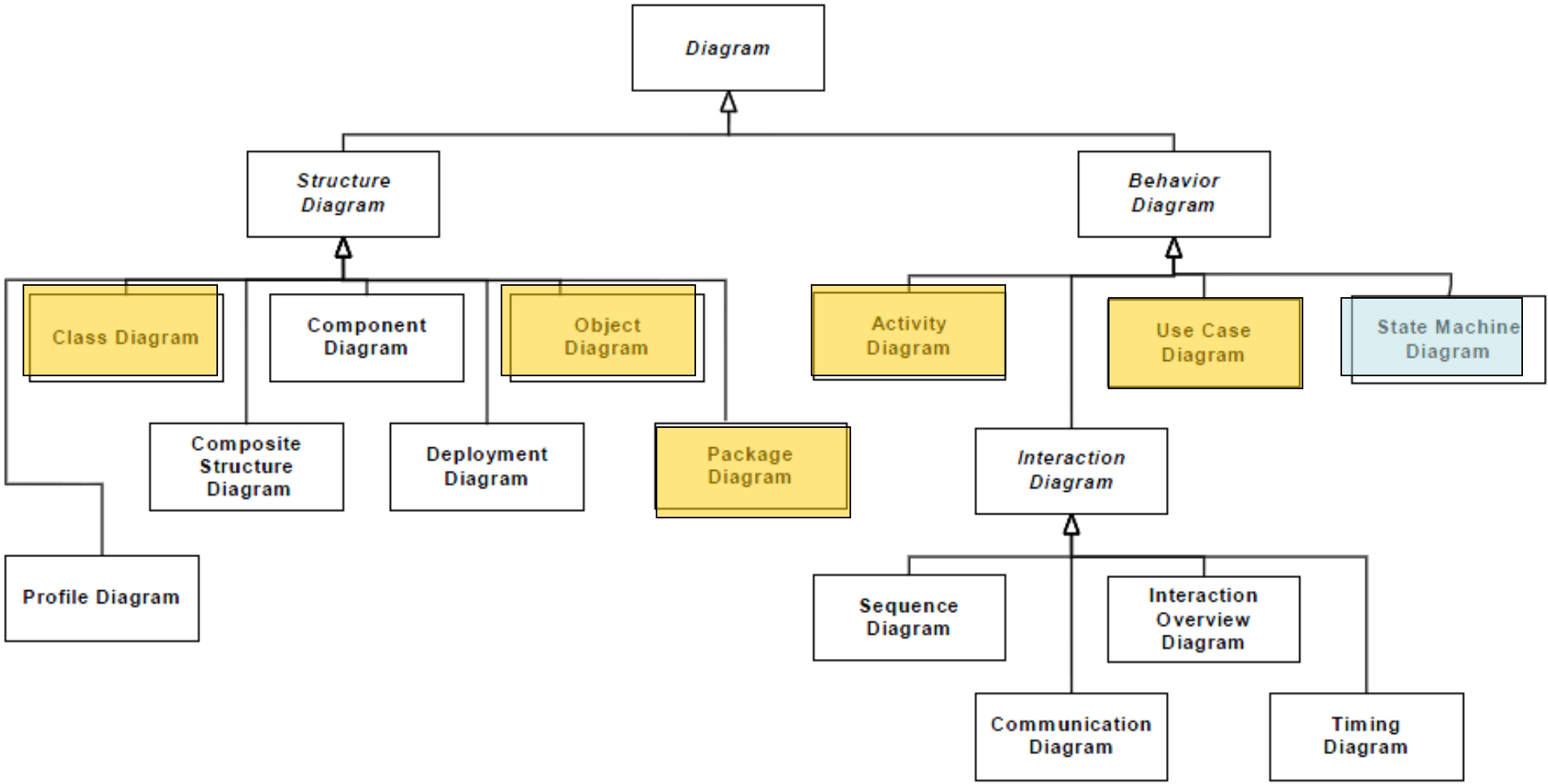
Object-Oriented Technology and UML

State Machine Diagram

State Machine Diagram



Topics

- Basic concepts
- Constituent elements
- Representation
- Reading method
- Modeling

Concept of the State

- A state of an object is a period of time during which it satisfies some condition, performs some activity, or waits for some event

