INTRODUCTION TO SYSTEMS ANALYSIS AND DESIGN:

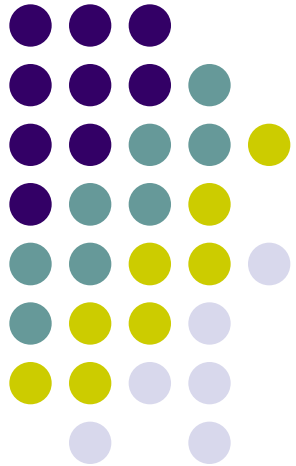
AN AGILE, ITERATIVE APPROACH

SATZINGER | JACKSON | BURD

CHAPTER 3

Use Cases

Chapter 3



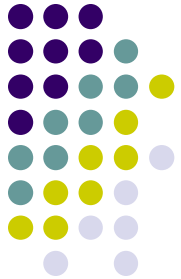
Introduction to Systems Analysis and Design: An Agile, Iteractive Approach 6th Ed

Satzinger, Jackson & Burd

Chapter 3 Outline

- Use Cases and User Goals
- Use Cases and Event Decomposition
- Use Cases and CRUD
- Use Cases in the Ridgeline Mountain Outfitters Case
- User Case Diagrams

Learning Objectives



- Explain why identifying use cases is the key to defining functional requirements
- Describe the two techniques for identifying use cases
- Apply the user goal technique to identify use cases
- Apply the event decomposition technique to identify use cases
- Apply the CRUD technique to validate and refine the list of use cases
- Describe the notation and purpose for the use case diagram
- Draw use case diagrams by actor and by subsystem

Overview

- Chapter 2 provided an overview of systems analysis activities, functional and non-functional requirements, modeling, and information gathering techniques
- This chapter focuses on identifying and modeling the key aspect of functional requirements– use cases
- In the RMO Tradeshow System from Chapter 1, some use cases are *Look up supplier*, *Enter/update product information*, *Enter/Update contact information*
- In this chapter's opening case Waiters on Call, examples of use cases are *Record an order*, *Record delivery*, *Update an order*, *Sign in driver*, *Reconcile driver receipts, Produce end of day deposit slip,* and *Produce weekly sales reports*

Use Cases

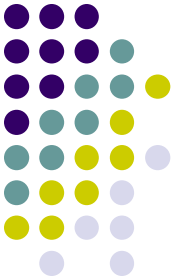
- Use case— an activity that the system performs, usually in response to a request by a user
- Use cases define functional requirements
- Analysts decompose the system into a set of use cases (functional decomposition)
- Two techniques for Identifying use cases
  - User goal technique
  - Event decomposition technique
- Name each use case using *Verb-Noun*

User Goal Technique

- This technique is the most common in industry
- Simple and effective
- Identify all of the potential categories of users of the system
- Interview and ask them to describe the tasks the computer can help them with
- Probe further to refine the tasks into specific user goals, "I need to *Ship items*, *Track a shipment*, *Create a return*"

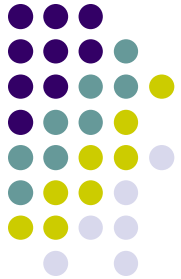
User Goal Technique

Some RMO CSMS Users and Goals



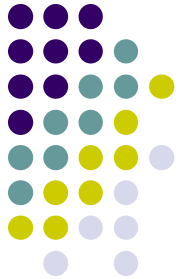
User	User goal and resulting use case
Potential customer	Search for item Fill shopping cart View product rating and comments
Marketing manager	Add/update product information Add/update promotion Produce sales history report
Shipping personnel	Ship items Track shipment Create item return

User Goal Technique: Specific Steps



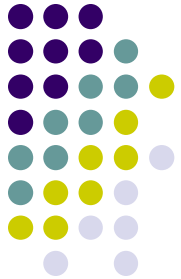
1. Identify all the potential users for the new system
2. Classify the potential users in terms of their functional role (e.g., shipping, marketing, sales)
3. Further classify potential users by organizational level (e.g., operational, management, executive)
4. For each type of user, interview them to find a list of specific goals they will have when using the new system (current goals and innovative functions to add value)

User Goal Technique Specific Steps (continued)



