Module 3:

Introduction to Use-Case Modeling

Boonprasert Surakratanaskul

Faculty of Information Technology

King Mongkut's Institute of Technology LADKRABANG

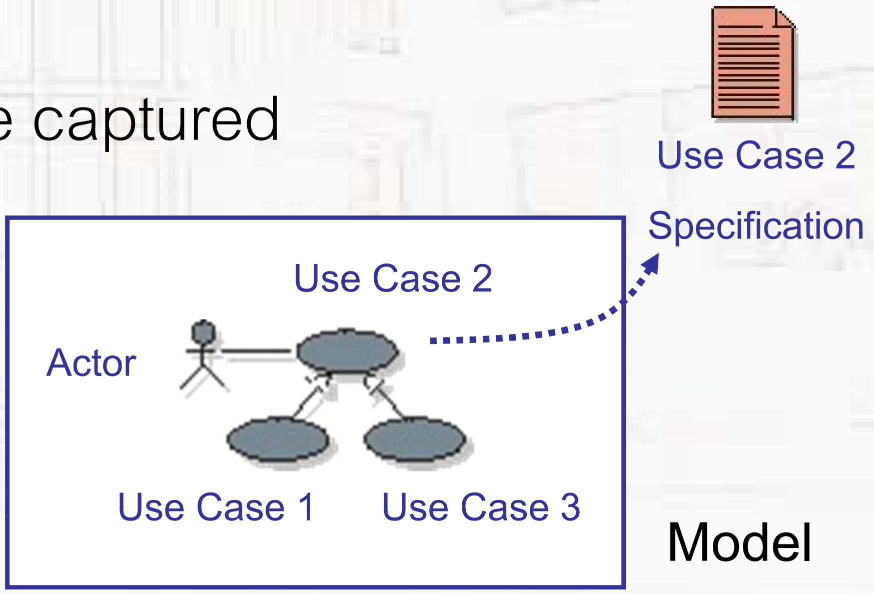
Objectives:

Introduction to Use-Case Modeling

- Define key concepts of use-case modeling
- List the benefits of use-case modeling
- Find and describe actors and use cases
- Sketch a use-case diagram

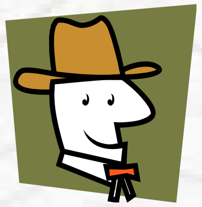
What is Use-Case Modeling?

- A means for capturing the desired behavior for the system under development
- A way to communicate the system's behavior
- Identifiers who or what interacts with the system and what the system should do
- A way to verify all requirements are captured
- A planning instrument



Who Reads Use Cases?

- Client team



Client

Users

- Developer team

Tester

Designer

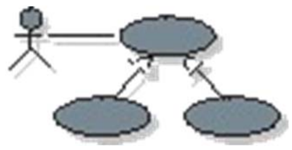
Requirements Specifier

Technical Writer

Project Manager

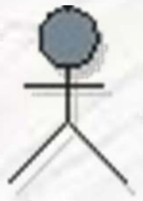
Benefits of Use Cases

- Give context for requirements
- Are easy to understand
- Facilitate agreement with customers
- Illustrate why the system is needed
  - Use cases: why the system is used
  - Actors: who/what wants to interact with the system

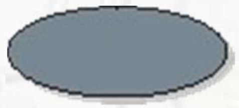


The idea behind use cases is to decide what the system will be used for <ins>before</ins> defining what the system is supposed to do.

Actors and Use Cases

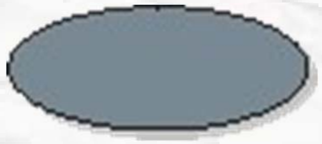


- Actor
  - Someone/something outside the system that interacts with the system



- Use Case
  - What an actor wants to use the system to do

What Is a Use Case?