Example of States

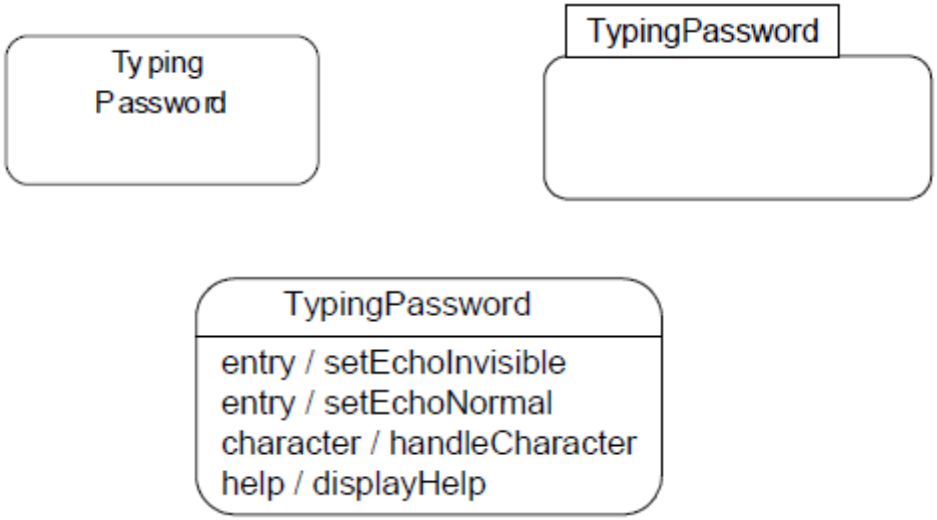
- A heater in a home might be in any of four states
  - Idle (waiting for a command to start heating the house)
  - Activating (its gas is on, but it's waiting to come up to temperature)
  - Active (its gas and blower are both on)
  - ShuttingDown (its gas is off but its blower is on, flushing residual heat from the system)

Concept of the State

- A state has several parts
  - Name
    - A textual string that distinguishes the state from other states; a state may be anonymous, meaning that it has no name
  - Entry/exit effects
    - Actions executed on entering and exiting the state, respectively
  - Internal transitions
    - Transitions that are handled without causing a change in state
  - Substates
    - The nested structure of a state, involving nonorthogonal (sequentially active) or orthogonal (concurrently active) substates
  - Deferred events
    - A list of events that are not handled in that state but, rather, are postponed and queued for handling by the object in another state

