SOFTWARE ENGINEERING
(15B11CI513)

Credits :- 4 **Contact Hours :-** 3-1-0

Activity, State, Deployment, & Package Diagrams

More UML Diagrams

- Activity diagrams
- Deployment diagrams
- Package diagrams
- State diagrams

Activity Diagrams - Topics

- What is an Activity?
- Activity Diagrams – Notation
- Activity Diagrams – 5 Examples

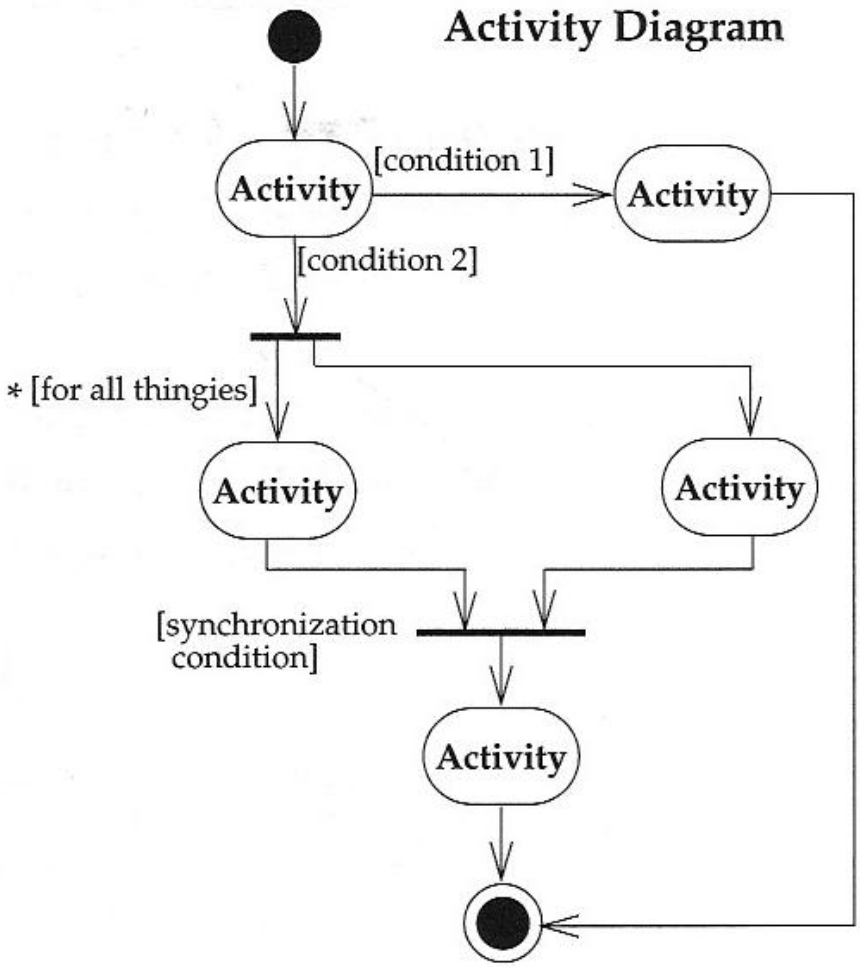
What is an Activity?

- Two definitions
  - In a conceptual diagram, an activity is some task that needs to be done, whether by a human or a computer
  - In a specification-perspective diagram or an implementation-perspective diagram, an activity is a method on a class
- Activity arrangement
  - Sequential – one activity is followed by another
  - Parallel – two or more sets of activities are performed concurrently, and order is irrelevant
    - Interleaving is permitted – we can jump between the parallel flows

What is an Activity? (concluded)

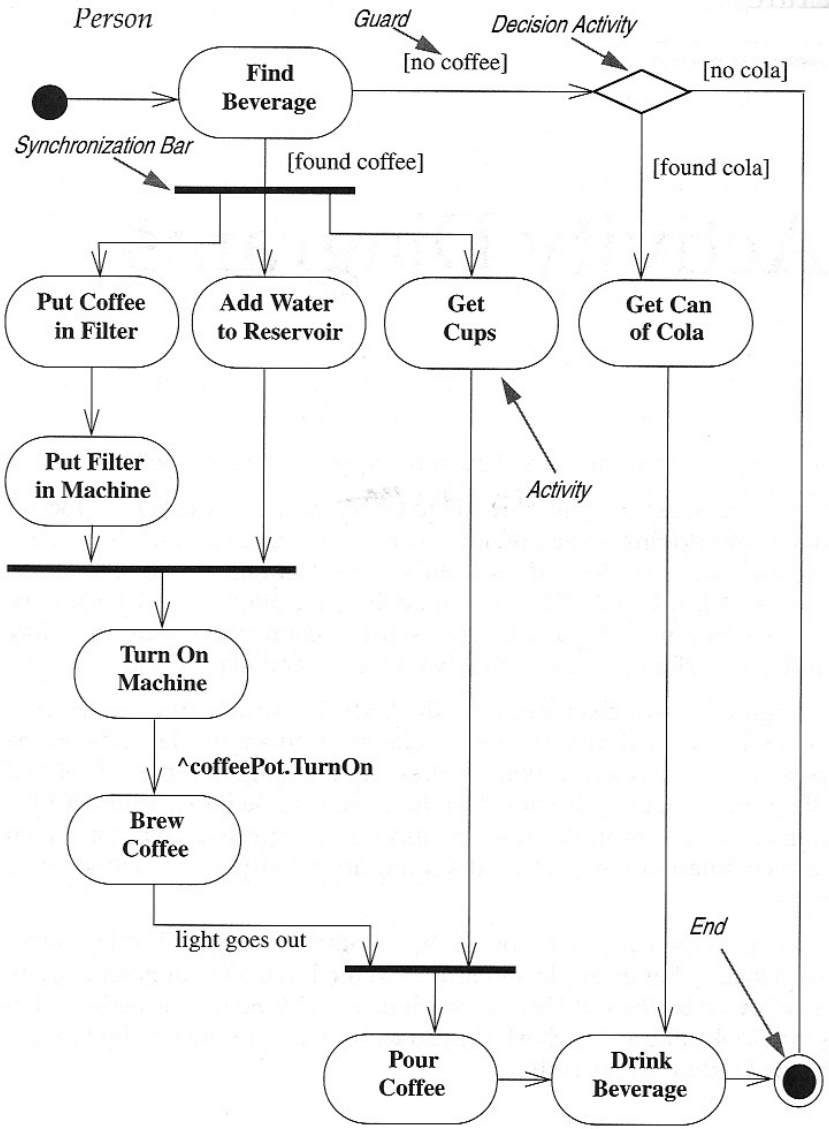
- Activity Diagrams are used to describe activities
  - Activity Diagrams are useful for describing complicated methods
  - Activity Diagrams are useful for describing use cases, since, after all, a use case is an interaction, which is a form of activity
- Using Activity Diagrams with Use Cases
  - Start with a coarse-grained use case, which is composed of subordinate use cases
  - For the complicated subordinate use cases, use Activity Diagrams rather than Use Case Diagrams
- Activity Diagrams are like Flow Charts, but Flow Charts are usually limited to sequential activities while Activity Diagrams can show parallel activities as well

