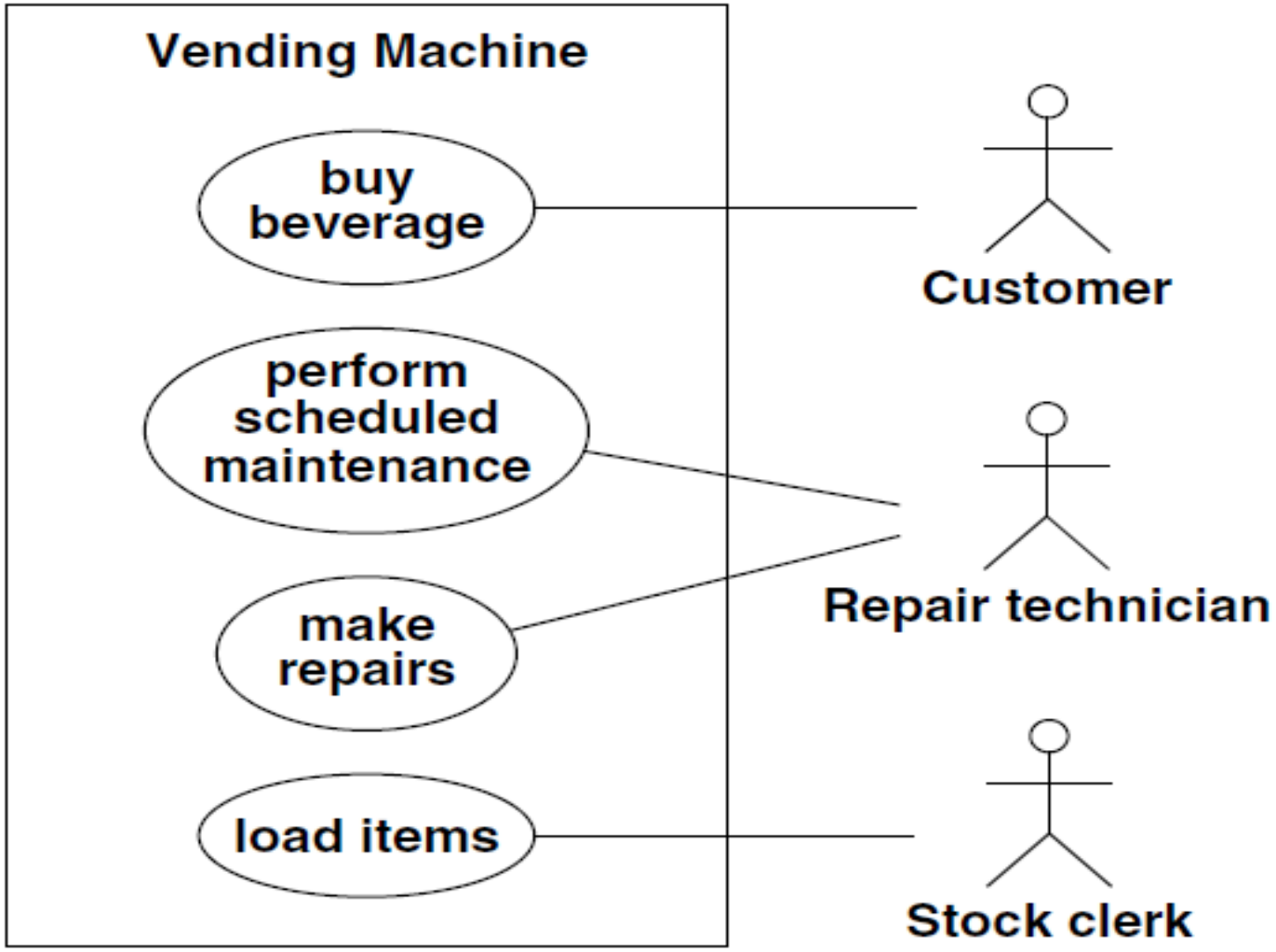
Use Case Model

Use Case Model

- Part of "Interaction Model"
- Describes how a system interacts with outside actors
- "Each use case is a piece of functionality provided by system to the users"
- Used in "Capturing Requirements"
- Basic notations:
  - Actors
  - Use cases
  - System Boundary