Concept of the State

- You represent a state as a rectangle with rounded corners



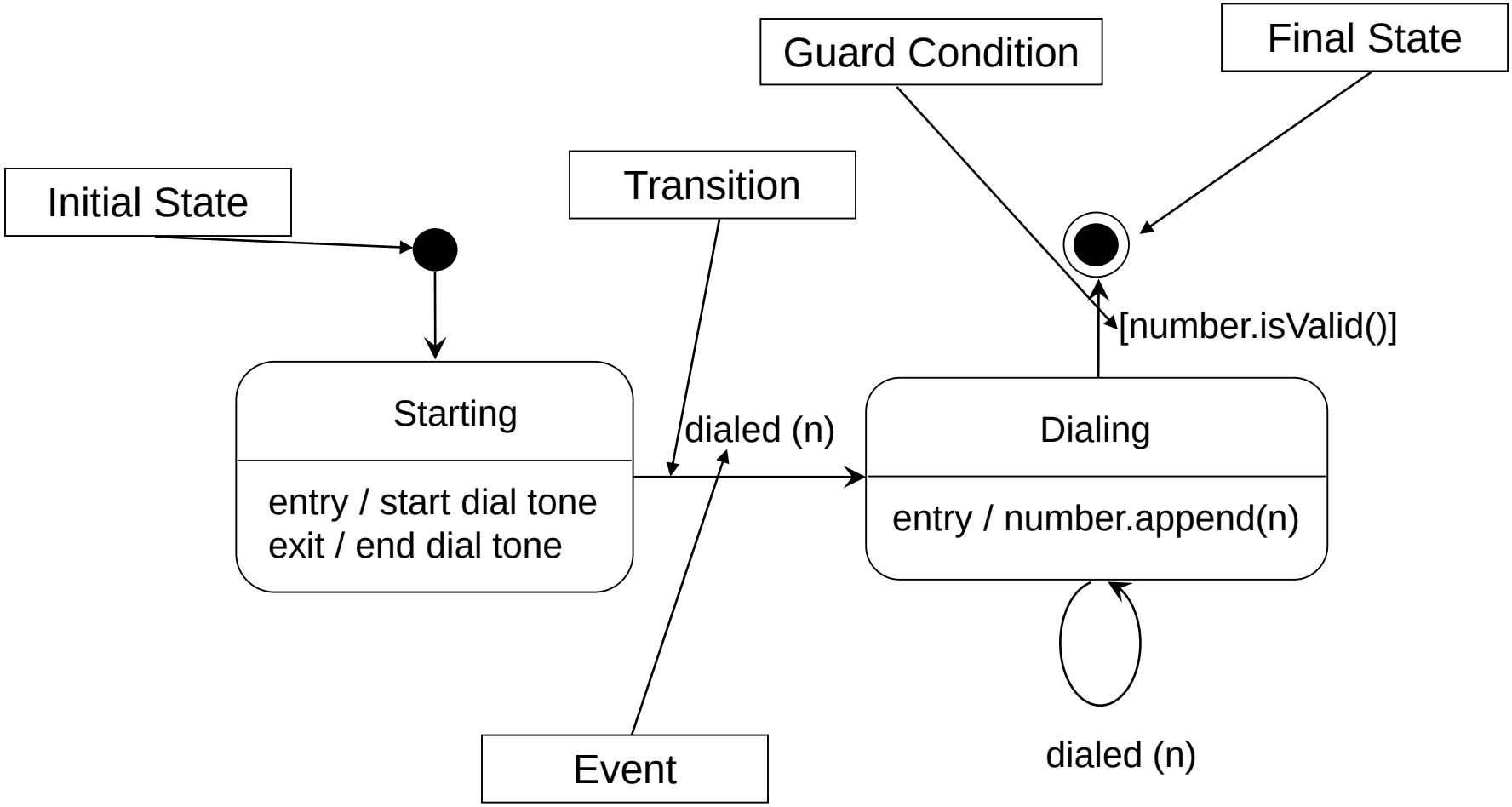
State Machine

- In the UML, you model the dynamic aspects of a system by using state machines
  - a state machine models the lifetime of a single object, whether it is an instance of a class, a use case, or even an entire system

State Machine Diagram

- State machines may be visualized using state machine diagrams
  - The UML provides a graphical representation of states, transitions, events, and effects
  - This notation permits you to visualize the behavior of an object in a way that lets you emphasize the important elements in the life of that object

Example of State Machine Diagram



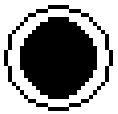
Constituent Elements of State Machine Diagram

- Initial state and final state
- Transition
  - Source and target state
  - Event trigger
  - Guard condition
  - Effect

Initial state and final state

- Initial state
  - Indicates the default starting place for the state machine or substate
  - An initial state is represented as a filled black circle

- Final state
  - Indicates that the execution of the state machine or the enclosing state has been completed
  - A final state is represented as a filled black circle surrounded by an unfilled circle (a bull's eye)



Transition

- A transition is a relationship between two states indicating that an object in the first state will perform certain actions and enter the second state when a specified event occurs and specified conditions are satisfied

Source and target state

- Source state
  - The state affected by the transition
- Target state
  - The state that is active after the completion of the transition

Event trigger

- In the context of state machines, an event is an occurrence of a stimulus that can trigger a state transition
- Events may include
  - Signals
  - Calls
  - The passing of time
  - A change in state

Guard condition

- A Boolean expression that is evaluated when the transition is triggered by the reception of the event trigger
  - If the expression evaluates true, the transition is eligible to fire
  - If the expression evaluates false, the transition does not fire
  - And if there is no other transition that could be triggered by that same event, the event is lost