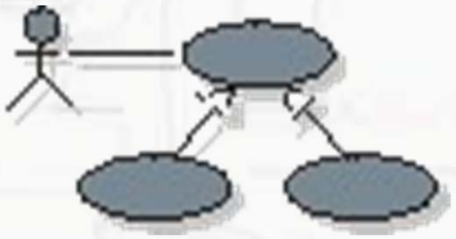


Use Case Name

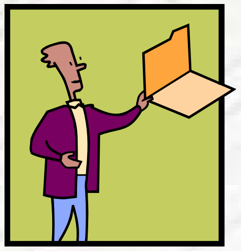
A use case defines a sequence of actions performed by a system that yields an observable result of value to an actor.

Use Cases Contain Software Requirements

- Each use case
  - Describe actions the system takes to deliver something of value to the actor
  - Shows the system functionally an actor uses
  - Models a dialog between the system and actors
  - Is a complete and meaningful flow of events from the perspective of a particular actor



Instances of Actors

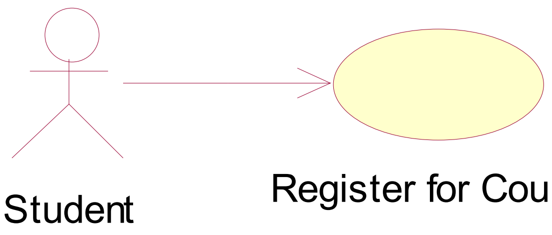


Sam

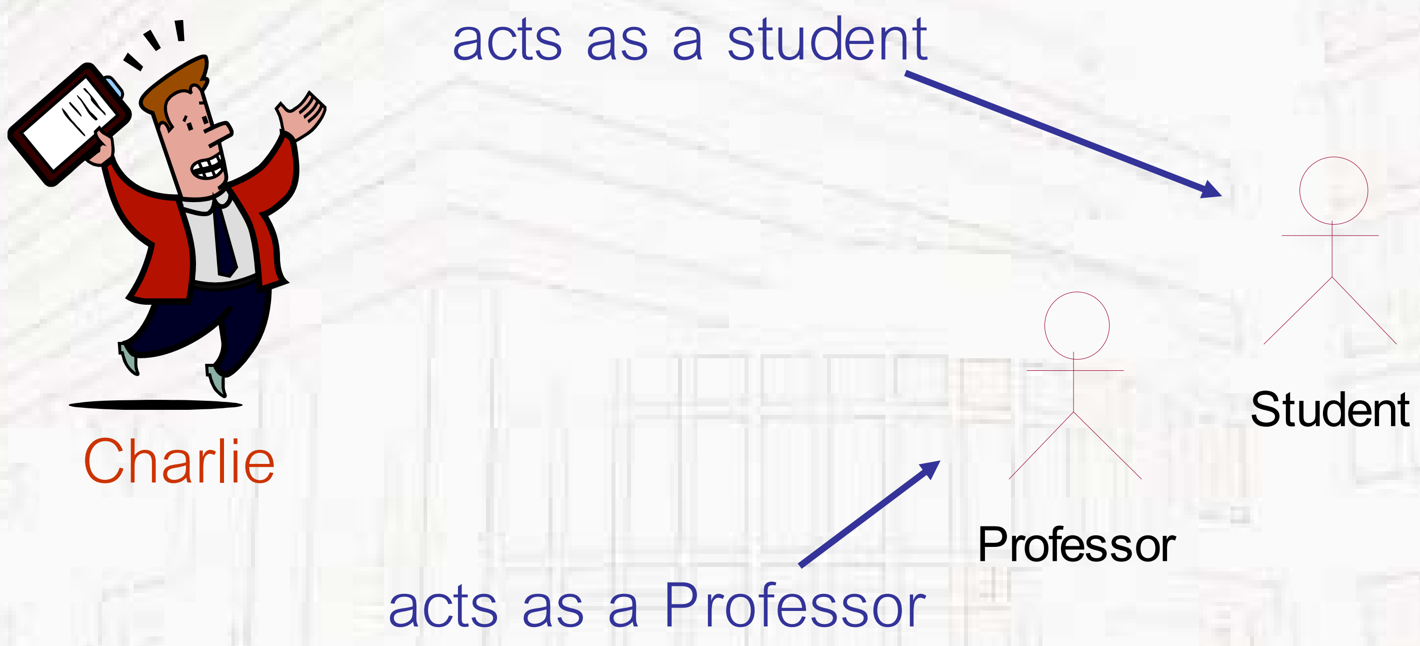
acts as a Student

Jody

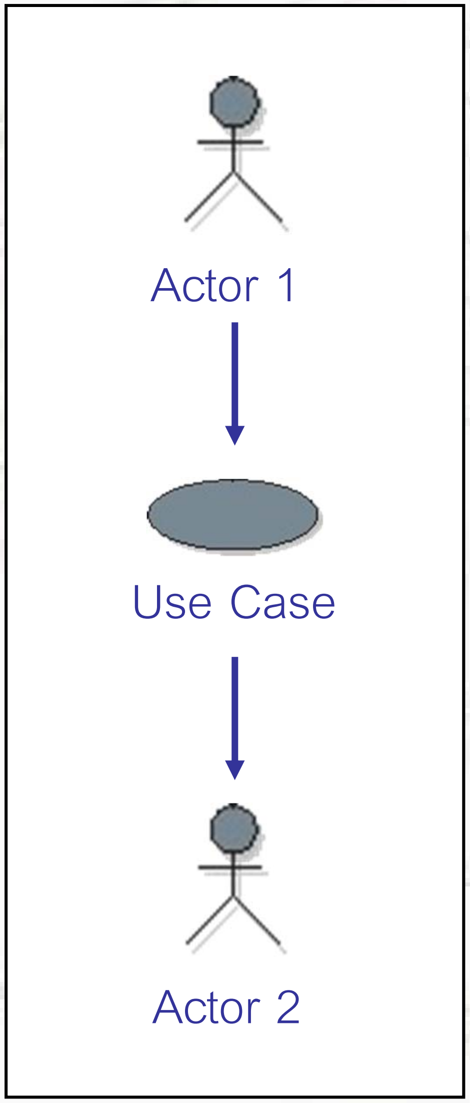
acts as a Student



A User can Act as Several Actors



Communicates-Association



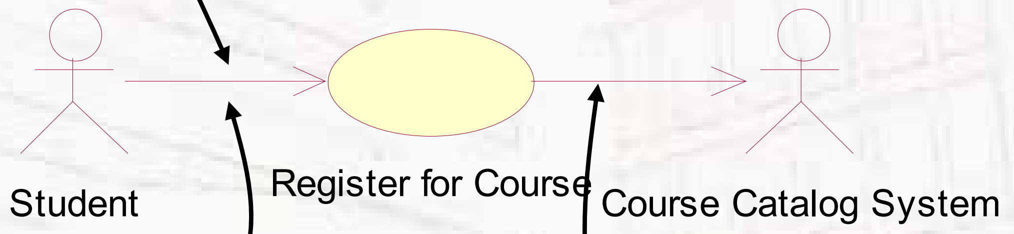
- A channel of communication between an actor and a use case
- A line is used to represent it
- An arrow indicates who initiates the communication