Activity Diagrams - Notation



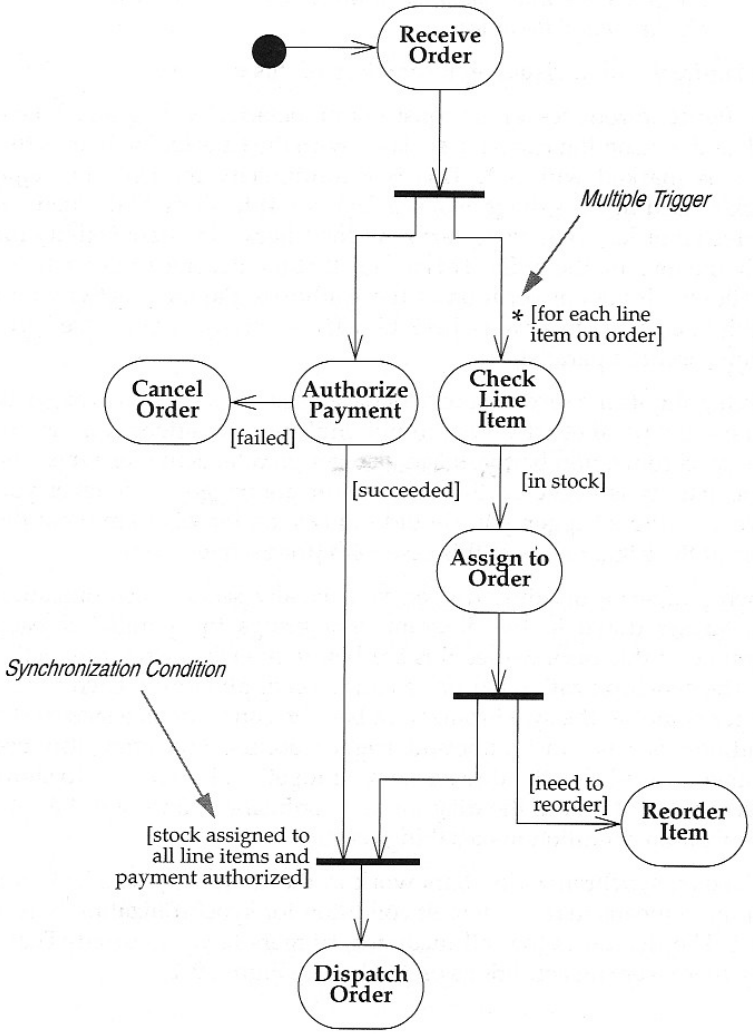
- Start at the top black circle
- If condition 1 is TRUE, go right; if condition 2 is TRUE, go down
- At first bar (a synchronization bar), break apart to follow 2 parallel paths
- At second bar, come together to proceed only when both parallel activities are done

Activity Diagrams – Notation (concluded)



- Activity – an oval
- Trigger – path exiting an activity
- Guard – each trigger has a guard, a logical expression that evaluates to "true" or "false"
- Synchronization Bar – can break a trigger into multiple triggers operating in parallel or can join multiple triggers into one when all are complete
- Decision Diamond – used to describe nested decisions (the first decision is indicated by an activity with multiple triggers coming out of it)

