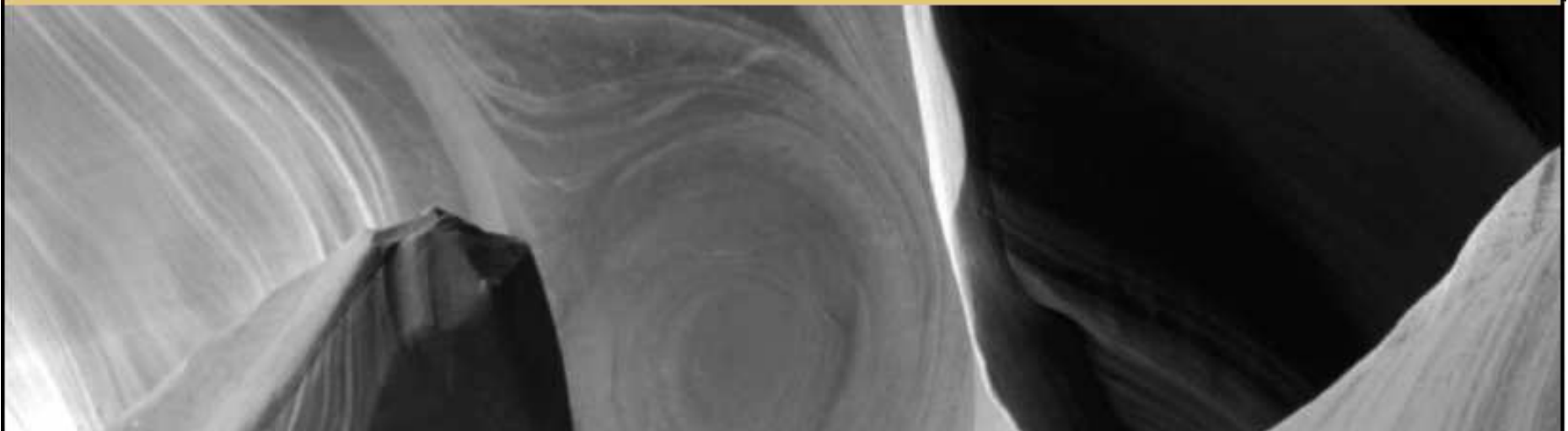


# *Systems Analysis and Design in a Changing World, Fourth Edition*

CHAPTER

7

THE OBJECT-ORIENTED  
APPROACH TO  
REQUIREMENTS



# Learning Objectives

- ◆ Develop use case diagrams
- ◆ Write use case and scenario descriptions
- ◆ Develop activity diagrams and system sequence diagrams
- ◆ Develop state machine diagrams to model object behavior
- ◆ Explain how UML diagrams work together to define functional requirements for the object-oriented approach

# Overview

- ◆ Objective of requirements definition is understanding users' needs, business processes, and system to support business processes
- ◆ Understand and define requirements for a new system using object-oriented analysis models and techniques
- ◆ Line between object-oriented analysis and object-oriented design is somewhat fuzzy
  - Iterative approach to development
  - Models built in analysis are refined during design

# Object-Oriented Requirements

- ◆ Object-oriented modeling notation is Unified Modeling Language (UML 2.0)
- ◆ UML was accepted by Object Management Group (OMG) as standard modeling technique
- ◆ Purpose of Object Management Group
  - Promote theory and practice of object-oriented technology for development of distributed systems
  - Provide common architectural framework for OO

## Object-Oriented Requirements (continued)

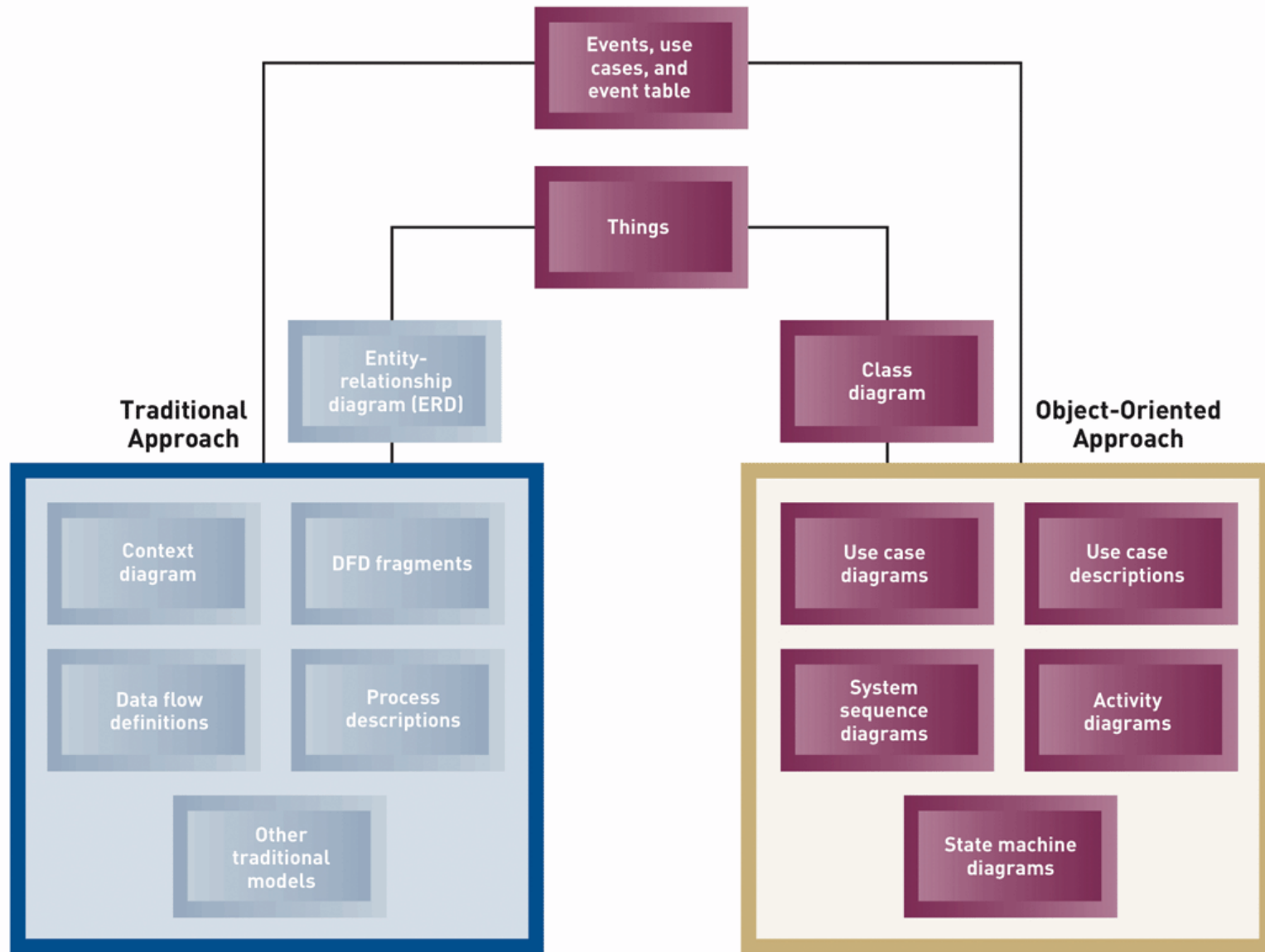
- ◆ Object-oriented system requirements are specified and documented through process of building models
- ◆ Modeling process starts with identification of use cases and problem domain classes (things in users' work environment)
- ◆ Business events trigger elementary business processes (EBP) that new system must address as use cases
- ◆ Use cases define functional requirements

# Object-Oriented Requirements Models

- ◆ **Use case diagrams** – identify actors and their use cases (goals)
- ◆ **Use case descriptions** – include details of a use case and how actors use the system
- ◆ **Systems sequence diagrams (SSDs)** – define inputs and outputs and sequence of interactions between user and system for a use case
- ◆ **Activity diagrams** – describe user and system activities for a use case
- ◆ **State machine diagrams** – describe states of each object

# Requirements Models—Traditional versus OO

(Figure 7-1)



# The System Activities— A Use Case/Scenario View

- ◆ Use case analysis used to identify and define all business processes that system must support
- ◆ **Use case** – an activity a system carried out, usually in response to a user request
- ◆ **Actor**
  - Role played by user
  - Outside automation boundary



# Techniques for Identifying Use Cases

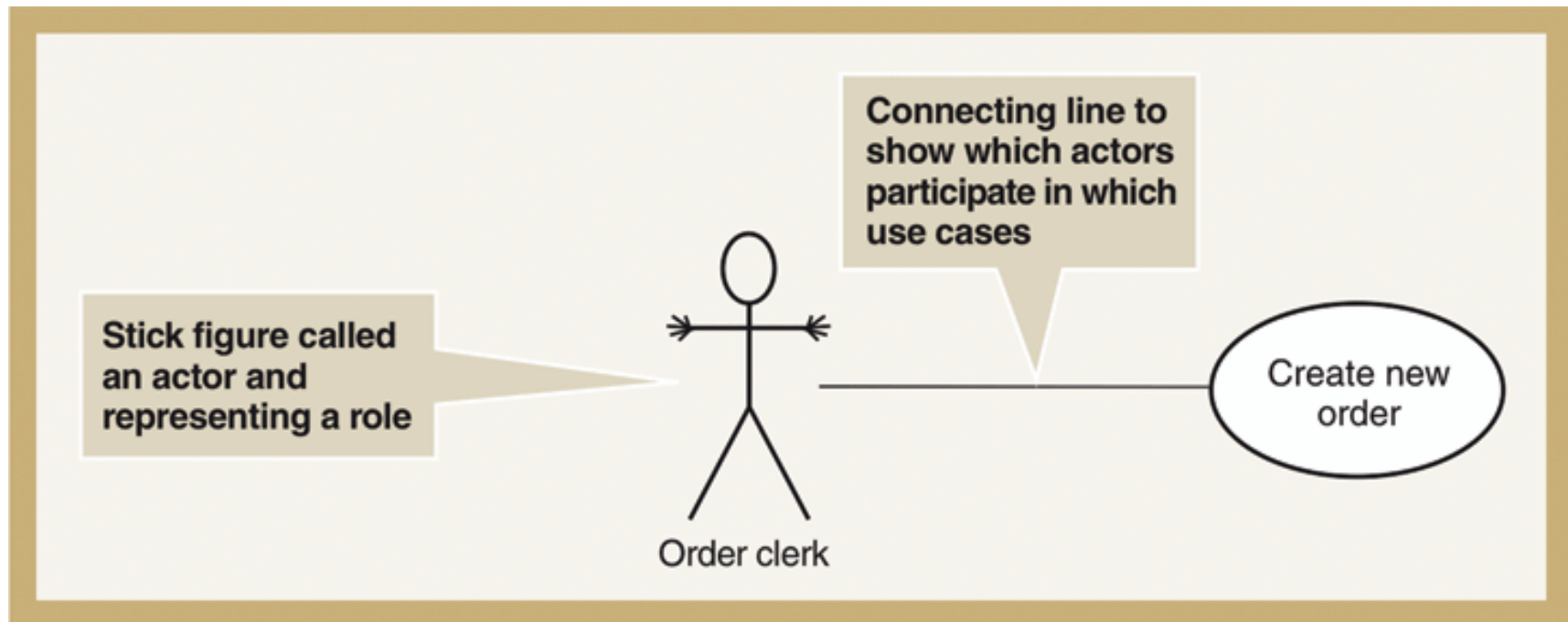
(Review from Chapter 5)

