

---

# CIS 422/522

## Requirements Overview Use Cases

# Two Kinds of Software Requirements

---

- Discussed last class period
- Concept of Operations: documents user needs and customer expectations
  - Link to organizational goals
  - Stated in terms the the users' / customer's can understand
- Technical Specifications: a concise and unambiguous statement of technical parameters for a system that will satisfy the operational requirements
  - Stated in the developers' terminology
  - Covers issues such as performance, interfaces, safety, security, and reliability

# For Your Project

---

- Apply Use Cases (scenarios) to describe the system's mission from the user's point of view
  - Answers the questions, “What is the system for?” and “How will the user use it?”
  - Tells a story
- For the “Functional Requirements” be as rigorous as possible
  - Use tables, bullets, or case-by-case behavior description
  - Purpose is to answer questions about the requirements quickly and precisely
    - Answers, “What should the system output in this circumstance?”
    - Reference, not a narrative, does not “tell a story”

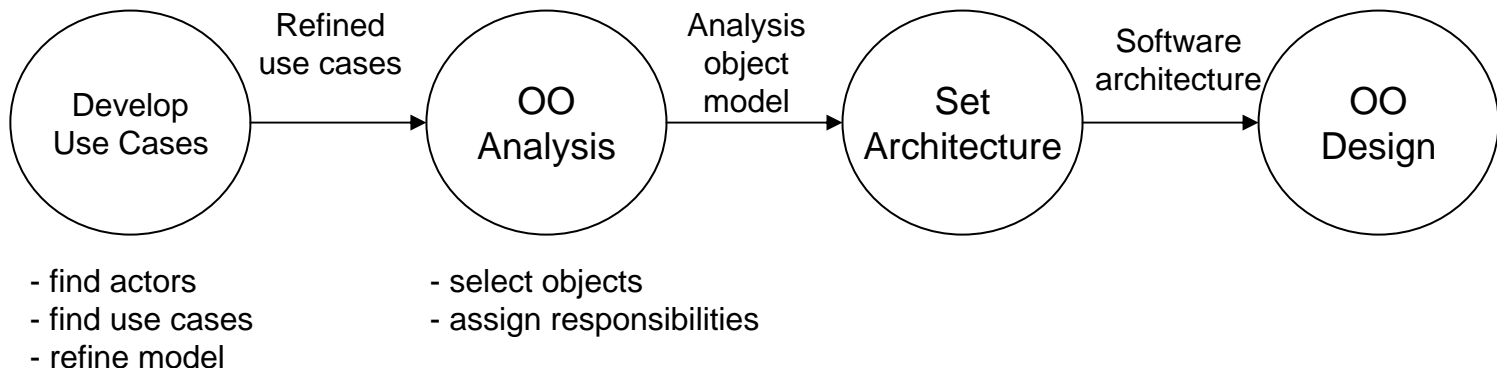
---

# Problem Analysis Using Use Cases

# Use Cases

---

- Often done as a prelude to OO modeling
- A form of *User Centered Analysis* – capturing requirements from the user’s point of view
  - Goal of helping identify user needs
  - Solve the right problem
- Best fit with operational requirements though may be used elsewhere



