CS2062 Object Oriented Software Development

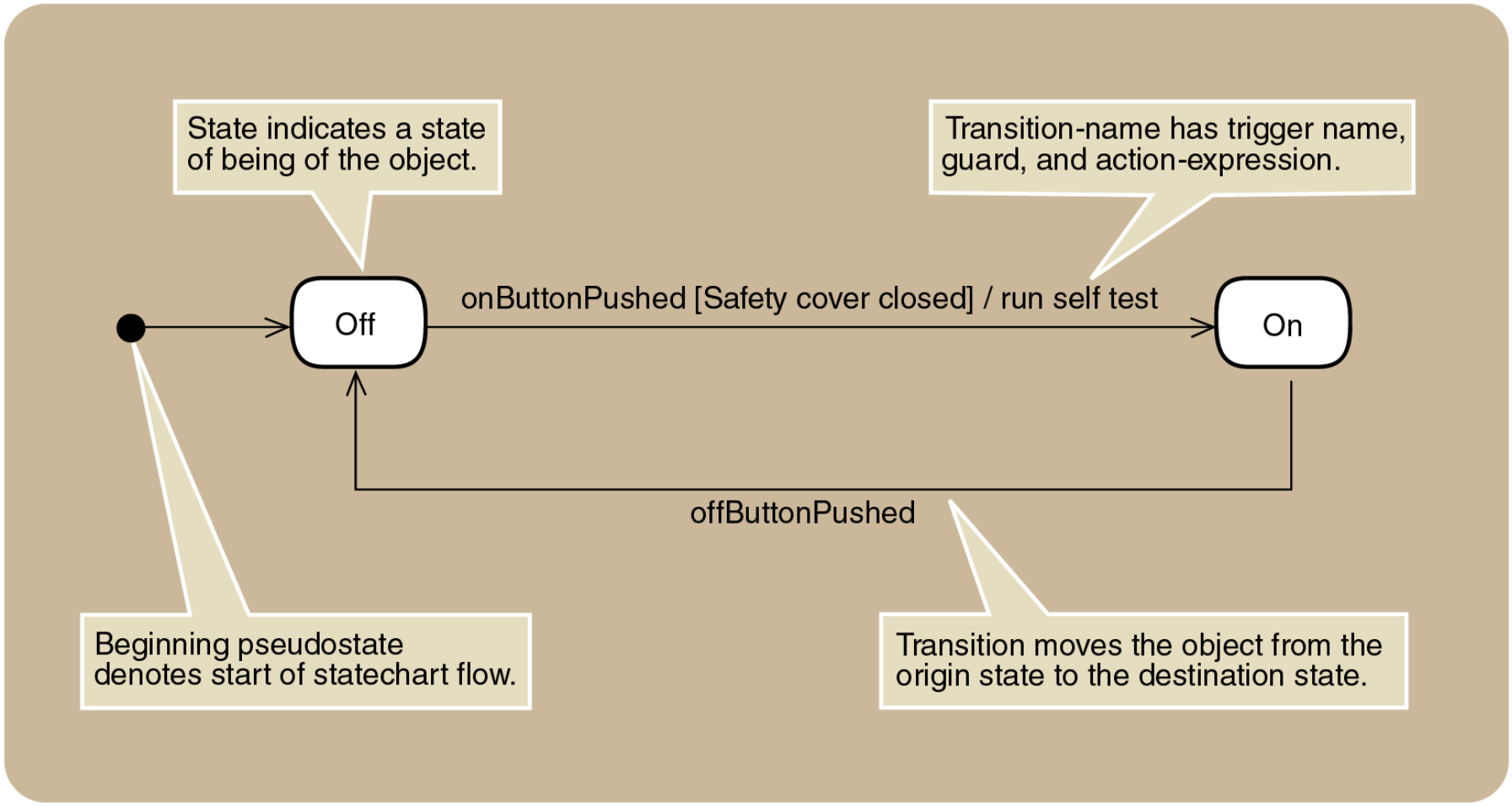
Lecture 6

State chart diagrams

Identifying the Object Behavior with State Charts

- Business **objects** may have **status conditions** which are **important** for the business / stakeholders and need to be **tracked**
  - Spans many business events
- State chart diagram
  - Developed for complex problem domain classes
  - Composed of ovals representing status of an object
  - Arrows represent transitions

Simple State chart for a Printer



Transition Format : transition-name (parameters)[guard-condition]/action expression

State and Transition

- State is a semi-permanent condition of an object
- External event can change the current state of an object
- Object remains in a state until a particular event occurs which moves it to another state
- Such a movement is called a transition
- Transitions are considered to be short in duration compared to states
- Transition name is the trigger / message event which causes the object to make the transition