Effect

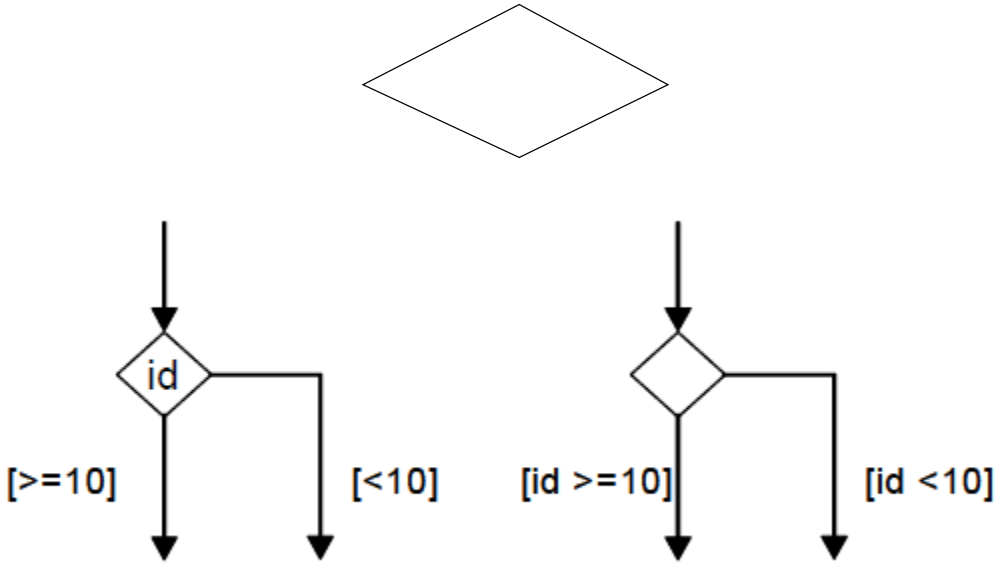
- An effect is a behavior that is executed when a transition fires
- Effects may include inline computation, operation calls (to the object that owns the state machine as well as to other visible objects), the creation or destruction of another object, or the sending of a signal to an object

Kinds of Transitions

- External transition
  - Include entry / exit activity
- Internal transition
  - Exclude entry / exit activity
- Local Transition
  - Local transitions execute the effect code for the transition and the exit and entry code for all sub states but not the exit and entry code for the state to which it is connected
  - Local transitions are hence only meaningful for composite states

Choice Pseudo-State

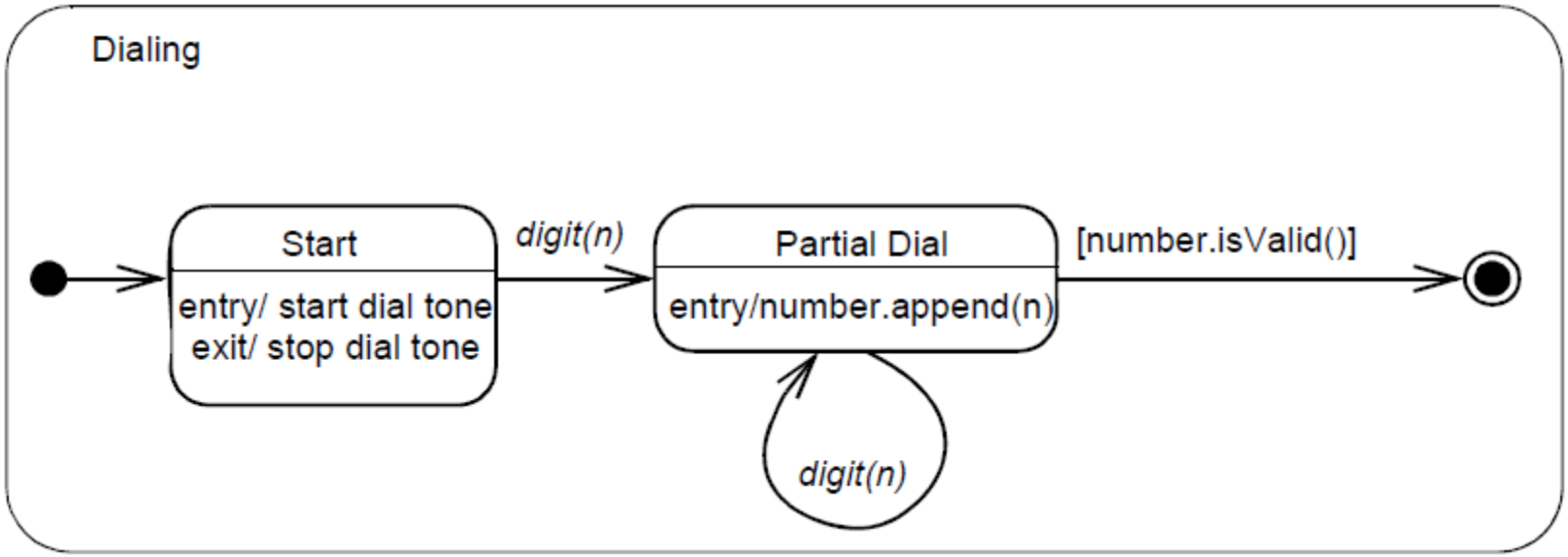
- A choice pseudo-state is shown as a diamond with one transition arriving and two or more transitions leaving

