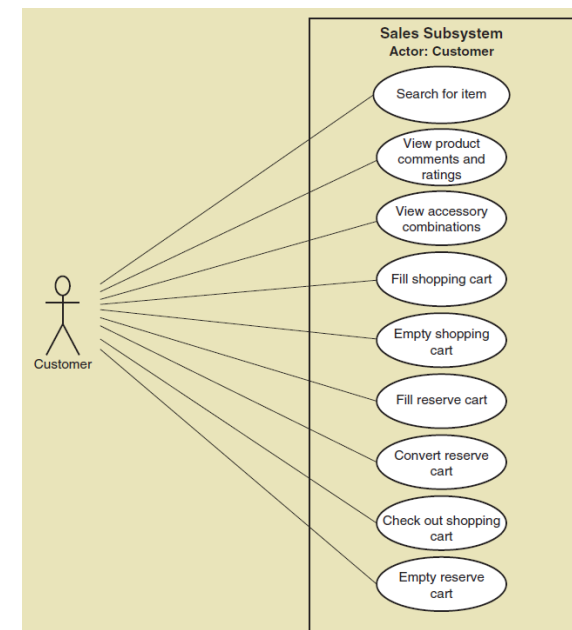
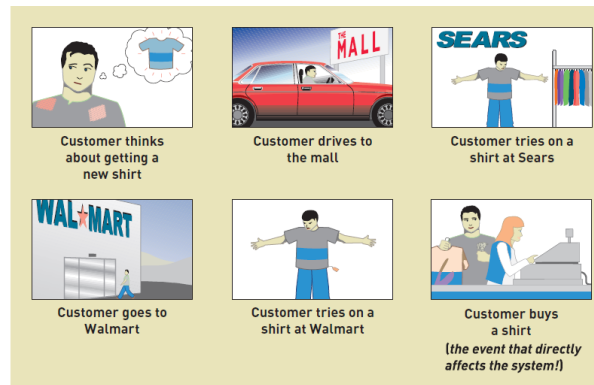


System Analysis Activities-

Identifying User Stories and Use Cases

Dr. Yuehua Wang
yuehua.wang@tamuc.edu



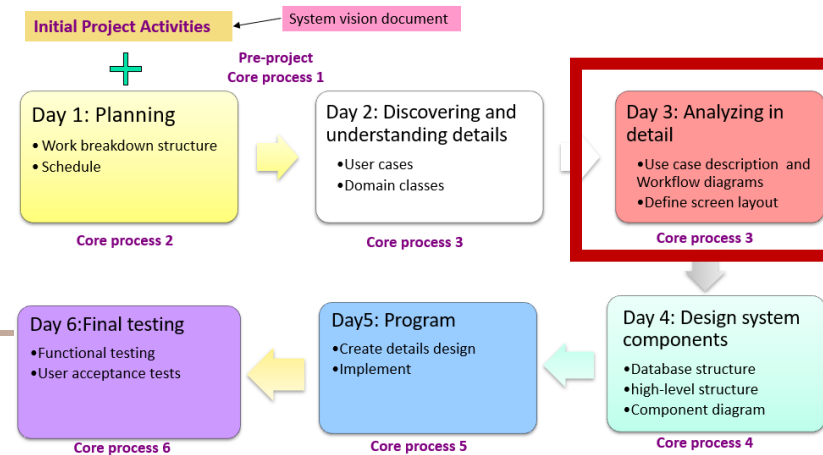
Learning Objectives

- Explain why identifying use cases is the key to defining functional requirements
- Write user stories with acceptance criteria
- 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
- Describe the notation and purpose for the use case diagram

What we have learned

- provided an overview of systems analysis activities, functional and non-functional requirements, modelling, and information gathering techniques
- Will focus on identifying and modelling the key aspect of functional requirements– use cases

Day 3 Activities



- Core Process 3: Discover and Understand Details
 - Do in-depth fact-finding to understand requirements
 - Understand and document the detailed workflow of each use case
- Core Process 4: Design System Components
 - Define the user experience with screens and report sketches

Supplier Information Subsystem

- Use cases:

- Look up supplier
- Enter/update supplier information
- Lookup contact information
- Enter/update contract information