Guard Condition and Action Expression

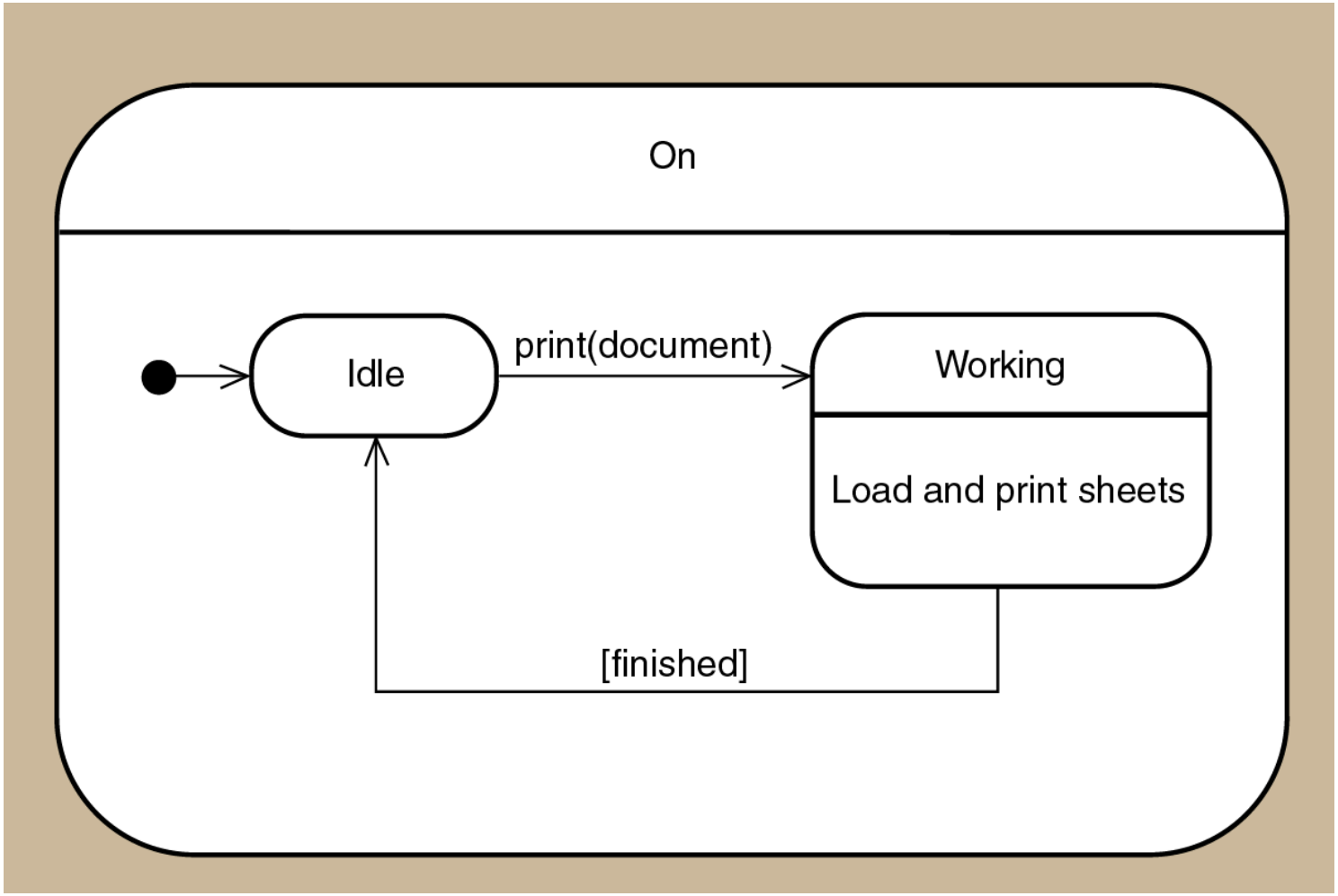
- Guard condition is a true/false condition which acts as a qualifier for the transition
- Condition should be true for the transition trigger to fire
- Action expression is a procedural expression that executes when the transition trigger fires