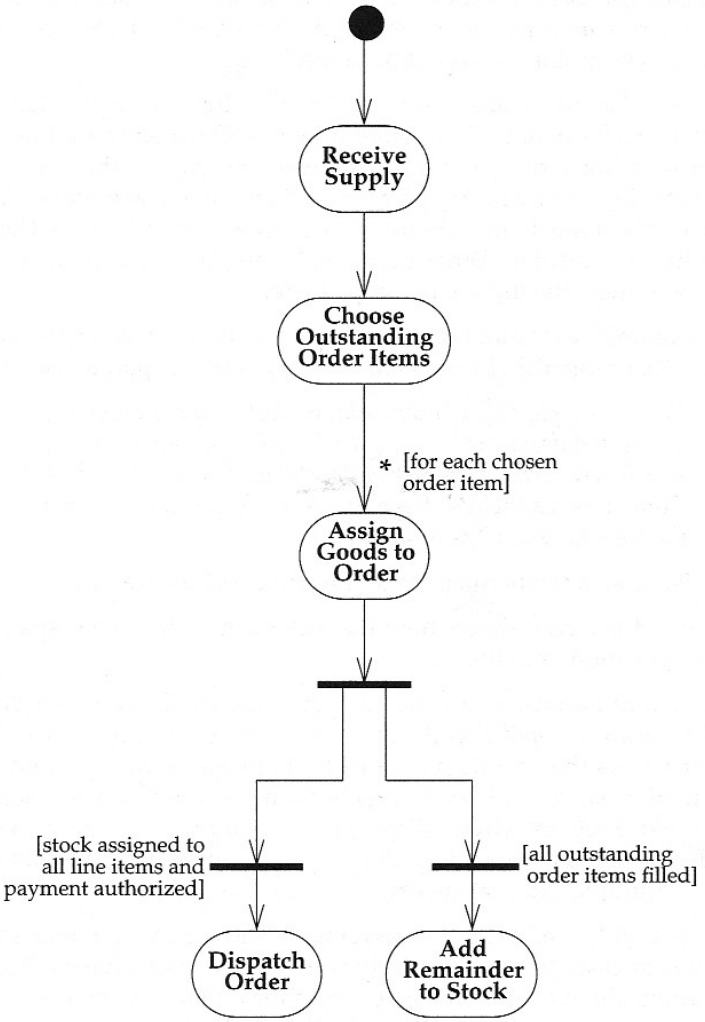
Activity Diagrams – Example 1 of 5



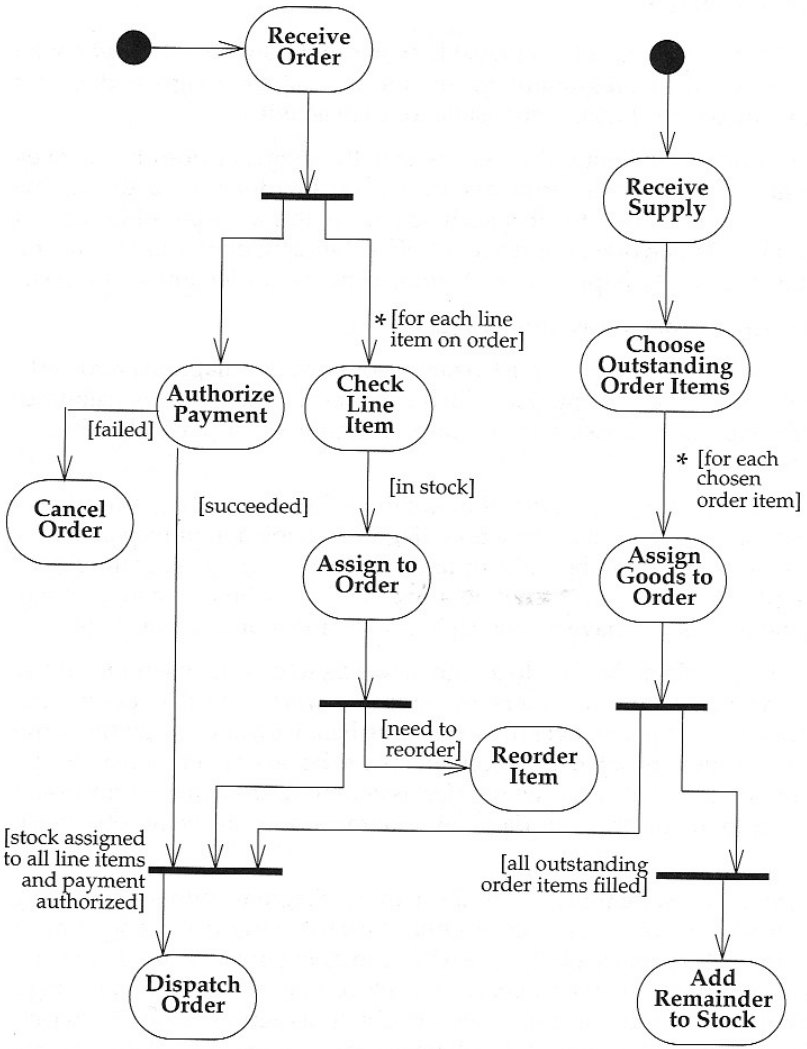
Use Case: Receiving an Order

Activity Diagram – Example 2 of 5

- Use Case: Receiving a Supply

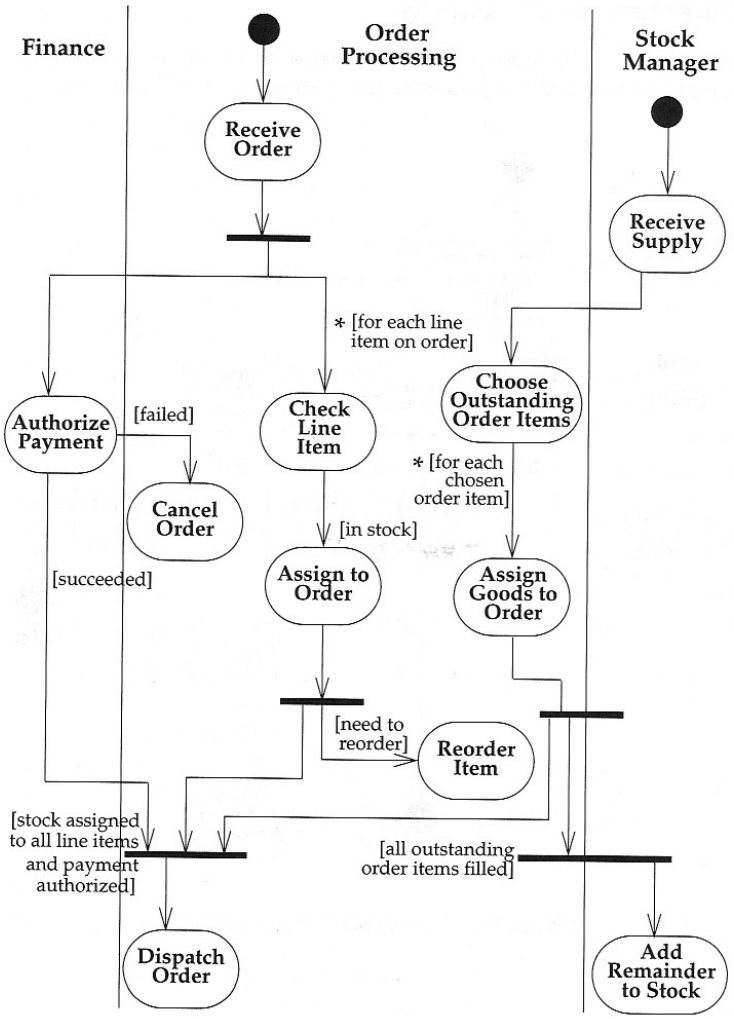


Activity Diagram – Example 3 of 5



Use Case: Receiving an Order and Receiving a Supply

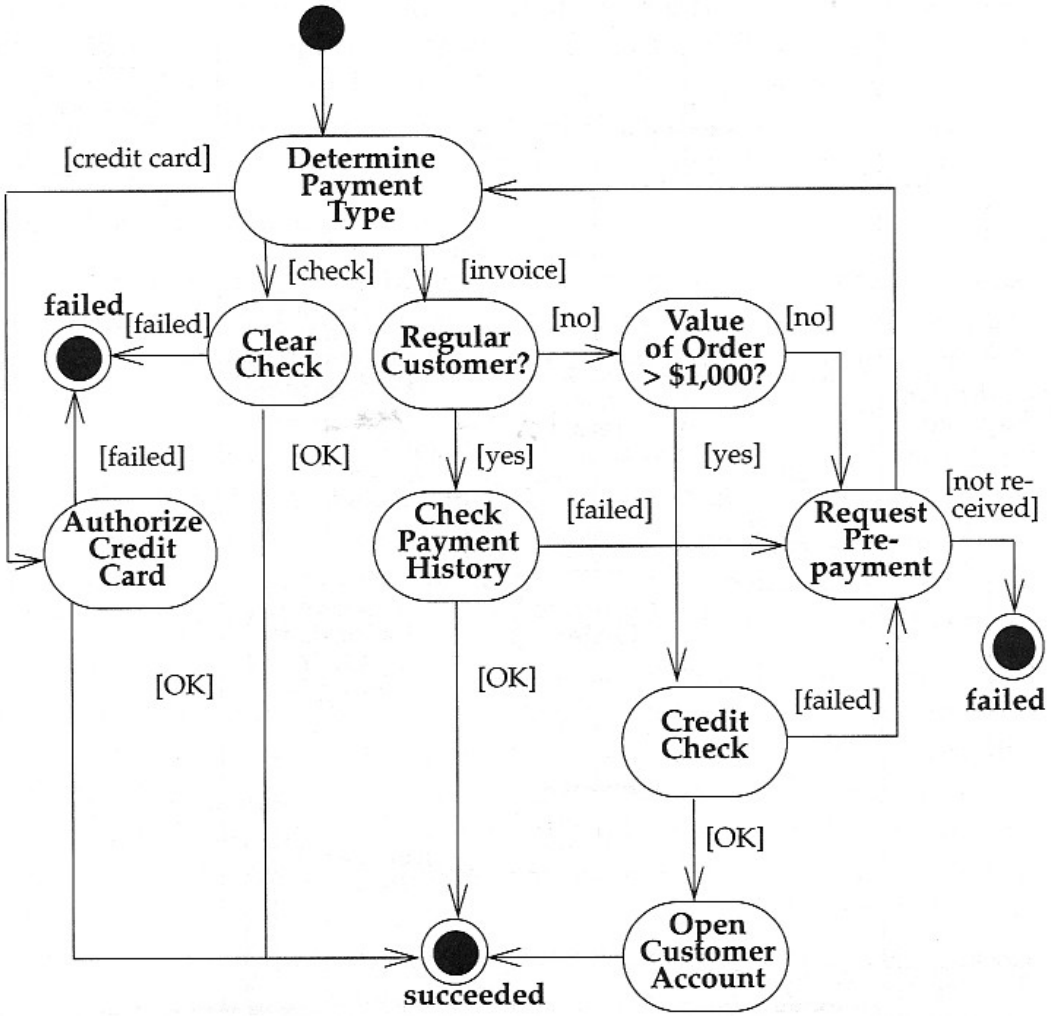
Activity Diagram – Example 4 of 5



Swimlanes - Activity Diagrams that show activities by class

- Arrange activity diagrams into vertical zones separated by lines
- Each zone represents the responsibilities of a particular class (in this example, a particular department)

Activity Diagram – Example 5 of 5



Decomposing an Activity