Use Case Diagram

Supplier information subsystem

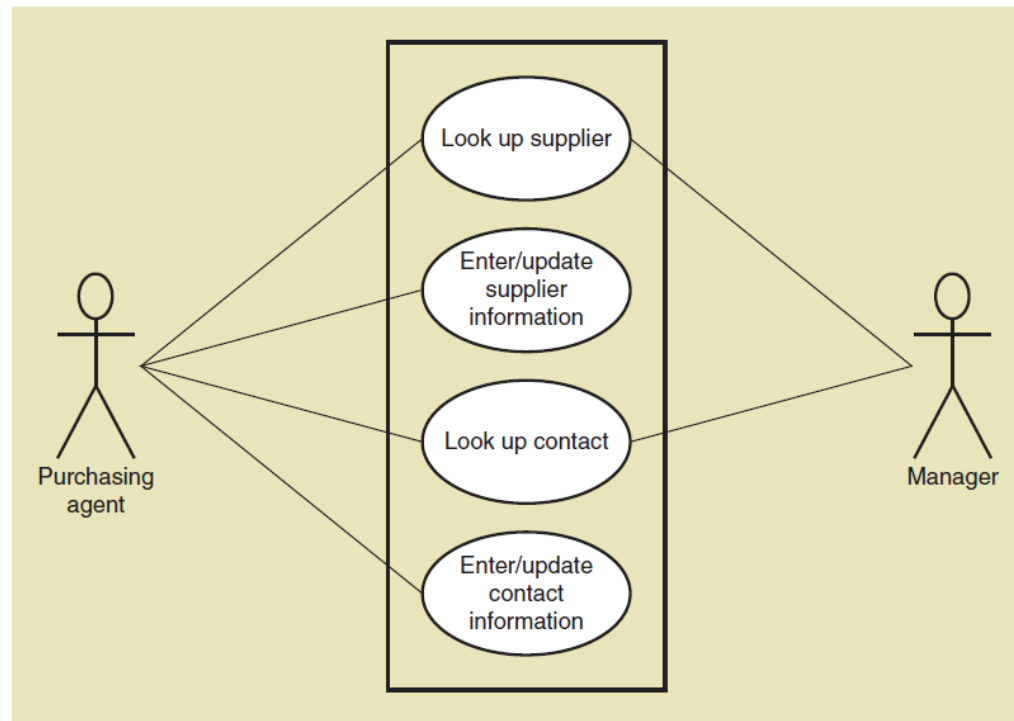


Figure 1-14

Activity Diagram

- *Also called Workflow diagram*
- *Look up supplier* use case

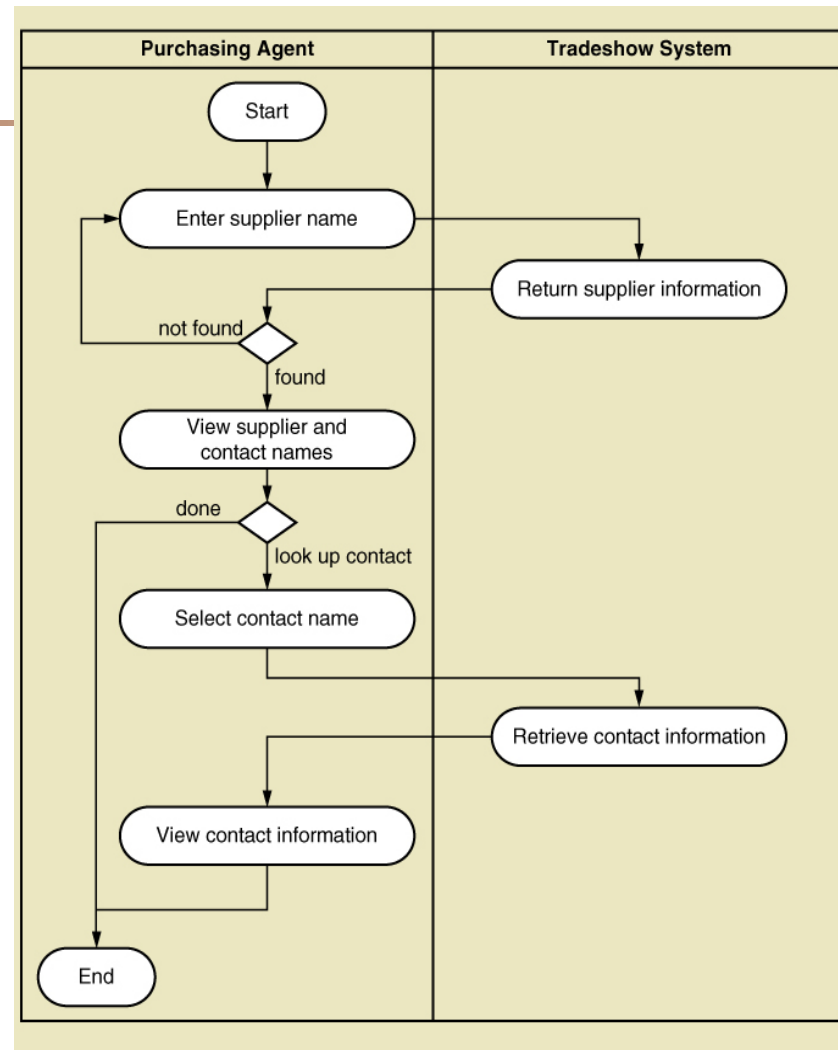


Figure 1-15

User Stories and Use Cases

- identifying user stories and use cases is **a key task**
 - when **defining functional requirements** because these form **the basis for the list of functions** the system needs to carry out.
- Same or different?
 - These two concepts are **similar** in that
 - they focus on **the goals of the user**, and
 - they show **the list of functions** at the appropriate level of detail.
 - They **differ in** the **approach** taken to identify them and in **the amount of detail** that is captured by the analyst.



User Stories and Use Cases

■ User Stories

- are favored by highly Agile system development methodologies, and
- are turned over to the programmer analyst much earlier than use cases are.
- The programmer analyst designs and codes each **user story** to discover needed details.

■ Use Cases

- focus on detailed steps carried out by the user and the system

User Stories



- A User Story is
 - a one short sentence description in the everyday language that states a work-related task done by a user to achieve some goal or result
- The template for a user story description is:

“As a <role played>, I want to <goal or desire> so that <reason/benefit>

- User stories are
 - a basic concept in Agile development because they focus on **simplicity**, **value added**, and **user collaboration**.
 - They document **the functional requirements quickly** and **less formally than traditional requirements modeling** by focusing on *who*, *what*, and *why* for each function.
 - **The users and stakeholders** are responsible for identifying the user stories.

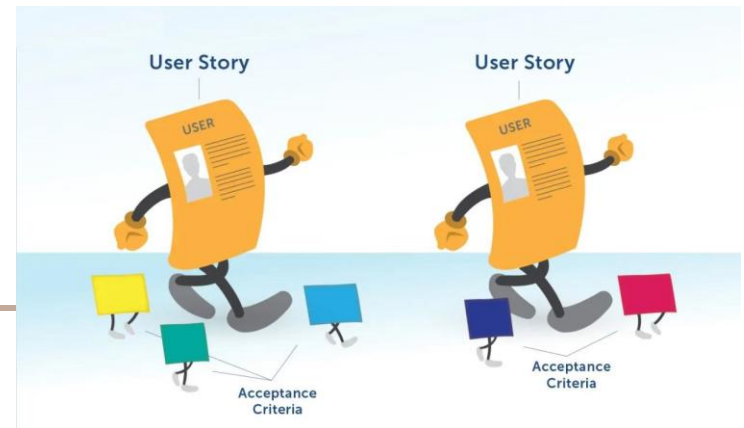


- The template for a user story description is:

"As a <role played>, I want to <goal or desire> so that <reason/benefit>"

- Examples:
 - some user stories for a bank teller might be:
 - "As a teller, I want to make a deposit to quickly serve more customers."
 - "As a teller, I want to balance the cash drawer to assure there were no errors."
 - As a customer of the bank using an ATM machine, some user stories might be:
 - "As a bank customer, I want to withdraw cash and feel confident the stack of cash I get is the correct amount."
 - "As a bank customer, I want to deposit a check and feel confident the deposit is recorded correctly."

User Stories



- **Acceptance Criteria**
 - identify the features that must be present for the user at the completion of the task
 - Focus on functionality, not on features or user-interface design
- The programmer analyst
 - uses the acceptance criteria to **clarify the expectations of the user** and to **verify the user is looking at the user story** at an appropriate level of analysis.
- When the user story is implemented and refined,
 - the acceptance criteria are used for testing.



How the client described it



How the designer designed it



How the engineer developed it



What the customer really wanted

Tree swing cartoon for product development

Sample User Story 1-Bank teller

User Story

As a teller, I want to make a deposit to quickly serve more customers.