Each Communicates-Association is a Whole Dialog

Student logs on to system

System approves log on

Student request course info



System displays course list

Student select course

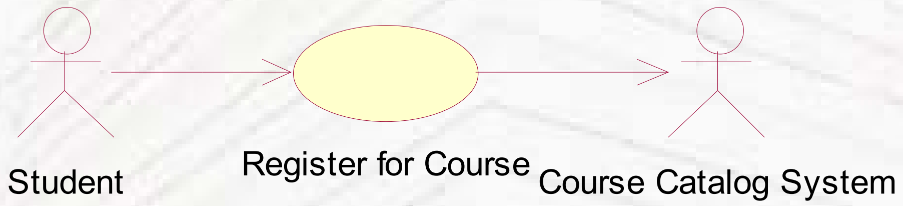
System confirms course availability

System displays approved schedule

System transmits request

Course Catalog returns course info

A Scenario is a Use-Case Instance



Scenario 1

Log on to system

Approve log on

Enter subject in search

Get course list

Display course list

Select course

Confirm available

Display final schedule

Scenario 2

Log on to system

Approve log on

Enter subject in search

Invalid subject

Re-enter subject

Get course list

Display course list

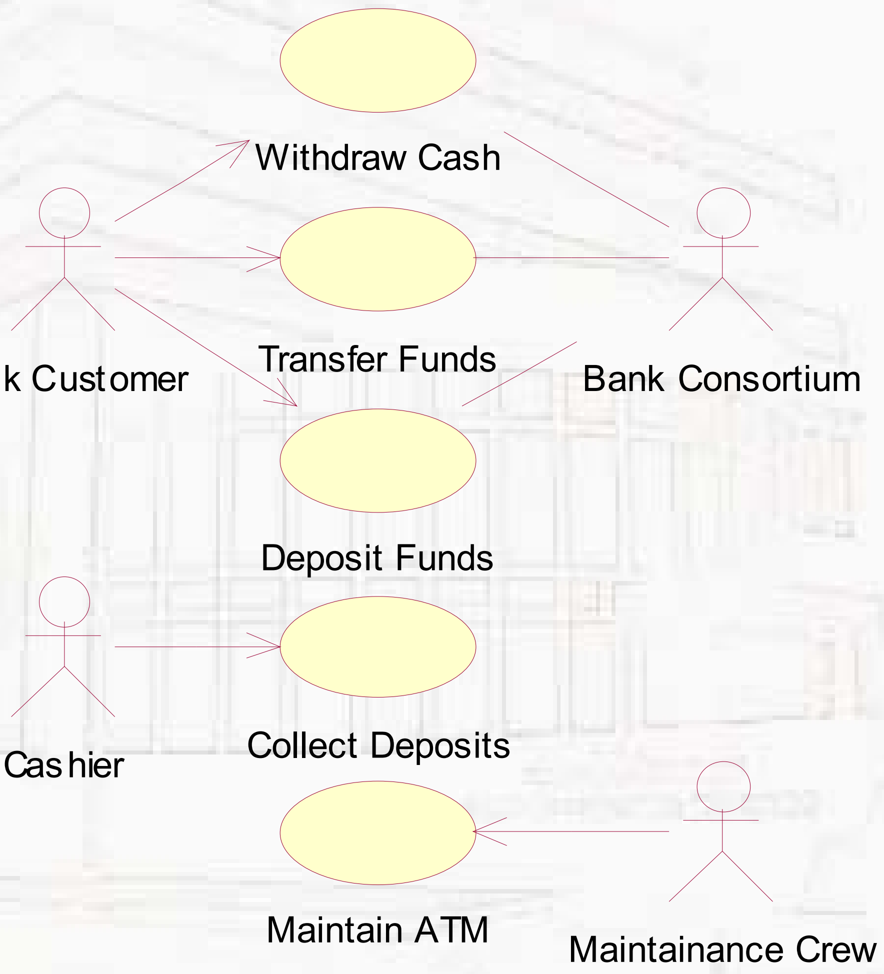
Select course

confirm available

Display final schedule

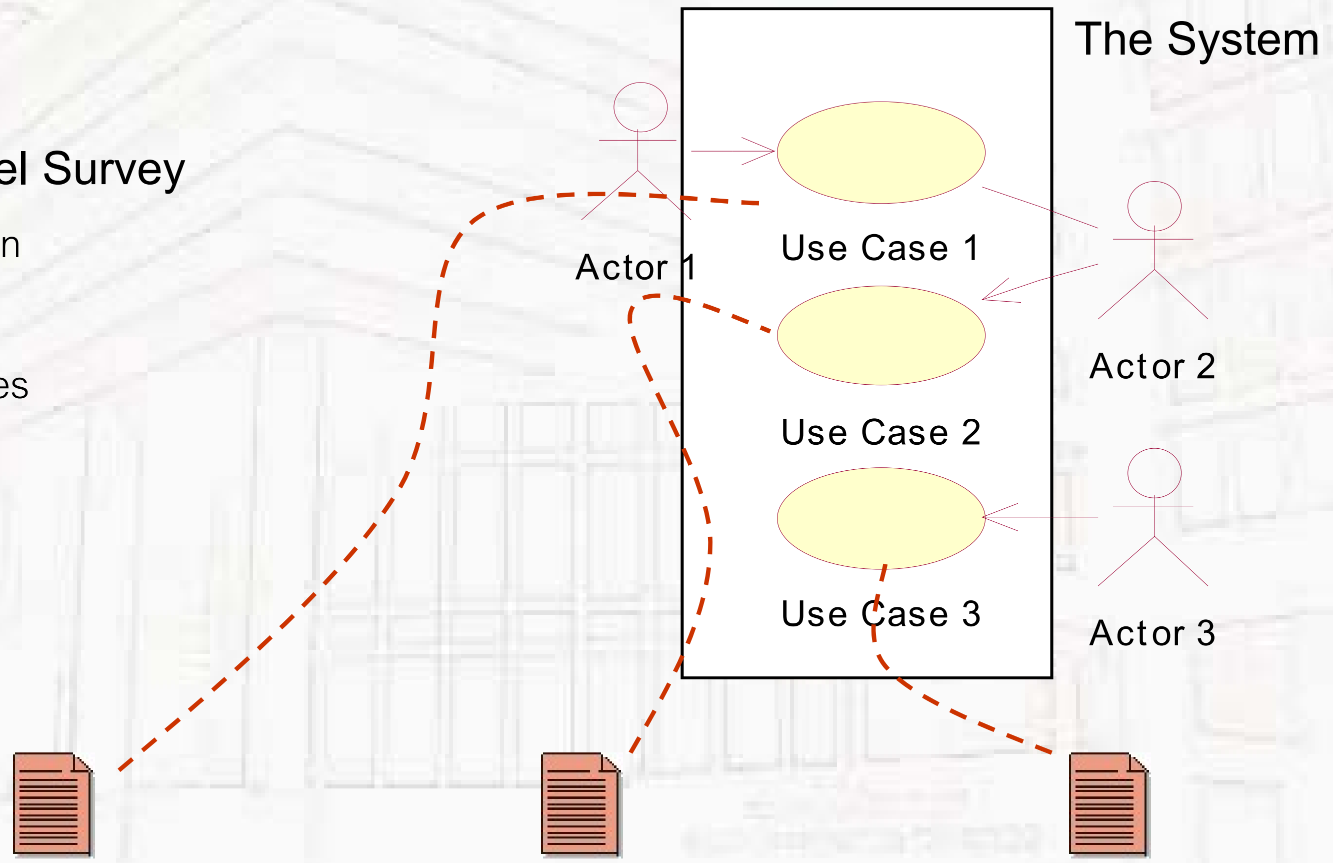
Use-Case Diagram

An Automated Teller Machine (ATM)



A Use-Case Model
Contains Diagrams and Text

Use-Case-Model Survey
- survey description
- list of all actors
- list of all use cases



