Systems Analysis & Design

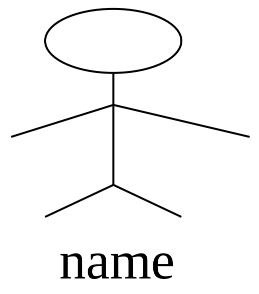
Use Case Diagram

Using Use Case Diagrams

- Use case diagrams are used to visualize, specify, construct, and document the (intended) behavior of the system, during requirements capture and analysis.
- Provide a way for developers, domain experts and end-users to Communicate.
- Serve as basis for testing.
- Use case diagrams contain use cases, actors, and their relationships.

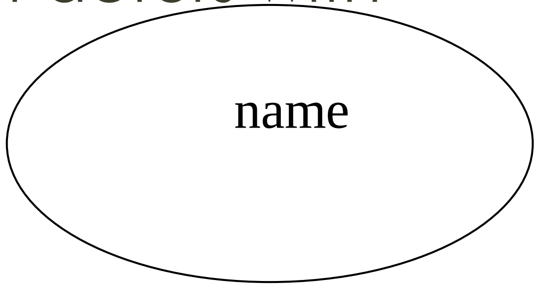
Actors

- An actor represents a set of roles that users of use case play when interacting with these use cases.
- Actors can be human or automated systems.
- Actors are entities which require help from the system to perform their task or are needed to execute the system's functions.
- Actors are not part of the system.



Use Case

- Use cases specify desired behavior.
- A use case is a description of a set of sequences of actions, including variants, a system performs to yield an observable result of value to an actor.
- Each sequence represent an interaction of actors with the system.

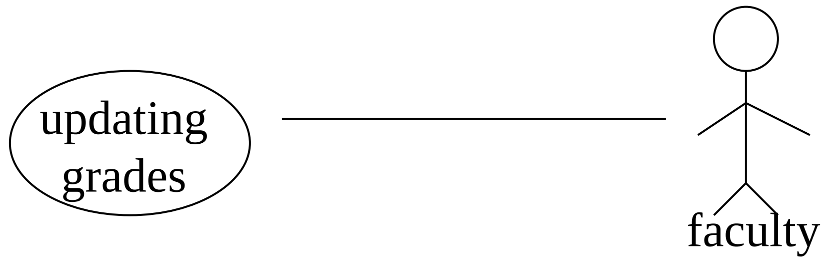


Specifying the Behavior of a Use Case

- Describing the flow of events within the use case.
- Can be done in natural language, formal language or pseudo-code.
- Includes: how and when the use case starts and ends; when the use case interacts with actors and what objects are exchanged; the basic flow and alternative flows of the behavior.

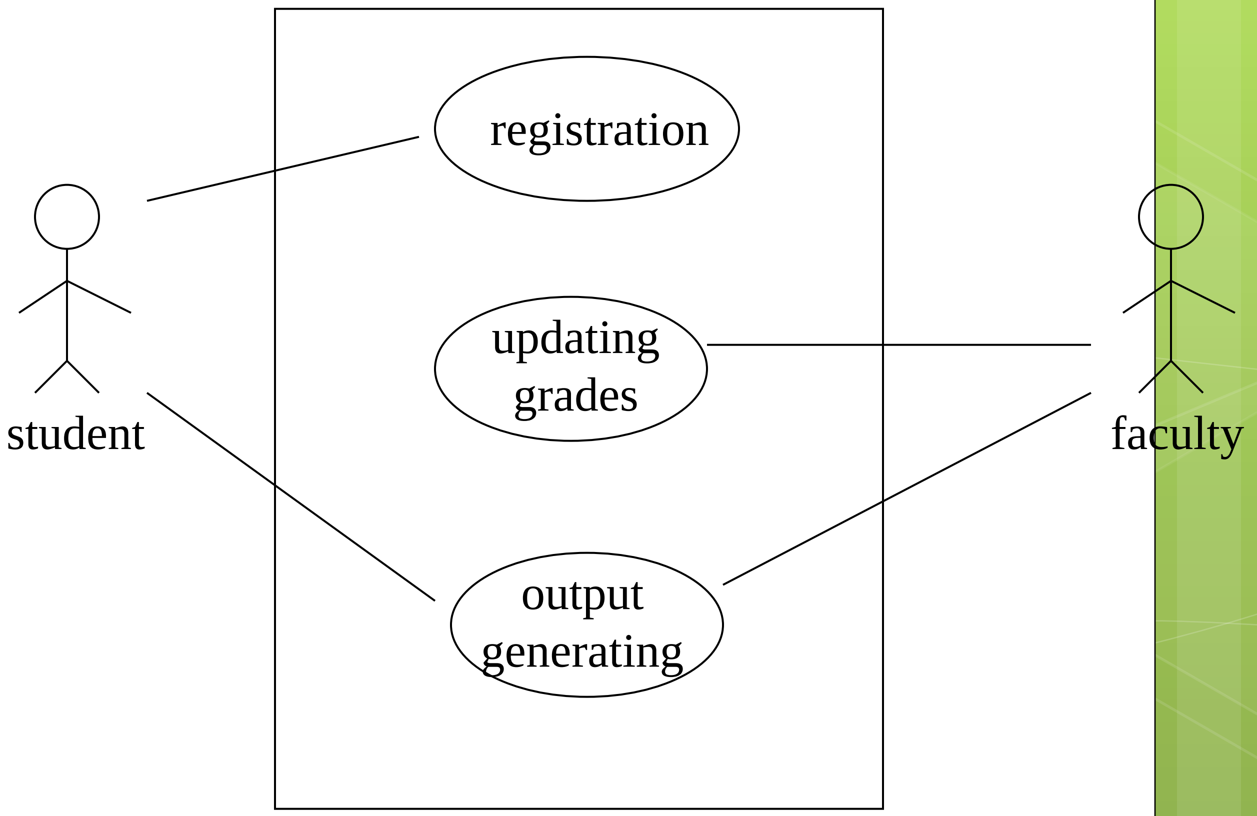
Relationships between Use Cases and Actors

- Actors may be connected to use cases by associations, indicating that the actor and the use case communicate with one another using messages.



Use Cases and Actors

- From the perspective of a given actor, a use case does something that is of value to the actor, such as calculate a result or change the state of an object.
- The Actors define the environments in which the system lives

