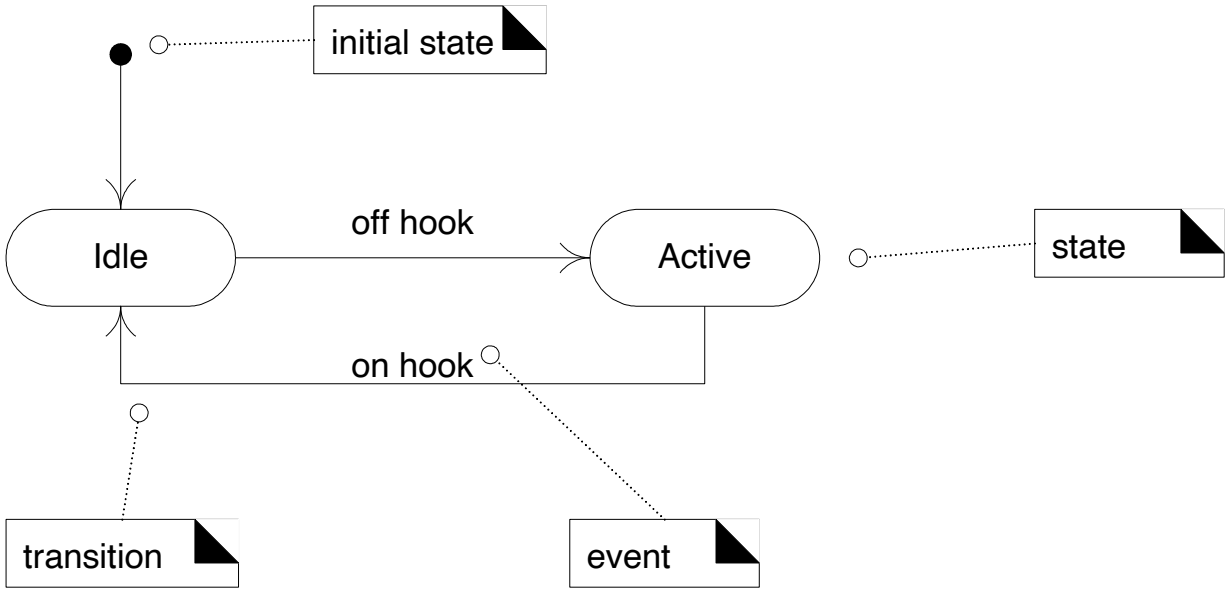
UML State Diagrams
And
State Pattern



- Shows lifecycle of an object
- Event: Significant occurrence
  - Example: Phone receiver taken off the hook
  - Example: User moves to next level
- State: Condition of an object (or group of objects) at a moment in time
- Transition: A relationship between states
  - One transition: Object moves from one state to the other when a particular event occurs

How to apply state machine diagrams?

- Object state independent:
  - Response to a given event always the same
- Object state dependent:
  - Based on state, object may respond differently to different events
  - Example: Vending Machine buy button
    - “state == “NO PAYMENT” → Response: “Pay First”
    - “state == “HAS PAYMENT” → Response: “Deliver”
  - “Use state machines for state-dependent objects with complex behavior

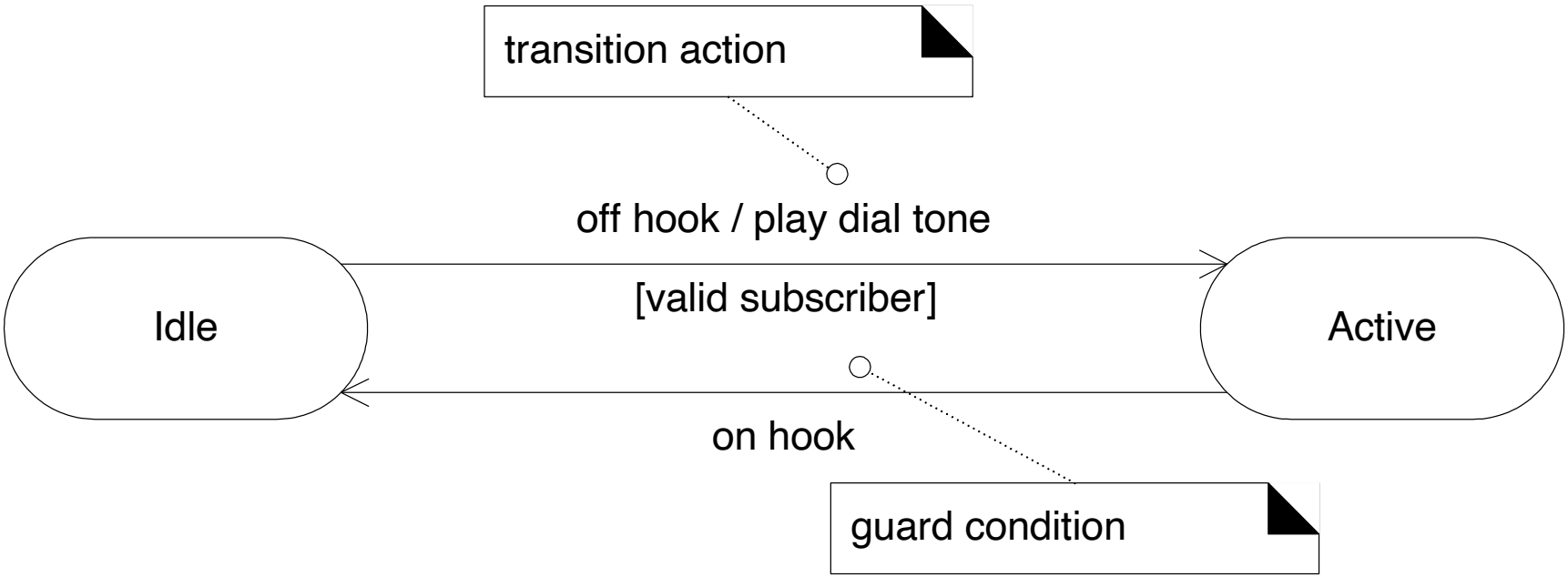
When to use state machine diagrams?

