CS342 Software Engineering

Dr. Ismail Hababeh
German Jordanian University

Lecture 6
Requirements & Use Cases

Adapted from Software Engineering, by Dr. Paul E. Young & slides by Dr. Mohammad Daoud

The Requirements Workflow

- The aim of the requirements workflow is to determine the **client needs (constraints).**
- The Requirements Workflow aims to answer the question:

  What must the product be able to do?

The Requirements Workflow

- Determining the client's constraints
  - Deadline
  - Parallel running (new + old) systems
  - Portability
  - Reliability
  - Rapid response time
  - Cost:

    A bidding procedure is used

The Classical Requirements Phase

- The classical approach to requirements
  - Requirements' introduction
  - Requirements analysis
  - Construction of a rapid prototype
  - (Human) Client and future user experiments with the rapid prototype

CASE Tools for the Requirements Workflow

- Graphical CASE environments are extended to support UML include
  - System Architect
  - Software through Pictures
- Object-oriented CASE environments include
  - IBM Rational Rose
  - Together
  - ArgoUML (open source)

Determining the Client Needs

- The client is the only source of this information
- Problem! Misconception
  - "I know you believe you understood what you think I said, but I am not sure you realize that what you heard is not what I meant!"

Determining the Client Needs

- Misconception solution:
  - Obtain initial information from the client
  - Use this initial information as input to the Unified Process
  - Follow the Unified Processes to determine the client's real needs
- An expert systems analyst is needed to extract the appropriate information from the client.

Determining the Client Needs

1. Understanding the software product *domain:*

   The specific environment in which the target product operate

2. Building a *business model*:

   Construct the client's business processes

3. Using the business model to determine the initial *requirements techniques*

4. Building the business model *use cases*

5. *Iterating* the above steps

1. Understanding the Software Product Domain

- Every member of the development team must become fully familiar with the application domain
- Correct terminology is essential
- Construct a glossary

  A list of technical words used in the domain, and their meanings

2. Building the Business Model

- *Business model:*
  - A description of the business processes of an organization
  - Gives an understanding of the client's business as a whole.
  - Different requirements' techniques are used.

2.a. Requirements Techniques

1. Interviewing is the primary requirements technique

- The requirements team meet with the client and users to extract all relevant information
- After the interview, the interviewer must prepare a written report
  - It is strongly advisable to give a copy of the report to the person who was interviewed

2.a. Requirements Techniques

2. A questionnaire

   useful when the opinions of hundreds of individuals need to be determined.

3. Examination of business forms

   shows how the client currently does business.

4. An observations

   direct observation of the employees while they perform their duties can be useful.

Initial Requirements

- A *functional requirement* specifies an actions that the software product must be able to perform
  - Often expressed in terms of inputs and outputs
  - handled as part of the requirements and analysis workflows
- A *nonfunctional requirement* specifies properties of the software product itself, such as
  - Platform constraints
  - Response times
  - Reliability
  - Some nonfunctional requirements should wait until the design workflow

Initial Requirements

- Based on the initial business model, the initial requirements are refined.
- The dynamic requirements are frequently change
  - Maintain a list of likely requirements, together with use cases of requirements approved by the client

Requirements Extraction - Example

Extract the functional requirements based on the following Car Rental Company information system components: cars, members, and car rentals.

Cars Rental Functionalities

The software system needs to provide the following functionalities:

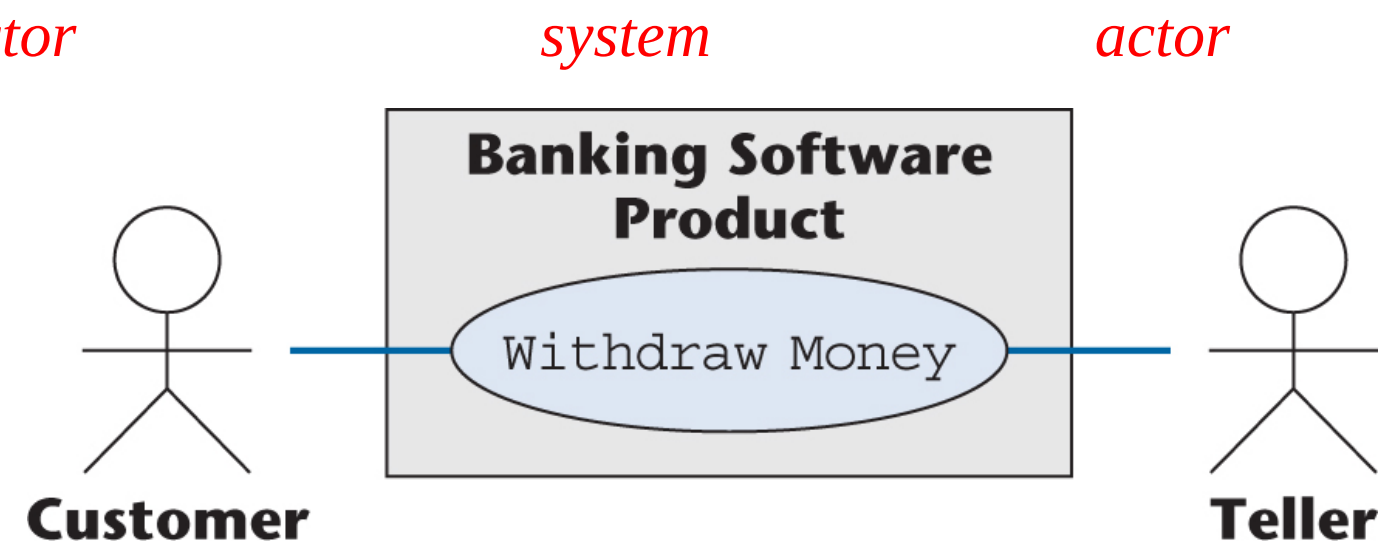
1. Add, modify and delete cars.
2. Add, modify and delete members' information.
3. Performs queries on cars by car id or car type.
4. Performs queries on clients' information by their id or name.
5. Provides a report of the cars owned by the company.
6. Provides a report of the members' rentals overdue.
7. Provides an invoice of a car rental.