- ◆ Identify user goals
  - Each goal at the elementary business process (EBP) level is a use case
  - EBP – task performed by one user in one place and in response to business event that adds measurable business value, and leaves system and data in consistent state
- ◆ Event decomposition technique (event table)
- ◆ CRUD analysis technique (create, read/report, update, delete) to ensure coverage

# Use Case Diagram

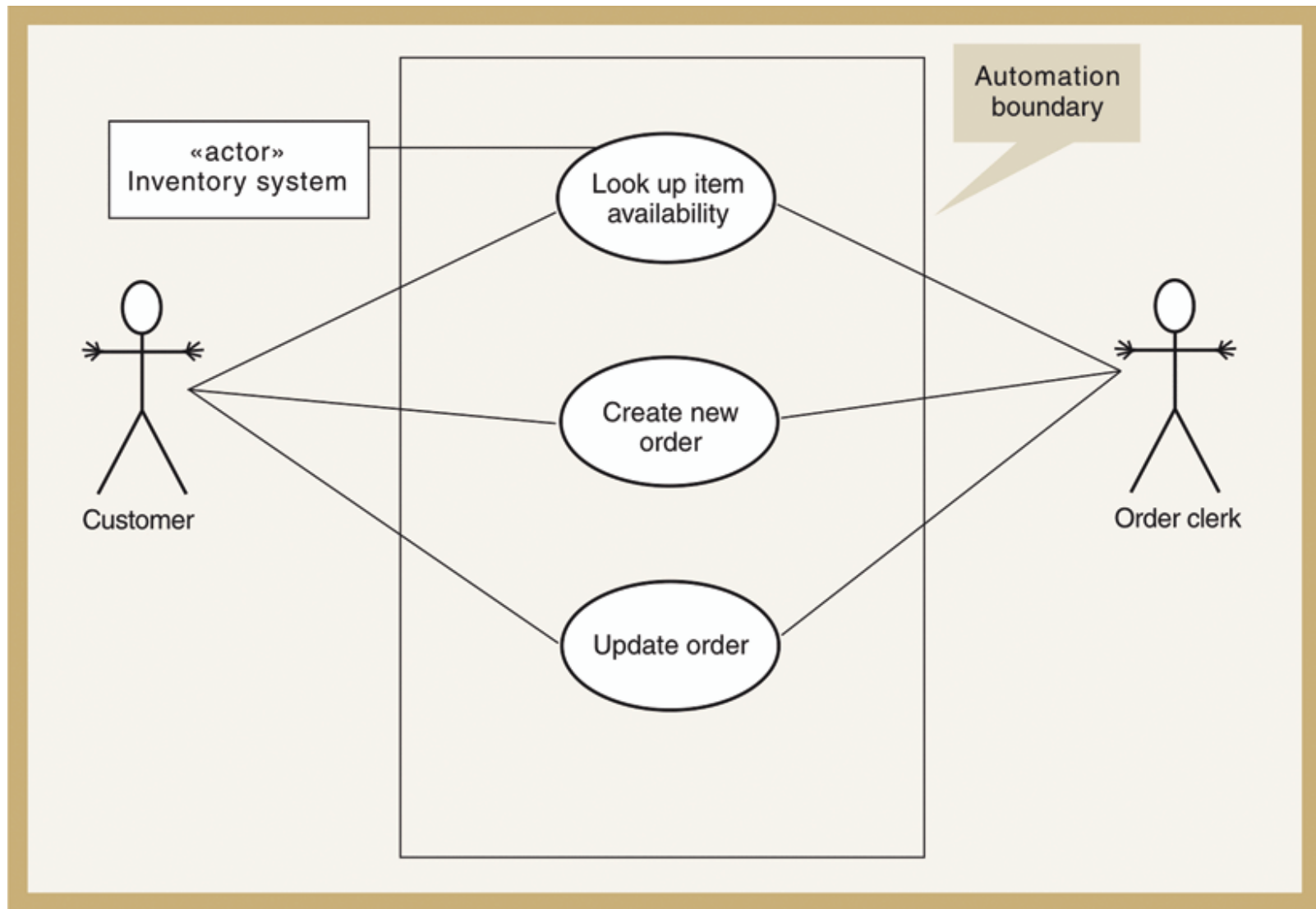
- ◆ Graphical UML diagram that summarizes information about actors and use cases
- ◆ Simple diagram shows overview of functional requirements
- ◆ Can have multiple use case diagrams
  - By subsystem
  - By actor

# Simple Use Case with an Actor (Figure 7-2)

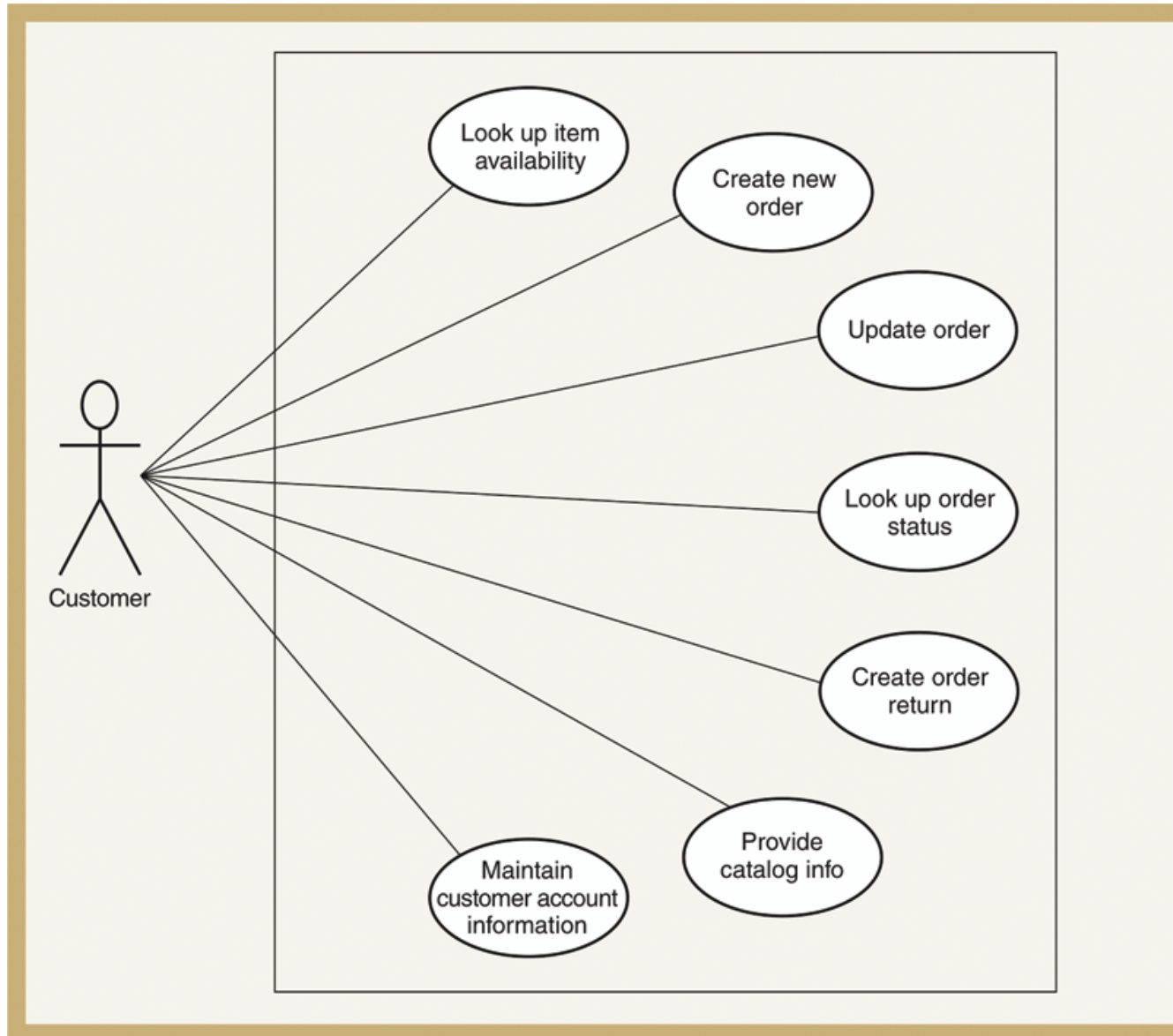


# Use Case Diagram with Automation Boundary and Alternate Actor Notation

(Figure 7-3)