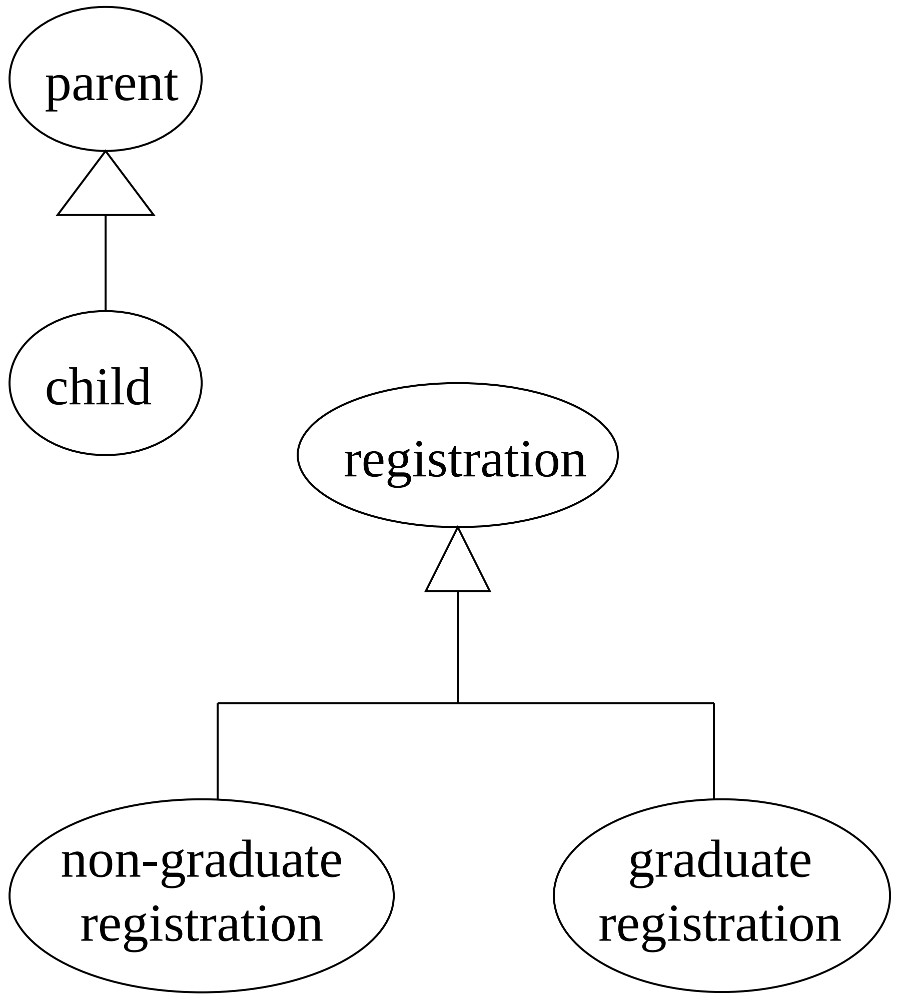


Relationships between Use Cases

1. **Generalization** - use cases that are specialized versions of other use cases.
2. **Include** - use cases that are included as parts of other use cases. Enable to factor common behavior.
3. **Extend** - use cases that extend the behavior of other core use cases. Enable to factor variants.

1. Generalization

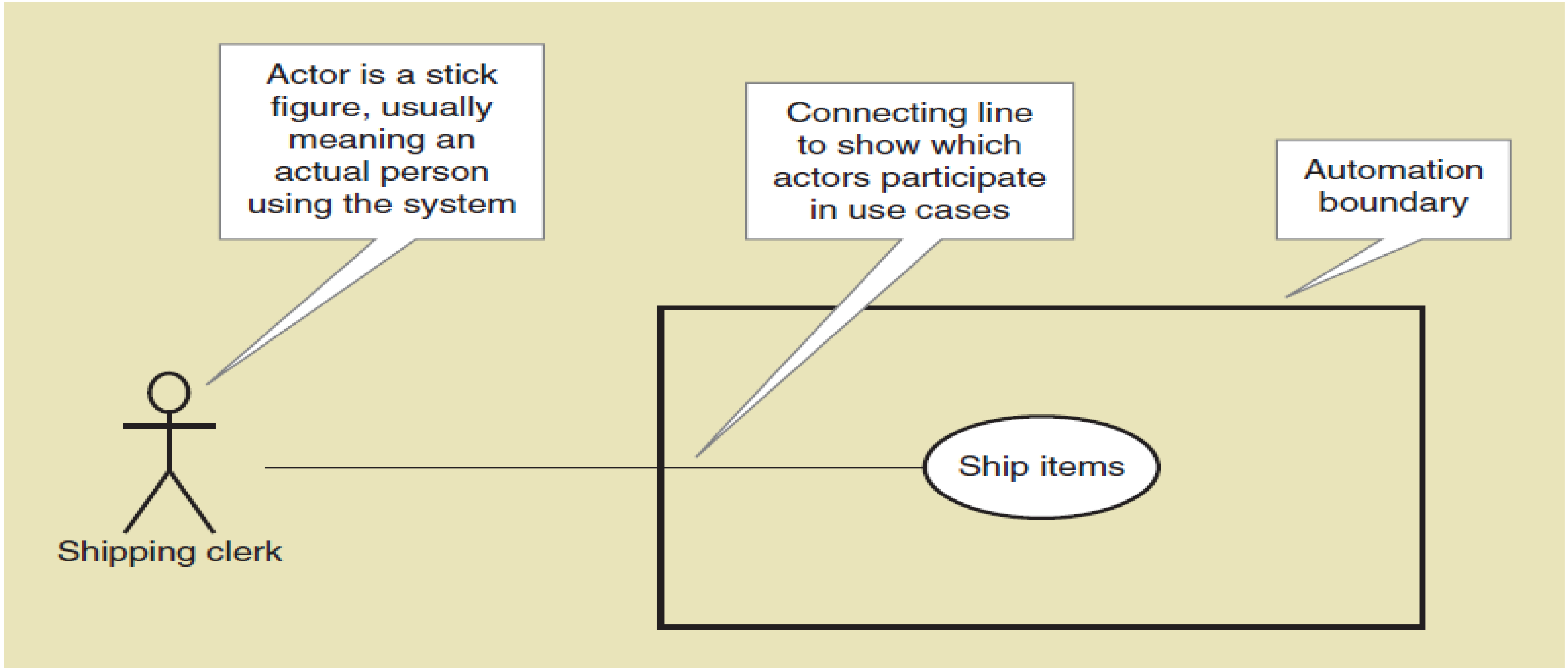
- The child use case inherits the behavior and meaning of the parent use case.
- The child may add to or override the behavior of its parent.



Use Case Diagrams

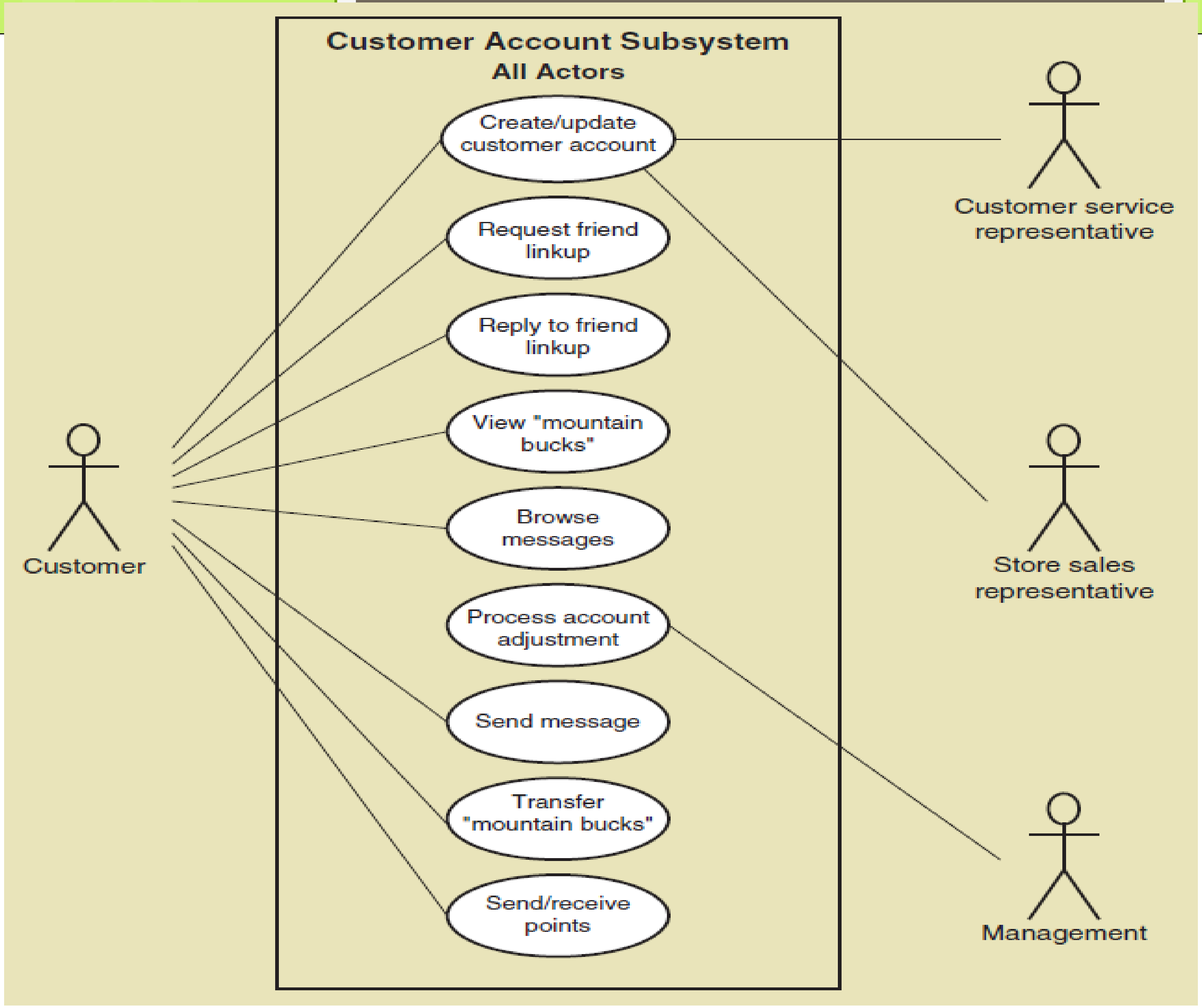
- Use case diagram— a UML model used to graphically show uses cases and their relationships to actors
- Recall UML is Unified Modeling Language, the standard for diagrams and terminology for developing information systems
- Actor is the UML name for a end user
- **Automation boundary**— the boundary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