- An activity can be decomposed into a further Activity Diagram
- When an Activity Diagram represents a decomposition of a higher-level activity, there can be only one start point

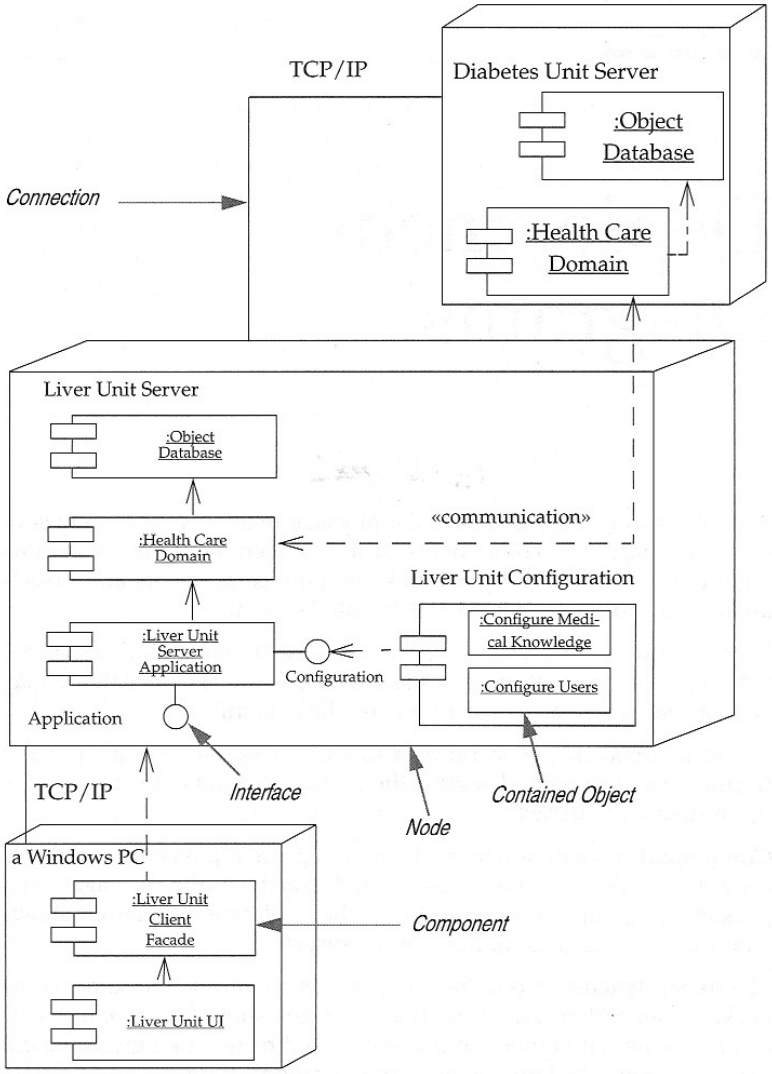
Deployment Diagrams - Topics

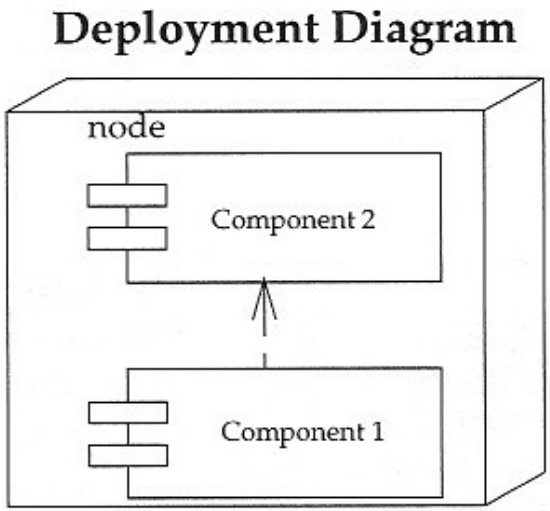
- What is a Deployment Diagram?
- Deployment Diagrams – Notation and Example

What is a Deployment Diagram?

- Deployment Diagram – a diagram that shows the physical relationships among software and hardware components in a system
  - Components – physical modules of code
  - Connections – show communication paths
  - Dependencies – show how components communicate with other components
  - Nodes – computational units, usually a pieces of hardware

