 2 State Machine Diagrams

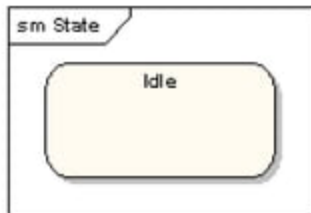
State Machine Diagrams

- UML state machine diagrams depict the various states that an object may be in and the transitions between those states.
- In other modeling languages, it is common for this type of a diagram to be called a state-transition diagram or even simply a state diagram.

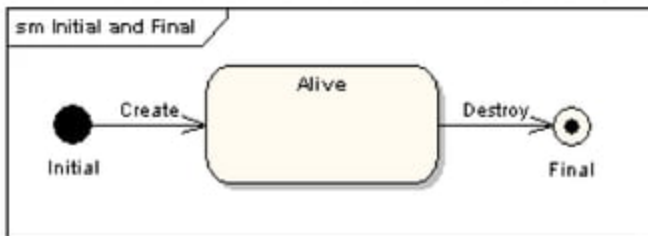


State Machine Diagrams

- **States** - A state is denoted by a round-cornered rectangle with the name of the state written inside it.

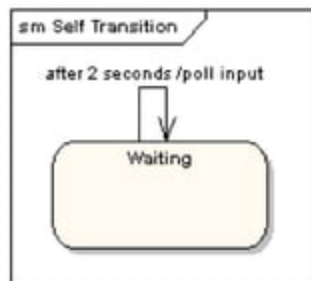
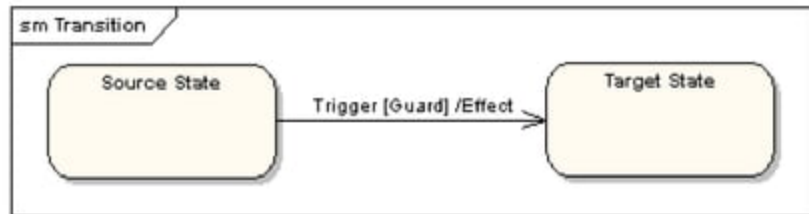


- **Initial and Final States** - The initial state is denoted by a filled black circle and may be labeled with a name. The final state is denoted by a circle with a dot inside and may also be labeled with a name.

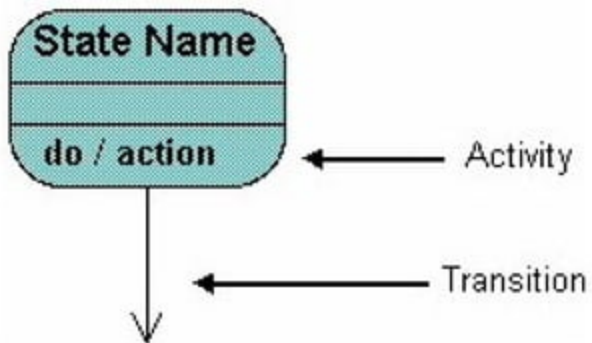


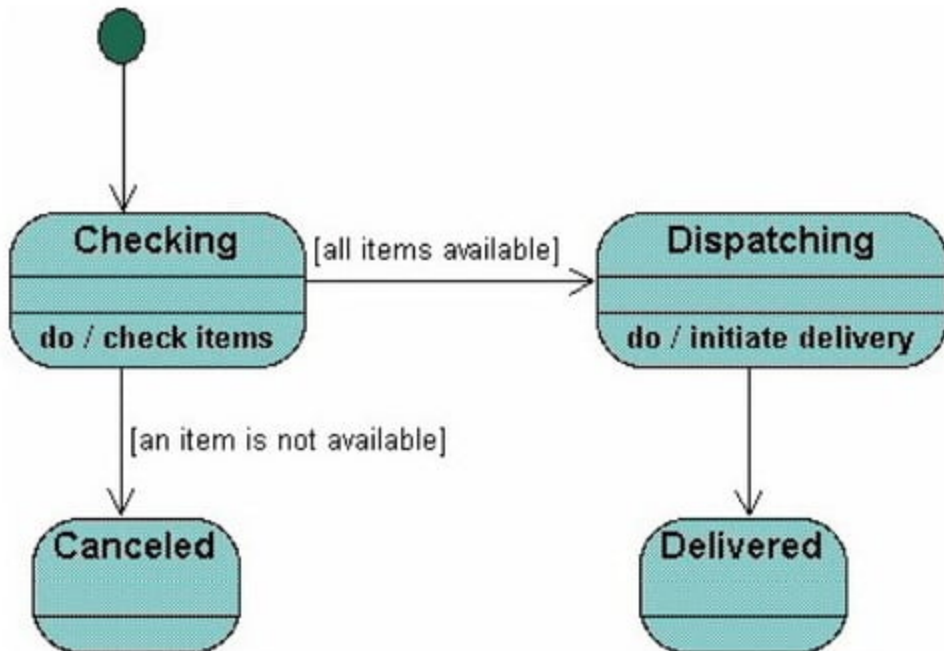
State Machine Diagrams

- **Transitions** - Transitions is a progression from one state to another are denoted by lines with arrowheads. A transition may have a trigger, a guard and an effect.
- **Self-Transitions** - A state can have a transition that returns to itself, as in the following diagram. This is most useful when an effect is associated with the transition.