Use Case Diagrams Symbols


Actor is a stick figure, usually meaning an actual person using the system
Connecting line to show which actors participate in use cases
Automation boundary
Shipping clerk
Ship items

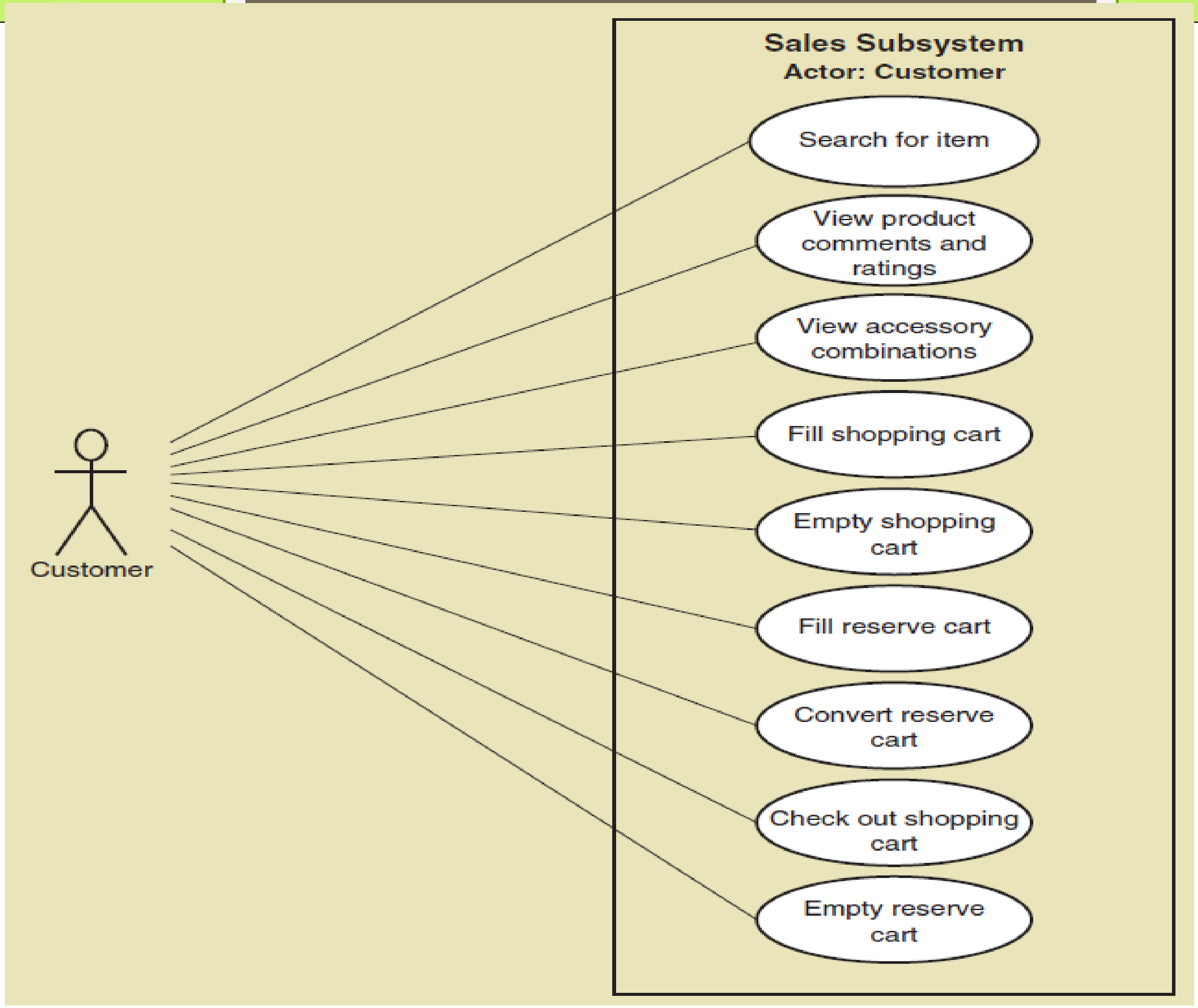
Use Case Diagrams

Draw for each subsystem

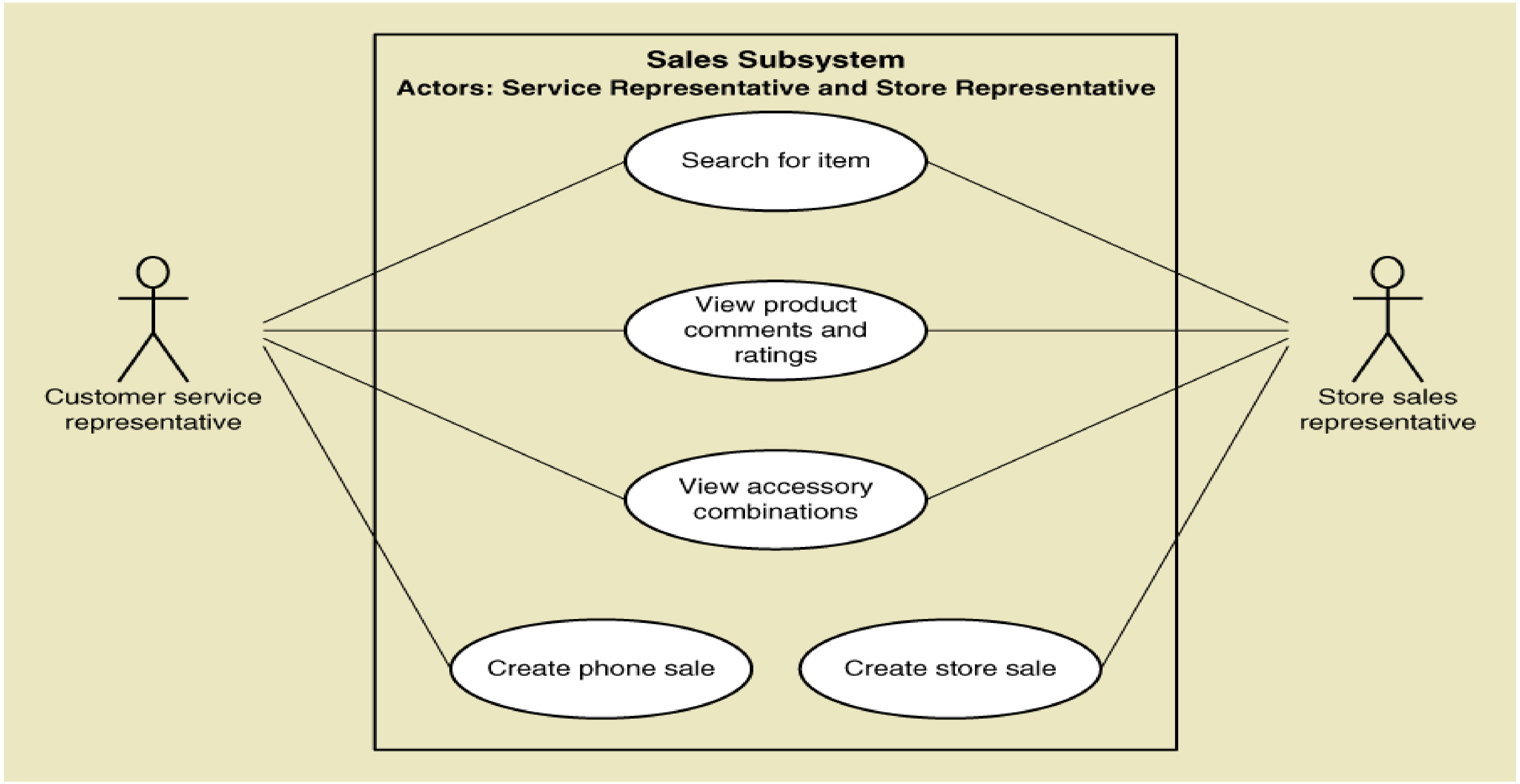

Customer Account Subsystem
All Actors
Create/update customer account
Request friend linkup
Reply to friend linkup
View "mountain bucks"
Browse messages
Process account adjustment
Send message
Transfer "mountain bucks"
Send/receive points
Customer
Customer service representative
Store sales representative
Management