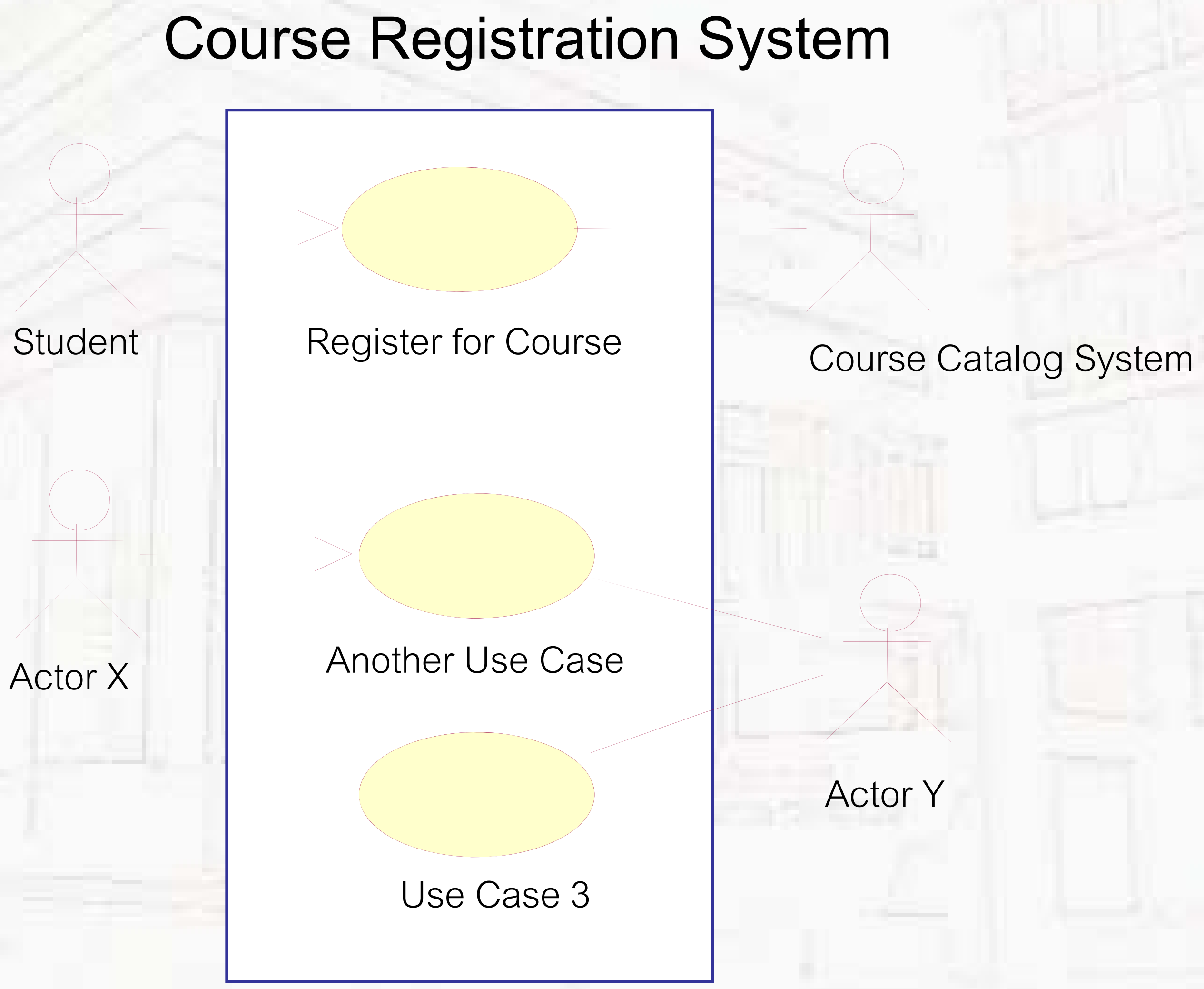
Use-Case 1 Spec
- brief description
- flow of events

Use-Case 2 Spec
- brief description
- flow of events

Use-Case 3 Spec
- brief description
- flow of events

Example:

Online Course Registration System



How Should I Name a Use Case?

- Indicate the value or goal
- Use the active form: begin with a verb
- Imagine a to-do list
- Examples of variations
  - Register for Courses
  - Registering for Courses
  - Acknowledge Registration
  - Course Registration
  - Use Registration System

Which variations show the value to the actor? Which do not?
Which would you choose as the use-case name? Why?