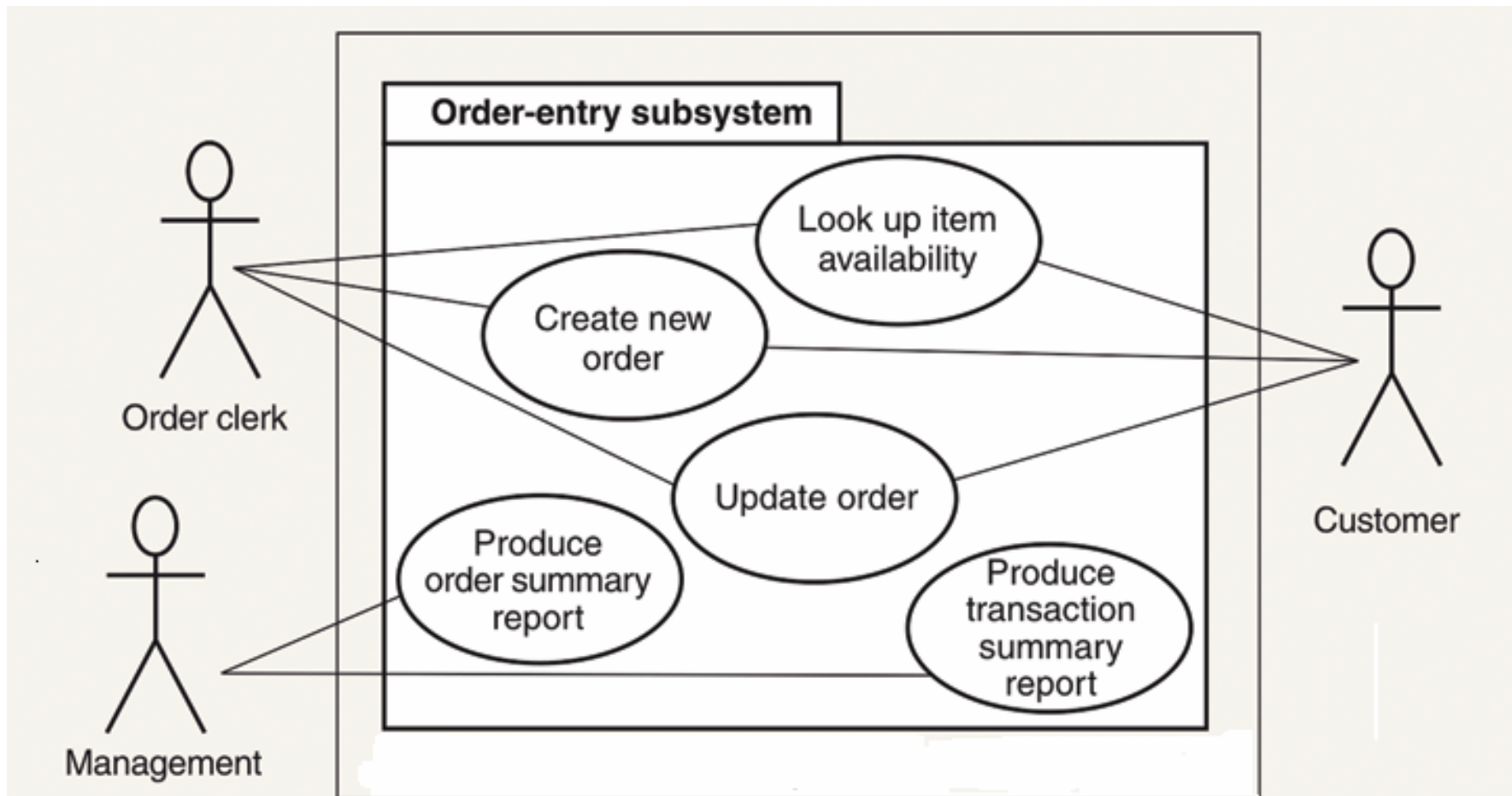


# All Use Cases Involving Customer as Actor (Figure 7-4)



# Use Cases of RMO Order Entry Subsystem

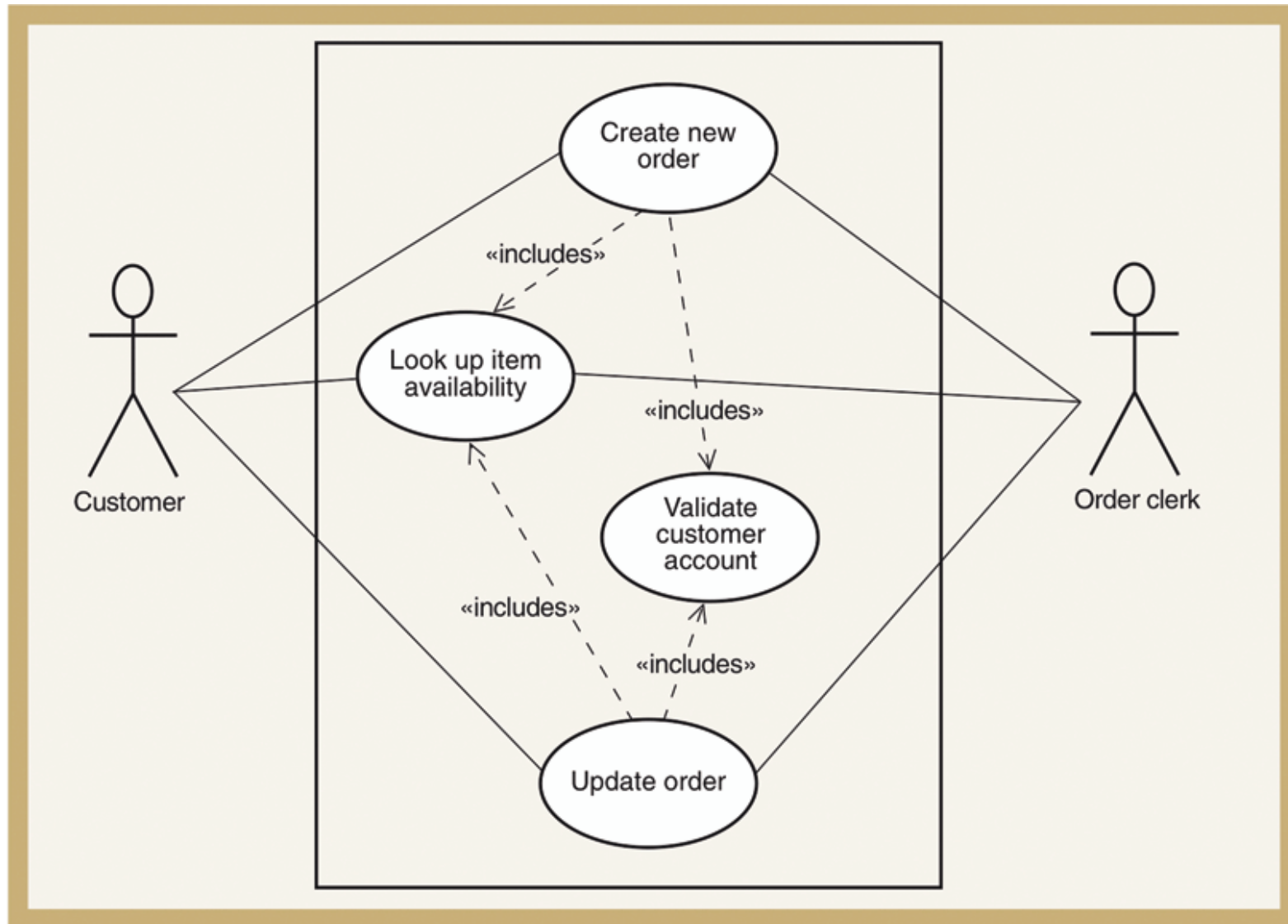
(Partial Figure 7-5 with package symbol)



## <<Includes>> Relationship

- ◆ Documents situation in which one use case requires the services of a common subroutine
- ◆ Another use case is developed for this common subroutine
- ◆ A common use case can be reused by multiple use cases

## Example of Order-Entry Subsystem with <<Includes>> Use Cases (Figure 7-6)





# CRUD Analysis for Identifying/Confirming Use Cases

- ◆ **CRUD** – create, read/report, update, delete
- ◆ Information Engineering (IE) technique to identify event table or directly develop use case diagram
- ◆ Compares identified use cases with domain model class diagram
- ◆ Every class in class diagram must have use cases to support creating, reading, reporting, updating, and deleting object instances
- ◆ Confirms system integration requirements

# Use Case Description

- ◆ **Use case description** provides details of preconditions, postconditions, sequence of activities, and exception conditions in use case
- ◆ Describes **actor** interacting with computer system step-by-step to carry out business activity
- ◆ May have several **scenarios** for a use case, each a specific **use case instance**
- ◆ Three levels of detail: brief, intermediate, and fully developed description
- ◆ Many analysts prefer to write narrative descriptions of use cases instead of drawing activity diagrams

# Brief Description of *Create New Order* Use Case (Figure 7-7)

## ***Create new order description***

When the customer calls to order, the order clerk and system verify customer information, create a new order, add items to the order, verify payment, create the order transaction, and finalize the order.

## Intermediate Description of the Telephone Order Scenario for *Create New Order* Use Case (Figure 7-8)

### Flow of activities for scenario of *Order Clerk creates telephone order*

#### Main Flow:

1. Customer calls RMO and gets order clerk.
2. Order clerk verifies customer information. If a new customer, invoke *Maintain customer account information* use case to add a new customer.
3. Clerk initiates the creation of a new order.
4. Customer requests an item be added to the order.
5. Clerk verifies the item and adds it to the order.
6. Repeat steps 4 and 5 until all items are added to the order.
7. Customer indicates end of order; clerk enters end of order; system computes totals.
8. Customer submits payment; clerk enters amount; system verifies payment.
9. System finalizes order.

#### Exception Conditions:

1. If an item is not in stock, then customer can
  - a. choose not to purchase item, or
  - b. request item be added as a back-ordered item.
2. If customer payment is rejected due to bad-credit verification, then
  - a. order is canceled, or
  - b. order is put on hold until check is received.

# Intermediate Description of the Web Order Scenario for *Create New Order* (Figure 7-9)

## Flow of activities for scenario of *Customer creates Web order*

### Main Flow:

1. Customer connects to the RMO home page and then links to the order page.
2. If this is a new customer, customer links to the customer account page and adds the appropriate information to establish a customer account.
  - 2a If existing customer, customer logs on.
3. The system starts a new order and displays the catalog frame.
4. Customer searches the catalog.
5. When customer finds the correct item, he/she requests it be added to the order; the system adds it to the shopping cart.
6. Customer repeats steps 4 and 5.
7. Customer requests end of order; system displays a summary of the ordered items.
8. Customer makes any changes.
9. Customer requests payment screen; system displays payment screen.
  - 9a Customer enters payment information; system displays summary information and sends confirmation e-mail.
10. System finalizes order.

### Exception Conditions:

1. If existing customer forgets password, then
  - a. customer can invoke forgotten password processing, or
  - b. customer can create a new customer account.
2. If customer payment is rejected due to bad-credit verification, then
  - a. customer can cancel the order, or
  - b. order is put on hold until check is received.



# Fully Developed Description of Telephone Order Scenario for *Create New Order Use Case*

(Figure 7-10)

<b>Use Case Name:</b>	<i>Create new order</i>	
<b>Scenario:</b>	Create new telephone order	
<b>Triggering Event:</b>	Customer telephones RMO to purchase items from the catalog.	
<b>Brief Description:</b>	When customer calls to order, the order clerk and system verify customer information, create a new order, add items to the order, verify payment, create the order transaction, and finalize the order.	
<b>Actors:</b>	Telephone sales clerk.	
<b>Related Use Cases:</b>	Includes: <i>Check item availability</i> .	
<b>Stakeholders:</b>	Sales department: to provide primary definition. Shipping department: to verify information content is adequate for fulfillment. Marketing department: to collect customer statistics for studies of buying patterns.	
<b>Preconditions:</b>	Customer must exist. Catalog, Products, and Inventory items must exist for requested items.	
<b>Postconditions:</b>	Order and order line items must be created. Order transaction must be created for the order payment. Inventory items must have the quantity on hand updated. The order must be related (associated) to a customer.	
<b>Flow of Activities:</b>	<b>Actor</b>	<b>System</b>
	<ol style="list-style-type: none"> <li>1. Sales clerk answers telephone and connects to a customer.</li> <li>2. Clerk verifies customer information.</li> <li>3. Clerk initiates the creation of a new order.</li> <li>4. Customer requests an item be added to the order.</li> <li>5. Clerk verifies the item (<i>Check item availability</i> use case).</li> <li>6. Clerk adds item to the order.</li> <li>7. Repeat steps 4, 5, and 6 until all items are added to the order.</li> <li>8. Customer indicates end of order; clerk enters end of order.</li> <li>9. Customer submits payment; clerk enters amount.</li> </ol>	<ol style="list-style-type: none"> <li>3.1 Create a new order.</li> <li>5.1 Display item information.</li> <li>6.1 Add create an order item.</li> <li>8.1 Complete order.</li> <li>8.2 Compute totals.</li> <li>9.1 Verify payment.</li> <li>9.2 Create order transaction.</li> <li>9.3 Finalize order.</li> </ol>
<b>Exception Conditions:</b>	<ol style="list-style-type: none"> <li>2.1 If customer does not exist, then the clerk pauses this use case and invokes <i>Maintain customer information</i> use case.</li> <li>2.2 If customer has a credit hold, then clerk transfers the customer to a customer service representative.</li> <li>4.1 If an item is not in stock, then customer can               <ol style="list-style-type: none"> <li>a. choose not to purchase item, or</li> <li>b. request item be added as a back-ordered item.</li> </ol> </li> <li>9.1 If customer payment is rejected due to bad-credit verification, then               <ol style="list-style-type: none"> <li>a. order is canceled, or</li> <li>b. order is put on hold until check is received.</li> </ol> </li> </ol>	

# Top Detail from Fully Developed Use Case Description (Figure 7-10)

<b>Use Case Name:</b>	<i>Create new order</i>
<b>Scenario:</b>	Create new telephone order
<b>Triggering Event:</b>	Customer telephones RMO to purchase items from the catalog.
<b>Brief Description:</b>	When customer calls to order, the order clerk and system verify customer information, create a new order, add items to the order, verify payment, create the order transaction, and finalize the order.
<b>Actors:</b>	Telephone sales clerk.
<b>Related Use Cases:</b>	Includes: <i>Check item availability.</i>
<b>Stakeholders:</b>	Sales department: to provide primary definition. Shipping department: to verify information content is adequate for fulfillment. Marketing department: to collect customer statistics for studies of buying patterns.
<b>Preconditions:</b>	Customer must exist. Catalog, Products, and Inventory items must exist for requested items.
<b>Postconditions:</b>	Order and order line items must be created. Order transaction must be created for the order payment. Inventory items must have the quantity on hand updated. The order must be related (associated) to a customer.