Deployment Diagrams – Notation and Example





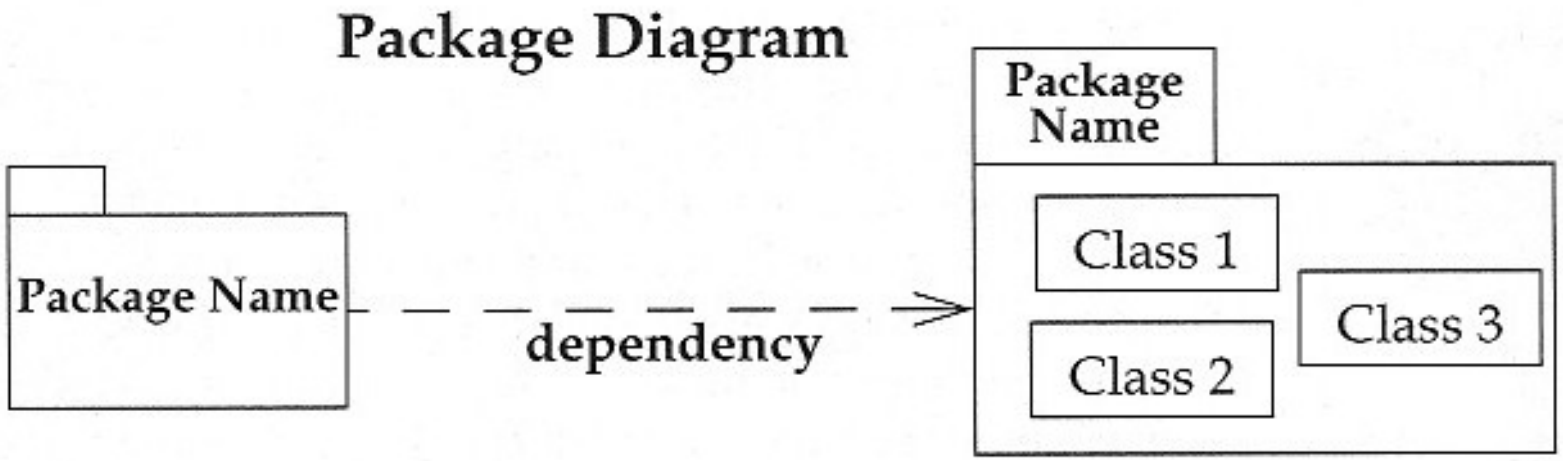
Package Diagrams - Topics

- What is a Package?
- Package Diagrams – Notation
- Package Diagrams – 2 Examples

What is a Package?

- Package – a grouping of classes (a conventional Package - a unit above a class in the abstraction hierarchy) and other packages (a Domain Package)
- Package Diagram – a UML diagram that shows packages of classes and the dependencies among them
  - A dependency exists between two elements if changes to the definition of one element may cause changes to the other
- Classes have dependencies for several reasons, including:
  - One class sends a message to another
  - One class has another as part of its data
  - One class mentions another as a parameter to an operation

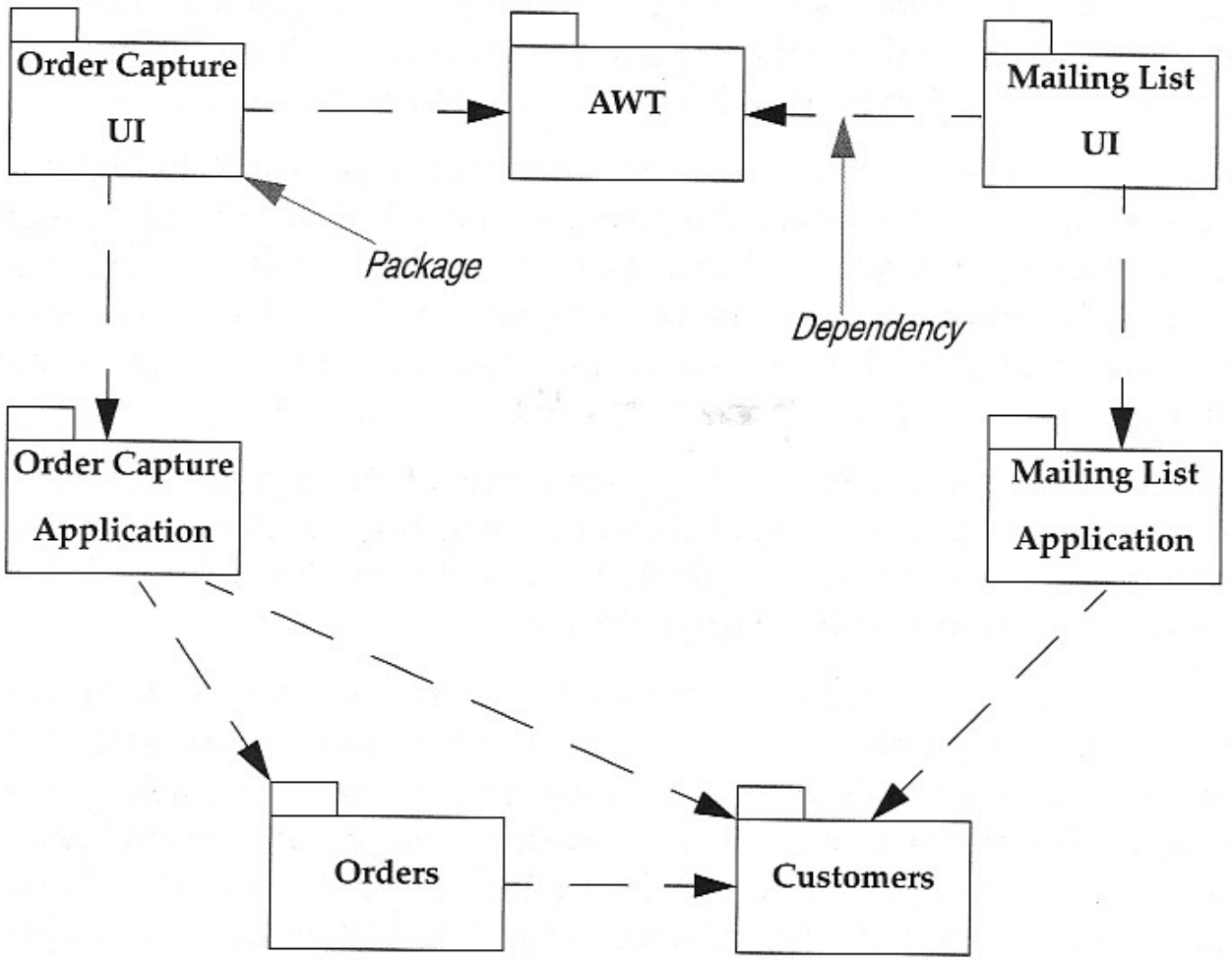
Package Diagrams - Notation



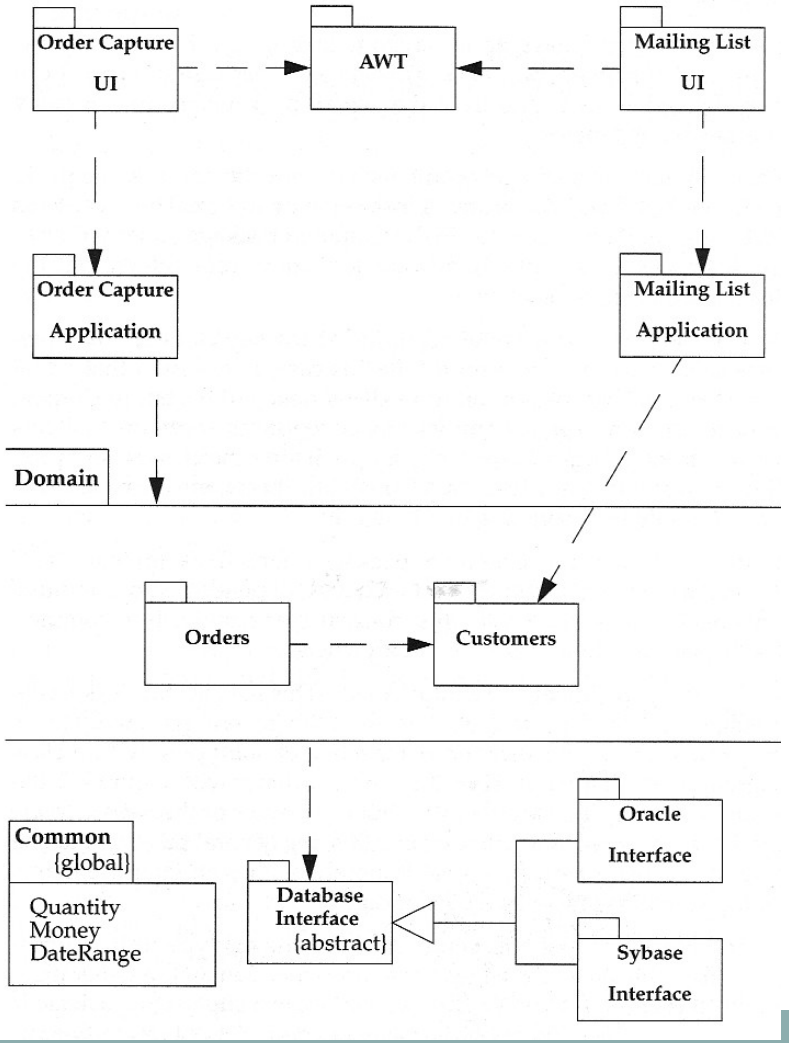
- Package – contains classes
- Dependency – changes to the definition (interface) of one package may cause changes in the other package