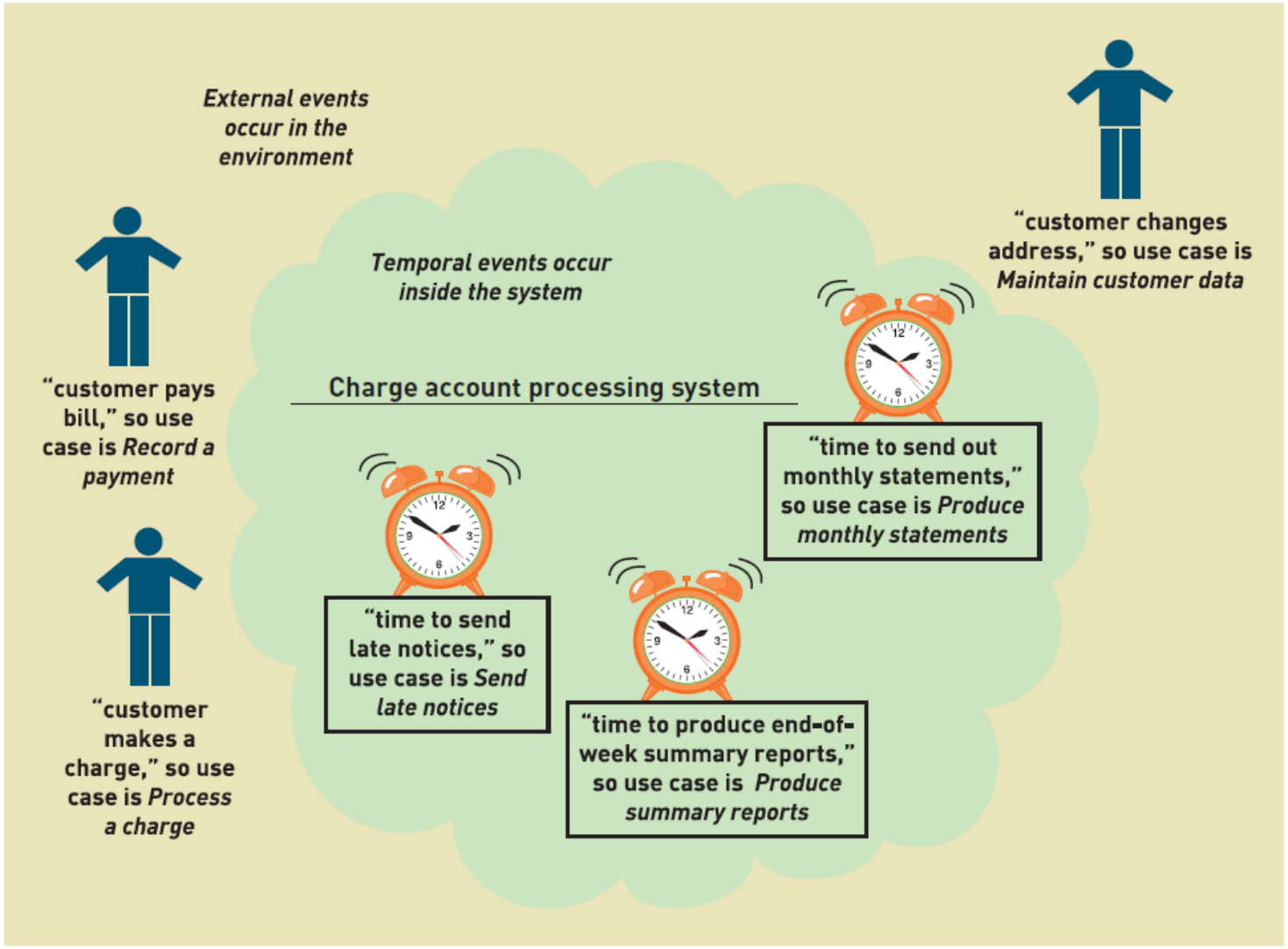
5. Create a list of preliminary use cases organized by type of user
6. Look for duplicates with similar use case names and resolve inconsistencies
7. Identify where different types of users need the same use cases
8. Review the completed list with each type of user and then with interested stakeholders

Event Decomposition Technique



- More Comprehensive and Complete Technique
  - Identify the events that occur to which the system must respond.
  - For each event, name a use case (verb-noun) that describes what the system does when the event occurs
- Event– something that occurs at a specific time and place, can be described, and should be remembered by the system

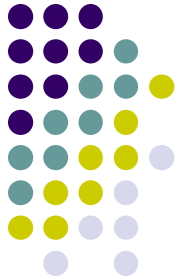
Events and Use Cases



Types of Events

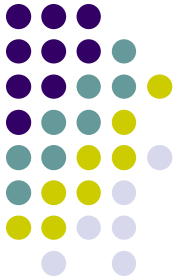
- External Event
  - an event that occurs outside the system, usually initiated by an external agent or actor
- Temporal Event
  - an event that occurs as a result of reaching a point in time
- State Event
  - an event that occurs when something happens inside the system that triggers some process
  - reorder point is reached for inventory item

External Event Checklist



- External agent or actor wants something resulting in a transaction
  - Customer buys a product
- External agent or actor wants some information
  - Customer wants to know product details
- External data changed and needs to be updated
  - Customer has new address and phone
- Management wants some information
  - Sales manager wants update on production plans

Temporal Event Checklist



- Internal outputs needed at points in time
  - Management reports (summary or exception)
  - Operational reports (detailed transactions)
  - Internal statements and documents (including payroll)
- External outputs needed at points of time
  - Statements, status reports, bills, reminders

Finding the actual event that affects the system

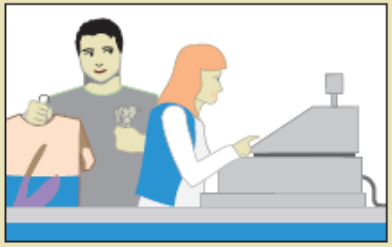
Customer thinks about getting a new shirt