Use Case Model



Use Case Model: Actors

- "Direct External user of a system"
- "An object or a set of objects that communicates directly with the system, but that is not part of the system"
- E.g.
  - Travel Agency System has actors as Traveler, Agent, Airline
  - DBMS has actors as Admin and User
- Actors can be Persons, Devices and Other Systems

Use Case Model: Use Cases

- "Coherent piece of functionality that a system can provide by interacting with actors"
- A use case involves a sequence of messages among the system and its actors.

- E.g. "Buy Item" is a use case which includes
  - Insert Money
  - Select Item
  - Receive Item

Use Case Model: Use Cases

- **Buy a beverage**. The vending machine delivers a beverage after a customer selects and pays for it.
- **Perform scheduled maintenance**. A repair technician performs the periodic service on the vending machine necessary to keep it in good working condition.
- **Make repairs**. A repair technician performs the unexpected service on the vending machine necessary to repair a problem in its operation.
- **Load items**. A stock clerk adds items into the vending machine to replenish its stock of beverages.

Use Case Model: Use Cases

- "A use case may involve one or more actors"
  - E.g. For Telephone System, "Make a call" relates Caller and Receiver
- "Every use case has sequence of messages between the system and its users"
  - E.g. "Buy Item"
- "Use case partitioning should be done at comparable level of abstraction"
  - E.g. "Make a call", "Record a message"– OK
  - But "Set volume to low", "Set volume to high" – Too narrow level

Use Case Model: Use case Description

- Every use case is supported by following description:
  - Use case: Name of the use case
  - Summary: Short overview of the use case
  - Actors: Names of actors involved
  - Pre conditions: Previous state of system, before this use case starts
  - Description: Explanation of entire scenario
  - Exceptions: Error conditions and explanation
  - Post conditions: State of system after use case execution

Use Case Model: Use case Description

Use Case: Buy a beverage

Summary: The vending machine delivers a beverage after a customer selects and pays for it.

Actors: Customer

Preconditions: The machine is waiting for money to be inserted.

Description: The machine starts in the waiting state in which it displays the message "Enter coins." A customer inserts coins into the machine. The machine displays the total value of money entered and lights up the buttons for the items that can be purchased for the money inserted. The customer pushes a button. The machine dispenses the corresponding item and makes change, if the cost of the item is less than the money inserted.

Use Case Model: Use case Description

Exceptions:

Canceled: If the customer presses the cancel button before an item has been selected, the customer's money is returned and the machine resets to the waiting state.

Out of stock: If the customer presses a button for an out-of-stock item, the message "That item is out of stock" is displayed. The machine continues to accept coins or a selection.

Insufficient money: If the customer presses a button for an item that costs more than the money inserted, the message "You must insert $*nn.nn* more for that item" is displayed, where *nn.nn* is the amount of additional money needed. The machine continues to accept coins or a selection.

No change: If the customer has inserted enough money to buy the item but the machine cannot make the correct change, the message "Cannot make correct change" is displayed and the machine continues to accept coins or a selection.

Postconditions: The machine is waiting for money to be inserted.

Exercise Time

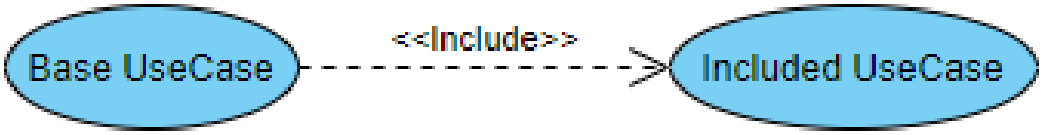
- Draw a basic use case diagram for following:
  - ATM system
  - Online Shopping system
  - Library Management system
  - Railway Reservation system

Use case Relationships

- Complex use cases can be built from smaller pieces with the
  - include,
  - extend, and
  - generalization relationships.