Reference: *UML Distilled*, Inside Cover

Package Diagrams – Example 1 of 2



Package Diagrams – Example 2 of 2



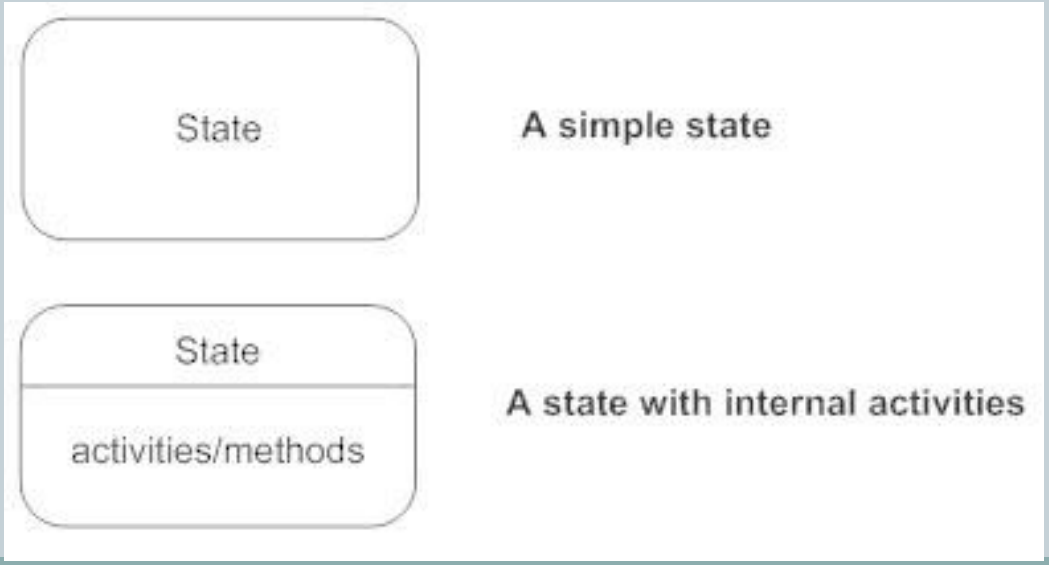
- Domain Package – a collection of related packages

State Diagrams - Topics

- What are State Diagrams?
- State Diagrams – Notation
- State Diagrams – 3 Examples

What is a state?

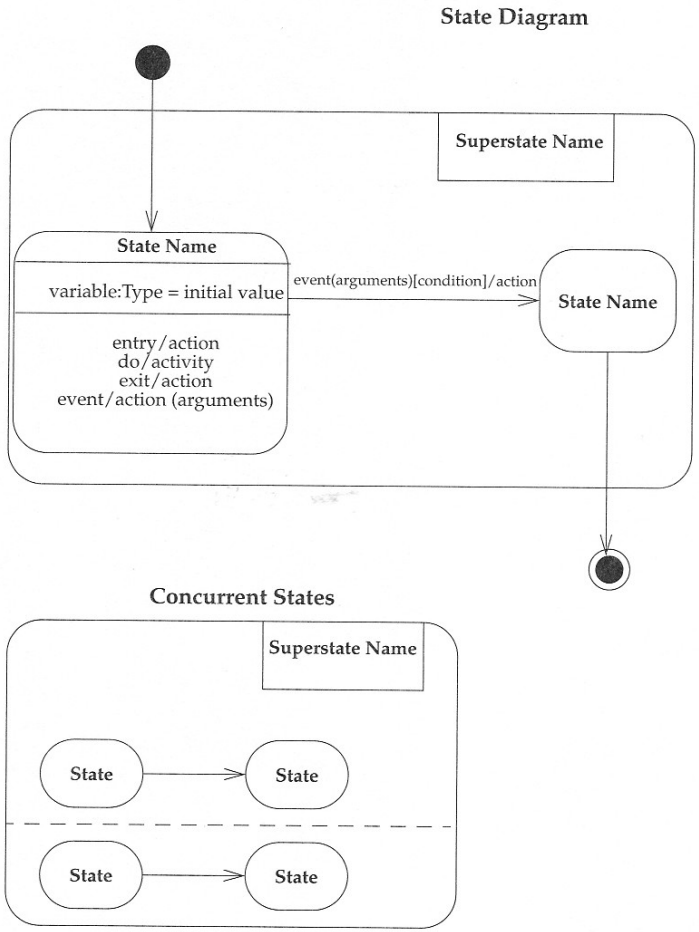
- *"A state is an abstraction of the attribute values and links of an object. Sets of values are grouped together into a state according to properties that affect the gross behavior of the object."*



What are State Diagrams?