Acceptance Criteria:

- 1. Customer lookup must be by name or by account number.*
- 2. Nice to display photo and signature of customer.*
- 3. Any check hold requirements must be indicated.*
- 4. Current balance and new balance must be displayed.*

Sample User Story 2- Shipping clerk

User Story

As a shipping clerk, I want to ship an order as accurately as possible as soon as the order details are available.

Acceptance Criteria:

- 1. Available order details must pop up on the screen when available.*
- 2. Portable display and scan device would cut time in half.*
- 3. Sort the items by bin location.*
- 4. Indicate number of items in stock for each item and mark backorder for those not available.*
- 5. Recommend shipper based on weight, size, and location.*
- 6. Print out shipping label for selected shipper.*

User Story vs. User Case

- | | |
|---|--|
| <ul style="list-style-type: none">■ Short description<ul style="list-style-type: none">■ Role■ Goal■ Reason/benefit■ Acceptance Criteria■ In any language | <ul style="list-style-type: none">■ Short description with more details■ More concrete■ User flow or interaction■ UML |
|---|--|

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*”

Name each use
case using
Verb-Noun

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 order Track shipment Create item return

RMO CSMS: RMO Consolidated Sales and Marketing System

User Goal Technique: 8 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

Name each use
case using
Verb-Noun

User Goal Technique

Some RMO CSMS Users and Goals

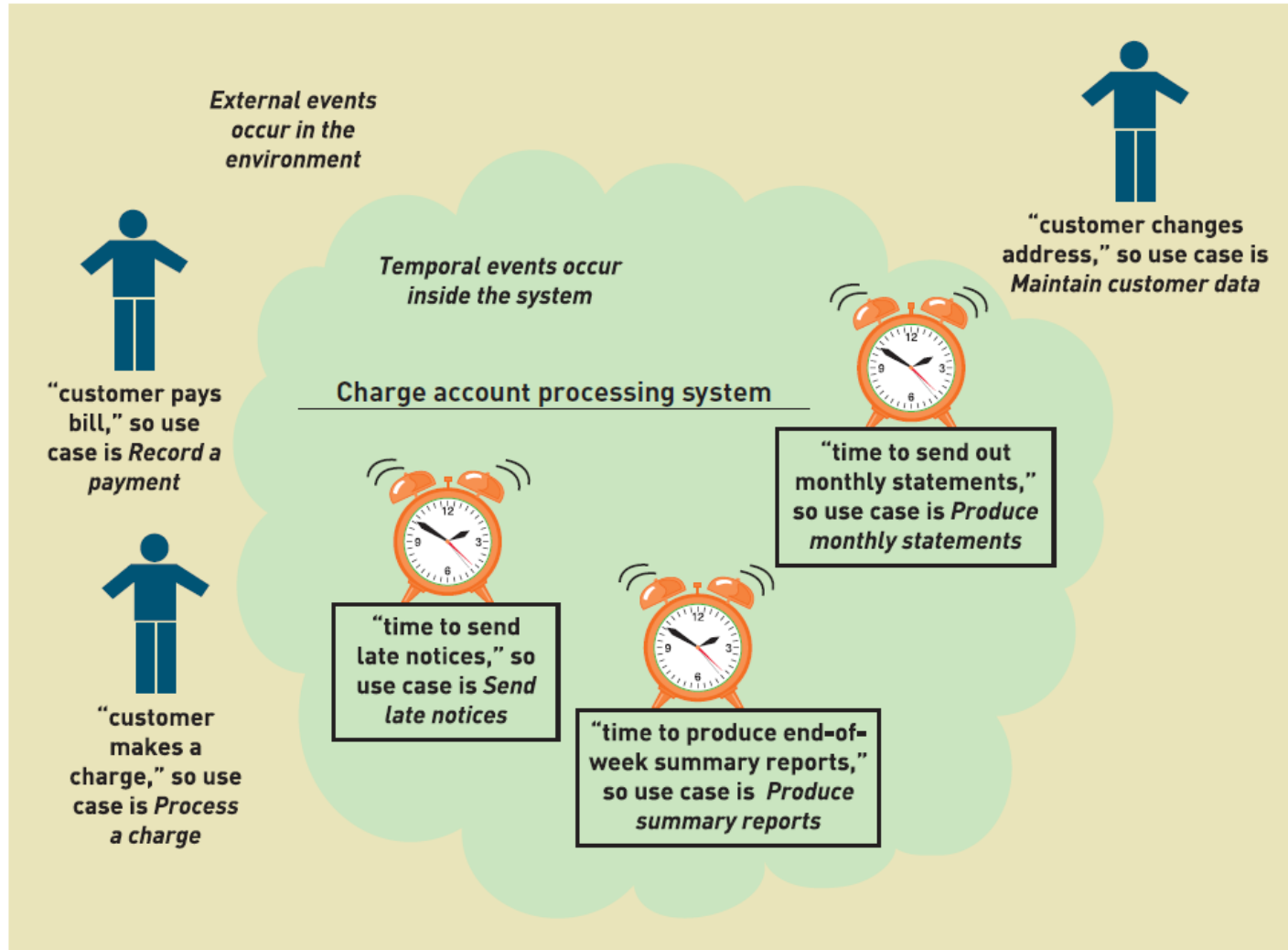
User	User goal and resulting use case
Potential customer	Search for item Fill shopping cart View product rating and comments
Marketing manager	Add/update product information Add/update promotion Produce sales history report
Shipping personnel	Ship order Track shipment Create item return

RMO CSMS: RMO Consolidated Sales and Marketing System

Event Decomposition Technique

- More Comprehensive and Complete Technique
 - Identify the events that occur to which the system must respond.
 - Using events to define functional requirements was first emphasized for real-time systems in the early 1980s.
 - Event
 - something that occurs at a specific time and place, can be described, and should be remembered by the system
 - For each event, name a use case (verb-noun) that describes what the system does when the event occurs

Events and Use Cases

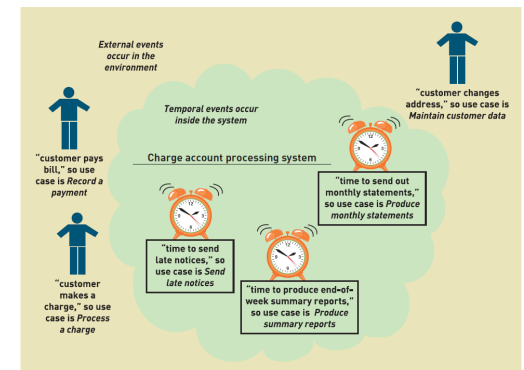


Types of Events

- 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 (internal event)
 - an event that occurs when something happens inside the system that triggers some process
 - reorder point is reached for inventory item