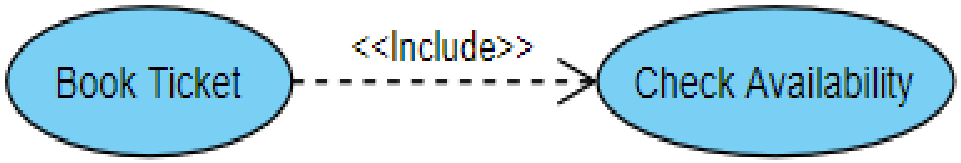
Use case Relationships: Include

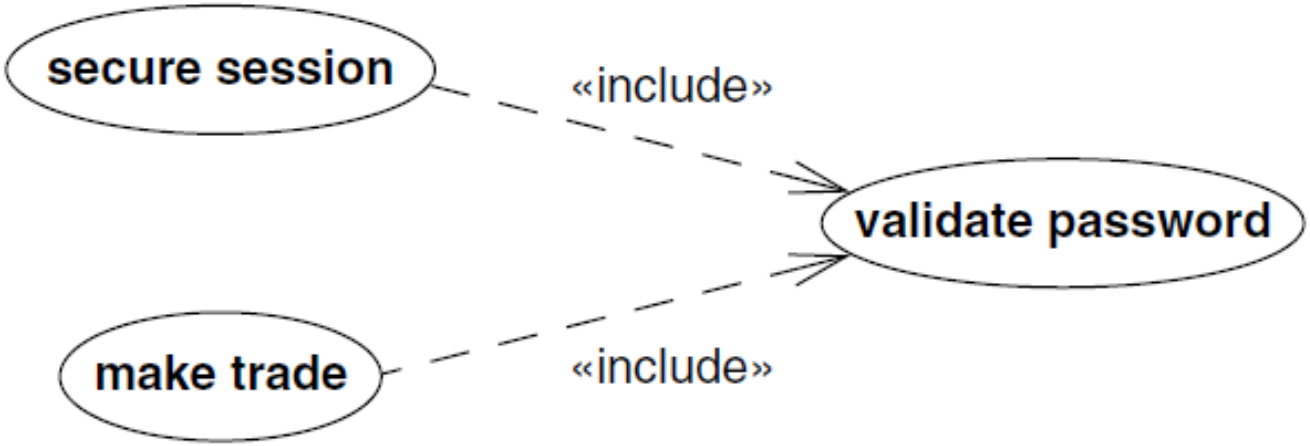
- The include relationship incorporates one use case within the behavior sequence of another use case.
- "The included use case may or may not be usable on its own."



- Include focuses on reuse of functionality (Update Product Detail and Delete Product Details both need Searching)