4. Building the business model using use cases

- A Use Case

  The interaction between the software product itself and its users (*actors*)

- Example:



Use Cases Principles

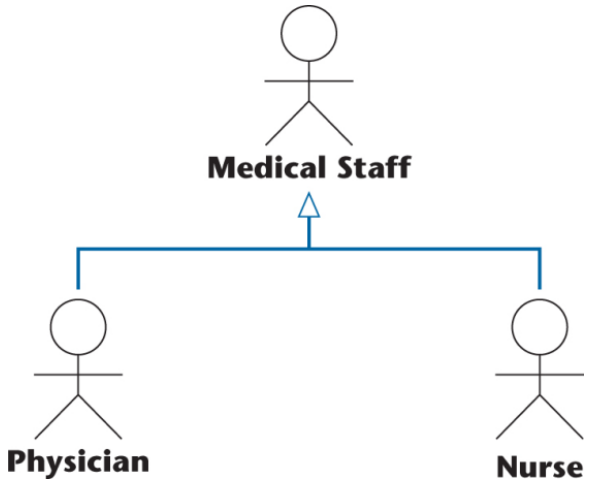
- A user (actor) of the system can play more than one role
  - Example: A customer of the bank can be a **Borrower** or a **Lender**
- One actor can be a participant in multiple use cases
  - Example: A **Borrower** may be an actor in the `Borrow Money`, the `Pay Interest on Loan`, and the `Repay Loan Principal` use cases
- The actor **Borrower** may stand for many thousands of bank customers

Use Cases Principles

- An actor need not be a human being
  - Example: An e-commerce information system has to interact with the credit card company information system
    - The credit card company information system is an actor from the viewpoint of the e-commerce information system
    - The e-commerce information system is an actor from the viewpoint of the credit card company information system

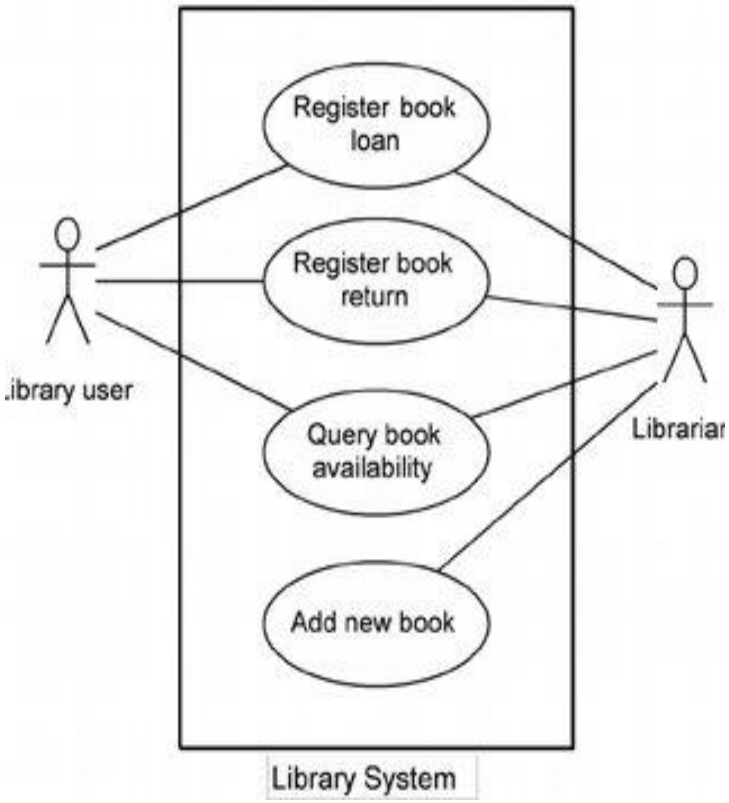
Use Cases Problems

- A potential problem when identifying actors
  - Overlapping actors
- Example: Hospital software product
  - One use case has actor **Nurse**
  - A different use case has actor **Medical Staff**
  - Better:
    - Actors: **Physician** and **Nurse**
- Alternatively:
  - Actor **Medical Staff** with two specializations: **Physician** and **Nurse**



Use Cases – Example 1

- Draw the use case of a simple library system that represents the following processes:
  - Register book loan
  - Register book return
  - Query book availability
  - Add new book



Use Cases – Example 2

- Draw the use case of a simple examination system that represents the following processes:
  - Student registration
  - Manage examination
  - Manage Question bank
  - System login
  - Tests
  - Reports