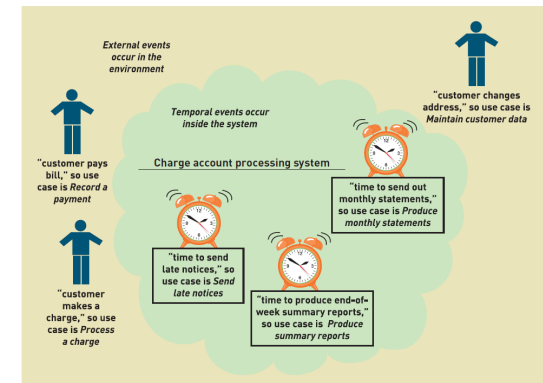
External Event Checklist

- 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

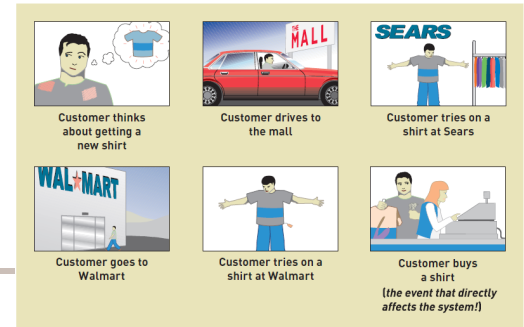


Temporal Event Checklist

- 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



Identifying Events



- Defining the events that affect a system is not always easy,
 - some guidelines can help an analyst think through the process.
- Guidelines:
 - Events Versus Prior Conditions and Responses
 - Consider a customer buying a shirt from a retail store
 - The Sequence of Events: Tracing a Transaction's Life Cycle

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!)*

Finding the actual event that affects the system



Customer thinks
about getting a
new shirt



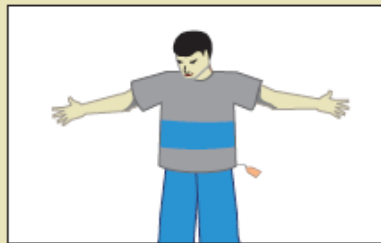
Customer drives to
the mall



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!)*

The system is not affected until the customer is in the store.

Tracing a sequence of transactions resulting in many events



Customer requests a catalog



Customer wants to check item availability



Customer places an order



Customer changes or cancels an order



Customer wants to check order status



Customer updates account information



Customer returns the item

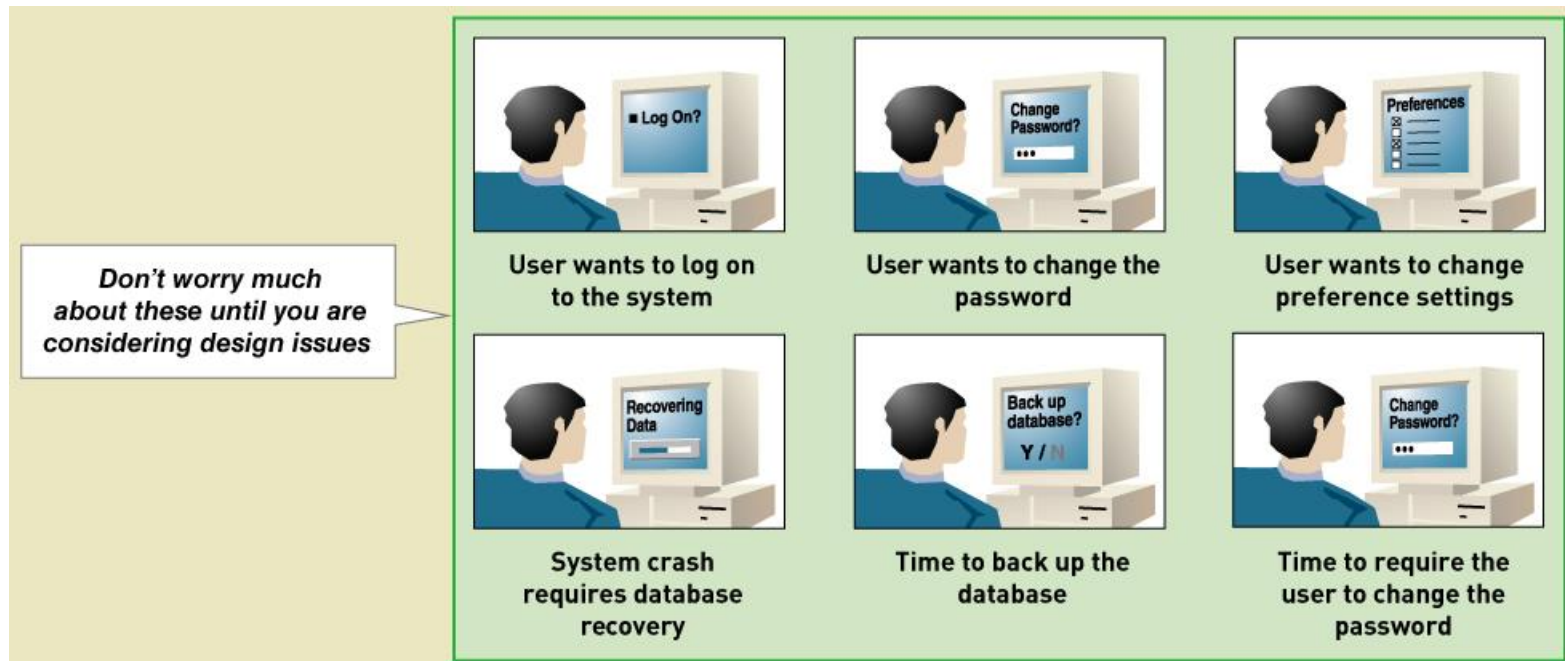
Ridgeline Mountain Outfitters' new CSMS, the analyst can think through all the possible transactions that might result from one new customer

Note

- Sometimes, the analyst is concerned about events that are important to the system, but do not directly concern users or transactions.
- Such events typically involve design choices or system controls.
- During analysis, the analyst should temporarily ignore these events.
- However, they are important later for design.