Use Case Diagrams

Draw for a single actor, such as customer

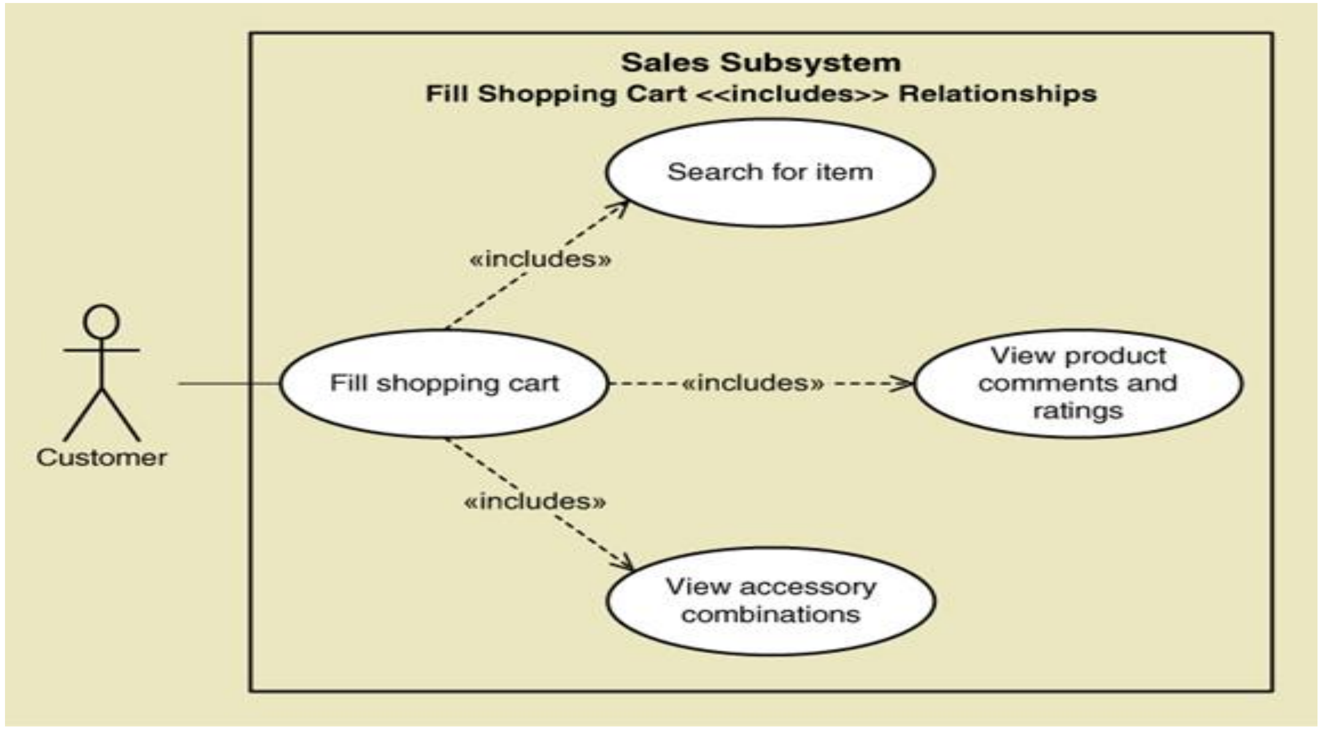


Use Case Diagrams
Draw for internal RMO actors


Sales Subsystem
Actors: Service Representative and Store Representative
Search for item
View product comments and ratings
View accessory combinations
Create phone sale
Create store sale
Customer service representative
Store sales representative

Use Case Diagrams— The <<Includes>> relationship

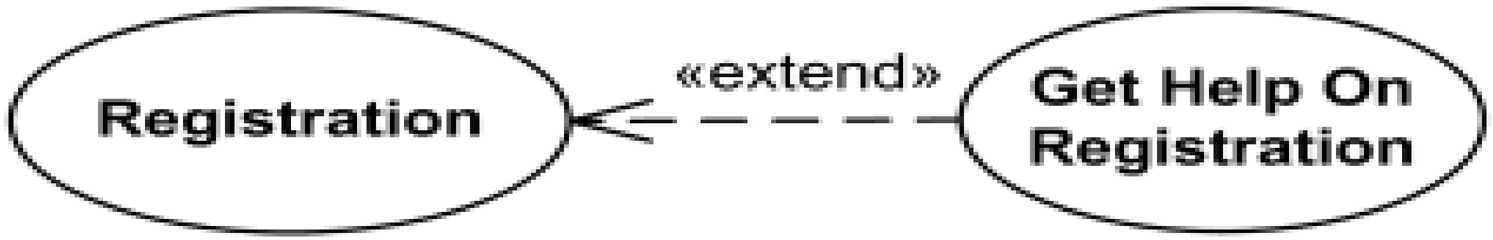
- A relationship between use cases where one use case is stereotypically included within the other use case— like a called subroutine. Arrow points to subroutine



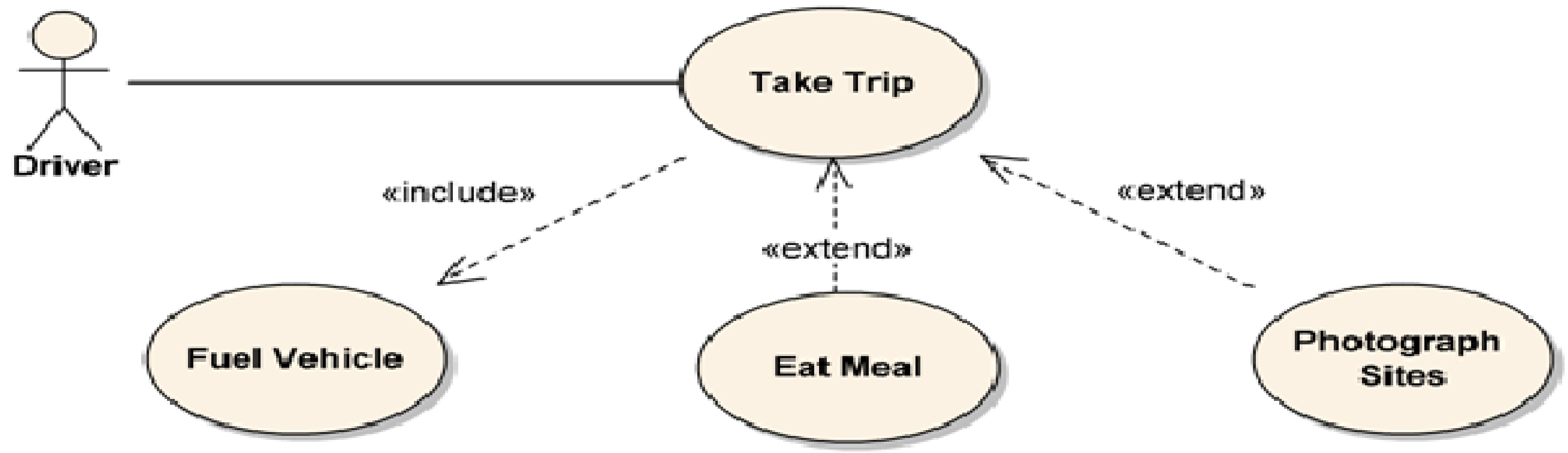
Sales Subsystem
Fill Shopping Cart <<includes>> Relationships
Search for item
«includes»
Customer
Fill shopping cart
«includes»
View product comments and ratings
«includes»
View accessory combinations