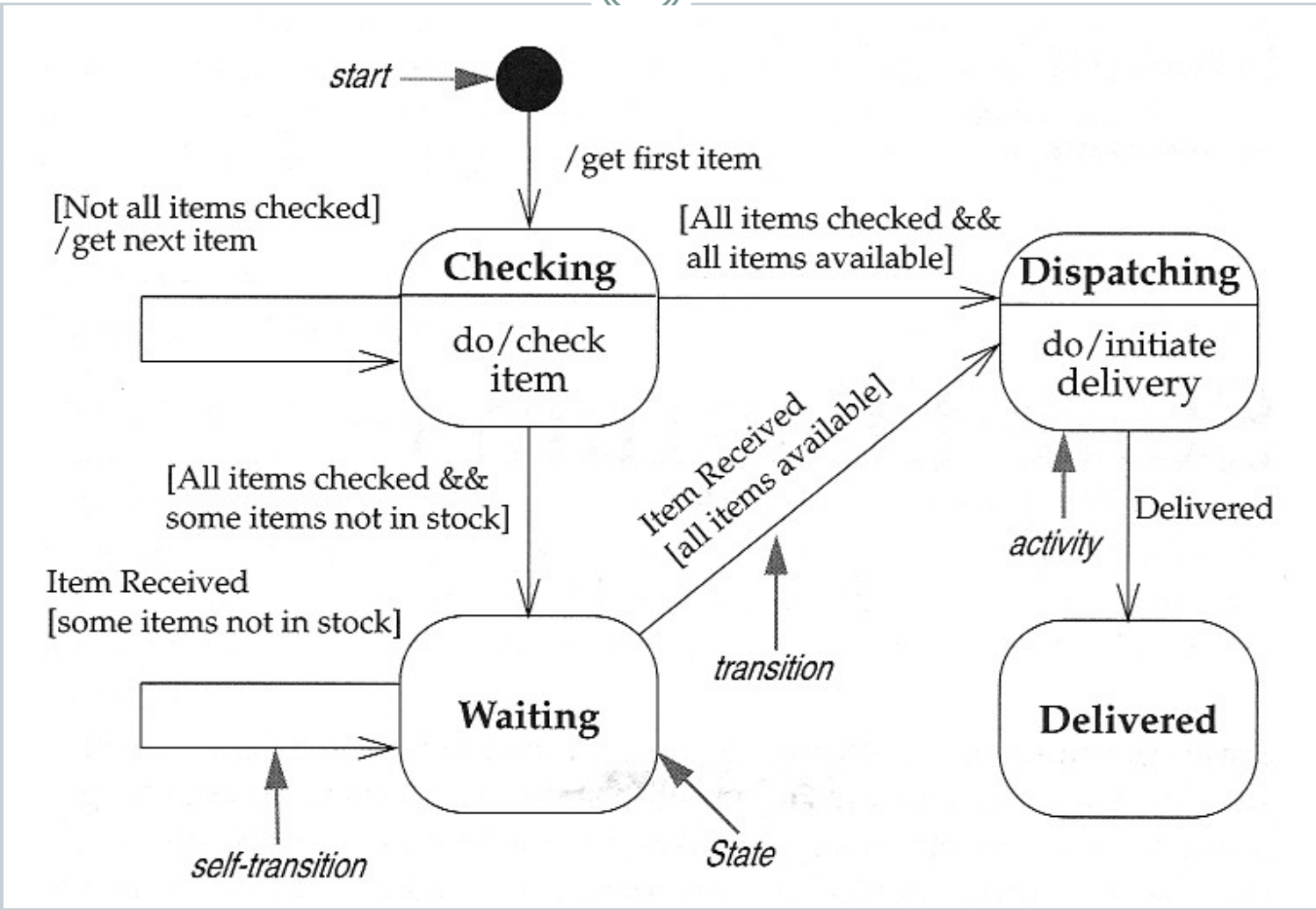
- State Diagrams – describe all the possible states an object can assume and how the object's state changes as a result of events that affect the object
  - State Diagrams are drawn for a single class to show the lifetime behavior of a single object
  - State Diagrams are good for describing the behavior of an object across several use cases
- In UML, State Diagrams
  - Support superstates (states which contain other states)
  - Support concurrency

State Diagrams - Notation

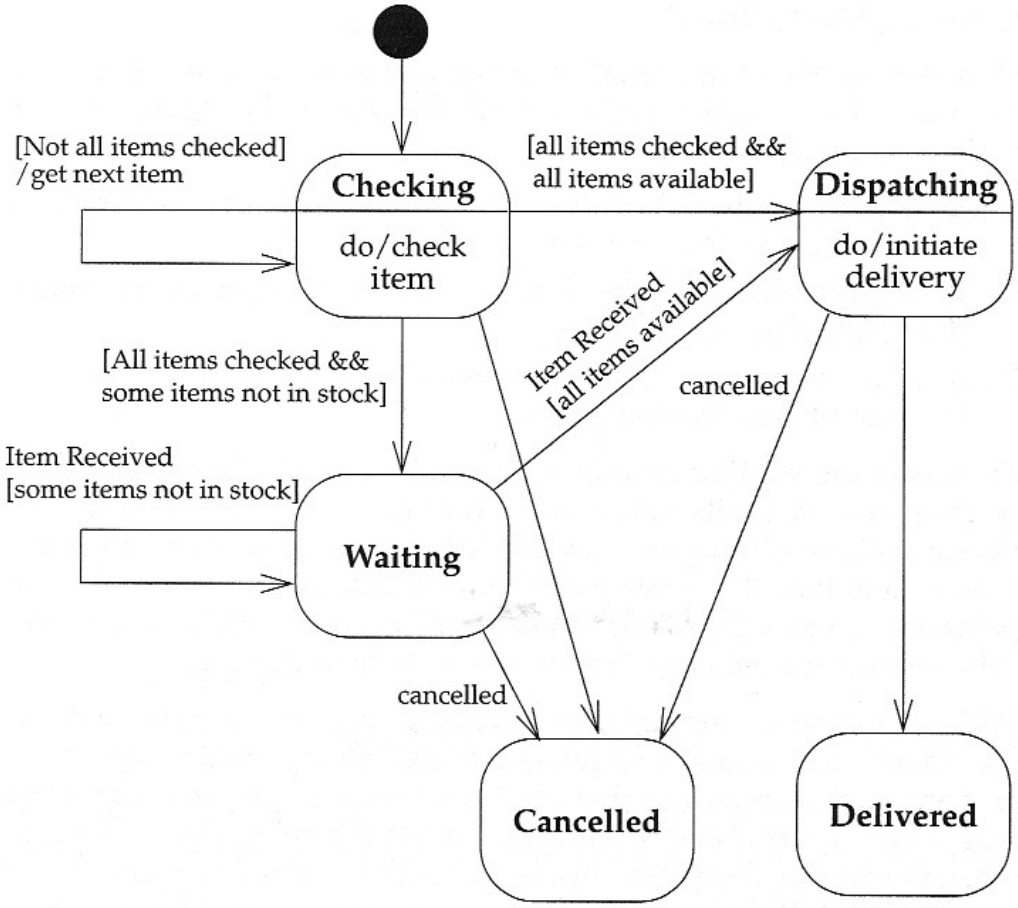


- Action – processes associated with transitions that occur quickly and are not interruptible
- Activity – processes associated with states that may take a while and my be interrupted by events
- Event – a stimulus that causes a transition or a self-transition to take place from one state to another
- Guard – a logical condition that returns “true” or “false”
- Superstate – a state that is itself a collection of states

State Diagrams – Notation (concluded)