Perfect Technology Assumption

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



Event Decomposition Technique: 7 Specific Steps

1. Consider the external events in the system environment that require a response from the system by using the checklist

External events to look for include:

- √ External agent wants something resulting in a transaction
- √ External agent wants some information
- √ Data changed and needs to be updated
- √ Management wants some information

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

Temporal events to look for include:

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

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

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 in the Ridgeline Mountain Outfitters

- The RMO CSMS involves a variety of use cases
 - It is important to recognize that this list of uses cases will continue to evolve as the project progresses.
- 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
<i>Create customer account</i>	User/actor enters new customer account data, and the system assigns account number, creates a customer record, and creates an account record.
<i>Look up customer</i>	User/actor enters customer account number, and the system retrieves and displays customer and account data.
<i>Process account adjustment</i>	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

Figure 3-10a

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

Figure 3-10b

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 partner credits	Customer
View "mountain bucks"	Customer
Transfer "mountain bucks"	Customer

Figure 3-10c

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

Figure 3-10d

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 promotional partner activity report	Management, marketing

Figure 3-10e

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

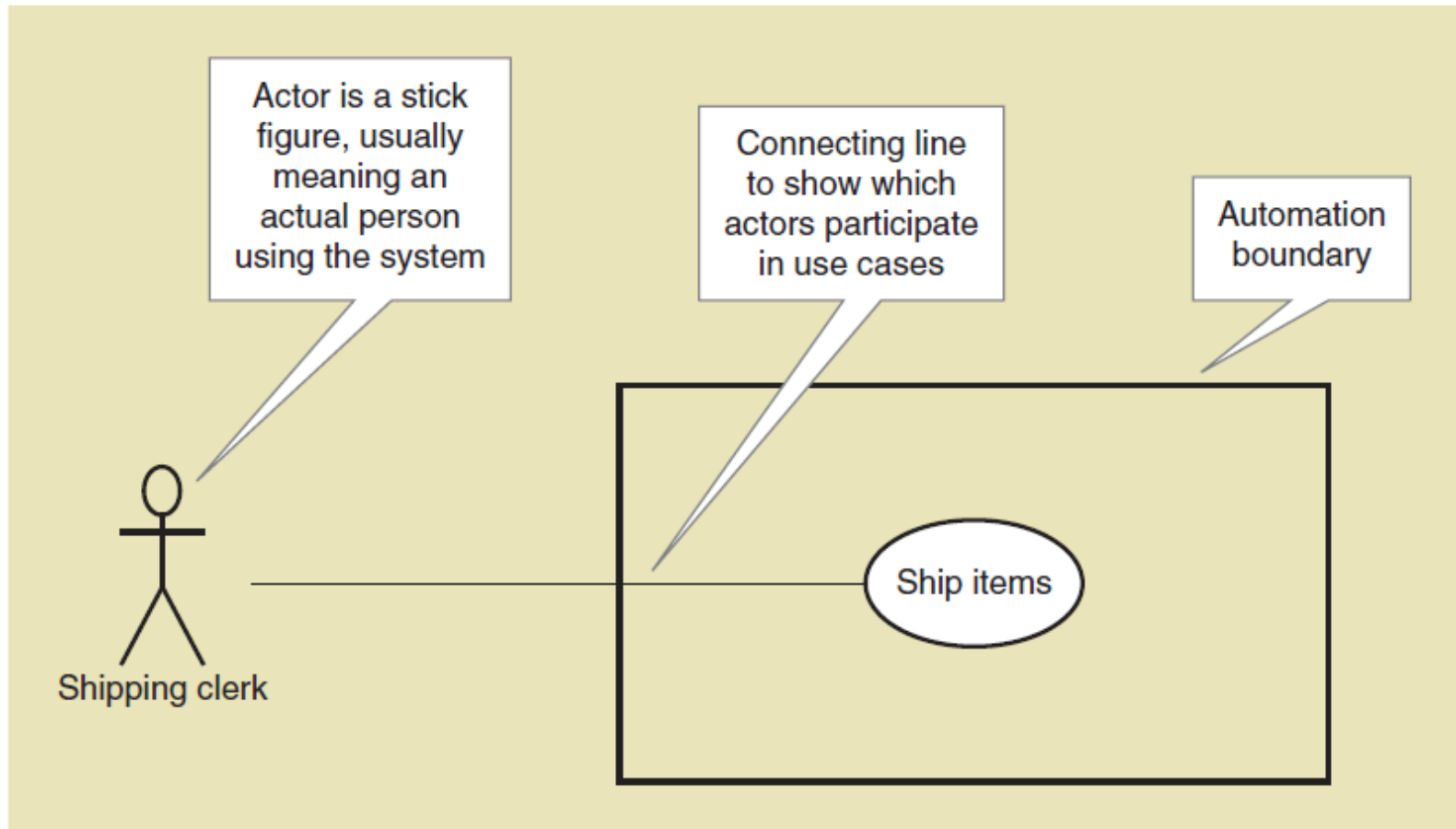
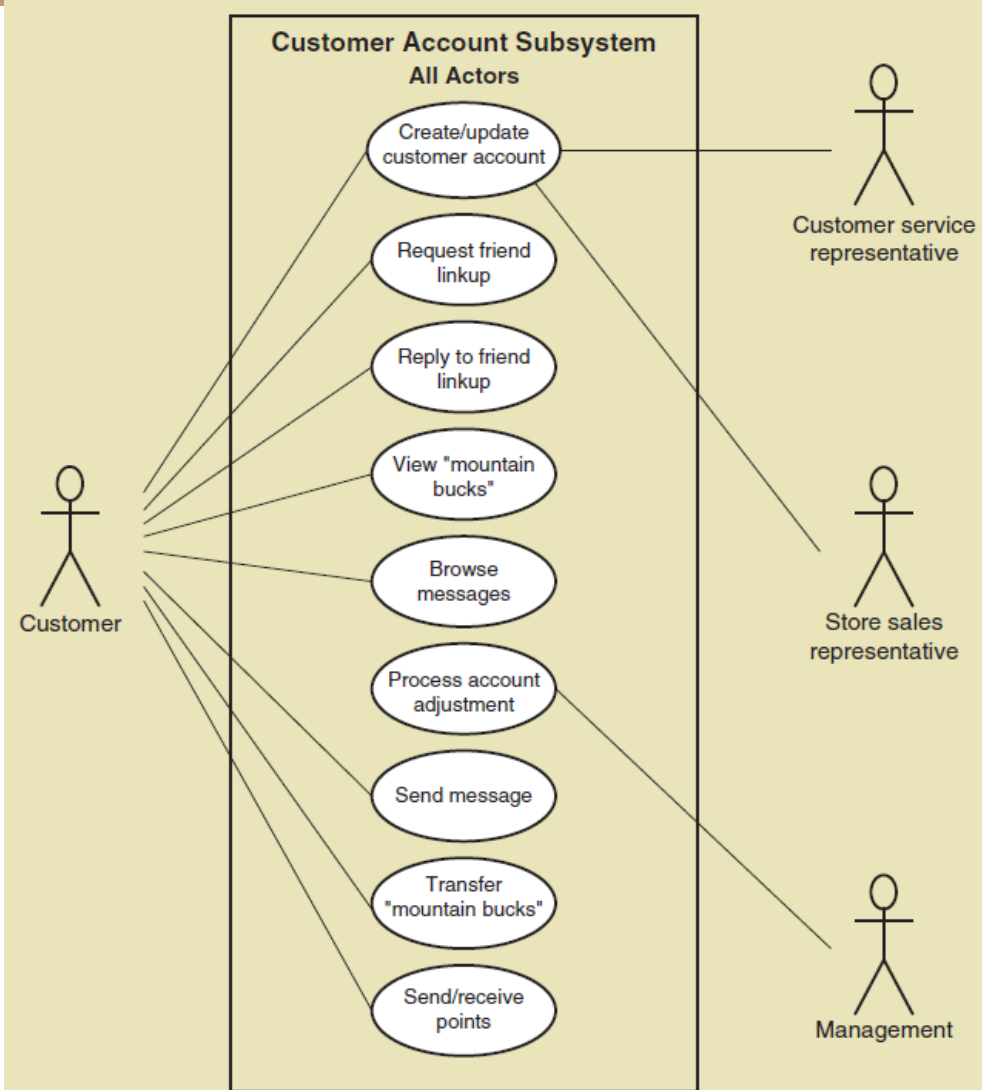