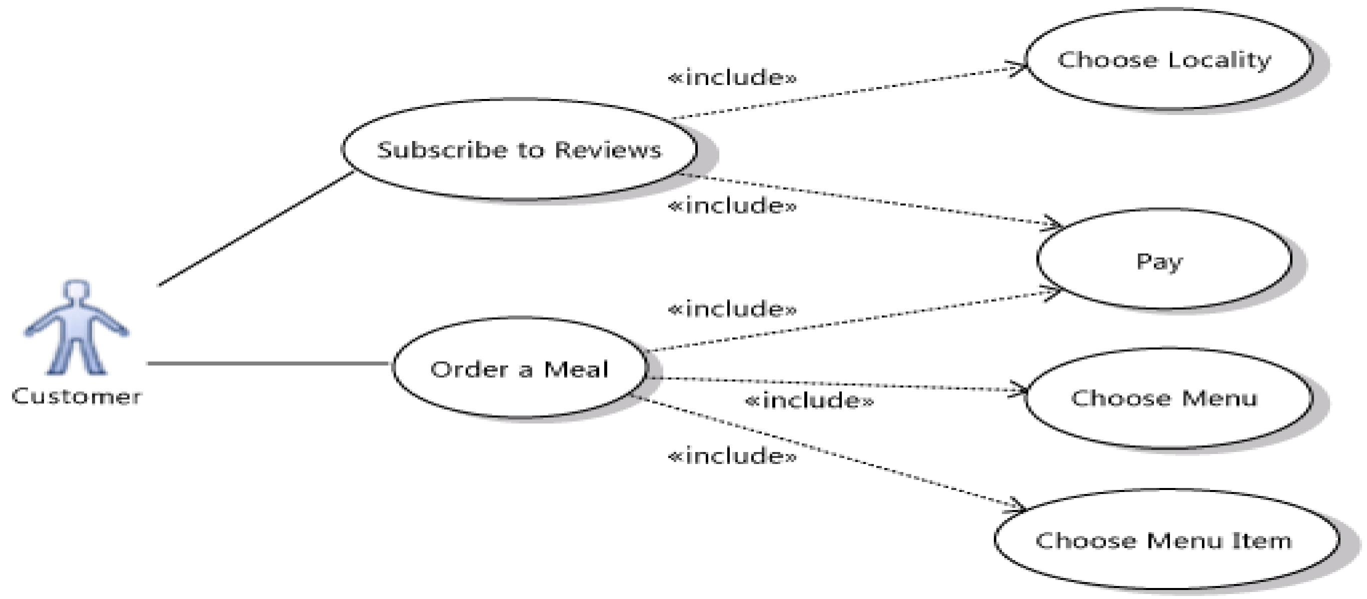
UML Use Case Extend

- **Extend** is a directed relationship that specifies how and when the behavior defined in usually supplementary (optional) **extending use case** can be inserted into the behavior defined in the **extended use case**. **Extended** use case is meaningful on its own, it is **independent** of the extending use case.
- **Extending** use case typically defines **optional** behavior that is not necessarily meaningful by itself. The extend relationship is **owned** by the extending use case. The same extending use case can extend more than one use case, and extending use case may itself be extended.

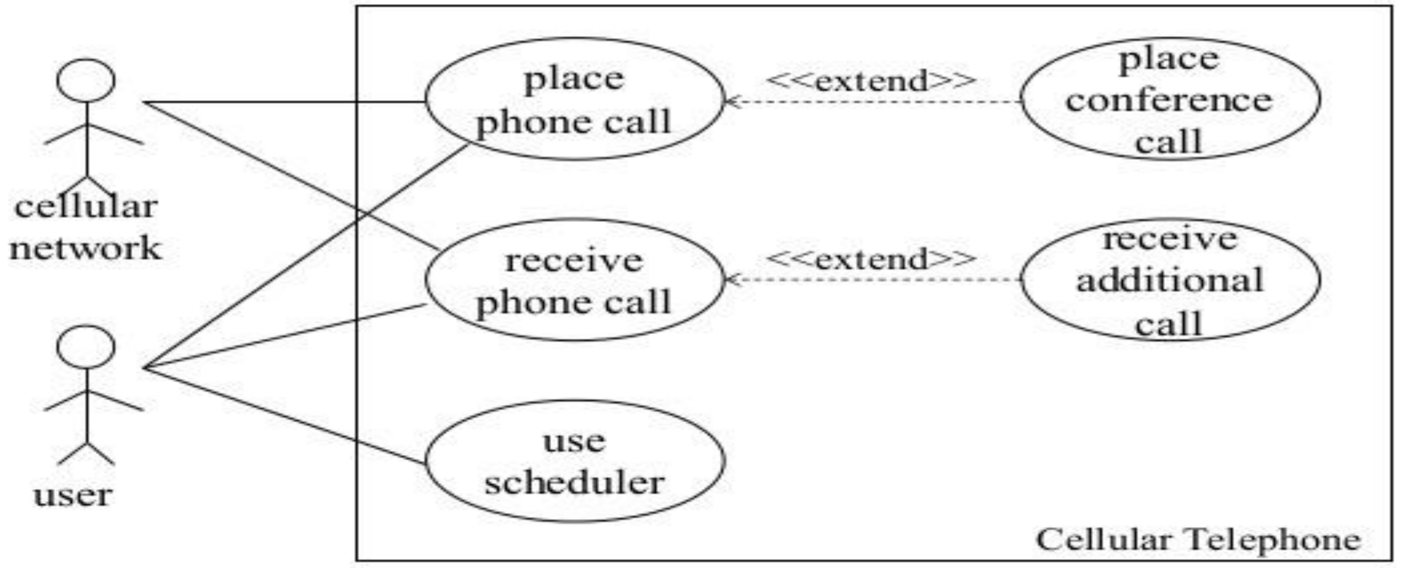


Registration *use case is complete and meaningful on its own.*
It could be extended with optional ***Get Help On Registration*** *use case.*


Driver
Take Trip
«include»
«extend»
«extend»
Fuel Vehicle
Eat Meal
Photograph Sites

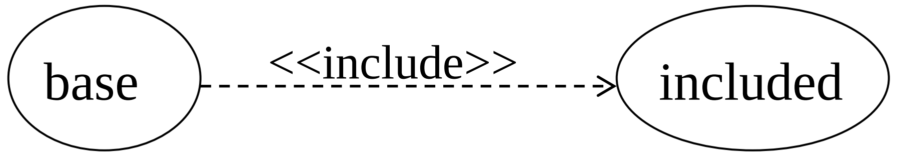

Customer
Subscribe to Reviews
Order a Meal
«include»
«include»
«include»
«include»
«include»
Choose Locality
Pay
Choose Menu
Choose Menu Item

Example #1


cellular network
user
place phone call
<<extend>>
place conference call
receive phone call
<<extend>>
receive additional call
use scheduler
Cellular Telephone

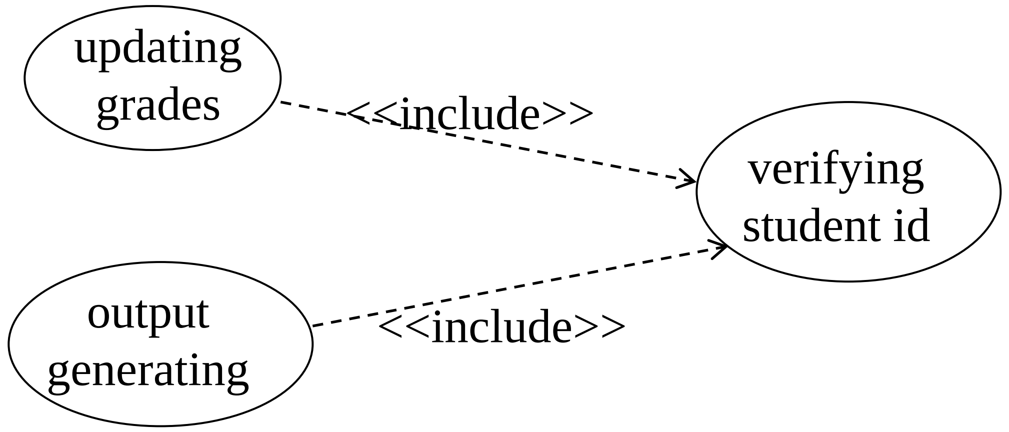
More about Include

- The base use case explicitly incorporates the behavior of another use case at a location specified in the base.
- The included use case never stands alone. It only occurs as a part of some larger base that includes it.