- "**Trigger**" is the cause of the transition, which could be a signal, an event, a change in some condition, or the passage of time. "**Guard**" is a condition which must be true in order for the trigger to cause the transition. "**Effect**" is an action which will be invoked directly on the object that owns the state machine as a result of the transition.

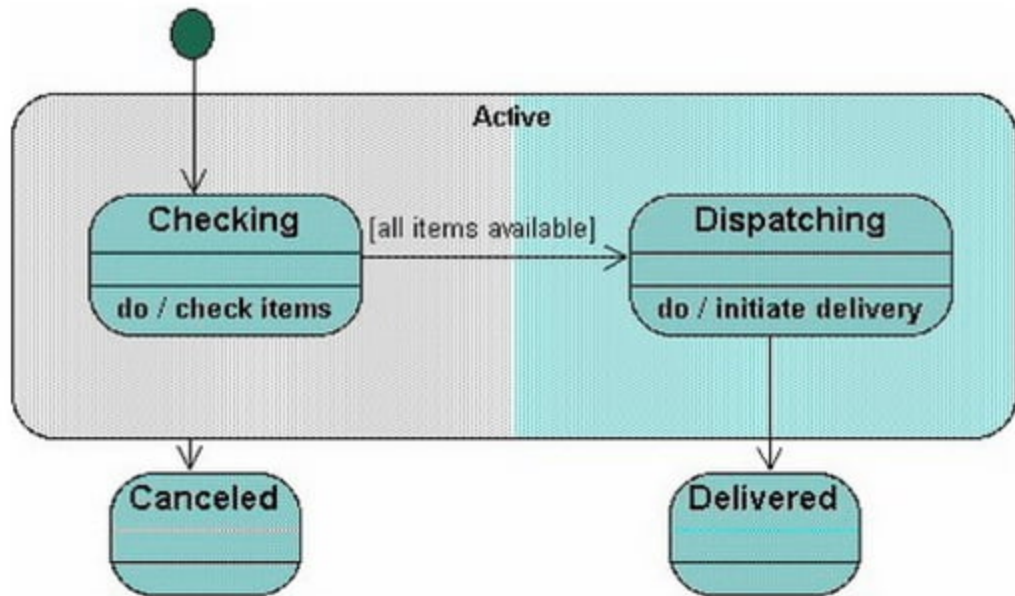




Super state

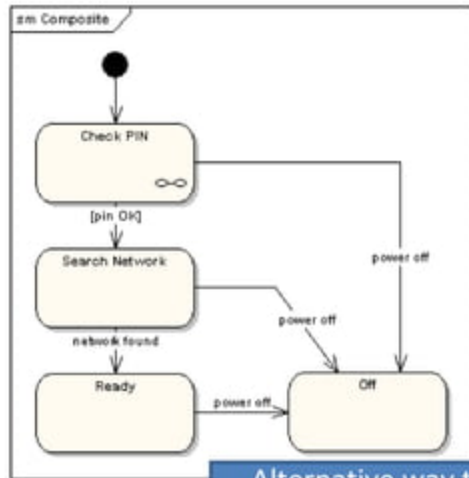
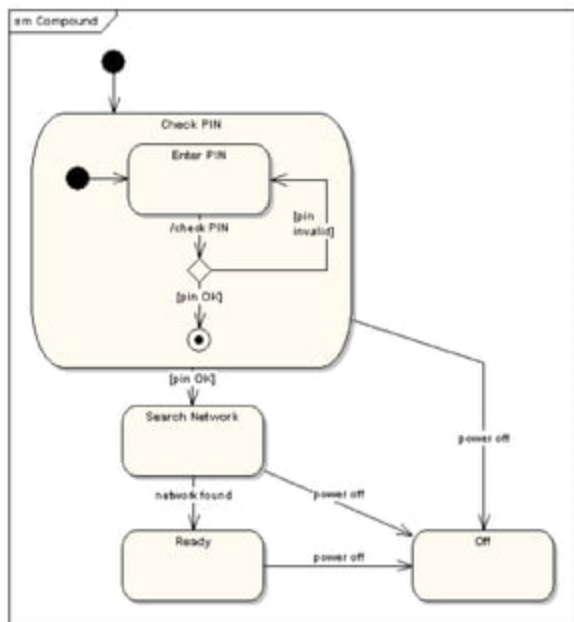
- A super-state is used when many transitions lead to the a certain state. Instead of showing all of the transitions from each state to the redundant state a super-state can be used to show that all of the states inside of the super-state can transition to the redundant state.
- This helps make the state diagram easier to read.