- Typically not useful in business information systems
- More useful in “time varying systems”
  - Process control
  - Device control
  - Protocol handlers
  - Games
  - Telecommunications programs

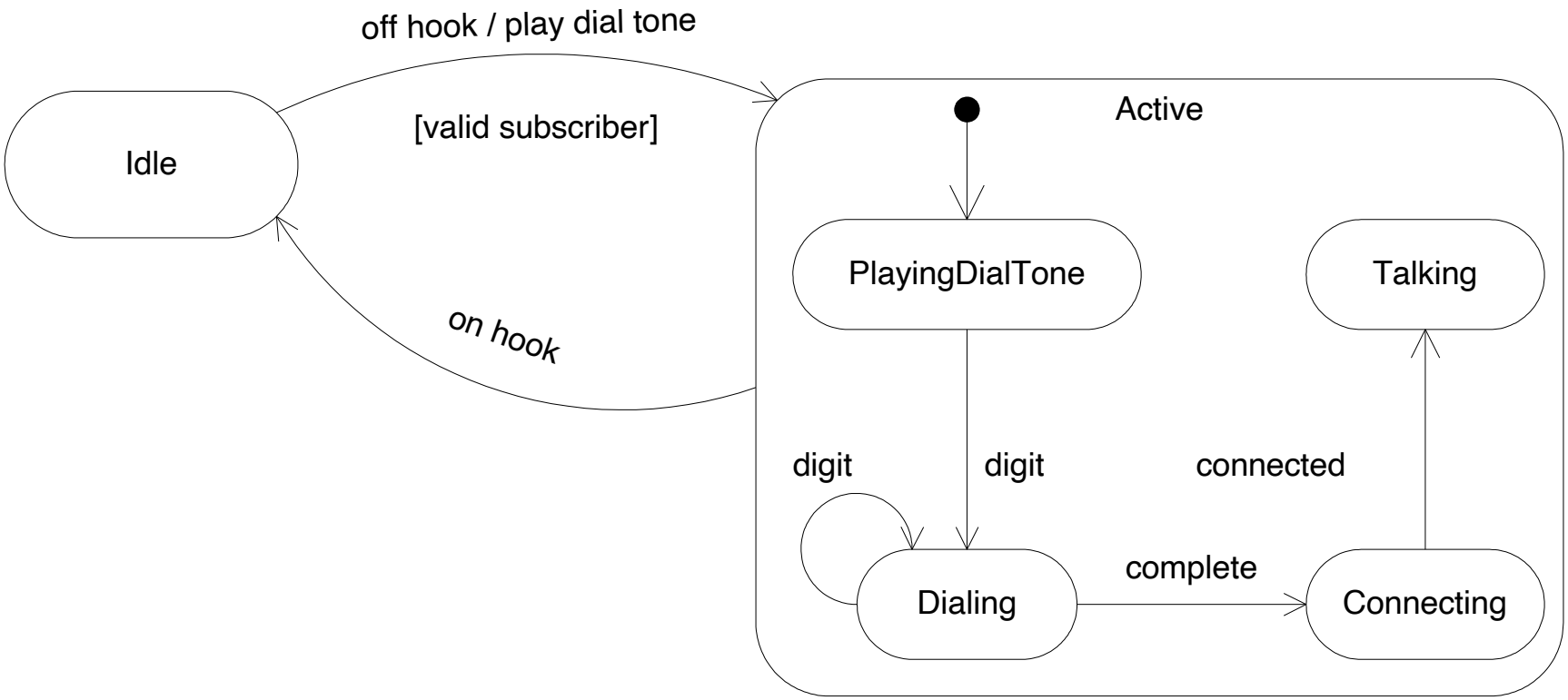
How are state machines used?

