Describing an Object Oriented System

UML (part II)

Behavior Diagrams

- Activity Diagrams - Like a flowchart; shows the sequence of events in an activity
- Use case diagrams - Involves the user, called an actor, and the system. *Note: the user does not have to be a human entity.* This diagram shows the interaction in a use case.
- State machine diagrams - Shows the sequence of states that an object goes through. It's a graph whose vertices are states and whose directed edges are transitions between edges.

Use Cases

- The various interactions of actors with a system are quantized into *use cases*
- A use case is a clear piece of functionality that a system provides by its interaction with *actors*
- As an example, take a vending machine:
  - The purchaser is an actor that can buy a snack from the machine by inserting money
  - A repair technician (actor) can repair the machine
  - The delivery person (actor) can re-stock the machine

Use Case Summary

- Buy a snack: The vending machine dispenses a snack after a purchaser makes their selection and inserts money
- Perform scheduled maintenance: A repair technician performs the periodic service on the vending machine to keep it working
- Make repairs: A repair technician performs unexpected repairs to fix a problem
- Re-stock (Load) items: A delivery person adds items into the machine

An Example

Use case: Buy a snack from the vending machine

Function Summary: The machine delivers a chosen snack after the purchaser (actor) selects and pays

Precondition: Machine in wait state for money

Description: Machine starts in the wait state; purchaser inserts money and machine displays total money entered; the purchaser then presses buttons for the item number selected; the machine dispenses item and delivers change (if necessary)

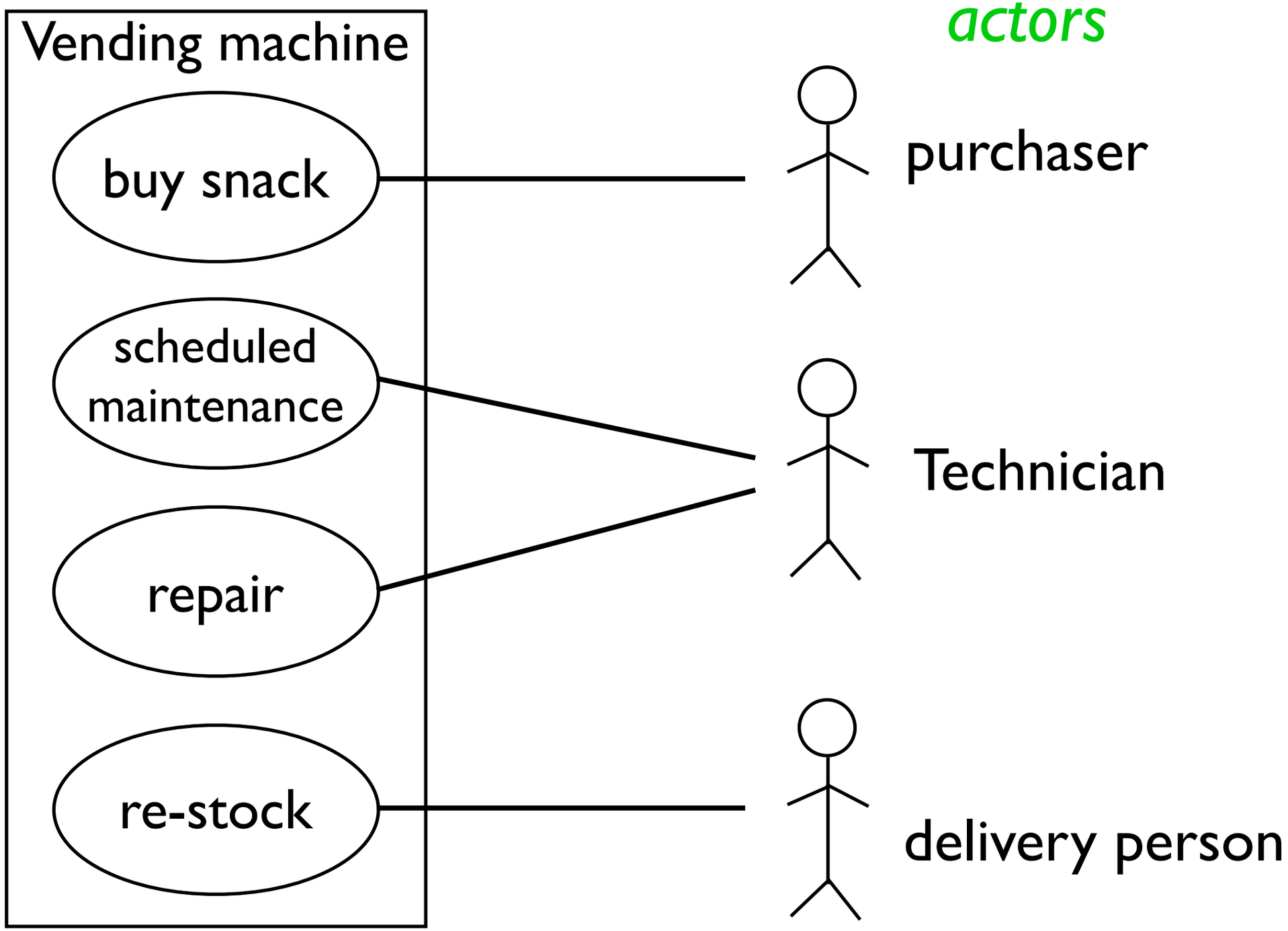
An Example (continued)