Figure 3-11 shows the basic parts of a use case diagram.

Use Case Diagrams

- Draw for each subsystem

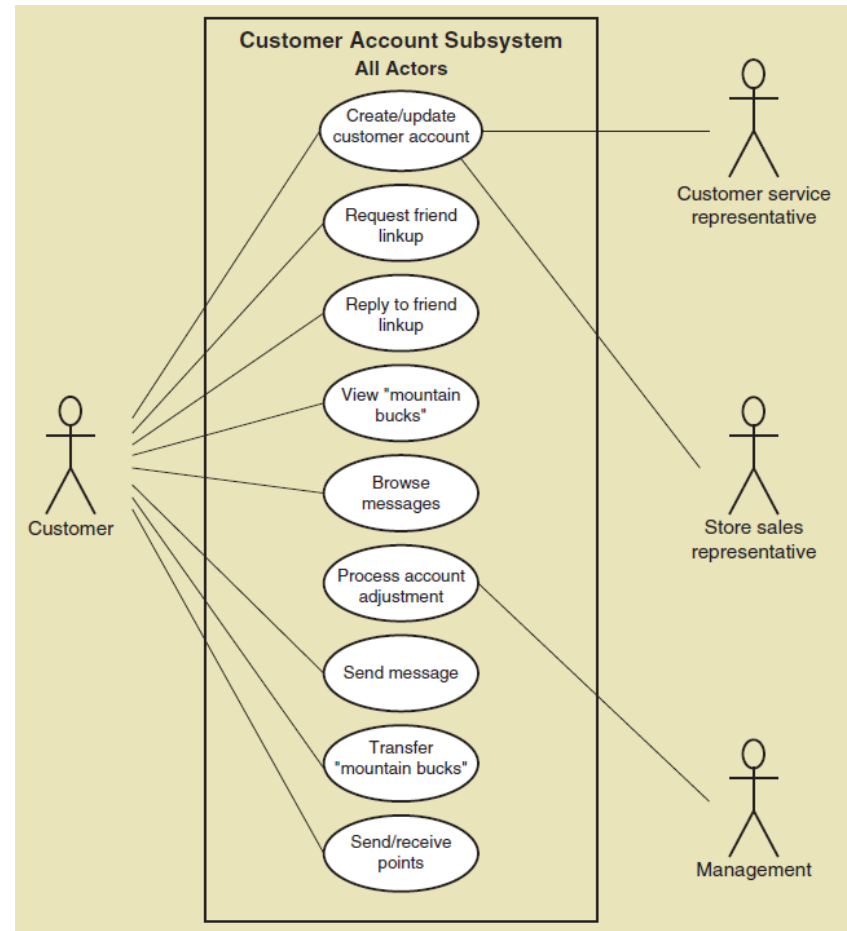


Use Case Diagrams

- Draw for each subsystem

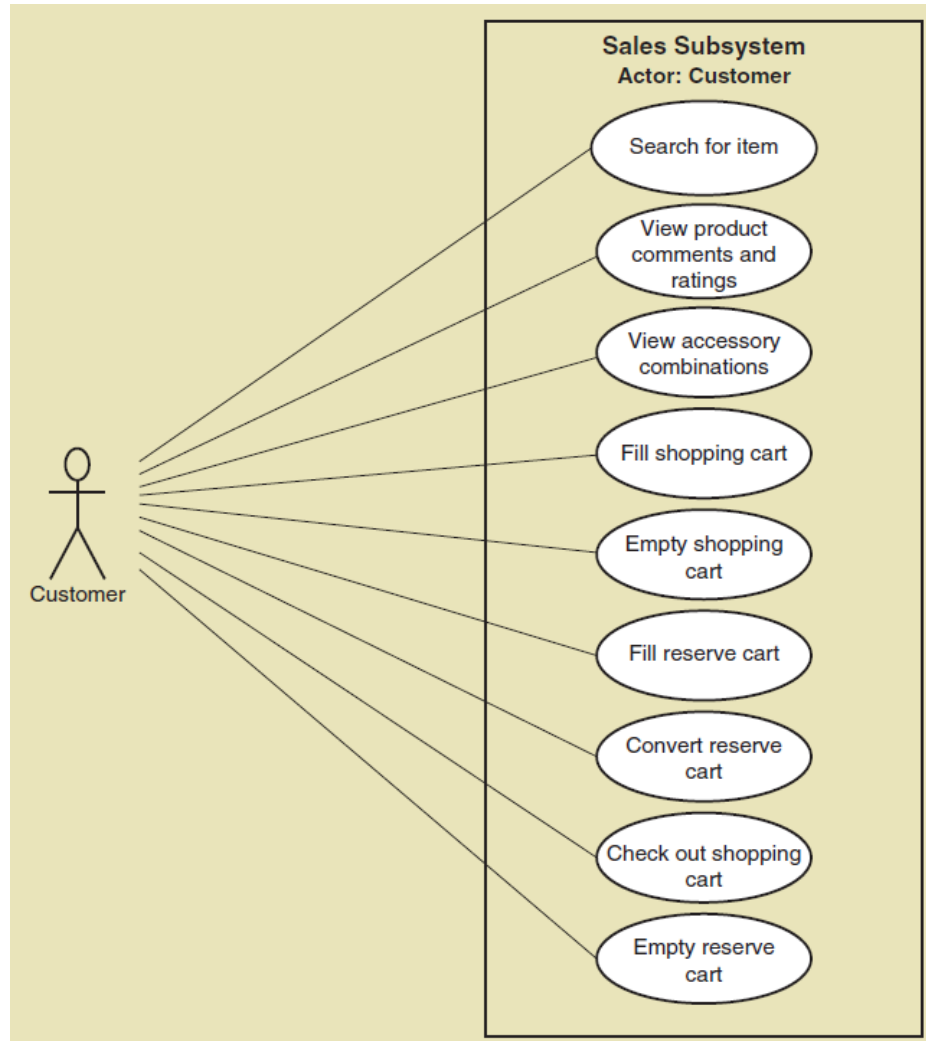
CSMS Customer Account Subsystem	
Use cases	Users/actors
Create/update customer account	Customer, customer service representative, store sales representative
Process account adjustment	Management
Send message	Customer
Browse messages	Customer
Request friend linkup	Customer
Reply to linkup request	Customer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