- To model the behavior of a complex reactive machine in response to events
  - Example: Vending Machine
  - Example: Navigation between web pages in a site
  - Example: Event handling based on mode or conditions (e.g. Edit-Paste button active only if clipboard has stuff in it)
- To model legal sequence of operations
  - Example: Protocol or language specifications

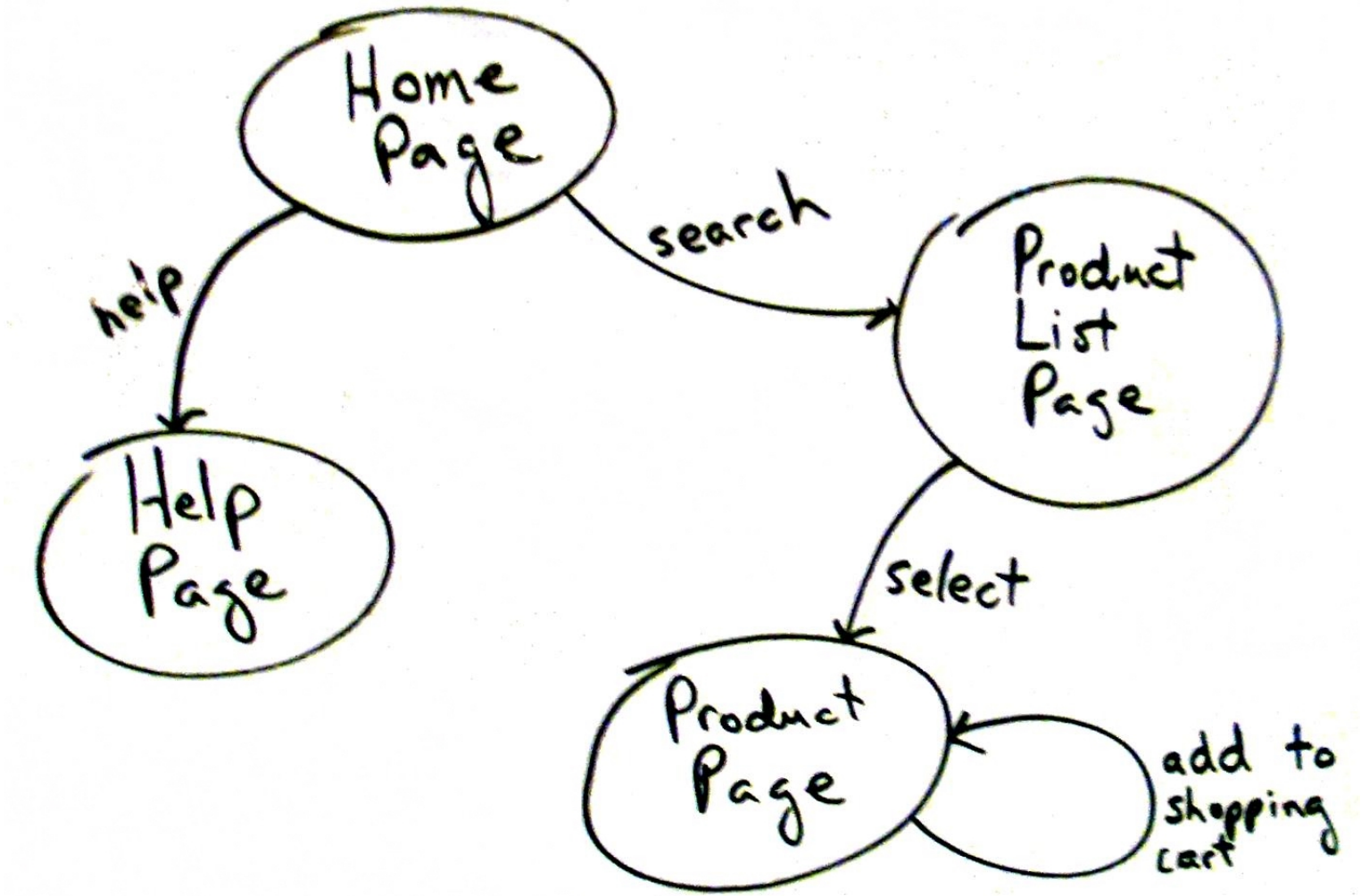
More state diagram notation



Nested (hierarchical states)



UI Navigation Example



Example - Su Otomatı (A Vending Machine)