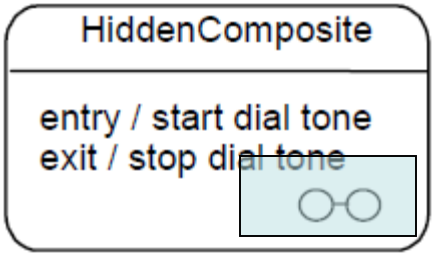


Composite state

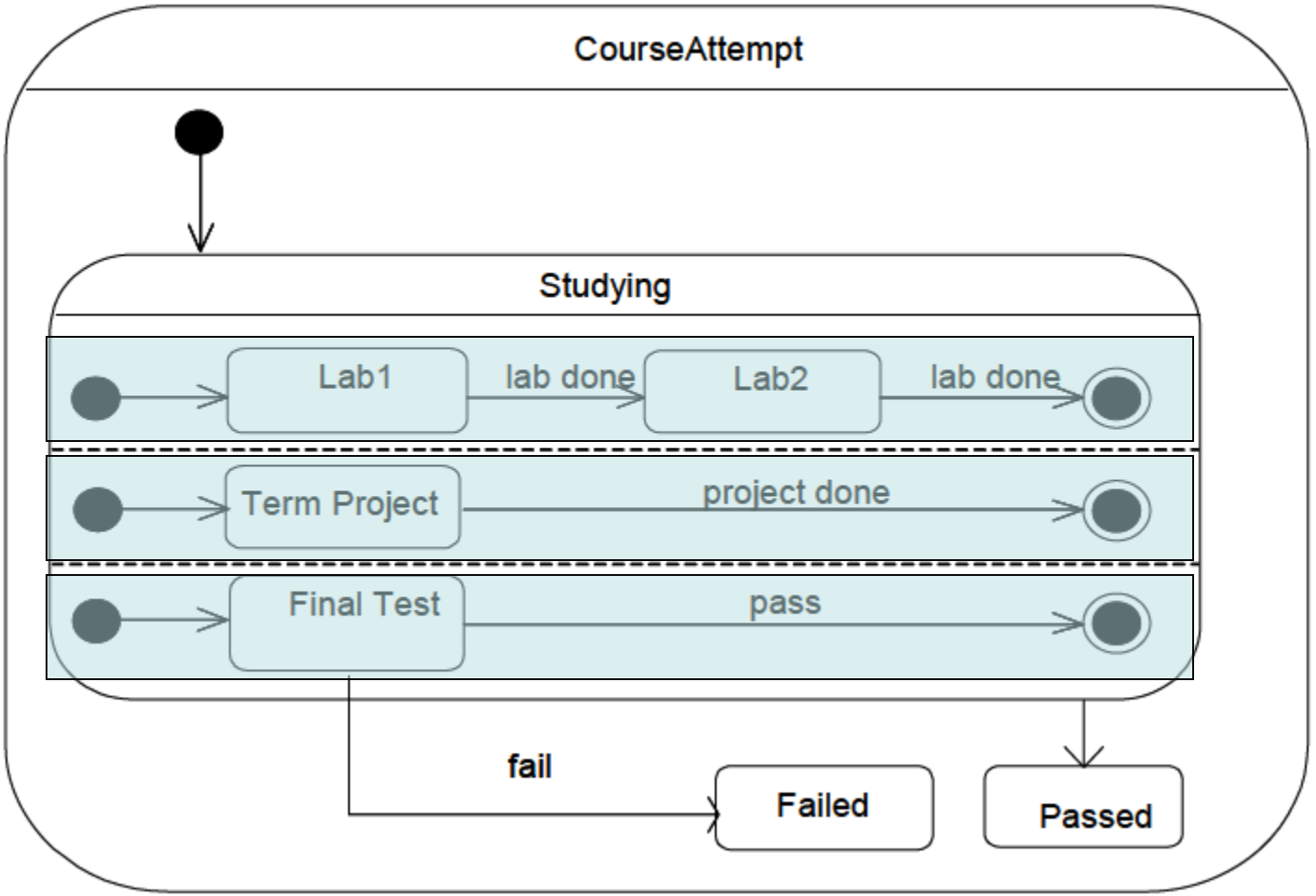
- A substate is a state that's nested inside another one
- A state that has substates (nested states) is called a composite state
- A composite state may contain either concurrent (orthogonal) or sequential (nonorthogonal) substates
- In the UML, you render a composite state just as you do a simple state, but with an optional graphic compartment that shows a nested state machine