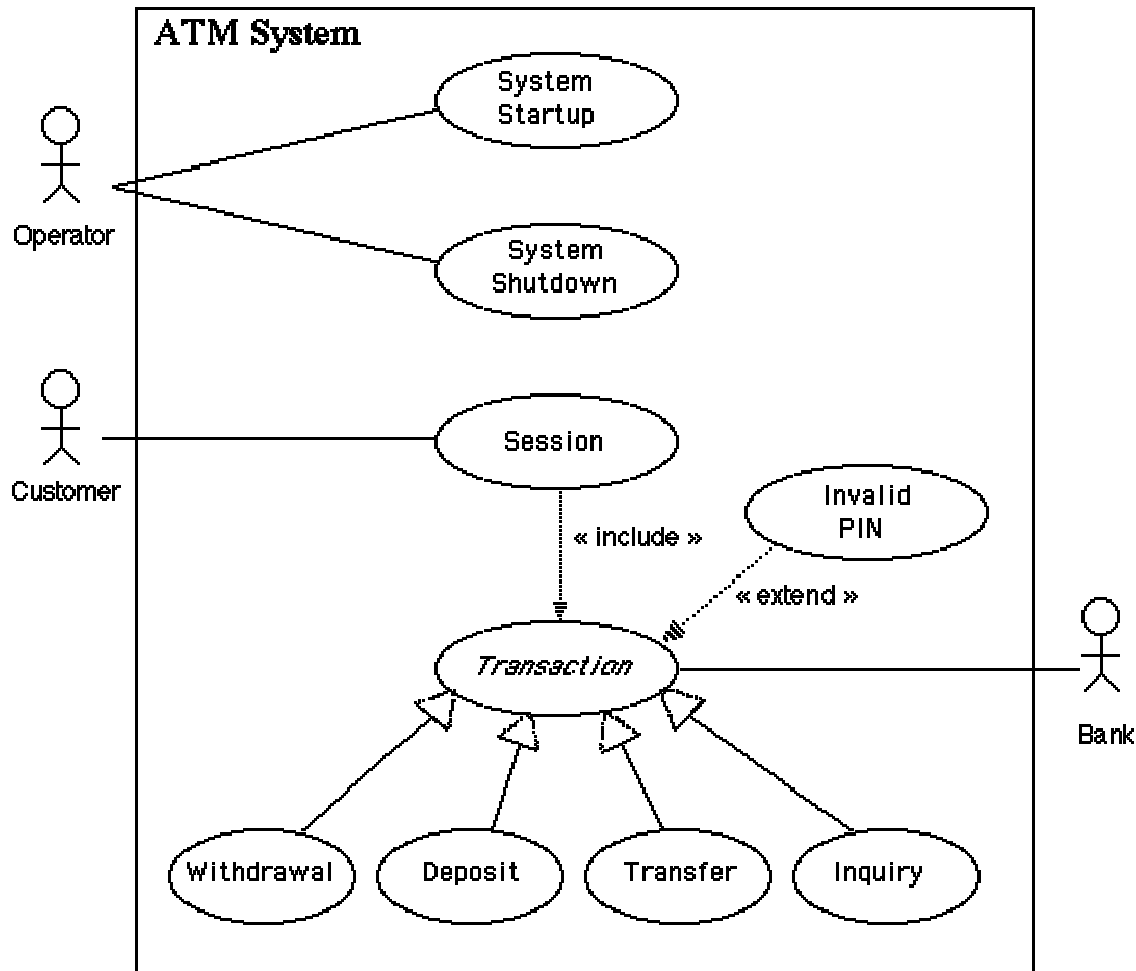
# Terminology

---

- Scenario – description of a sequence of interactions between a user and the system from the user's POV
  - What does the user see or do
  - What does the system do in response
- Use Case – a set of scenarios related by a common user goal
  - Goal – an objective the user is employing the system to achieve
  - Scenarios represent different possible outcomes (nominal case, error case, etc.)

# UML Graphic Example

<http://www.math-cs.gordon.edu/local/courses/cs211/ATMExample/>



# ATM Session Use Case (Informal)

---

**A session is started when a customer inserts an ATM card into the card reader slot of the machine. The ATM pulls the card into the machine and reads it. (If the reader cannot read the card due to improper insertion or a damaged stripe, the card is ejected, an error screen is displayed, and the session is aborted.) The customer is asked to enter his/her PIN, and is then allowed to perform one or more transactions, choosing from a menu of possible types of transaction in each case. After each transaction, the customer is asked whether he/she would like to perform another. When the customer is through performing transactions, the card is ejected from the machine and the session ends. If a transaction is aborted due to too many invalid PIN entries, the session is also aborted, with the card being retained in the machine.**

**The customer may abort the session by pressing the Cancel key when entering a PIN or choosing a transaction type**

<http://www.math-cs.gordon.edu/local/courses/cs211/ATMExample/>



# 1 Use Case: Manage Reports

## 1.1 Description

This Use Case describes operation for Creating, Saving, Deleting, Printing, Exiting and Displaying reports.

## 1.2 Actors

User  
Project database

## 1.3 Triggers

Program Manager selects operations from menu.

## 1.4 Flow of events

### 1.4.1 Basic Flow

1. User chooses desired report by selecting “Report” -> “Open” from the menu bar
2. System displays report to screen
3. User selects desired report layout using Use Case Specify Report
4. Steps 2 and 3 are repeated until user is satisfied
5. User can Save or Print report using use case Save Report or Print Report
6. User Exits report by selecting “Exit” from the “File” menu

### 1.4.2 Alternative Flows

#### 1.4.2.1 Create New Report

1. User selects “Create New Report” from file menu
2. ...

#### 1.4.2.2 Delete Report

.....

#### 1.4.3 Preconditions

etc

**A more systematic approach**

- **Uses a standard template**
- **Easier to make sure it is complete**

# Finding Use Cases

---

- First, identify the actors
- Then consider each actor in turn
  - consider all the ways in which an actor will interact with the system, one at a time
- Read the requirements, observe work practices, study similar systems, etc from the viewpoint of each actor, in turn
- Identify real-world entities and logical relationships
- Developing use cases may identify new actors, which may, in turn, lead to new use cases

# Use Case Contents (Generic)

---

- Use case identifier
- Summary – summary of use case
- Actors – roles enacting use case
- Scenarios
  - Basic scenario – the normal case
  - Alternative scenarios – other ways to reach goal
  - Exceptions – problem scenarios
- Trigger – what causes the use case to start
- Assumptions
- Preconditions – what must be true before the interaction can occur?
- Post conditions – what must be true after the interaction occurs

# Use Case Contents

---

- Actors – identifies a role the user plays with respect to the system
  - Roles represent different classes of users (use the system with different goals)
  - Actors carry out use cases
  - Primarily useful in identifying different use cases
    - “How would depositors use the system?”
    - “How would a library patron use the system?”

# Basic Scenario

---

- Sequence when the user proceeds to his goal as system designers intend

1. Customer puts card into ATM card slot and enters PIN number.
2. Card verified and main menu presented.
3. Customer selects the transaction services menu and the corresponding menu is displayed.
4. Customer selects "automated payment service" and is prompted for the recipient's account number.
5. Customer enters recipient's account number.
6. Account verified and a menu with payment schedules is presented.
- 7.....

# Exceptions

---

- What is the scenario if the customer enters the wrong PIN?

1. Customer puts card into ATM card slot and enters PIN number.
2. Incorrect PIN identified and error message “Incorrect PIN” and error menu displayed.
3. Customer selects the “try again” option
- 4.....

# Uses and Limitations

---

- What are the limitations?
  - Expressive power?
  - Ambiguity
  - Completeness
  - Etc?
- What can use cases help with?

# Summary

---

- Use cases can be an effective tool for identifying:
  - The actors,
  - Major analysis objects,
  - Interactions between actors and analysis objects, and
  - Collaborations among analysis objects
- Use case analysis is a generally applicable technique that can be done independent of OO design and implementation
  - use cases reduce the accidental difficulties and help to control the essential difficulties of software development
- Generally inadequate for detailed technical requirements



# Assignment

---

- Faulk Paper