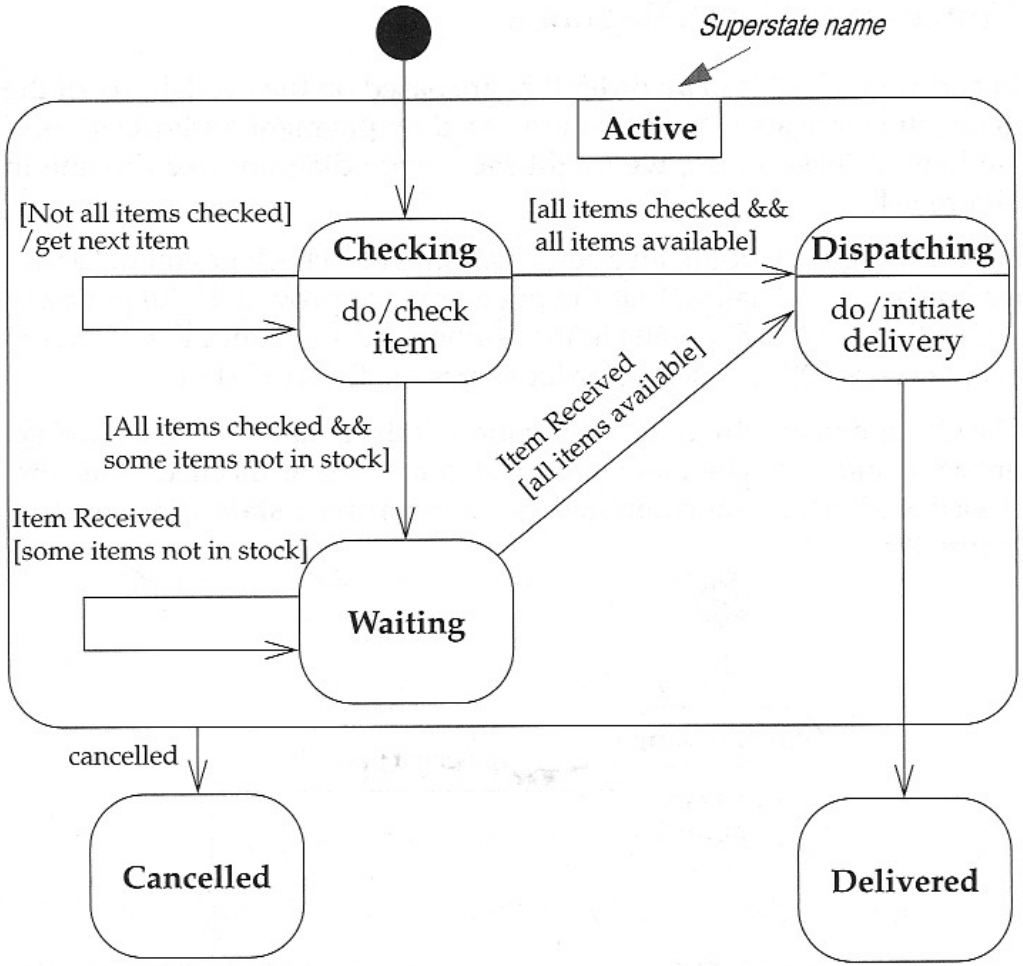


State Diagrams – Example 1 of 3



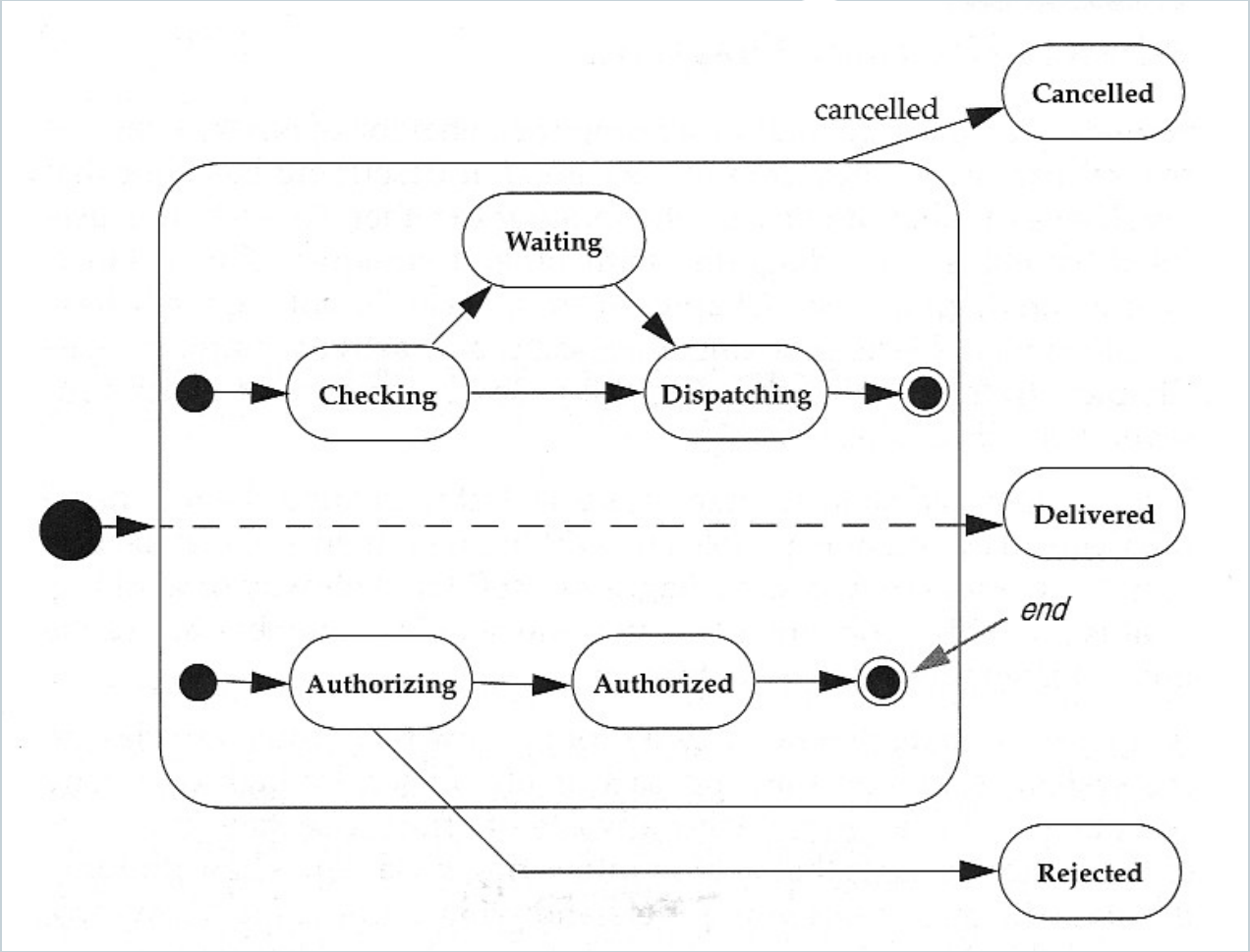
State Diagram without Superstates

State Diagrams – Example 2 of 3



State Diagram with a Superstate

State Diagrams – Example 3 of 3



Concurrent State Diagram