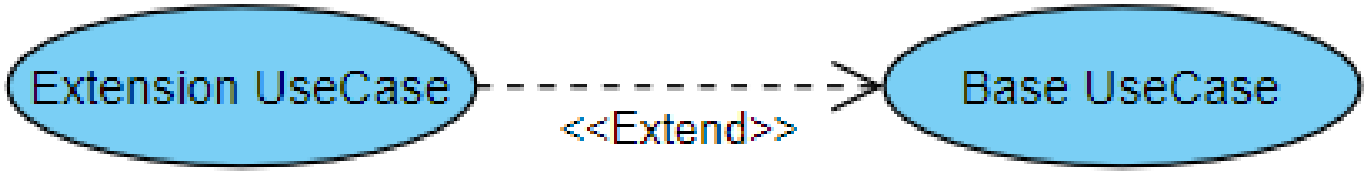
Use case Relationships: Include





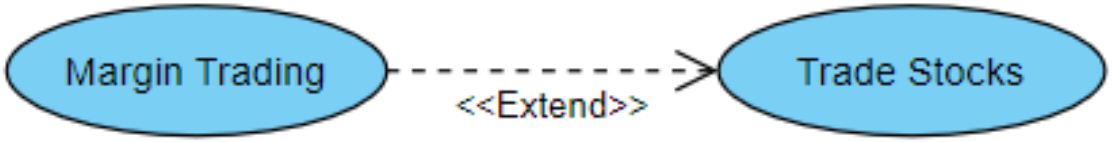
Use case Relationships: Extend

- The extend relationship adds incremental behavior to a use case.
- "It is like an include relationship looked at from the opposite direction",
  - in which the extension adds itself to the base, rather than the base explicitly incorporating the extension.



Use case Relationships: Extend

- Extension use case cannot appear alone
- Base use case must be valid in absence of any extensions
- Extends focuses on adding functionality to base use case



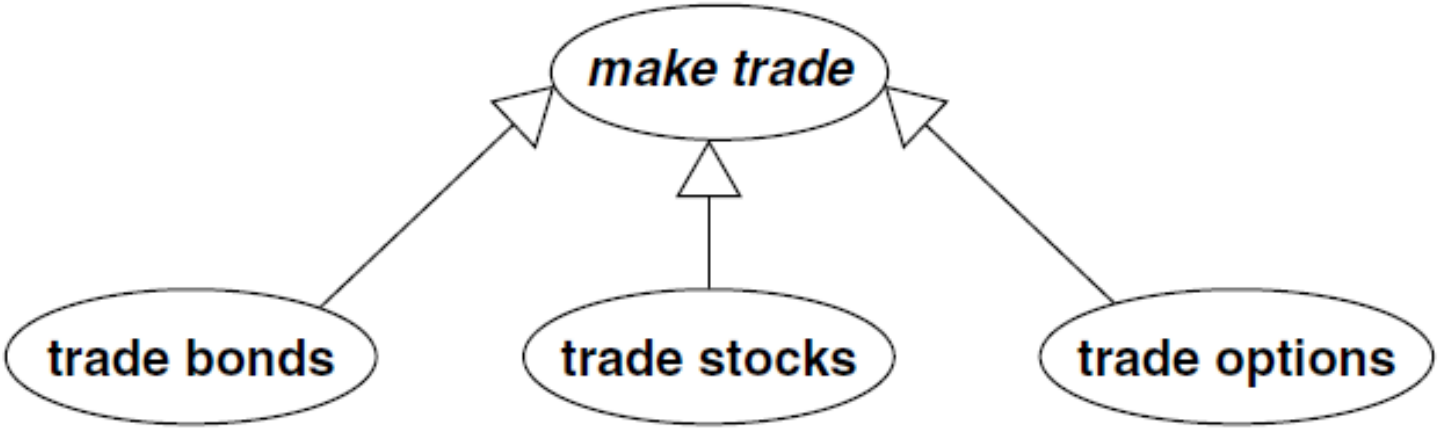
- Here, margin trading allows to buy stocks on loan if account has insufficient cash

Use case Relationships: Generalization

- Generalization can show specific variations on a general use case, analogous to generalization among classes.
- A parent use case represents a general behavior sequence.
- Child use cases specialize the parent by inserting additional steps or by refining steps.

Use case Relationships: Generalization

- Parent use case must be abstract or concrete
- Child use case can freely substitute for parent use case



When to use the different types?

- Include:
  - To extract common behavior out from a use case which can be reused
- Extend:
  - When a use case has additional behavior
- Generalization:
  - When a use case comes with several variations

Include vs. Extend

- Included behavior is necessary for base use case
- Extended behavior is optional

Exercise Time (Repeat with use case relationships)

- Draw a basic use case diagram for following:
  - ATM system
  - Online Shopping system
  - Library Management system
  - Railway Reservation system
  - Hospital Appointment Management system
  - Restaurant Management system