Customer drives to the mall

Customer tries on a shirt at Sears

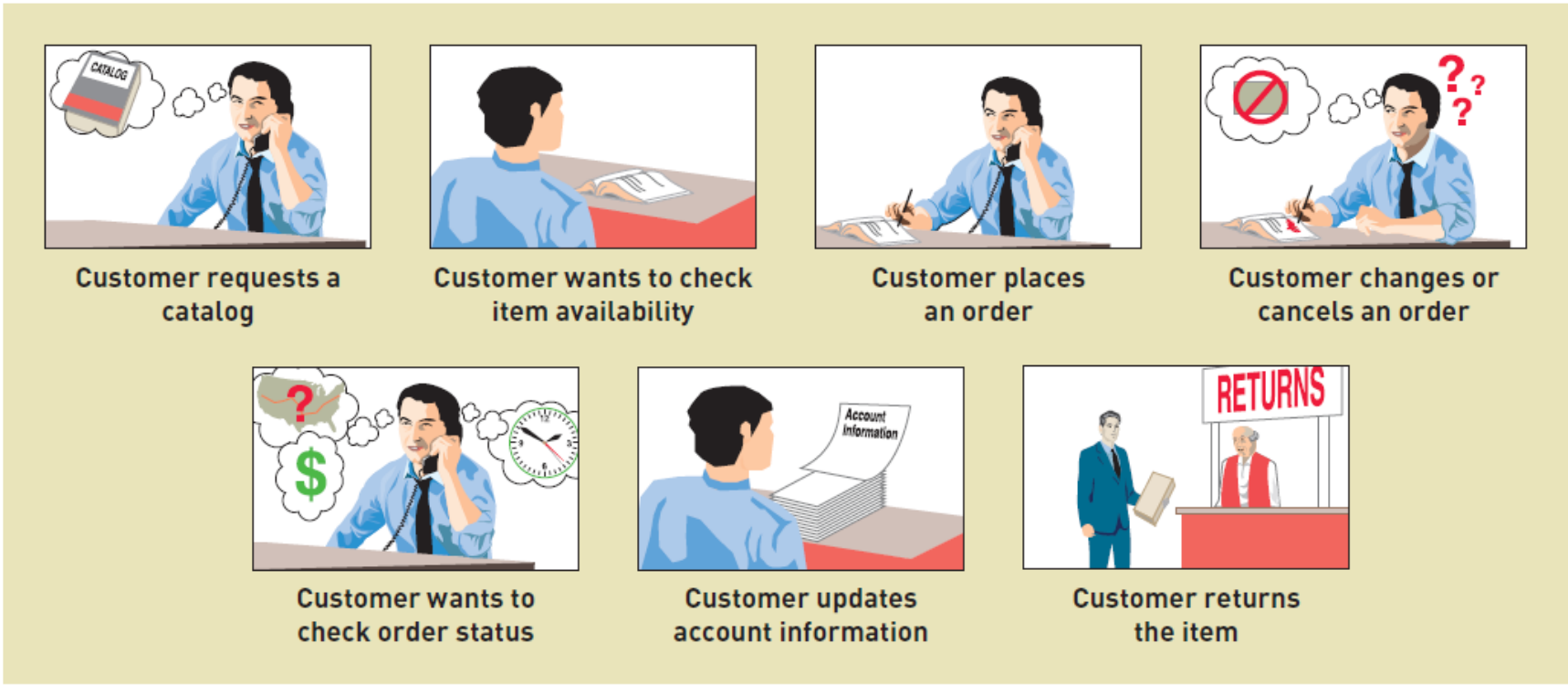
Customer goes to Walmart

Customer tries on a shirt at Walmart



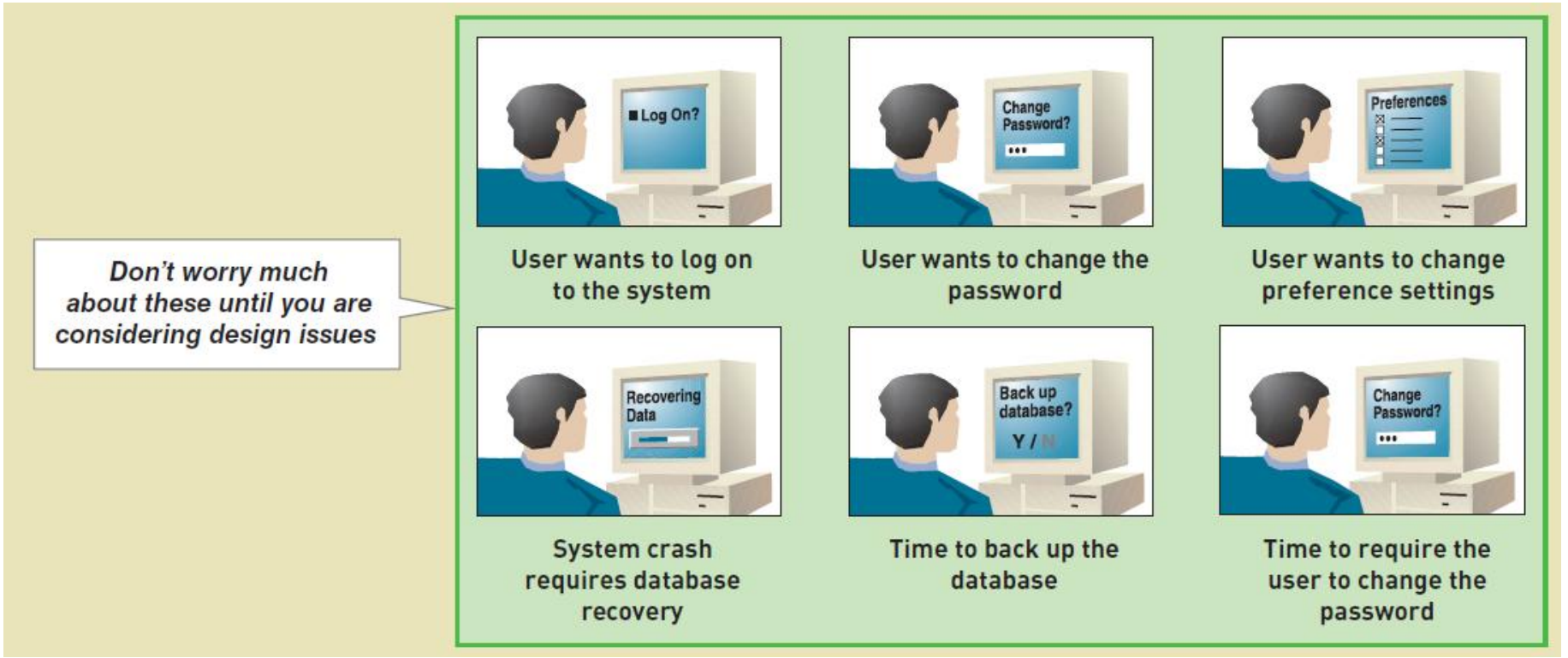
Customer buys a shirt
(the event that directly affects the system!)