# Middle Detail from Fully Developed Use Case Description (Figure 7-10)

Flow of Activities:	Actor	System
	<ol style="list-style-type: none"> <li>1. Sales clerk answers telephone and connects to a customer.</li> <li>2. Clerk verifies customer information.</li> <li>3. Clerk initiates the creation of a new order.</li> <li>4. Customer requests an item be added to the order.</li> <li>5. Clerk verifies the item (<i>Check item availability use case</i>).</li> <li>6. Clerk adds item to the order.</li> <li>7. Repeat steps 4, 5, and 6 until all items are added to the order.</li> <li>8. Customer indicates end of order; clerk enters end of order.</li> <li>9. Customer submits payment; clerk enters amount.</li> </ol>	<ol style="list-style-type: none"> <li>3.1 Create a new order.</li> <li>5.1 Display item information.</li> <li>6.1 Add create an order item.</li> <li>8.1 Complete order.</li> <li>8.2 Compute totals.</li> <li>9.1 Verify payment.</li> <li>9.2 Create order transaction.</li> <li>9.3 Finalize order.</li> </ol>



# Bottom Detail from Fully Developed Use Case Description (Figure 7-10)

<b>Exception Conditions:</b>	<p>2.1 If customer does not exist, then the clerk pauses this use case and invokes <i>Maintain customer information</i> use case.</p> <p>2.2 If customer has a credit hold, then clerk transfers the customer to a customer service representative.</p> <p>4.1 If an item is not in stock, then customer can</p> <ul style="list-style-type: none"><li>a. choose not to purchase item, or</li><li>b. request item be added as a back-ordered item.</li></ul> <p>9.1 If customer payment is rejected due to bad-credit verification, then</p> <ul style="list-style-type: none"><li>a. order is canceled, or</li><li>b. order is put on hold until check is received.</li></ul>
------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

# Use Case Description Components

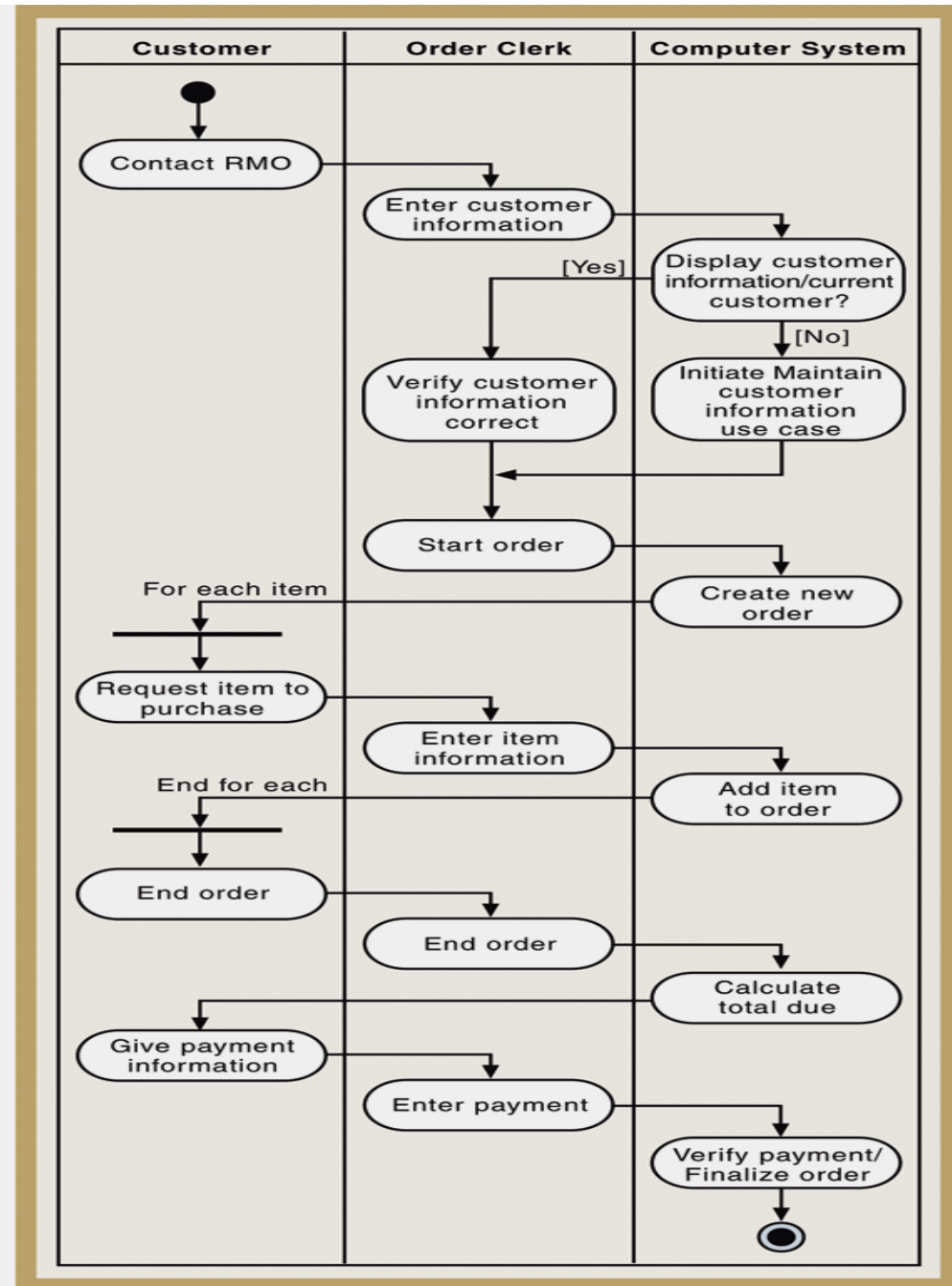
- ◆ Use case name/scenario name
- ◆ Actors/stakeholders
- ◆ Related use cases
- ◆ Preconditions – set of criteria that must be true prior to initiation of the use case
- ◆ Postconditions – set of criteria that must be true upon completion of the use case
- ◆ Flow of activities (steps in one column or two)
- ◆ Exception conditions

# Activity Diagrams

- ◆ Used to document workflow of business process activities for each use case or scenario
- ◆ Standard UML 2.0 diagram as seen in Chapter 4
- ◆ Can support any level of use case description; a supplement to use case descriptions
- ◆ Helpful in developing system sequence diagrams

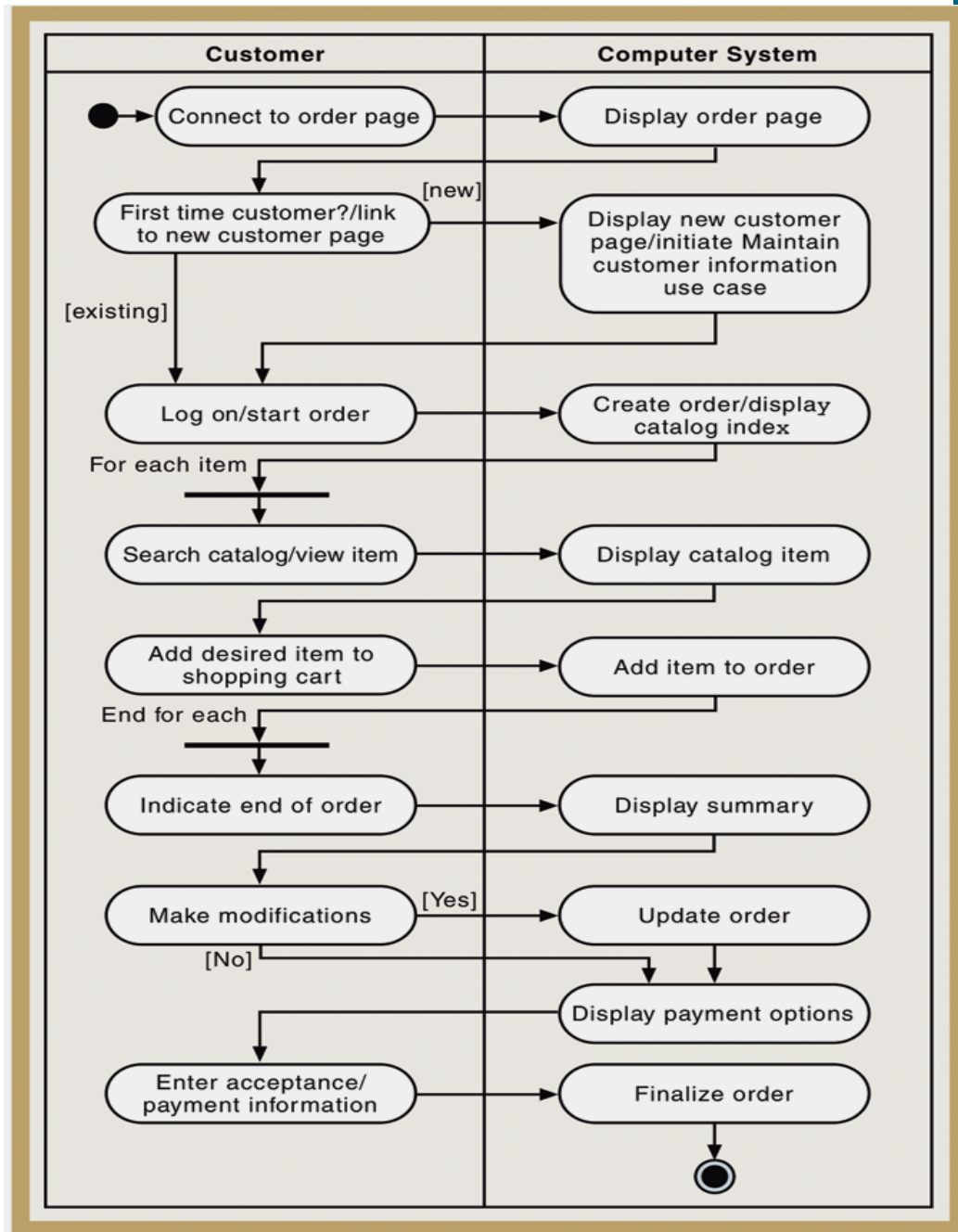
# Activity Diagram— Telephone Order Scenario

(Figure 7-12)