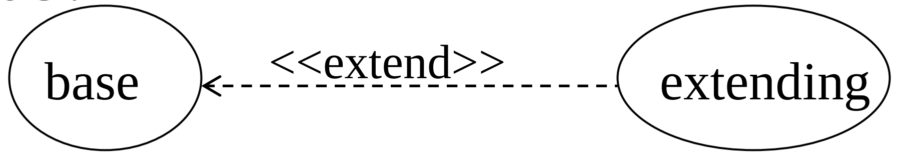
More about Include

- Enables to avoid describing the same flow of events several times by putting the common behavior in a use case of its own.



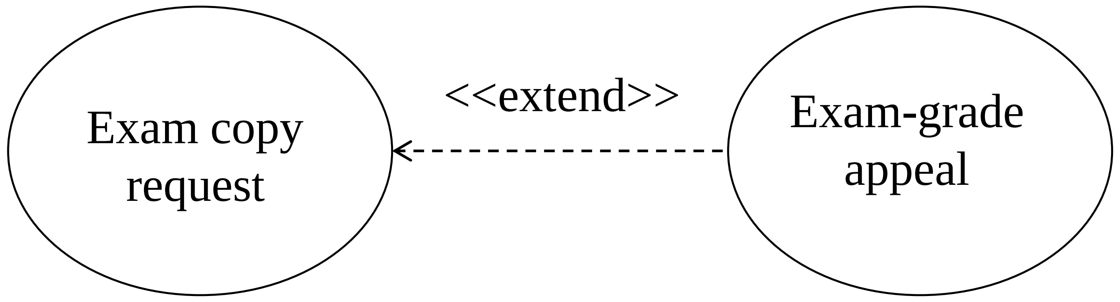
More about Extend

- The base use case implicitly incorporates the behavior of another use case at certain points called extension points.
- The base use case may stand alone, but under certain conditions its behavior may be extended by the behavior of another use case.



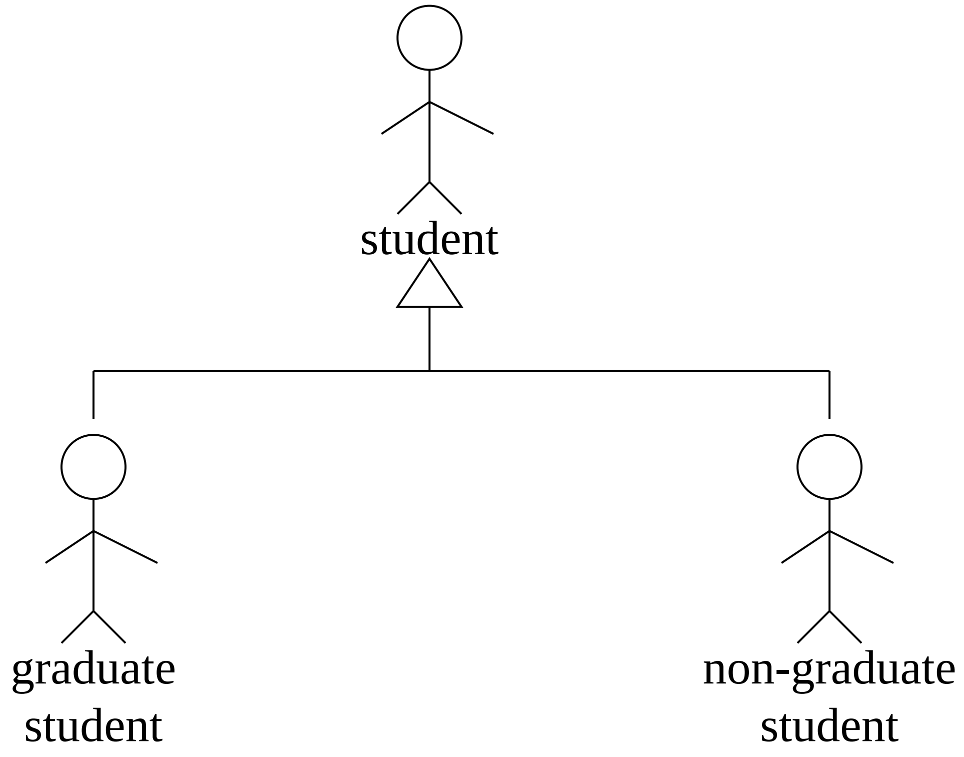
More about Extend

- Enables to model optional behavior or branching under conditions.