- Exceptions:
  - Cancelled: customer presses the cancel button and money is returned
  - Out of stock (empty): Display message and continue to accept money
  - Insufficient money: Display message "You must insert more money or the amount required"
  - Insufficient change: "Insert correct amount" the machine cannot make change.
- Postcondition: Machine in wait state for money

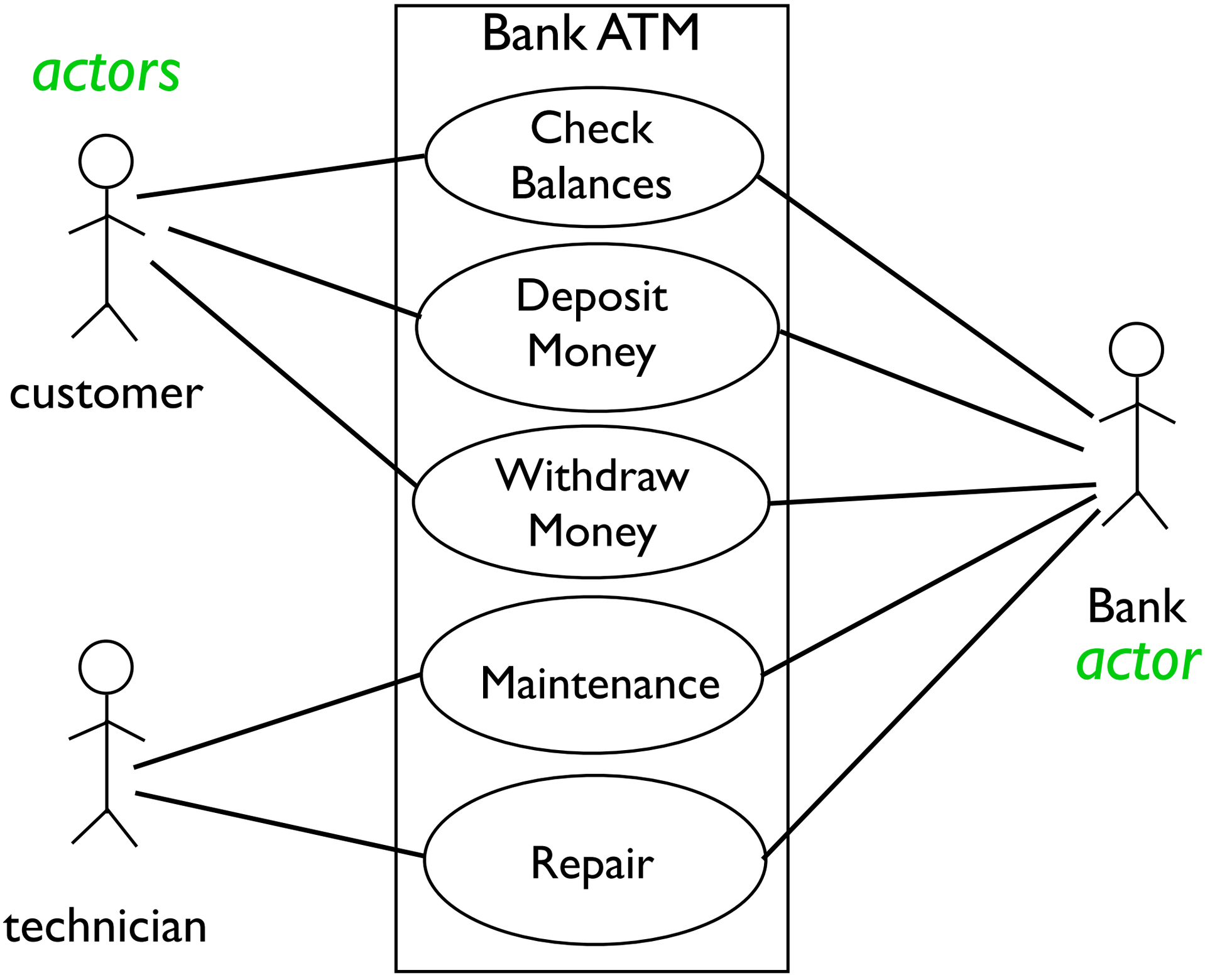
Use Case Diagrams

- Involves a set of *actors*
- Each use case represents a piece of what the functionality of the system provides
- Each actor represents one kind of object for which the behavior of the system can perform

The Use Case Diagram



Another Example: Bank ATM



Guidelines for Case Use Models

- Determine the System Boundary: Determine or decide what the system includes and what it omits. Knowing this, you can treat the system like a "Black Box". That is, as a single entity whose internal details are hidden.

- Make sure the actors are focused: Each actor should have a single, clear purpose. If an object embodies multi purposes, use separate actors.

Guidelines for Case Use Models (continued)

- Each use case must provide some value to users: A use case should represent a complete transaction that provides a value to users.

- Relate use cases and actors: Every case should have at least one actor and every actor should participate in at least one use case. Of course, as we've seen, a use case can involve more than one actor and an actor may be involved in more than one case.