# Activity Diagram— Web Order Scenario

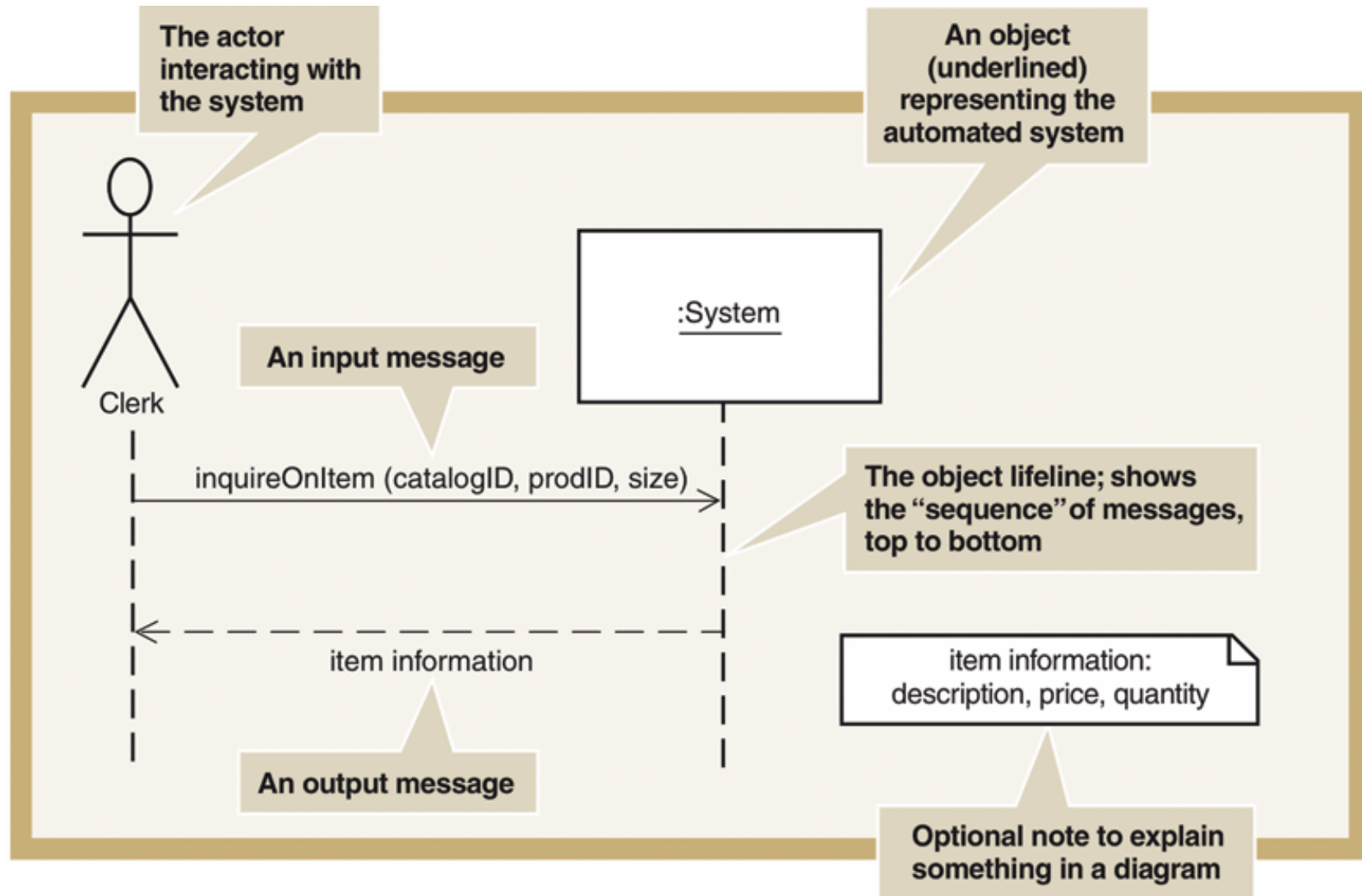
(Figure 7-13)



# Identifying Inputs and Outputs— The System Sequence Diagram

- ◆ **System sequence diagram (SSD)** is type of UML 2.0 interaction diagram
- ◆ Used to model input and output messaging requirements for a use case or scenario
- ◆ Shows actor interacting with system
- ◆ Shows sequence of interactions as messages during flow of activities
- ◆ System is shown as one object: a “black box”

# System Sequence Diagram (SSD) Notation (Figure 7-14)





# SSD Notation

- ◆ **Actor** represented by a stick figure – a person (or role) that interacts with system by entering input data and receiving output data
- ◆ **Object** is a rectangle with name of object underlined – shows individual object and not class of all similar objects ( :System for SSD )
- ◆ **Lifeline** or **object lifeline** is a vertical line under object or actor to show passage of time for object
- ◆ **Message** is labeled on arrows to show messages sent to or received by actor or system



# SSD Lifelines

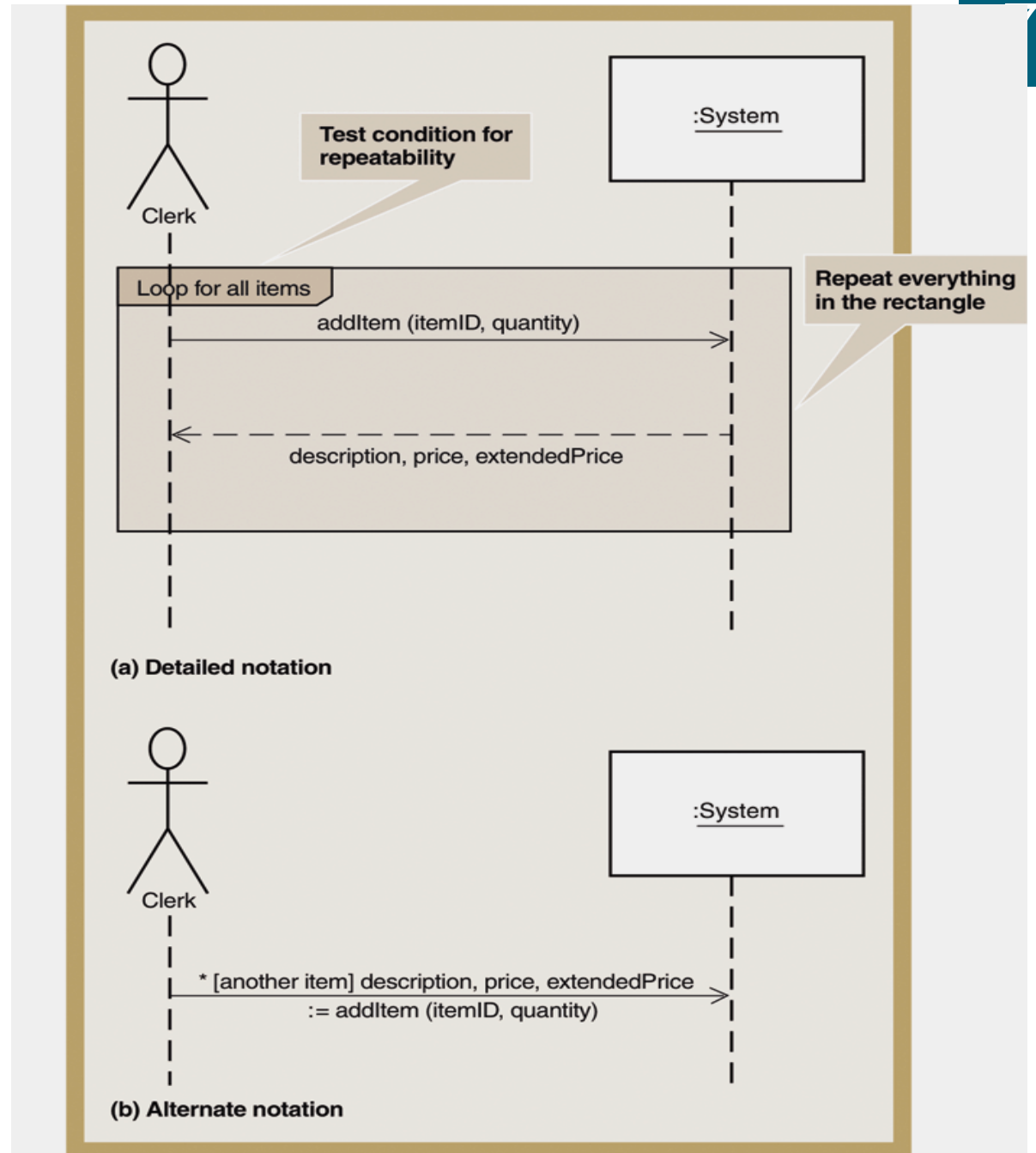
- ◆ Vertical line under object or actor
  - Shows passage of time
- ◆ If vertical line dashed
  - Creation and destruction of thing is not important for scenario
- ◆ Long narrow rectangles
  - Activation lifelines emphasize that object is active only during part of scenario

# SSD Messages

- ◆ Internal events identified by the flow of objects in a scenario
- ◆ Requests from one actor or object to another to do some action
- ◆ Invoke a particular method

# Repeating Message

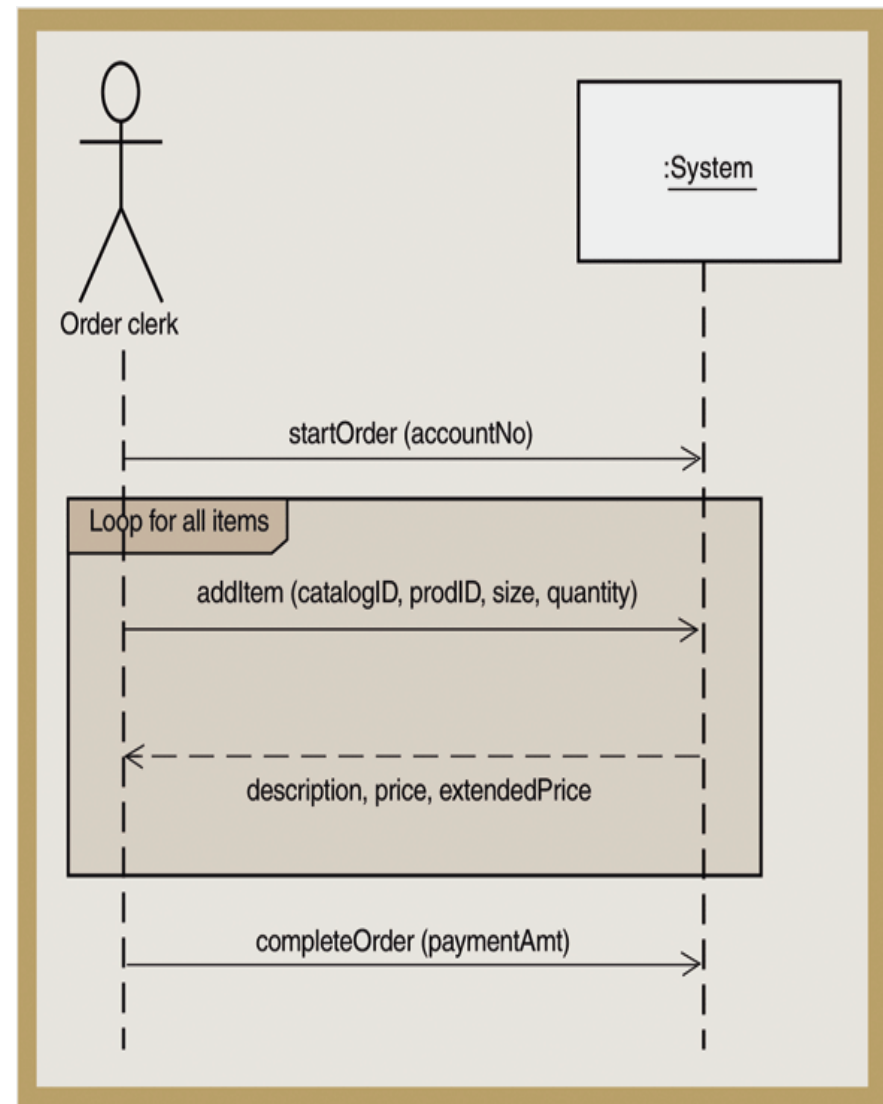
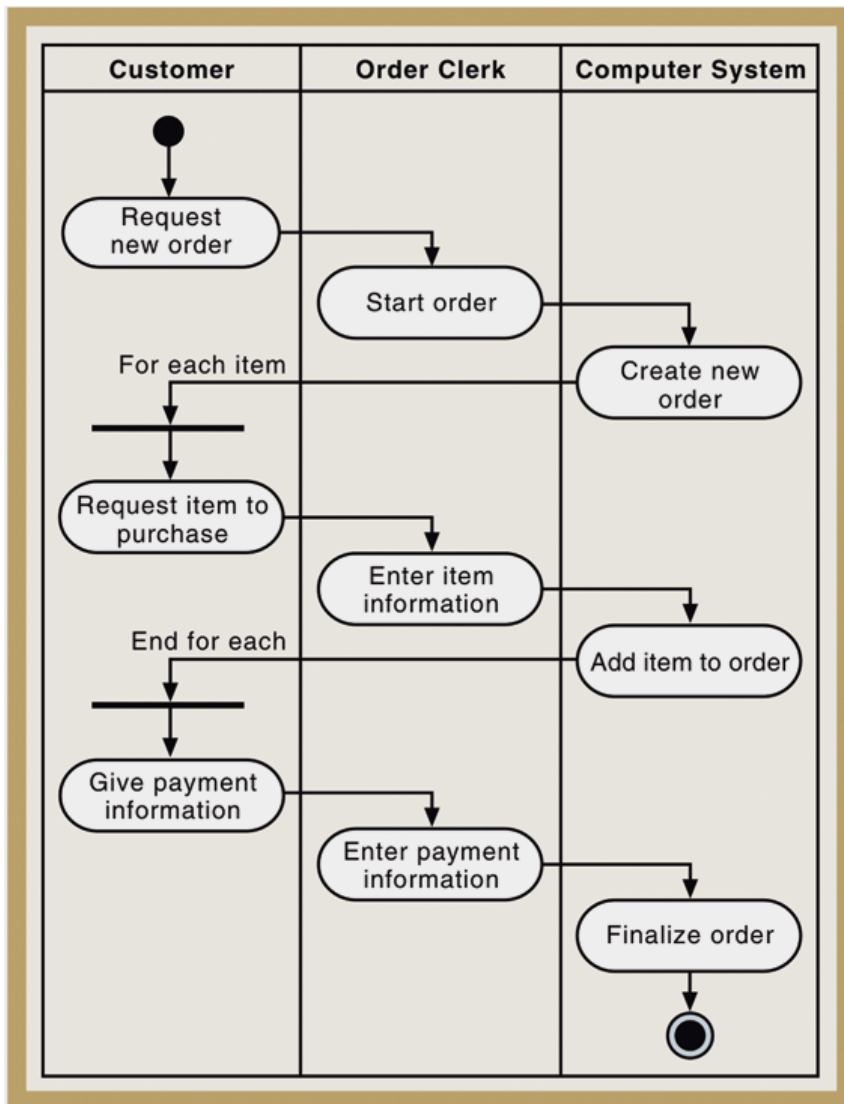
(Figure 7-15)