Example of Composite state





Example of Composite state

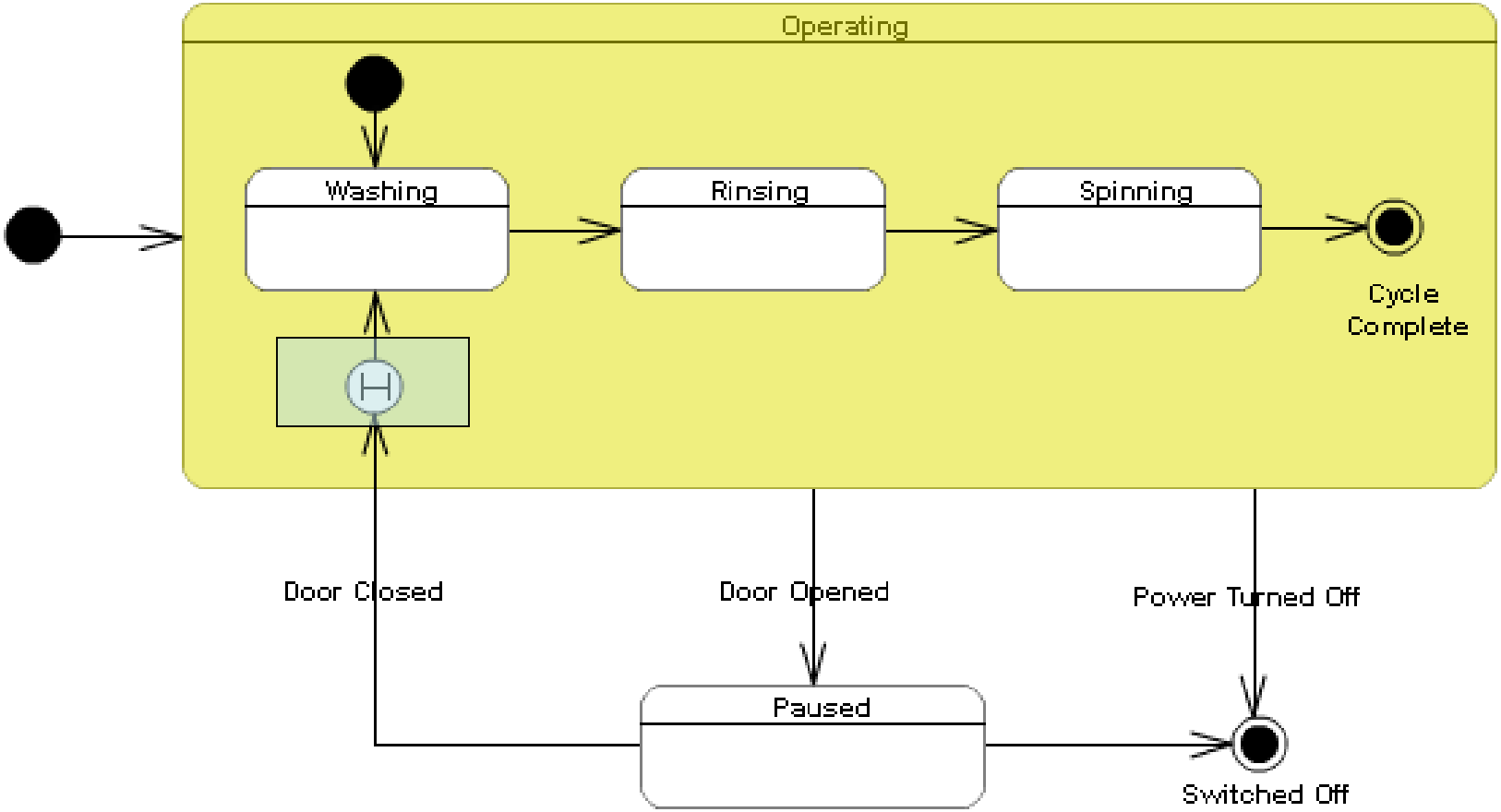


History State

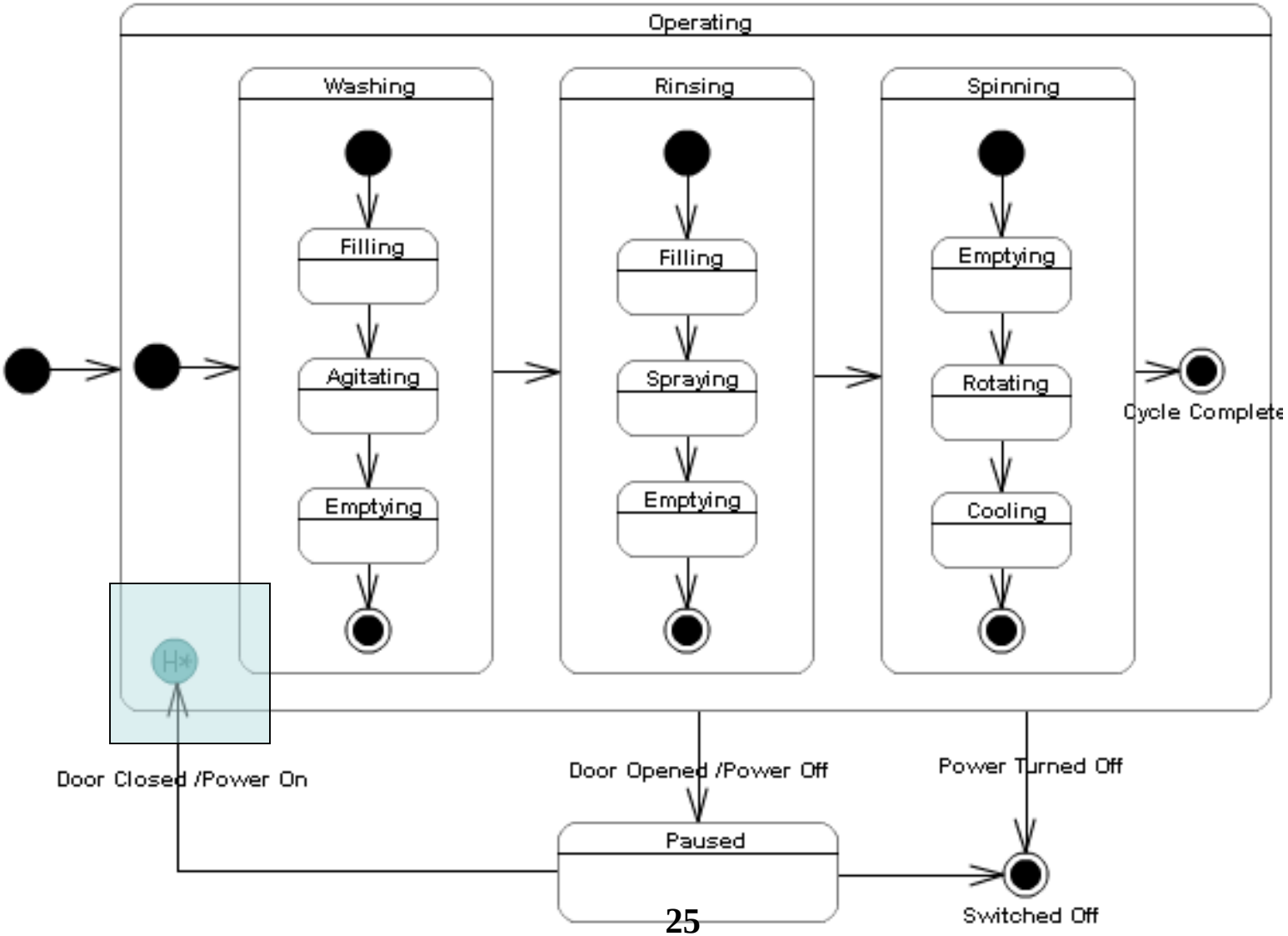
- A history state is used to remember the previous state of a state machine when it was interrupted
- Representation
  - Shallow history 