Super state



State Machine Diagrams

- Compound States** - A state machine diagram may include sub-machine diagrams, as in the example below.

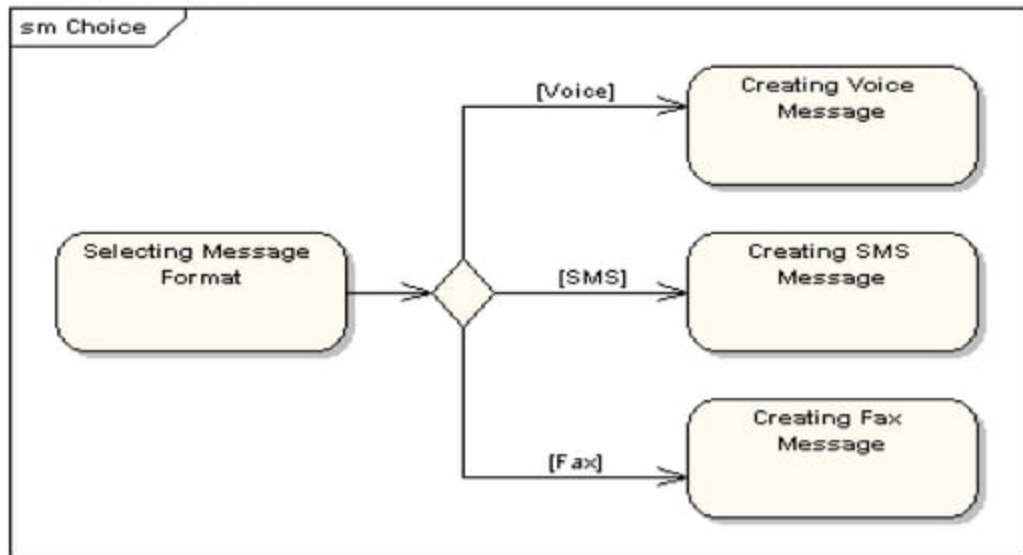


Alternative way to show the same information

- The ∞ symbol indicates that details of the Check PIN sub-machine are shown in a separate diagram.

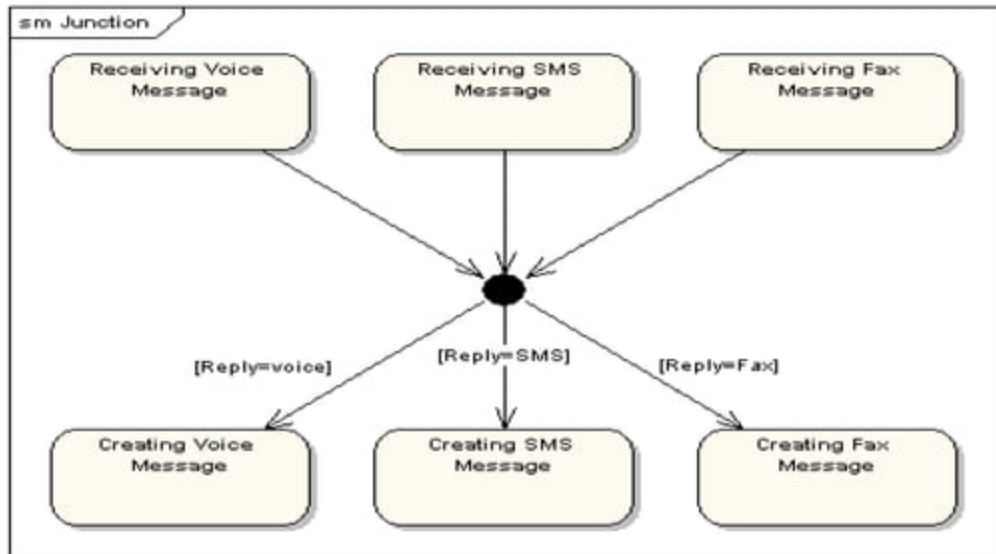
State Machine Diagrams

- 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.



State Machine Diagrams

- Junction Pseudo-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; 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.



State Machine Diagrams

- Concurrent Regions** - 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.