Identifying states -Guidelines

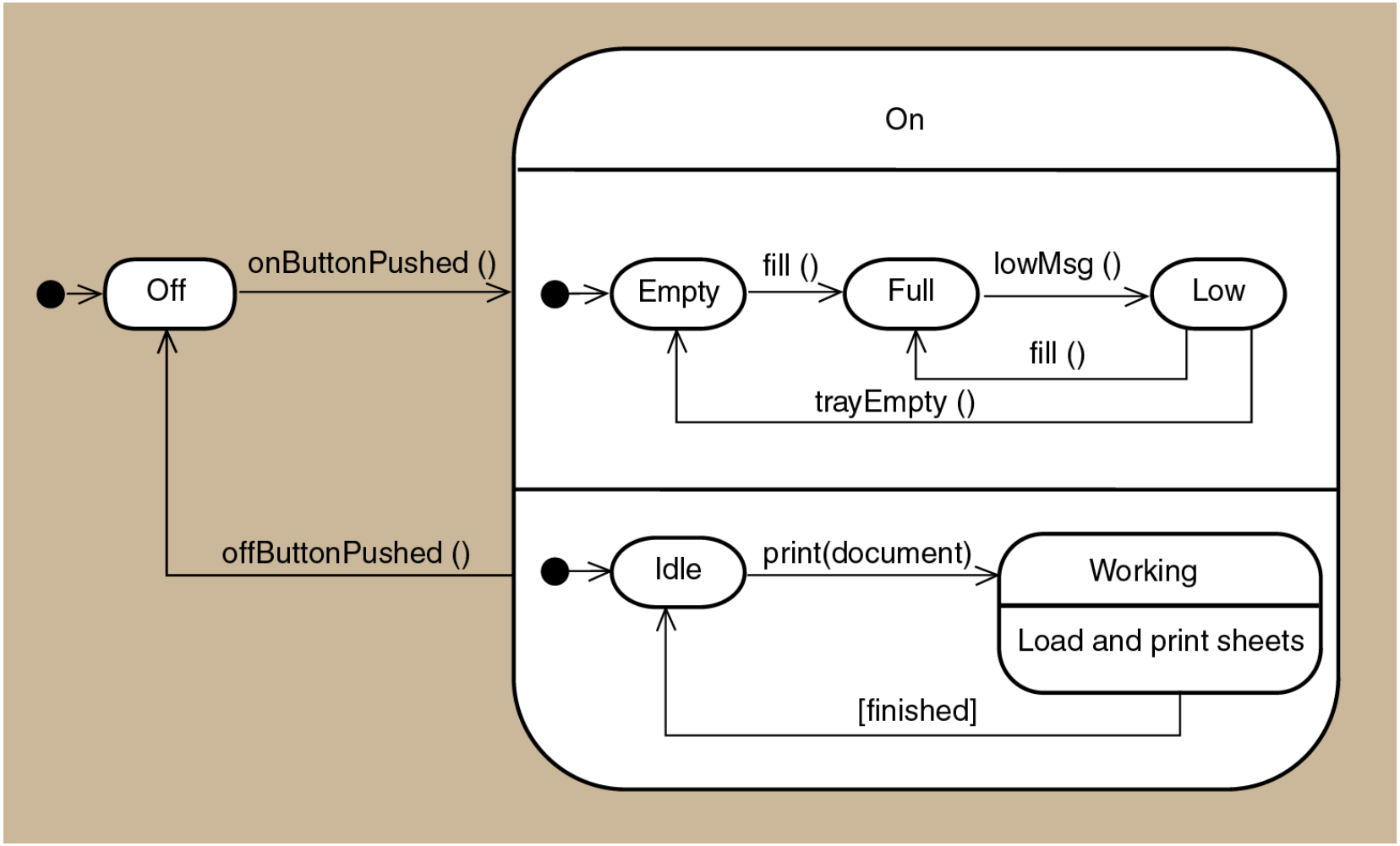
- Check naming convention for status conditions to identify valid states
- Simple states reflect simple conditions such as “On”
- Complex states labeled with gerunds or verb phrases
  - Example: “Being shipped”
- Active states usually not labeled with nouns
- Status conditions reported to management / customers
  - Example: “Shipped”

- Concurrency: condition of being in more than one state at a time
- Two modes of representation
  - Use synchronization bars and concurrent paths
  - Nest low-level states inside higher-level states
- Higher-level states also called composite states
  - Complex structure of sets of states and transitions
  - Represent a higher level of abstraction

Sample Composite States for the Printer Object



Concurrent Paths for the Printer in the On State



Rules for Developing Statecharts

1. Select the classes that will require state charts
2. List all the status conditions for selected classes
3. Specify transitions that cause object to leave the identified state
4. Sequence state-transition combinations in correct order
5. Identify concurrent paths
6. Look for additional transitions
7. Expand each transition as appropriate
8. Review and test each statechart