Tracing a sequence of transactions resulting in many events

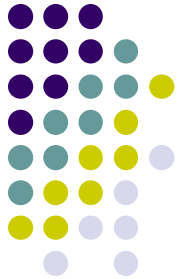


Perfect Technology Assumption

- Don’t worry about functions built into system because of limits in technology and people. Wait until design.

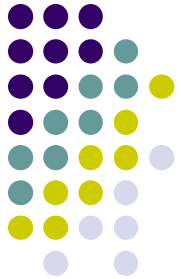


Event Decomposition Technique: Specific Steps



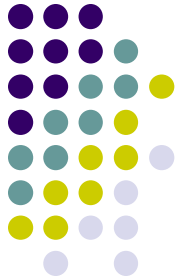
1. Consider the external events in the system environment that require a response from the system by using the checklist shown in Figure 3-3
2. For each external event, identify and name the use case that the system requires
3. Consider the temporal events that require a response from the system by using the checklist shown in Figure 3-4
4. For each temporal event, identify and name the use case that the system requires and then establish the point of time that will trigger the use case

Event Decomposition Technique: Specific Steps (continued)



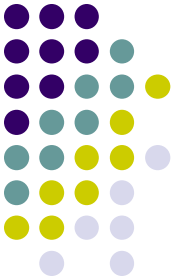
5. Consider the state events that the system might respond to, particularly if it is a real-time system in which devices or internal state changes trigger use cases.
6. For each state event, identify and name the use case that the system requires and then define the state change.
7. When events and use cases are defined, check to see if they are required by using the perfect technology assumption. Do not include events that involve such system controls as login, logout, change password, and backup or restore the database, as these are put in later.

Event Decomposition Technique: Benefits



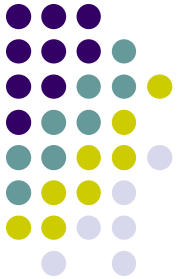
- Events are broader than user goal: Capture temporal and state events
- Help decompose at the right level of analysis: an elementary business process (EBP)
- EBP is a fundamental business process performed by one person, in one place, in response to a business event
- Uses perfect technology assumption to make sure functions that support the users work are identified and not additional functions for security and system controls

Use Cases and CRUD Technique



- CRUD is Create, Read/Report, Update, and Delete (archive)
- Often introduced in database context
- Technique to validate, refine or cross-check use cases
- NOT for primarily identifying use cases

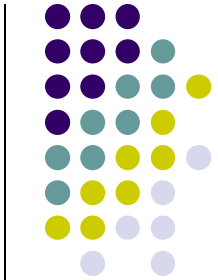
Use Cases and CRUD Technique



- For Customer domain class, verify that there are use cases that create, read/report, update, and delete (archive) the domain class

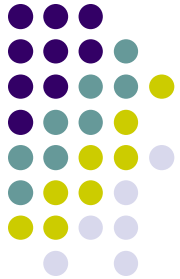
Data entity/domain class	CRUD	Verified use case
Customer	Create	Create customer account
	Read/report	Look up customer Produce customer usage report
	Update	Process account adjustment Update customer account
	Delete	Update customer account (to archive)

CRUD Technique Steps



1. Identify all the data entities or domain classes involved in the new system. (more in Chapter 4)
2. For each type of data (data entity or domain class), verify that a use case has been identified that creates a new instance, updates existing instances, reads or reports values of instances, and deletes (archives) an instance.
3. If a needed use case has been overlooked, add a new use case and then identify the stakeholders.
4. With integrated applications, make sure it is clear which application is responsible for adding and maintaining the data and which system merely uses the data.

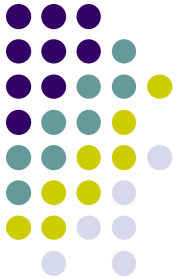
CRUD Technique
Use Case vs. Domain Class Table



- To summarize CRUD analysis results, create a matrix of use cases and domain classes indicating which use case C, R, U, or D a domain class

Use case vs. entity/domain class	Customer	Account	Sale	Adjustment
Create customer account	C	C		
Look up customer	R	R		
Produce customer usage report	R	R	R	
Process account adjustment	R	U	R	C
Update customer account	UD (archive)	UD (archive)		

Use Cases and Brief Use Case Descriptions



- Brief use case description is often a one sentence description showing the main steps in a use case

Use case	Brief use case description
Create customer account	User/actor enters new customer account data, and the system assigns account number, creates a customer record, and creates an account record.
Look up customer	User/actor enters customer account number, and the system retrieves and displays customer and account data.
Process account adjustment	User/actor enters order number, and the system retrieves customer and order data; actor enters adjustment amount, and the system creates a transaction record for the adjustment.