  - Deep history 

Example of History State



Example of History State



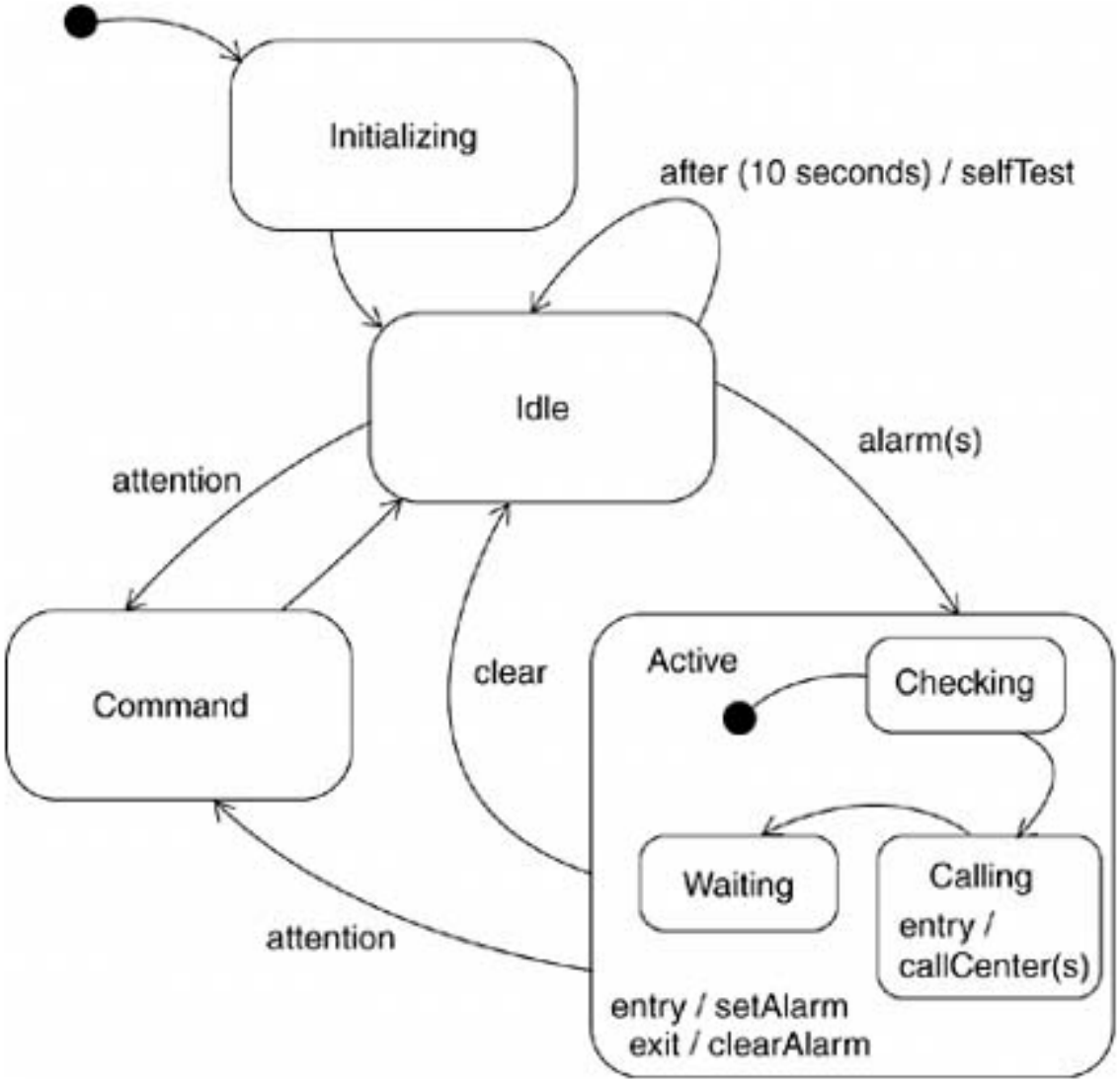
Application of State Machine Diagram

- Modeling the Lifetime of an Object
  - When you model the lifetime of an object, you essentially specify three things
    - The events to which the object can respond
    - The response to those events
    - And the impact of the past on current behavior

Steps of State Machine Diagram Modeling

- Find the major states
- Determine the transitions between states
- Refine the actions and transitions within the states
- Expand the details by using the composite state

Example of State Machine Diagram



Style of UML State Machine Diagram

- Do Not Overlap Guards
  - The guards on similar transitions leaving a state must be consistent with one another
  - For example
    - x < 0, x = 0 and x > 0
    - x <= 0 and x >= 0

Style of UML State Machine Diagram

- Never Place a Guard on an Initial Transition
- Indicate Entry Actions Only When Applicable to All Entry Transitions
- Indicate Exit Actions Only When Applicable to All Exit Transitions

Style of UML State Machine Diagram

- Question “Black-Hole” States
  - A black-hole state is one that has transitions into it but none out of it, something that should be true only of final states
  - This is an indication that you have missed one or more transitions
- Question “Miracle” States
  - A miracle state is one that has transitions out of it but none into it, something that should be true only of start points
  - This is also an indication that you have missed one or more transitions