# Developing a System Sequence Diagram

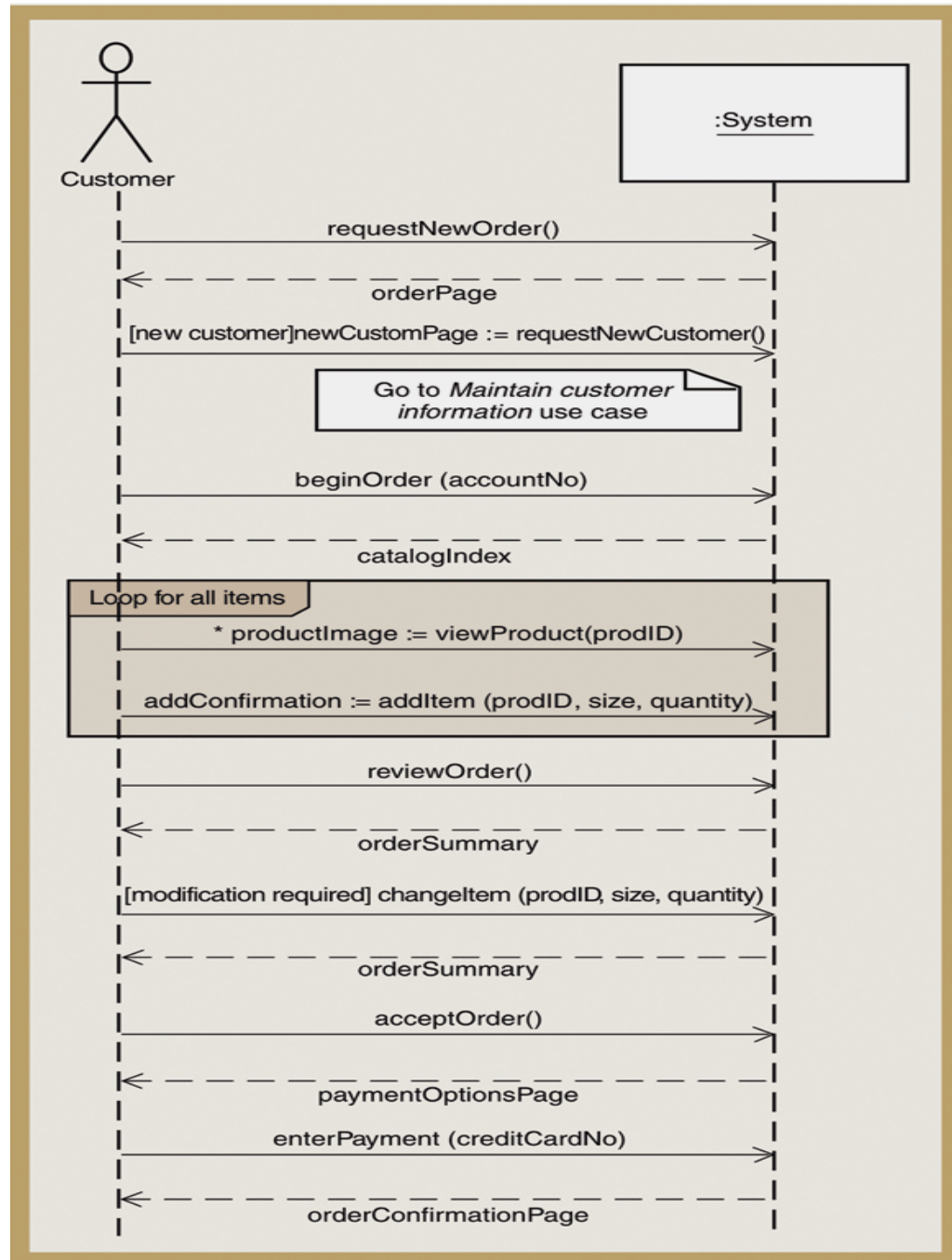
- ◆ Begin with detailed description of use case from fully developed form or activity diagram
- ◆ Identify input messages
- ◆ Describe message from external actor to system using message notation
- ◆ Identify and add any special conditions on input message, including iteration and true/false conditions
- ◆ Identify and add output return messages

# Activity Diagram and Resulting SSD for Telephone Order Scenario (Figures 7-16 and 7-17)



# SSD of the Web Order Scenario for the *Create New Order* Use Case

(Figure 7-18)



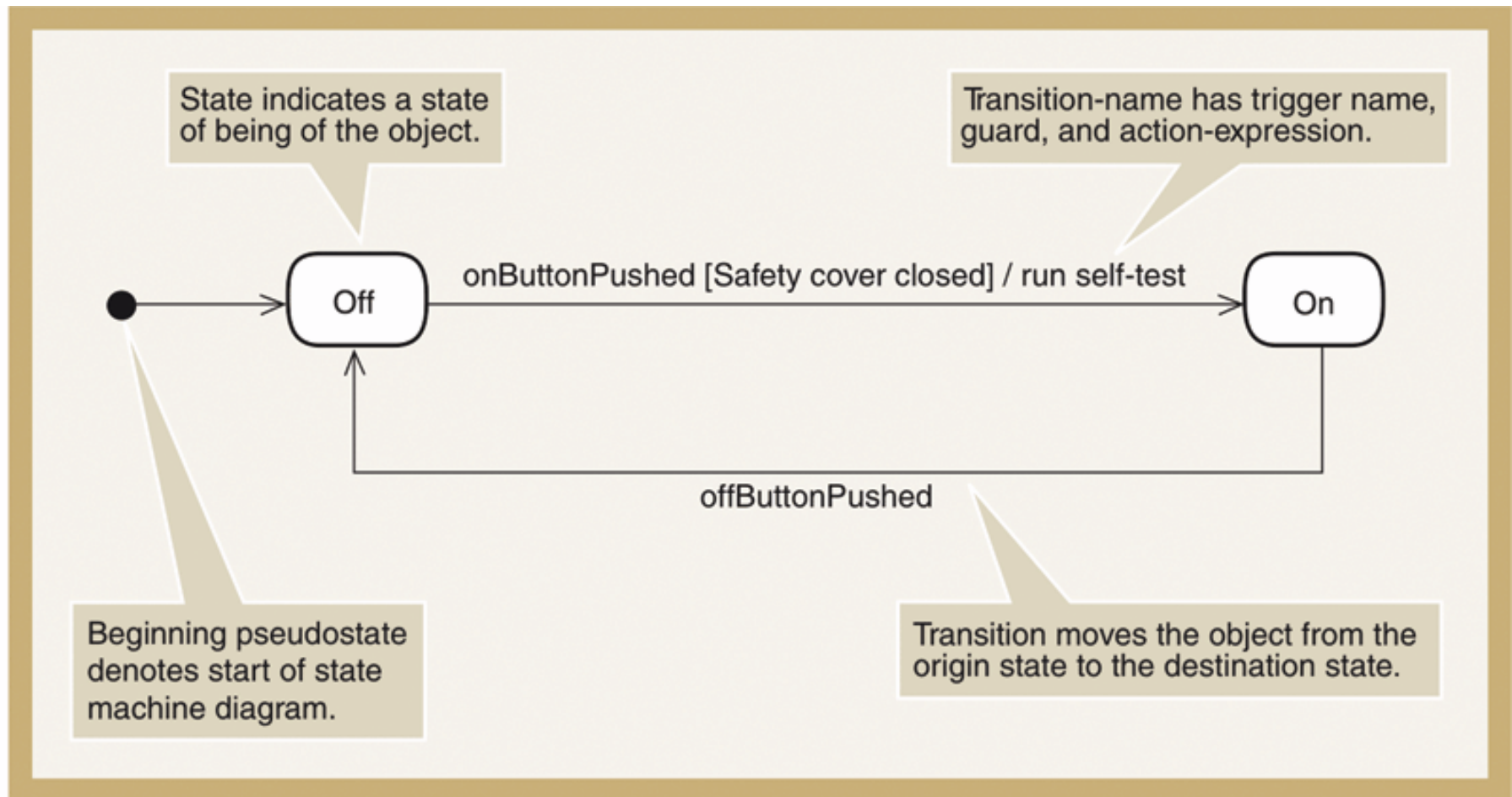
# Identifying Object Behavior— The State Machine Diagram

- ◆ State machine diagram is UML 2.0 diagram that models object states and transitions
  - Complex problem domain classes can be modeled
- ◆ State of an object
  - A condition that occurs during its life when it satisfies some criterion, performs some action, or waits for an event
  - Each state has unique name and is a semipermanent condition or status
- ◆ Transition
  - The movement of an object from one state to another state