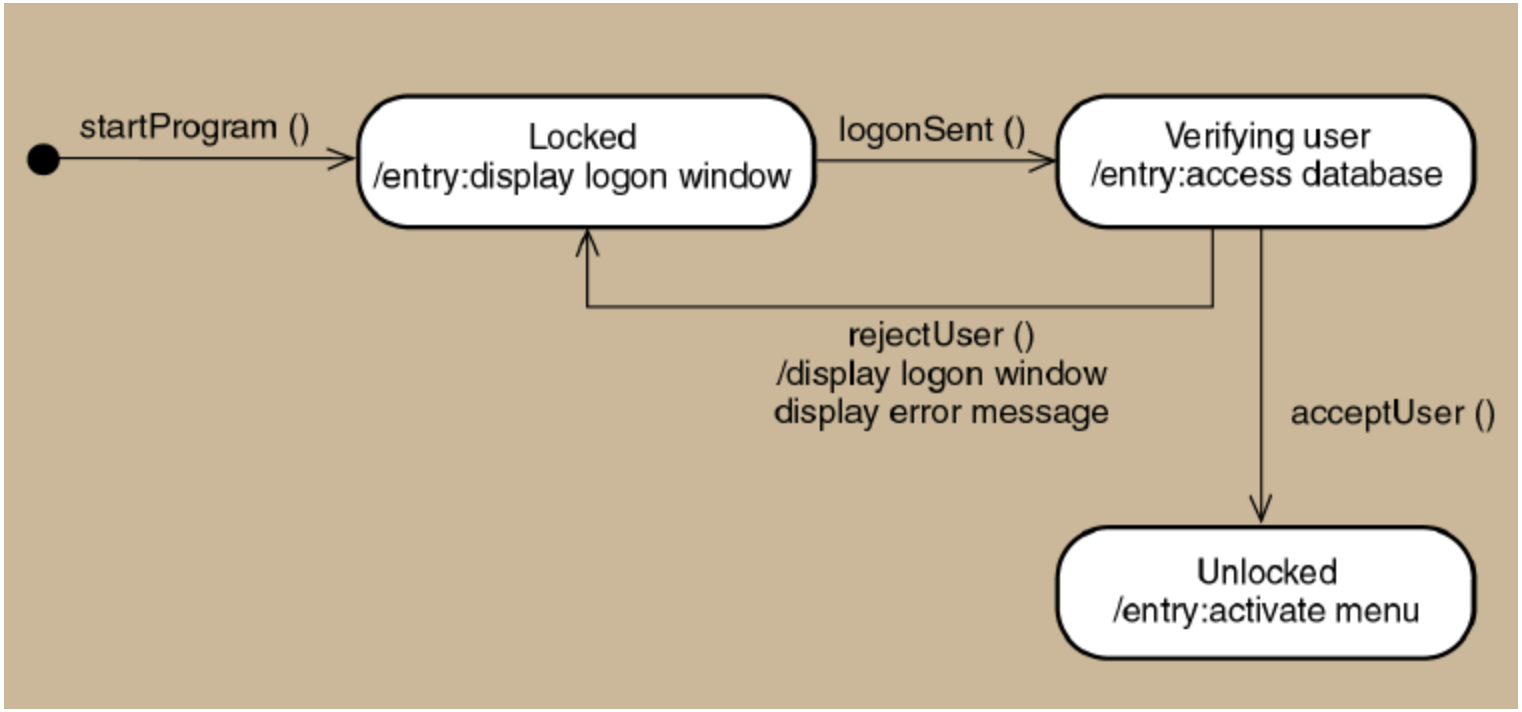
Modeling System Behavior and Method Logic with Statecharts

- State charts were used in the requirements discipline to capture status conditions of objects
- State charts are used in the design discipline to
  - Define the behavioral constraints of the system classes created during design
    - Windows menus, authentication and security classes, error-handling classes
- Describe the method logic within class methods
  - Action-expressions contain information that help define method logic

System-level Statecharts

- Designate the states and processing controls of a system
  - Which menu items are enabled or disabled
  - Which windows are modal
  - Which tool bars and icons are active
- Document possible options presented to users
- Use state variables to record the value of the current state

Logon statechart for a computer system



More requirements for flexi taxi

- Driver can update the information on current status of the ride to Flexi Taxi, such as ‘Taxi is on the way to pick up customer’, ‘Customer Picked up’, ‘On the way to destination’, ’Arrived at destination, and ‘Journey End’. This is done through a mobile device attached to the taxi
- How can you model these requirements? How will you incorporate this information to your design?