RMO CSMS Project Use Cases

CSMS sales subsystem	
Use cases	**Users/actors**
Search for item	Customer, customer service representative, store sales representative
View product comments and ratings	Customer, customer service representative, store sales representative
View accessory combinations	Customer, customer service representative, store sales representative
Fill shopping cart	Customer
Empty shopping cart	Customer
Check out shopping cart	Customer
Fill reserve cart	Customer
Empty reserve cart	Customer
Convert reserve cart	Customer
Create phone sale	Customer service representative
Create store sale	Store sales representative

RMO CSMS Project Use Cases

CSMS order fulfillment subsystem	
Use cases	**Users/actors**
Ship items	Shipping
Manage shippers	Shipping
Create backorder	Shipping
Create item return	Shipping, customer
Look up order status	Shipping, customer, management
Track shipment	Shipping, customer, marketing
Rate and comment on product	Customer
Provide suggestion	Customer
Review suggestions	Management
Ship items	Shipping
Manage shippers	Shipping

RMO CSMS Project Use Cases

CSMS Customer account subsystem	
Use cases	**Users/actors**
Create/update customer account	Customer, customer service representative, store sales representative
Process account adjustment	Management
Send message	Customer
Browse messages	Customer
Request friend linkup	Customer
Reply to linkup request	Customer
Send/receive points	Customer
View "mountain bucks"	Customer
Transfer "mountain bucks"	Customer

RMO CSMS Project Use Cases

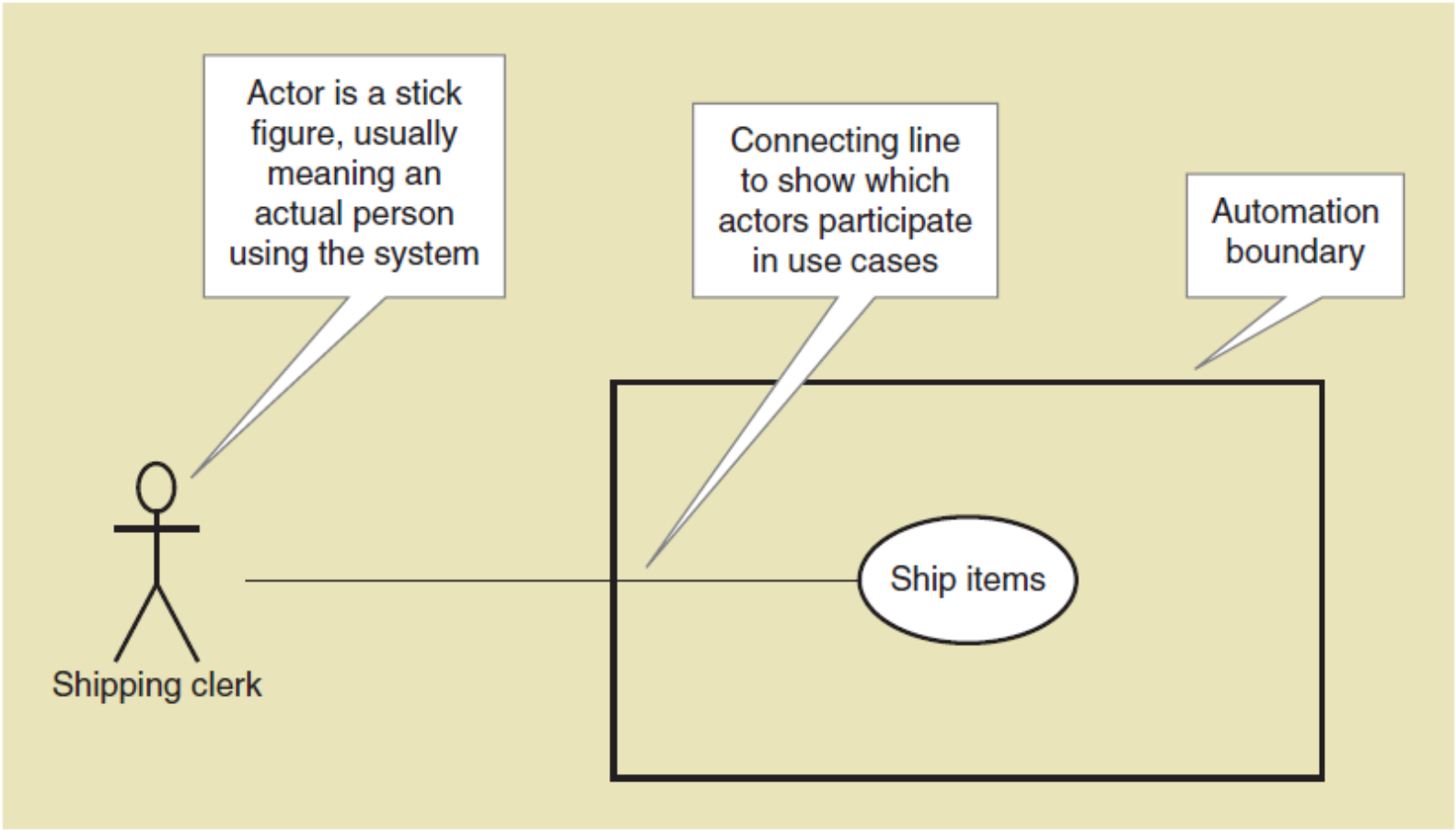
CSMS marketing subsystem	
Use cases	**Users/actors**
Add/update product information	Merchandising, marketing
Add/update promotion	Marketing
Add/update accessory package	Merchandising
Add/update business partner link	Marketing

CSMS reporting subsystem	
Use cases	**Users/actors**
Produce daily transaction summary report	Management
Produce sales history report	Management, marketing
Produce sales trends report	Marketing
Produce customer usage report	Marketing
Produce shipment history report	Management, shipping
Produce promotion impact report	Marketing
Produce business partner activity report	Management, marketing