# Simple State Machine Diagram for a Printer

(Figure 7-19)

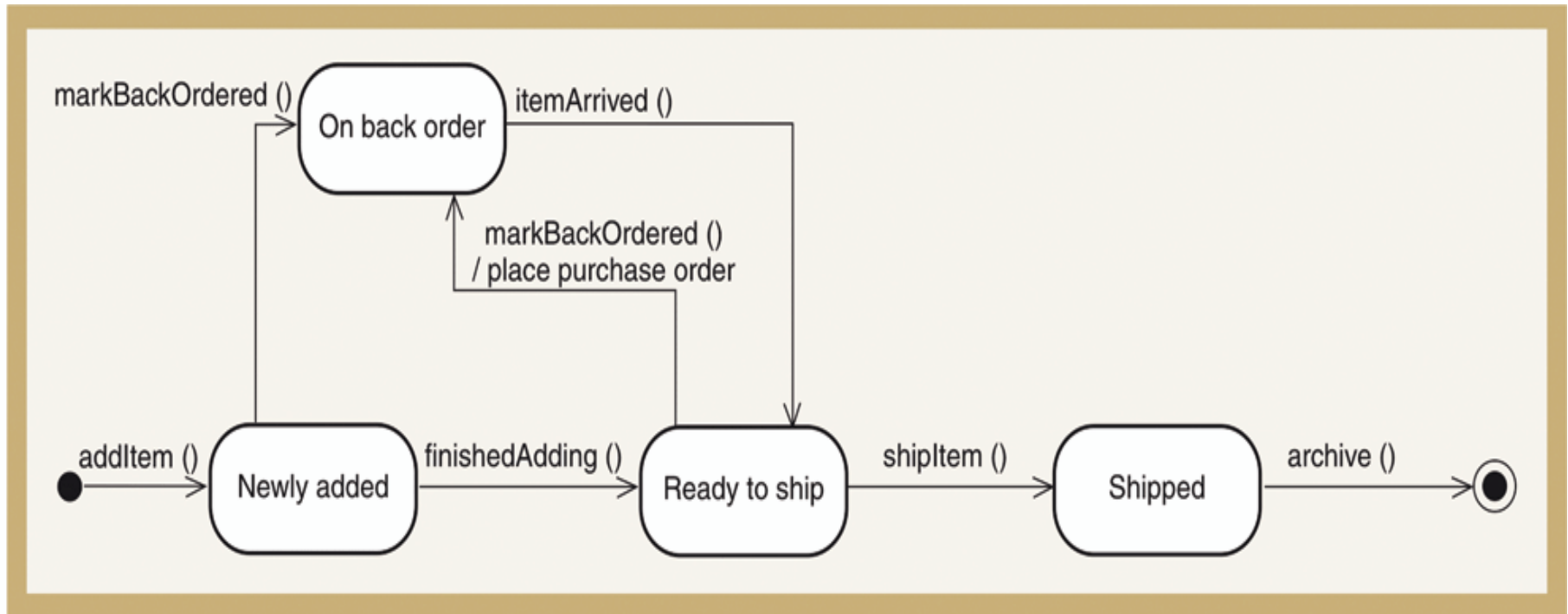




# State Machine Terminology

- ◆ Pseudo state – the starting point of a state machine, indicated by a black dot
- ◆ Origin state – the original state of an object from which the transition occurs
- ◆ Destination state – the state to which an object moves after the completion of a transition
- ◆ Message event – the trigger for a transition, which causes the object to leave the origin state
- ◆ Guard condition – a true/false test to see whether a transition can fire
- ◆ Action expression – a description of the activities performed as part of a transition

# RMO State Machine Diagram for OrderItem Problem Domain Class



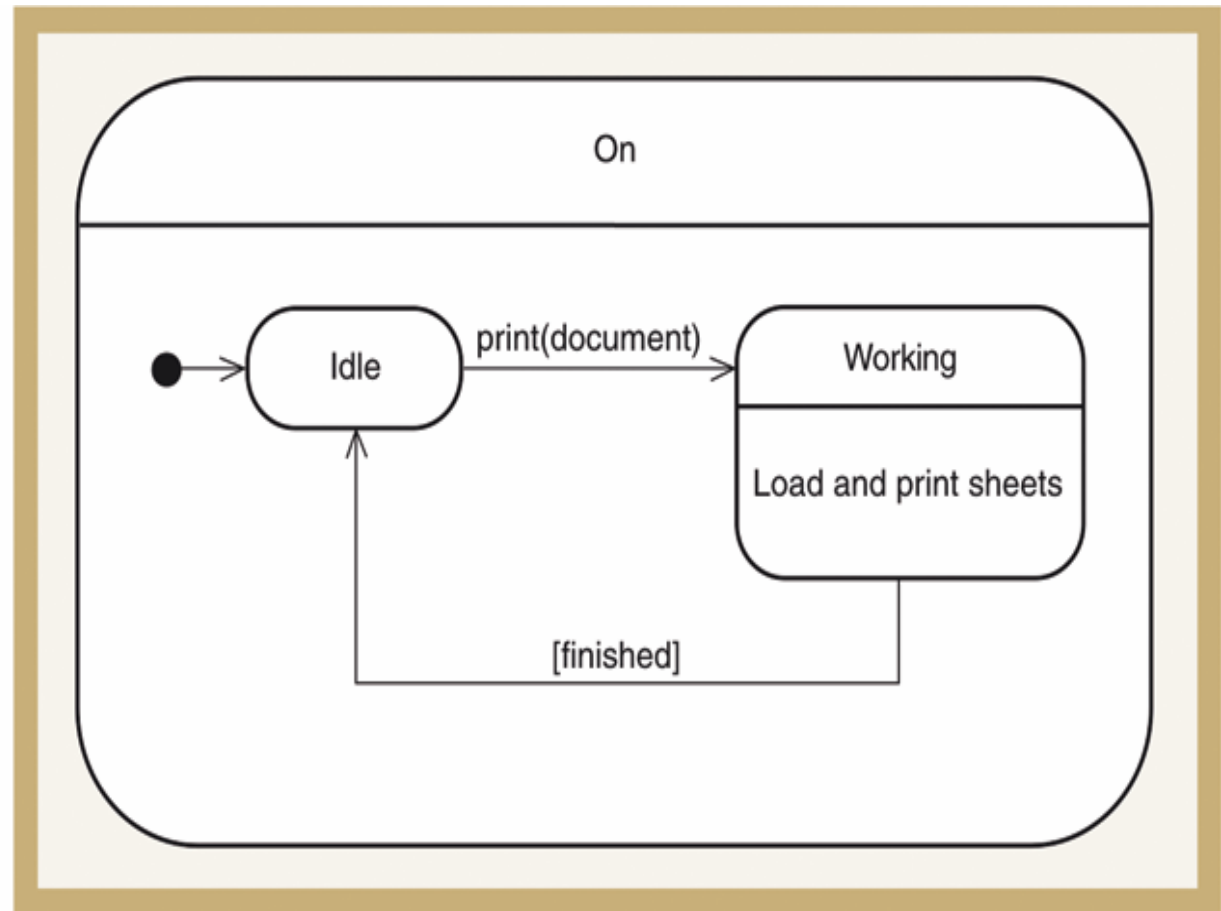
**Figure 7-24**

Final state machine diagram for OrderItem

# Composite States and Concurrency— States within a State

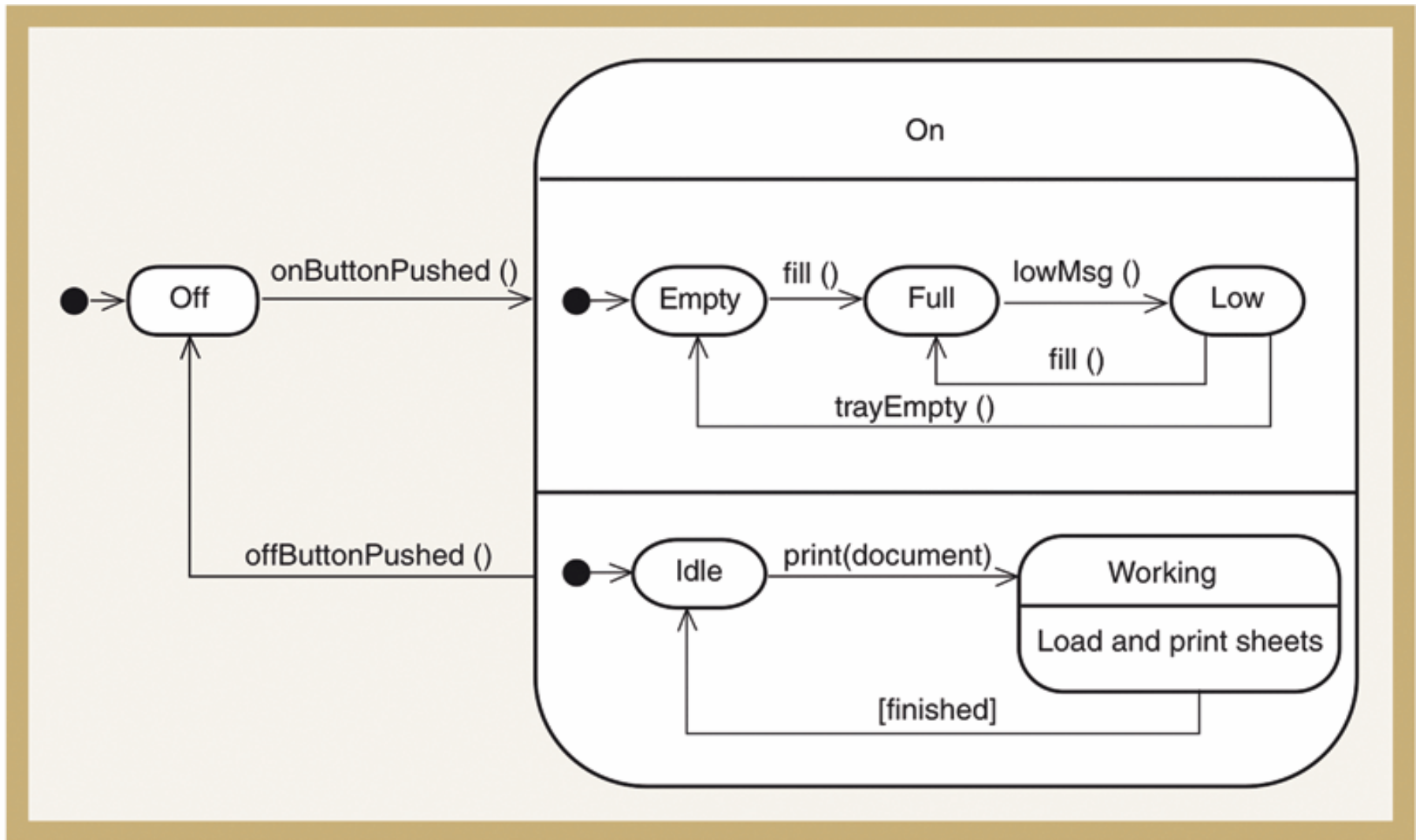
**Figure 7-20**

Sample composite states  
for the printer object



# Concurrent Paths for Printer in the *On* State

(Figure 7-21)