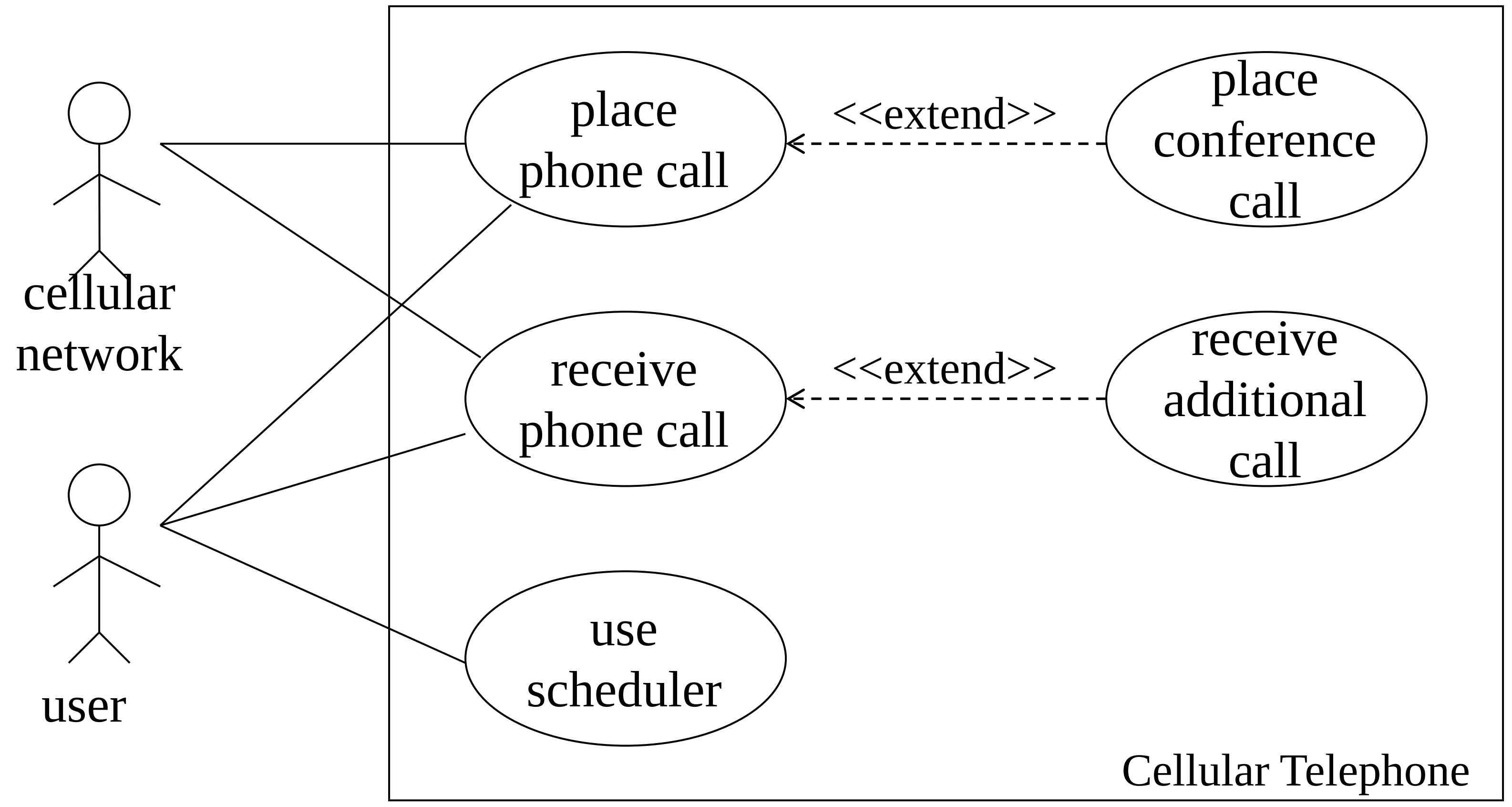


Relationships between Actors

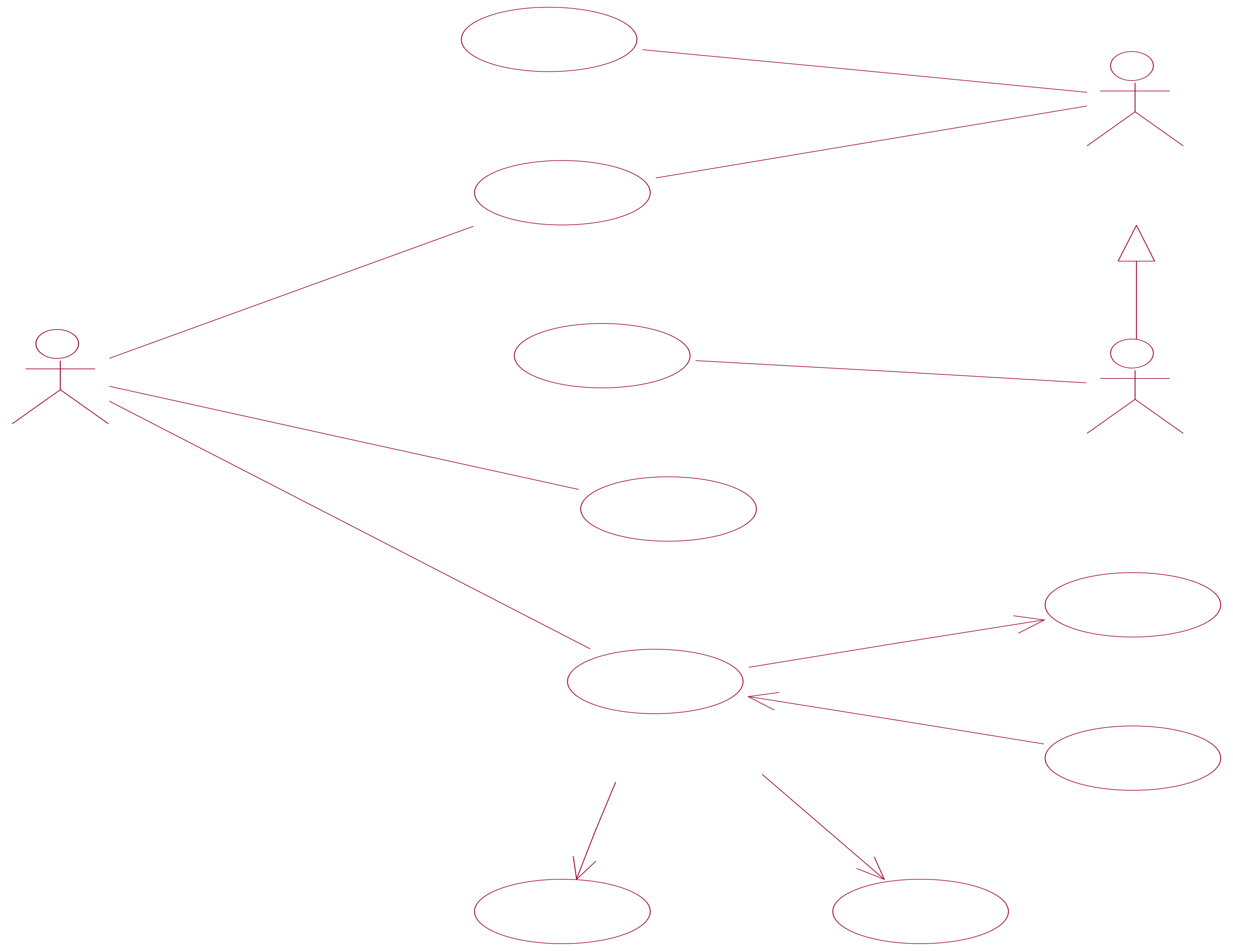
- Generalization.



Example


cellular network
user
place phone call
receive phone call
use scheduler
<<extend>>
place conference call
<<extend>>
receive additional call
Cellular Telephone

A More Complicate Example



Use Case Description

Each use case may include all or part of the following:

Title or Reference Name	- meaningful name of the UC
Author/Date	- the author and creation date
Modification/Date	- last modification and its date
Purpose	- specifies the goal to be achieved
Overview	- short description of the processes
Cross References	- requirements references
Actors	- agents participating
Pre Conditions	- must be true to allow execution
Post Conditions	- will be set when completes normally
Normal flow of events	- regular flow of activities
Alternative flow of events	- other flow of activities
Exceptional flow of events	- unusual situations
Implementation issues	- foreseen implementation problems

Example- Money Withdraw

- **Use Case**: Withdraw Money
- **Author**: ZB
- **Date**: 1-AUG-2015
- **Purpose**: To withdraw some cash from user's bank account
- **Overview**: The use case starts when the customer inserts his credit card into the system. The system requests the user PIN. The system validates the PIN. If the validation succeeded, the customer can choose the withdraw operation else alternative 1 – validation failure is executed. The customer enters the amount of cash to withdraw. The system checks the amount of cash in the user account, its credit limit. If the withdraw amount in the range between the current amount + credit limit the system dispense the cash and prints a withdraw receipt, else alternative 2 – amount exceeded is executed.
- Cross References: R1.1, R1.2, R7

Example- Money Withdraw (cont.)

- **Actors:** Customer
- **Pre Condition**:
  - The ATM must be in a state ready to accept transactions
  - The ATM must have at least some cash on hand that it can dispense
  - The ATM must have enough paper to print a receipt for at least one transaction
- **Post Condition**:
  - The current amount of cash in the user account is the amount before the withdraw minus the withdraw amount
  - A receipt was printed on the withdraw amount
  - The withdraw transaction was audit in the System log file

Example- Money Withdraw (cont.)