Use Case Diagrams

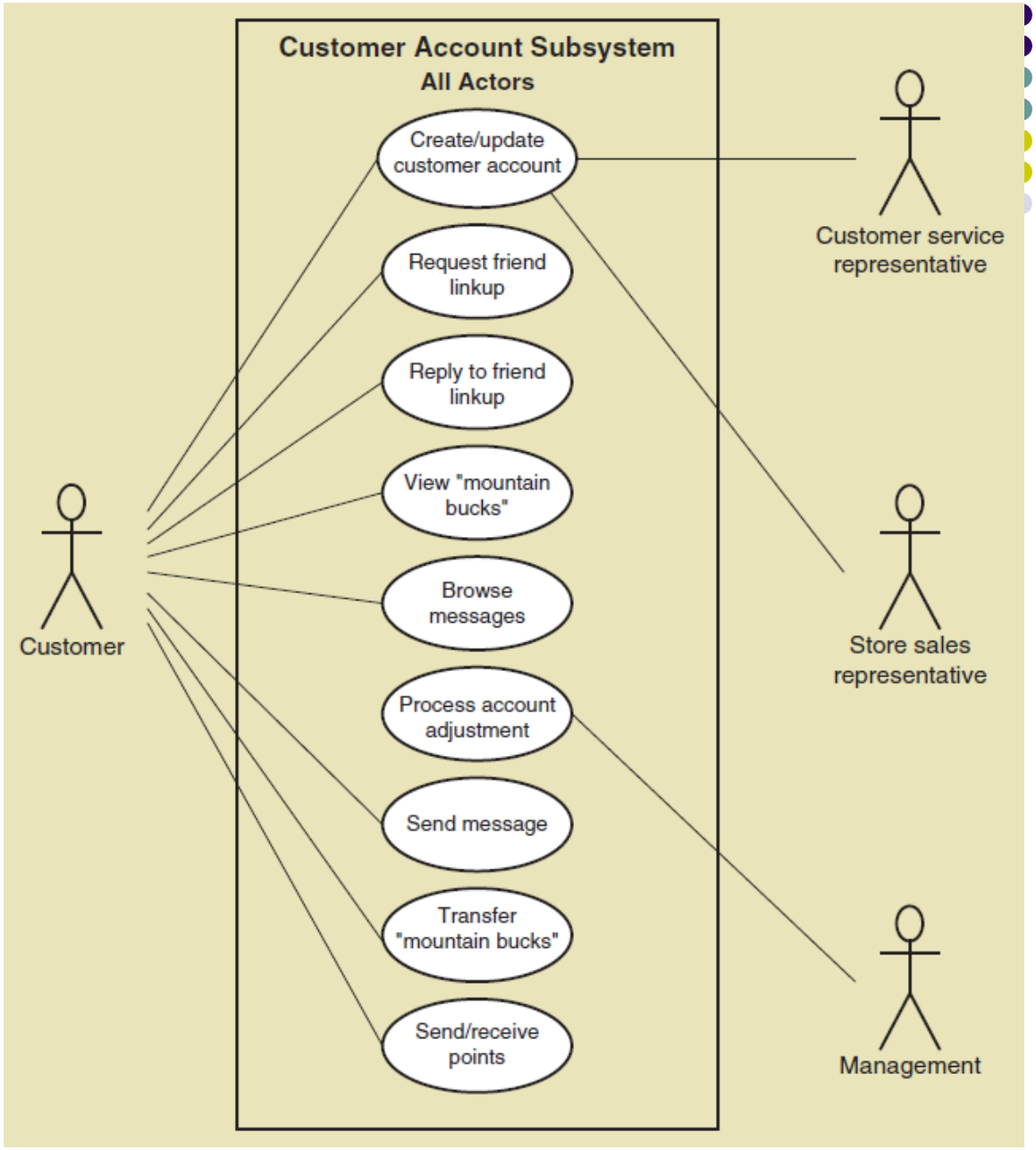
- Use case diagram— a UML model used to graphically show uses cases and their relationships to actors
- Recall UML is Unified Modeling Language, the standard for diagrams and terminology for developing information systems
- Actor is the UML name for a end user
- Automation boundary— the boundary between the computerized portion of the application and the users who operate the application