Figure 3-10c



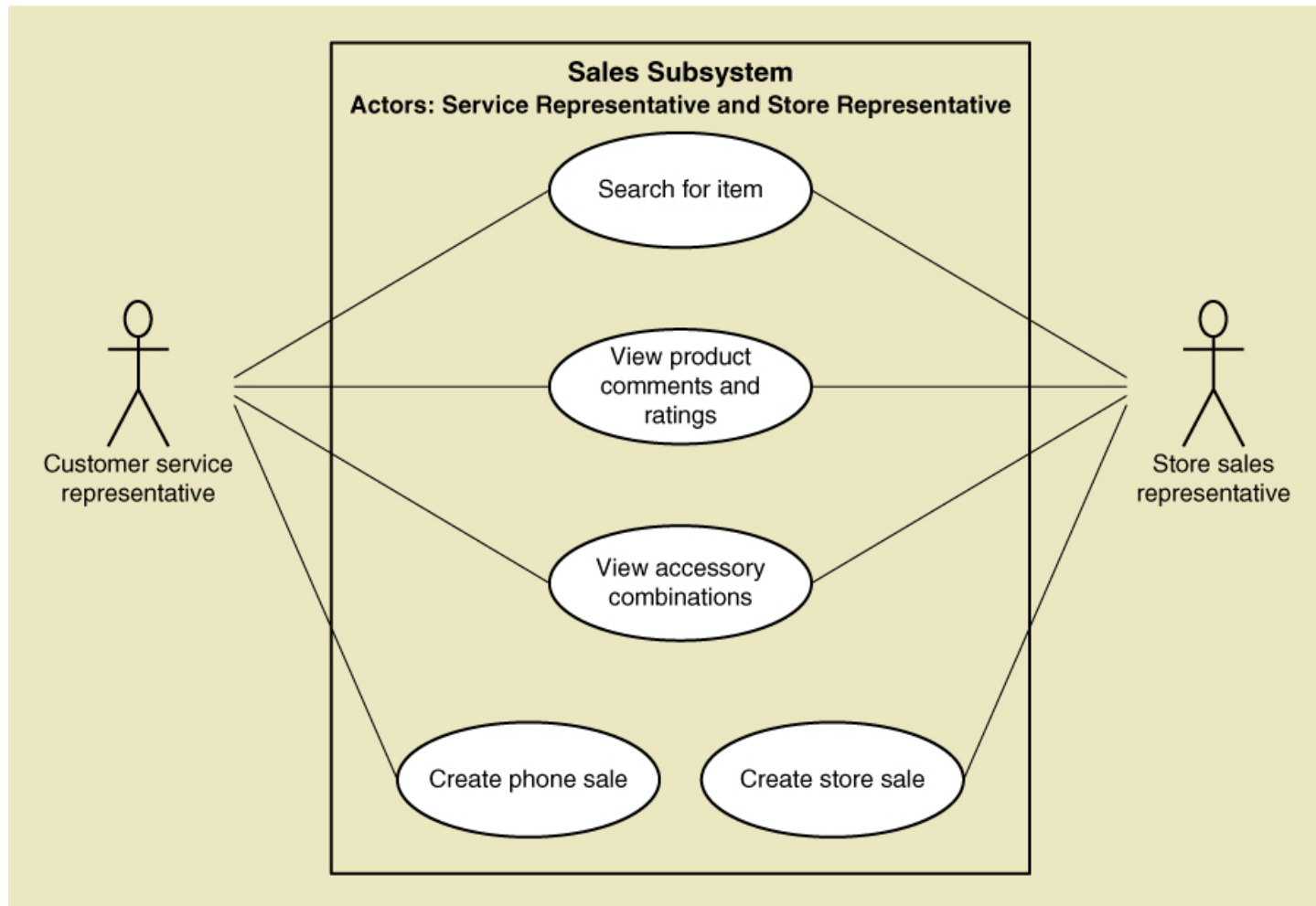
Use Case Diagrams

- Draw for a single actor, such as customer



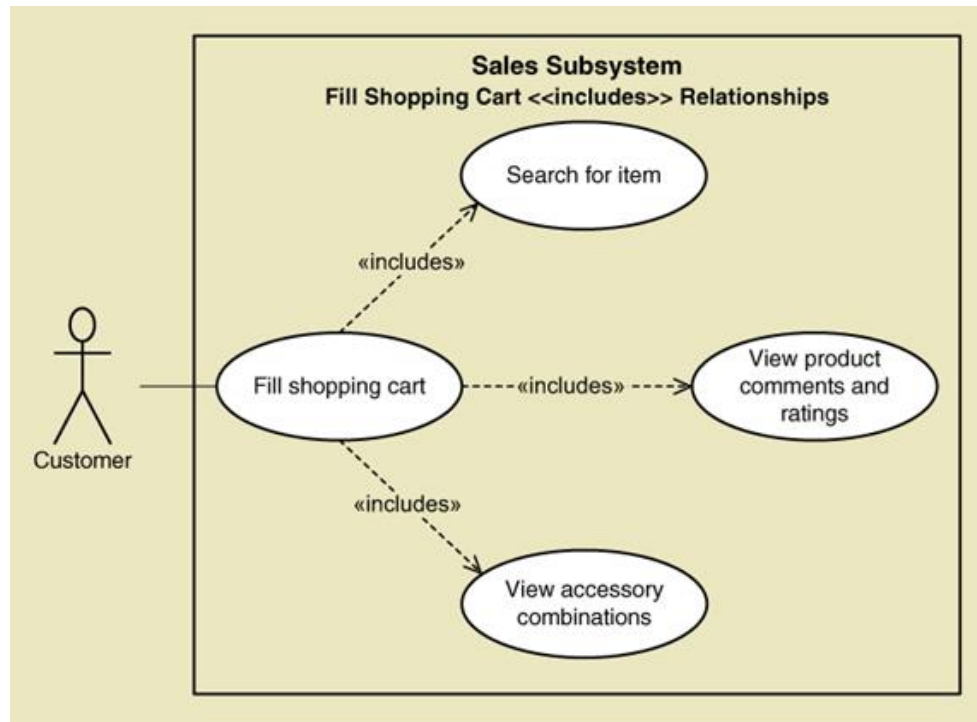
Use Case Diagrams

Draw for internal RMO actors



Use Case Diagrams— The <<Includes>> relationship

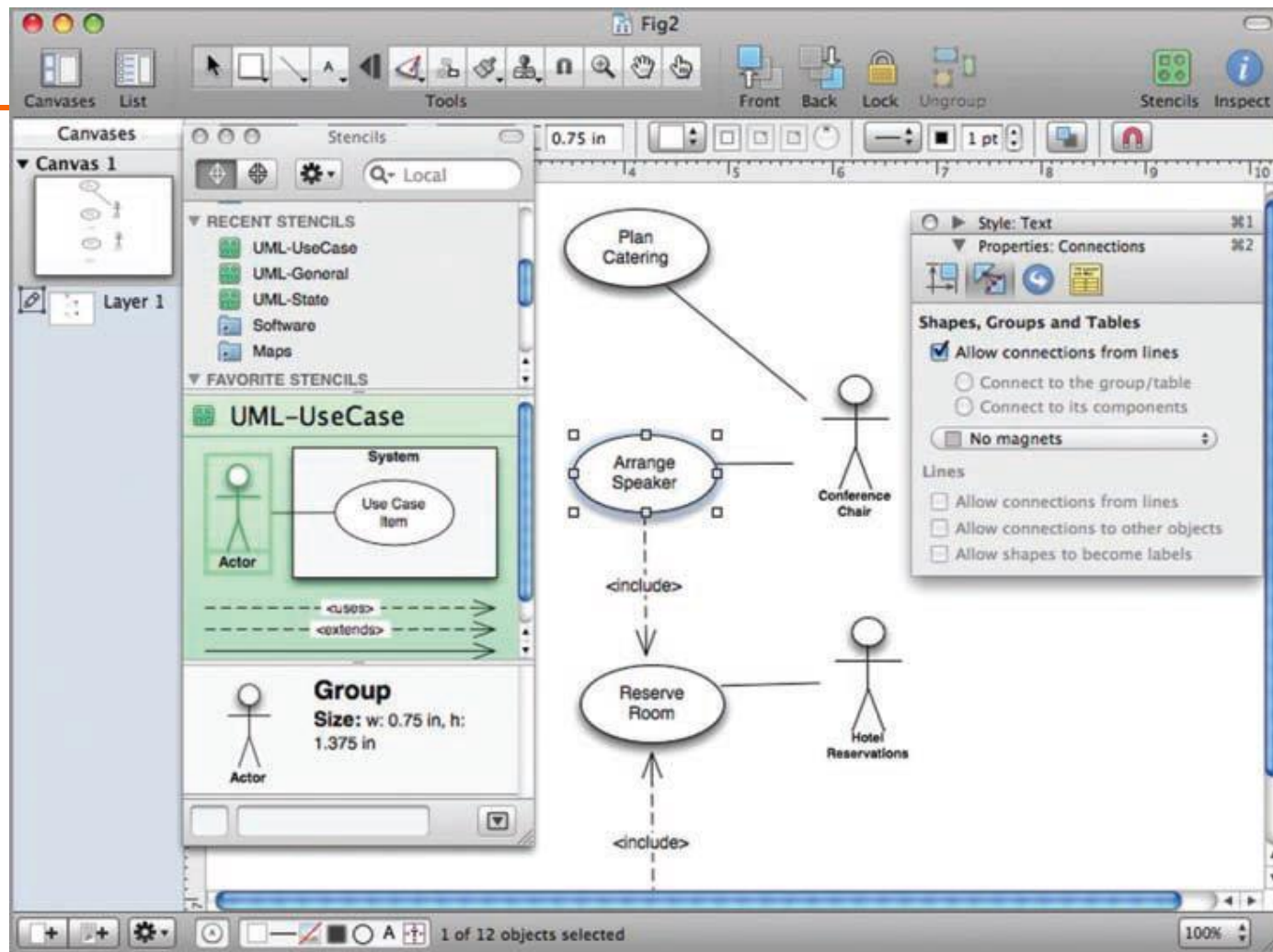
- A relationship between use cases where one use case is stereotypically included within the other use case— like a called subroutine. Arrow points to subroutine



Developing Use Case Diagrams: 4 Steps

- Identify **all the stakeholders and users** who would benefit by seeing a use case diagram
- 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
- 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
- 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

-
- Microsoft Visio makes it easy for a systems analyst to draw E-R diagrams as well as most of the other diagrams found in this book, but it is available only for PCs. Mac users have an alternative, OmniGraffle Professional. OmniGraffle is easier to use than Microsoft Visio because its drag-anddrop interface is smoother and more intuitive.



Summary

- This lecture 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 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

A green rectangular road sign with rounded corners and a white border, mounted on two wooden posts. The word "Ethics" is written in large, white, sans-serif capital letters. The sign is tilted slightly to the right. The background is a bright blue sky with scattered white clouds.

Ethics



A question of ethics

- Your supervisor manages the corporate office where you work as a systems analyst.
- Several weeks ago, after hearing rumors of employee dissatisfaction, he asked you to create a survey for all IT employees. After the responses were returned and tabulated, he was disappointed to learn that many employees assigned low ratings to morale and management policies.
- This morning he called you into his office and asked whether you could identify the departments that submitted the lowest ratings. No names were used on the individual survey forms. However, with a little analysis, you probably could identify the departments because several questions were department-related.

-
- Now you are not sure how to respond. The expectation was that the survey would be anonymous.
 - Even though no individuals would be identified,
 - **Would it be ethical to reveal which departments sent in the low ratings?**
 - **Would your supervisor's motives for wanting this information matter?**





The motives don't matter