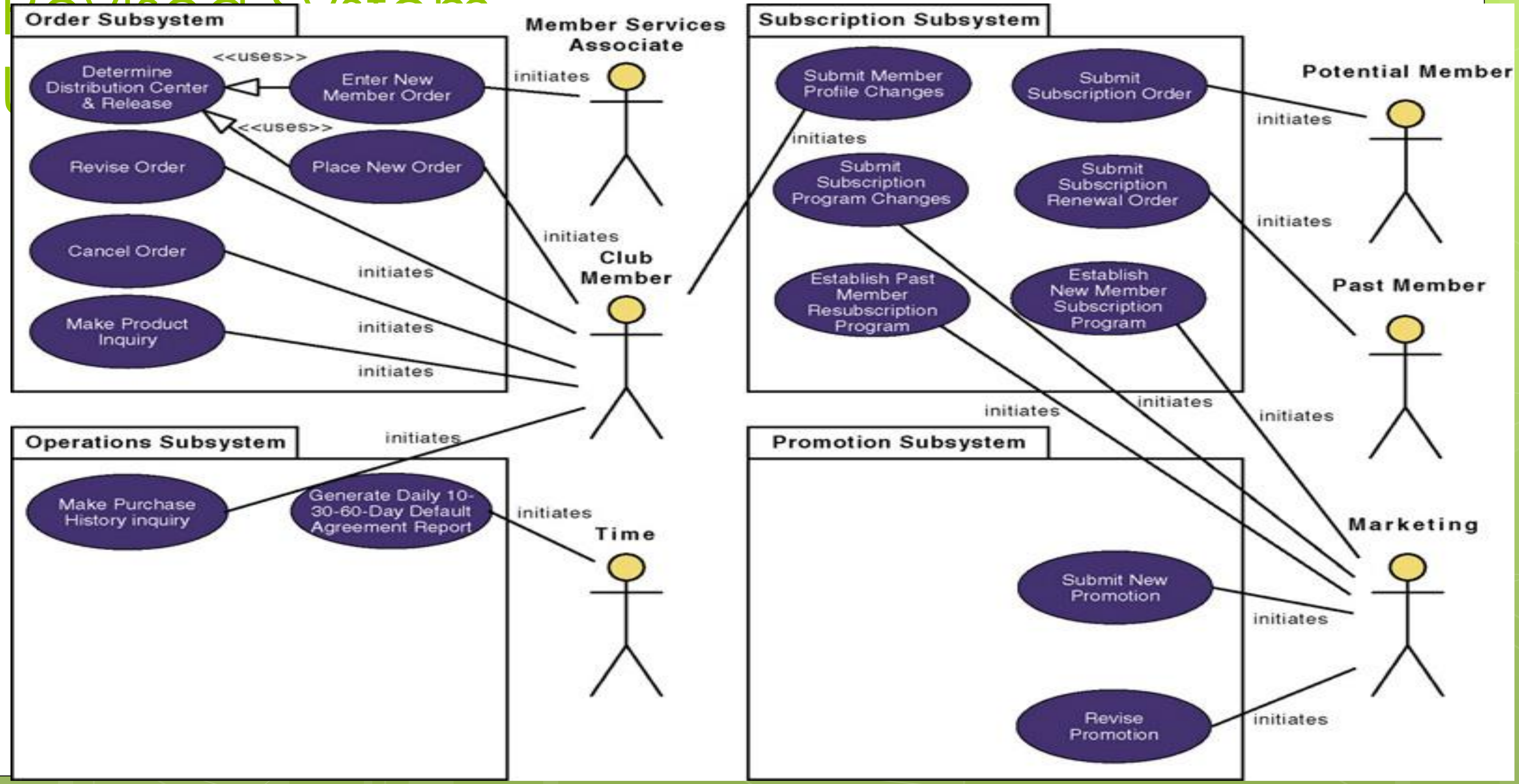
- **Typical Course of events:**

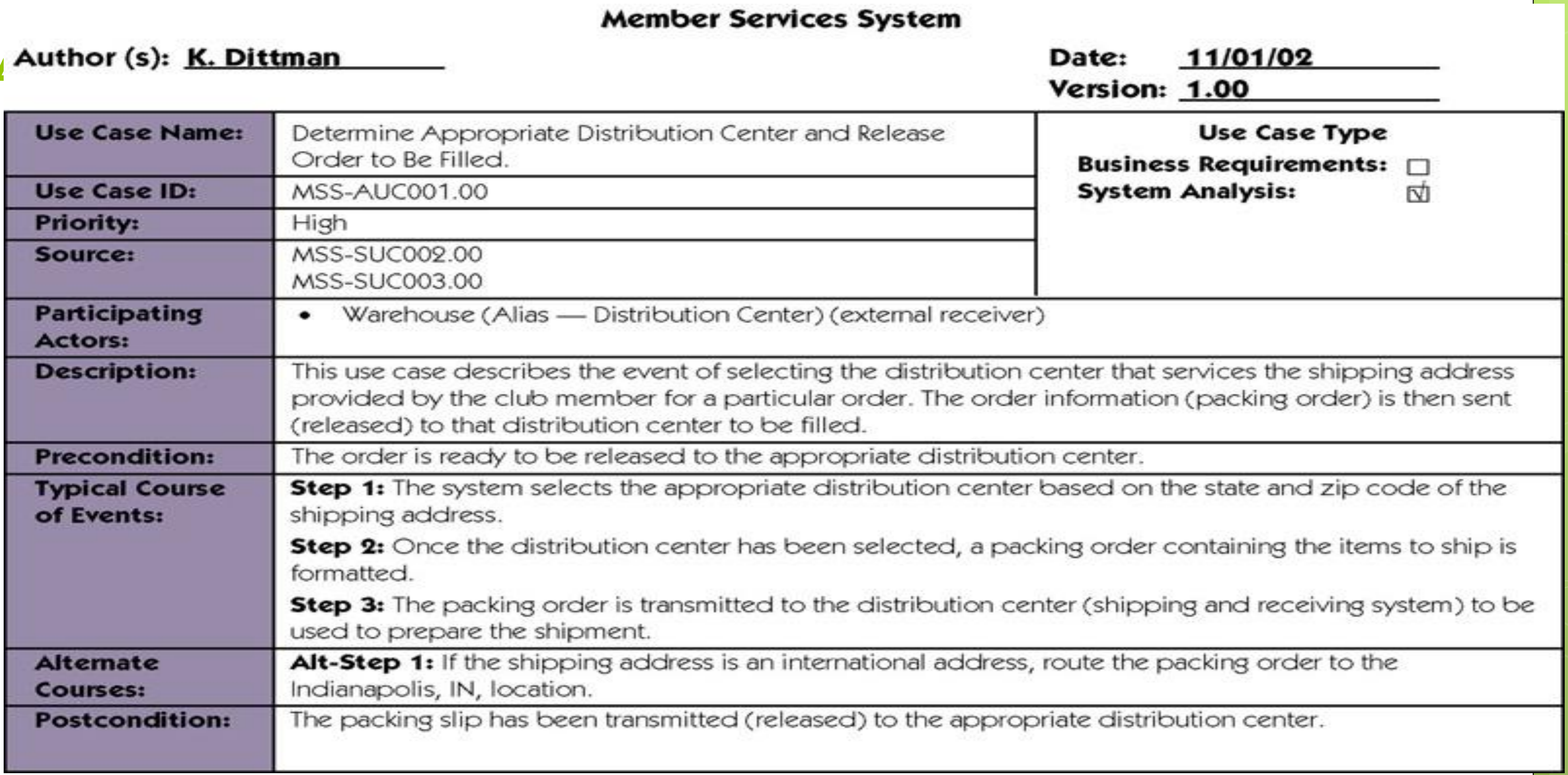
Actor Actions	System Actions
1. Begins when a Customer arrives at ATM	
2. Customer inserts a Credit card into ATM	3. System verifies the customer ID and status
5. Customer chooses “Withdraw” operation	4. System asks for an operation type
7. Customer enters the cash amount	6. System asks for the withdraw amount
	8. System checks if withdraw amount is legal
	9. System dispenses the cash
	10. System deduces the withdraw amount from account
	11. System prints a receipt
13. Customer takes the cash and the receipt	12. System ejects the cash card

Example- Money Withdraw (cont.)

- Alternative flow of events:
  - Step 3: Customer authorization failed. Display an error message, cancel the transaction and eject the card.
  - Step 8: Customer has insufficient funds in its account. Display an error message, and go to step 6.
  - Step 8: Customer exceeds its legal amount. Display an error message, and go to step 6.
- Exceptional flow of events:
  - Power failure in the process of the transaction before step 9, cancel the transaction and eject the card

Revised System





Member Services System

Author (s): K. Dittman **Date: 11/01/02**
Version: 1.00

Use Case Name:	Determine Appropriate Distribution Center and Release Order to Be Filled.	**Use Case Type** **Business Requirements:** ☐ **System Analysis:** ☑
Use Case ID:	MSS-AUC001.00	
Priority:	High	
Source:	MSS-SUC002.00 MSS-SUC003.00	
Participating Actors:	• Warehouse (Alias — Distribution Center) (external receiver)	
Description:	This use case describes the event of selecting the distribution center that services the shipping address provided by the club member for a particular order. The order information (packing order) is then sent (released) to that distribution center to be filled.	
Precondition:	The order is ready to be released to the appropriate distribution center.	
Typical Course of Events:	**Step 1:** The system selects the appropriate distribution center based on the state and zip code of the shipping address. **Step 2:** Once the distribution center has been selected, a packing order containing the items to ship is formatted. **Step 3:** The packing order is transmitted to the distribution center (shipping and receiving system) to be used to prepare the shipment.	
Alternate Courses:	**Alt-Step 1:** If the shipping address is an international address, route the packing order to the Indianapolis, IN, location.	
Postcondition:	The packing slip has been transmitted (released) to the appropriate distribution center.	