# Rules for Developing State Machine Diagram

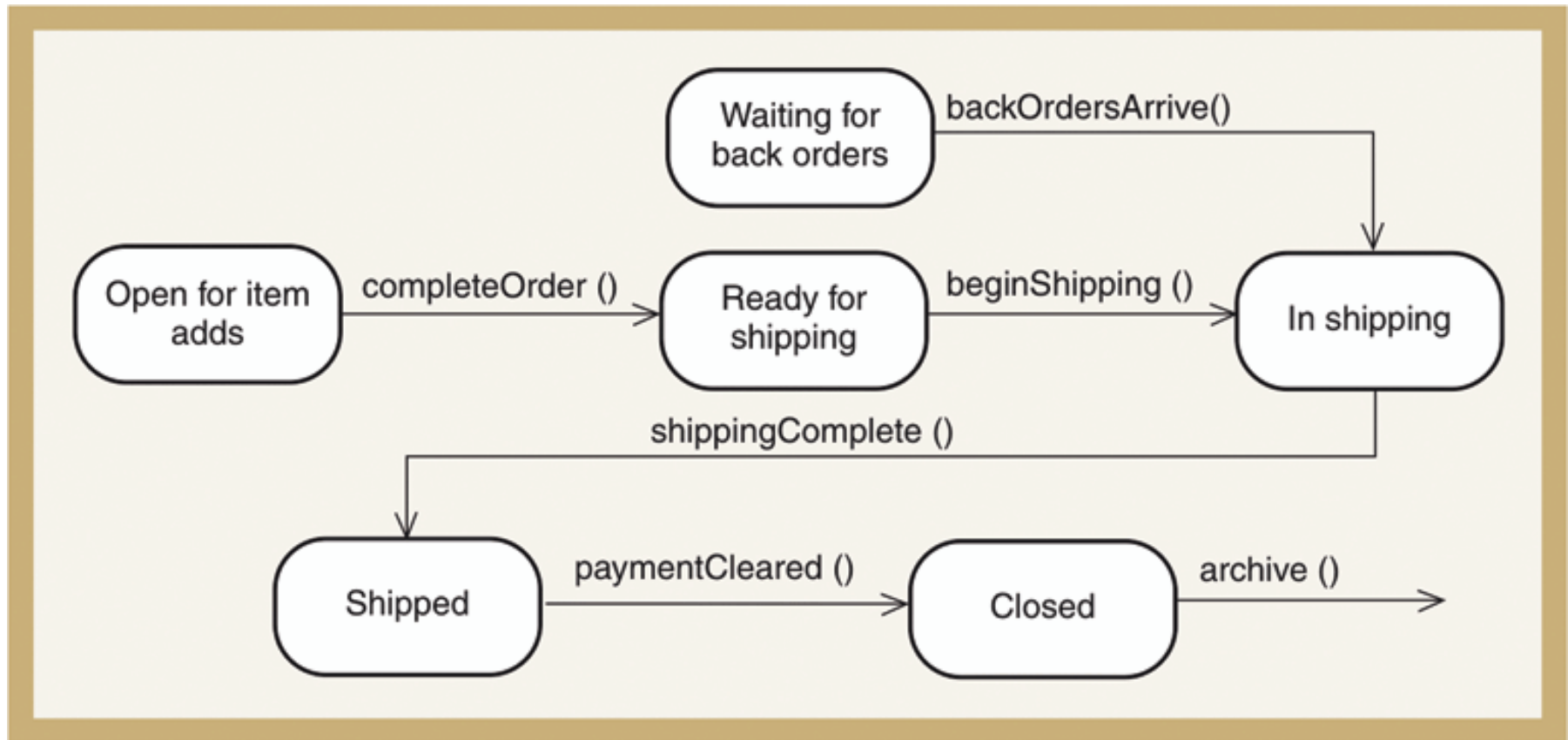
- ◆ Review domain class diagram, select important ones, and list all state and exit conditions
- ◆ Begin building state machine diagram fragments for each class
- ◆ Sequence fragments in correct order and review for independent and concurrent paths
- ◆ Expand each transition with message event, guard-condition, and action-expression
- ◆ Review and test each state machine diagram

## Order Domain Class for RMO— States and Exit Transitions (Figure 7-25)

State	Exit transition
Open for item adds	completeOrder
Ready for shipping	beginShipping
In shipping	shippingComplete
Waiting for back orders	backOrdersArrive
Shipped	paymentCleared
Closed	archive

# First-Cut State Machine Diagram for Order

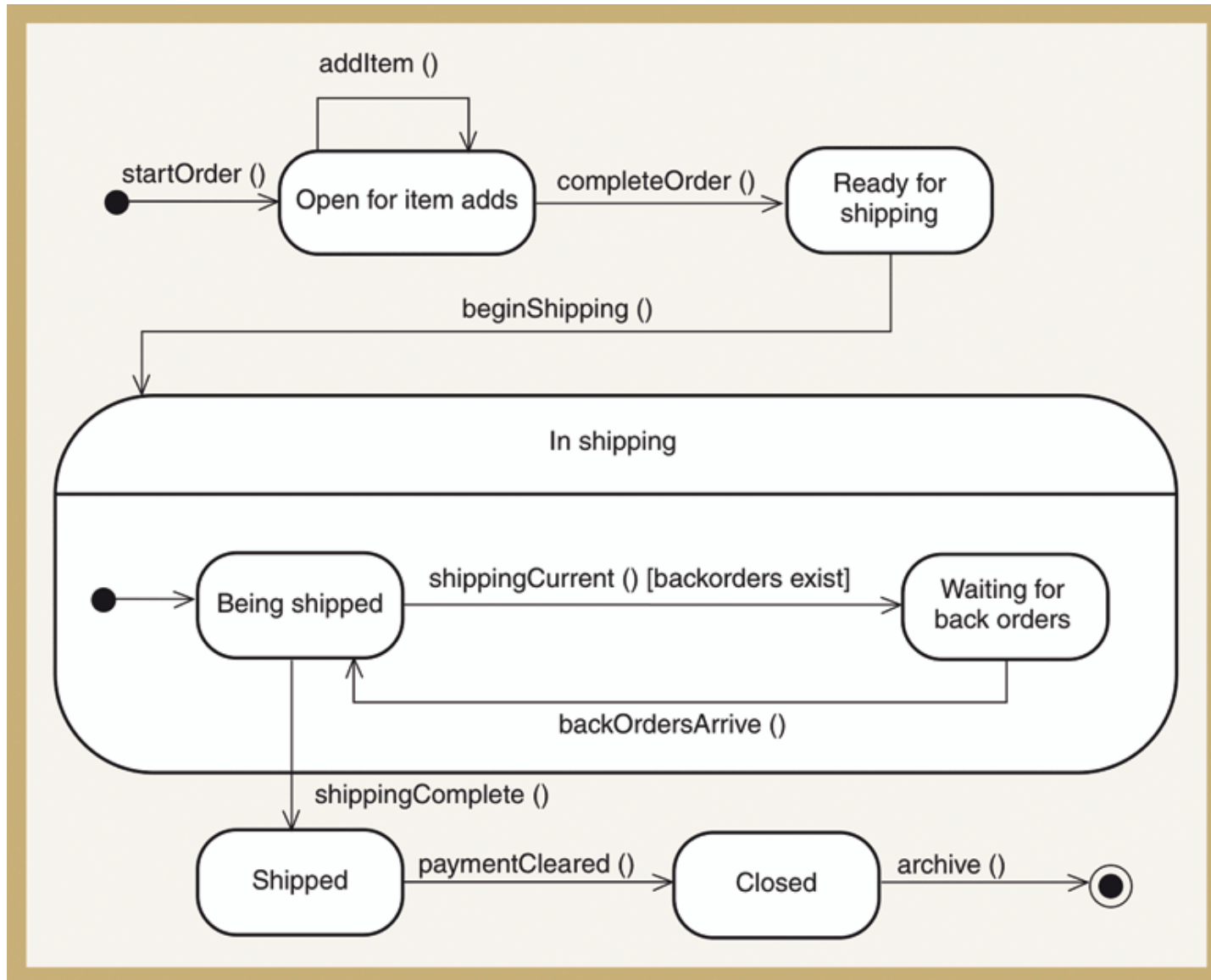
(Figure 7-26)





# Second-Cut State Machine Diagram for Order

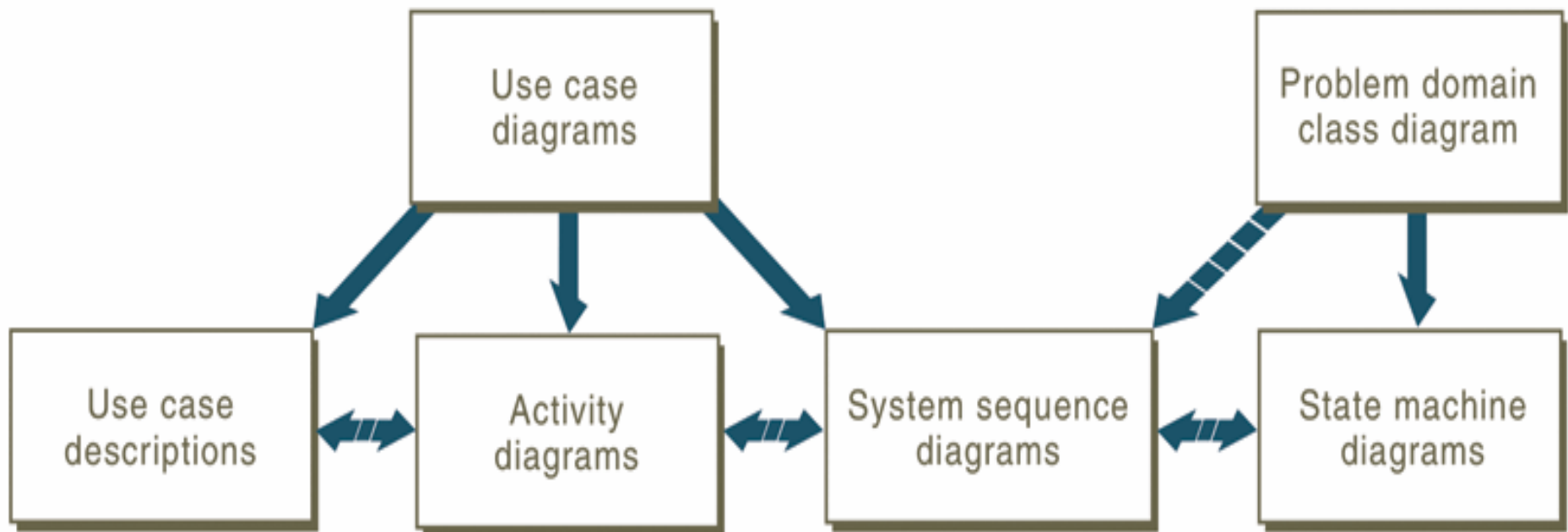
(Figure 7-26)



# Integrating Object-Oriented Models

- ◆ Complete use case diagram is needed to understand total scope of new system
- ◆ Domain model class diagrams should also be as complete as possible for entire system
- ◆ With iterative approach, only construct use case descriptions, activity diagrams, and system sequence diagrams for use cases in iteration
- ◆ Development of a new diagram often helps refine and correct previous diagrams

# Relationships Between OO Requirements Models (Figure 7-28)



# Summary

- ◆ Object-oriented approach has complete set of diagrams that define system requirements
- ◆ Requirements specified using following models
  - Domain model class diagram (Chapter 5)
  - Use case diagrams (Chapter 7)
  - Use case detailed models, either descriptive formats or activity diagrams (Chapter 7)
  - System sequence diagrams (Chapter 7)
  - State machine diagrams (Chapter 7)