Use Case Diagrams Symbols



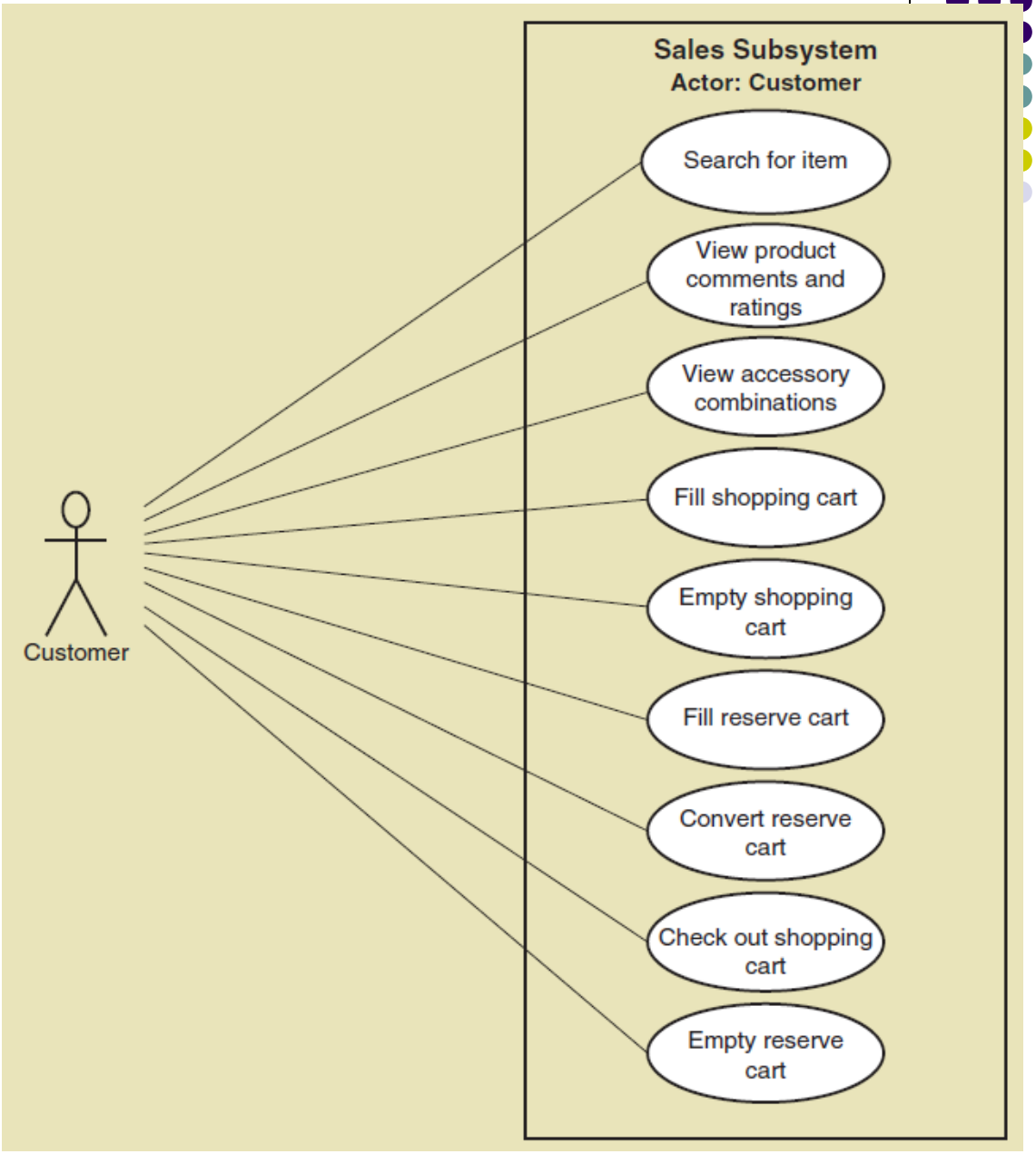
Use Case Diagrams

Draw for each subsystem

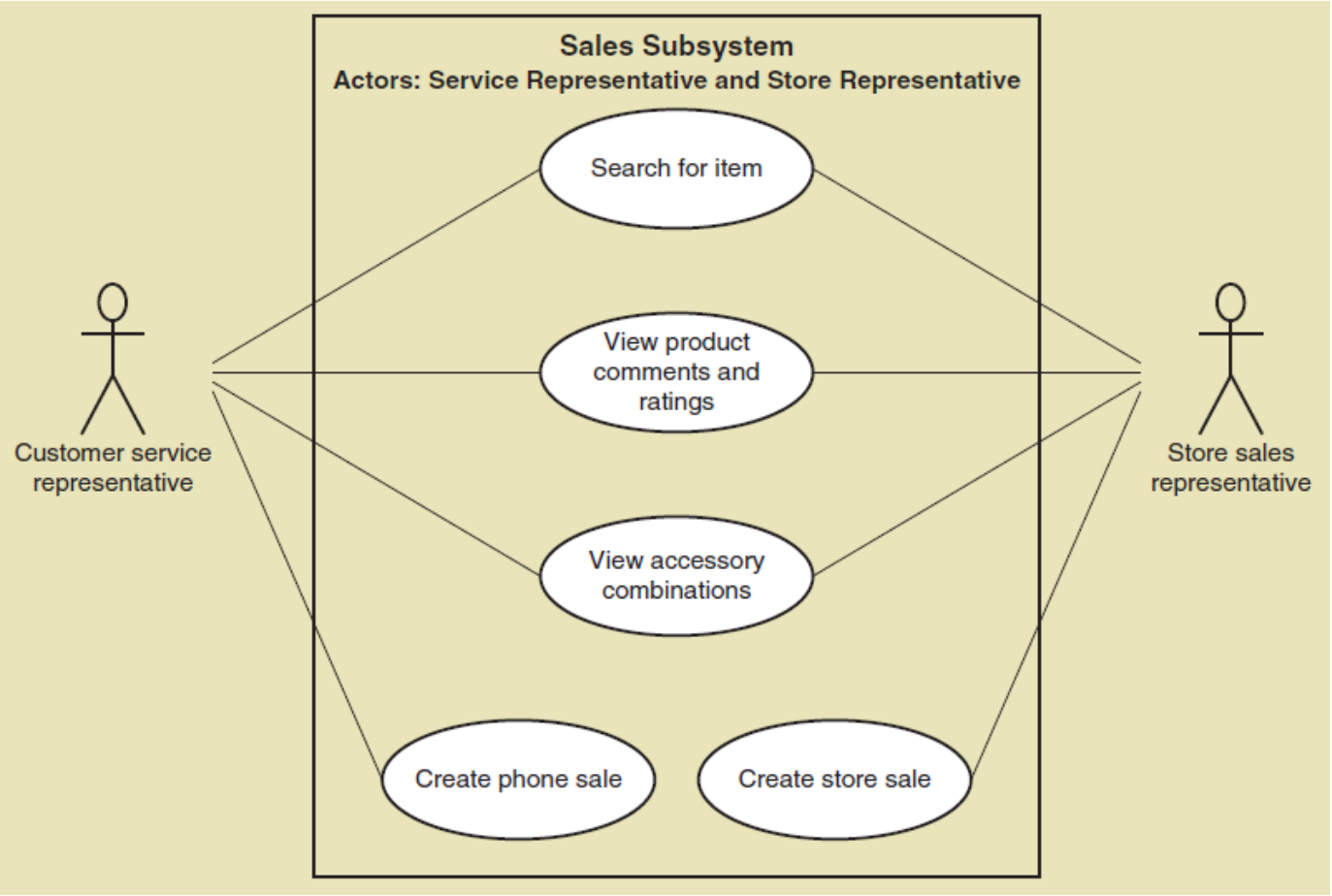


Use Case Diagrams

Draw for actor, such as customer



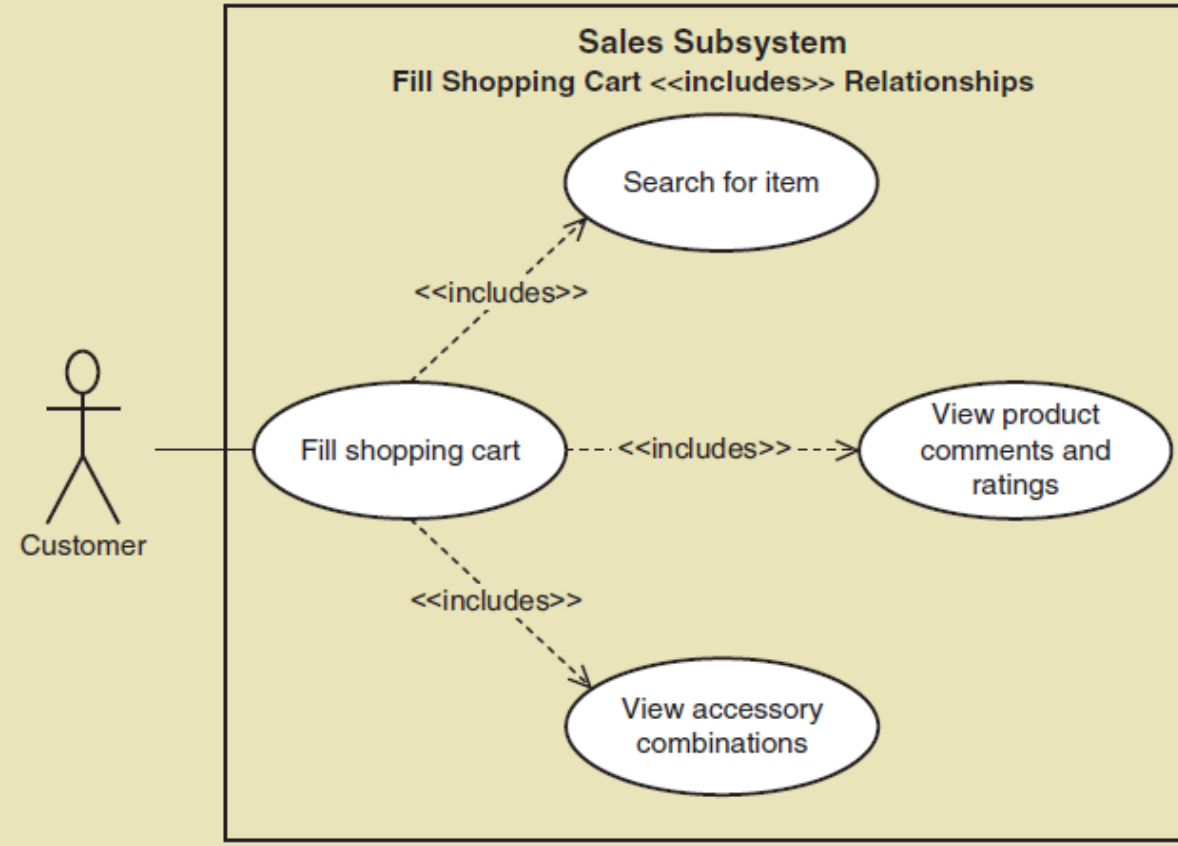
Use Case Diagrams
Draw for internal RMO actors



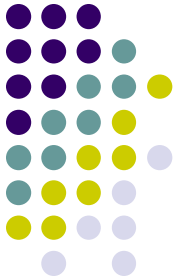
Use Case Diagrams

The <<Includes>> relationship

- A relationship between use cases where one use case is stereotypically included within the other use case— like a called subroutine. Arrow points to subroutine

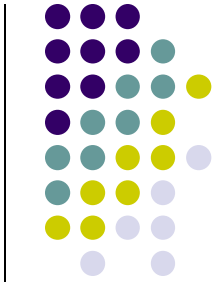


Use Case Diagrams: Steps



1. Identify all the stakeholders and users who would benefit by seeing a use case diagram
2. Determine what each stakeholder or user needs to review in a use case diagram: each subsystem, for each type of user, for use cases that are of interest
3. For each potential communication need, select the use cases and actors to show and draw the use case diagram. There are many software packages that can be used to draw use case diagrams
4. Carefully name each use case diagram and then note how and when the diagram should be used to review use cases with stakeholders and users

Summary



- This chapter is the first of three that focuses on modeling functional requirements as a part of systems analysis
- Use cases are the functions identified, the activities the system carries out usually in response to a user request
- Two techniques for identifying use cases are the user goal technique and the event decomposition technique
- The user goal technique begins by identifying end users called actors and asking what specific goals they have when interacting with the system
- The event decomposition technique begins by identifying events that occur that require the system to respond.

Summary

- Three types of events include external, temporal, and state events
- Brief use case descriptions are written for use cases
- The CRUD technique is used to validate and refine the use cases identified
- The use case diagram is the UML diagram used to show the use cases and the actors
- The use case diagram shows the actors, the automation boundary, the uses cases that involve each actor, and the <<includes>> relationship.
- A variety of use case diagrams are draw depending on the presentation needs of the analysis