Use Case Tips

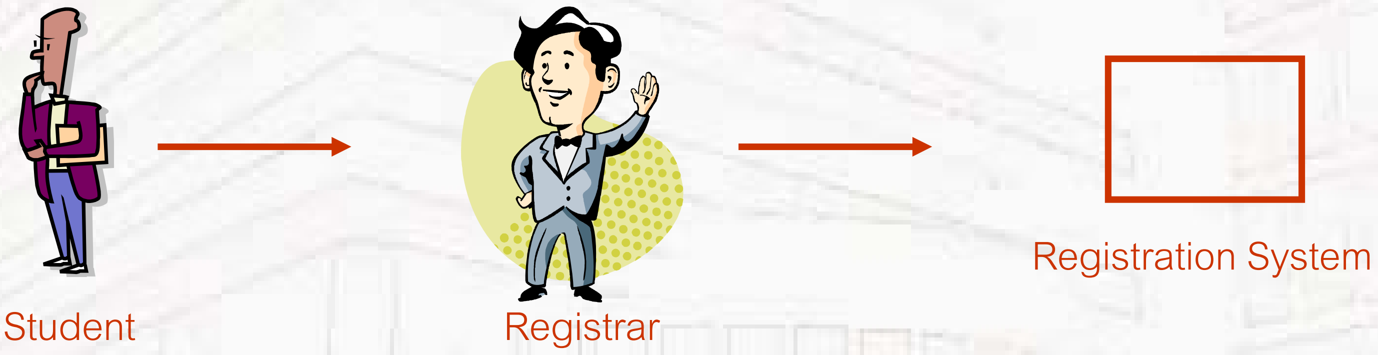
- Describe only the events visible to the actor:
  - What the actor does
  - What the system does in response
- Make use case provide value to an actor.
- Detail until everyone has a common understanding of the requirements, then stop.
- Sketch the user interface, but don't detail it
- Use agreed-upon terms and vocabulary.
- Use precise language.

Steps for Creating a Use-Case Model

1. Find actors and use cases
   - Identify and describe actors
   - Identify and describe use cases
2. Write the use cases
   - Outline all use cases
   - Prioritize and detail the use cases

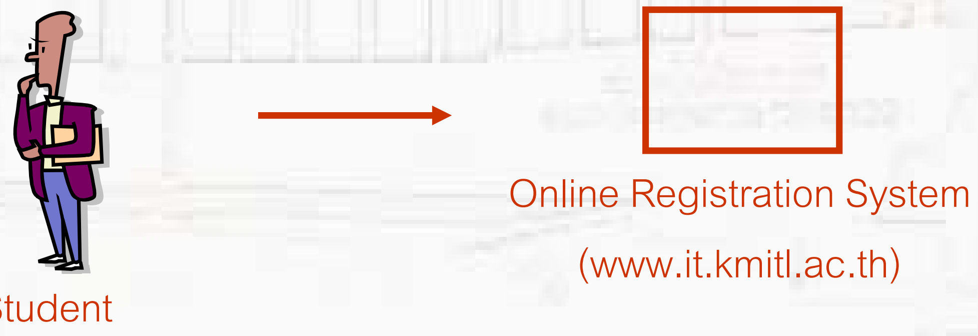
Find Actor

Who is pressing the keys (interacting with the system)?



The student never touches this system; the registrar operates it.

Or perhaps you are building an Internet application?

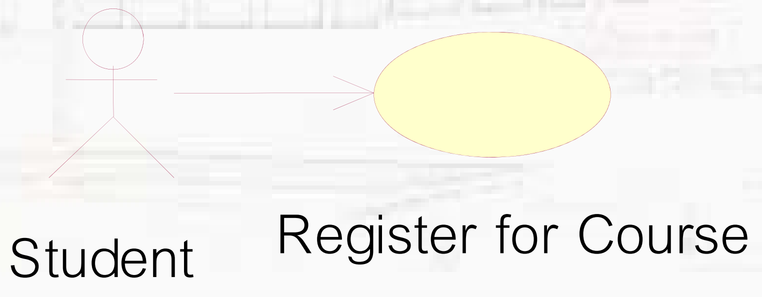


Identify Actors

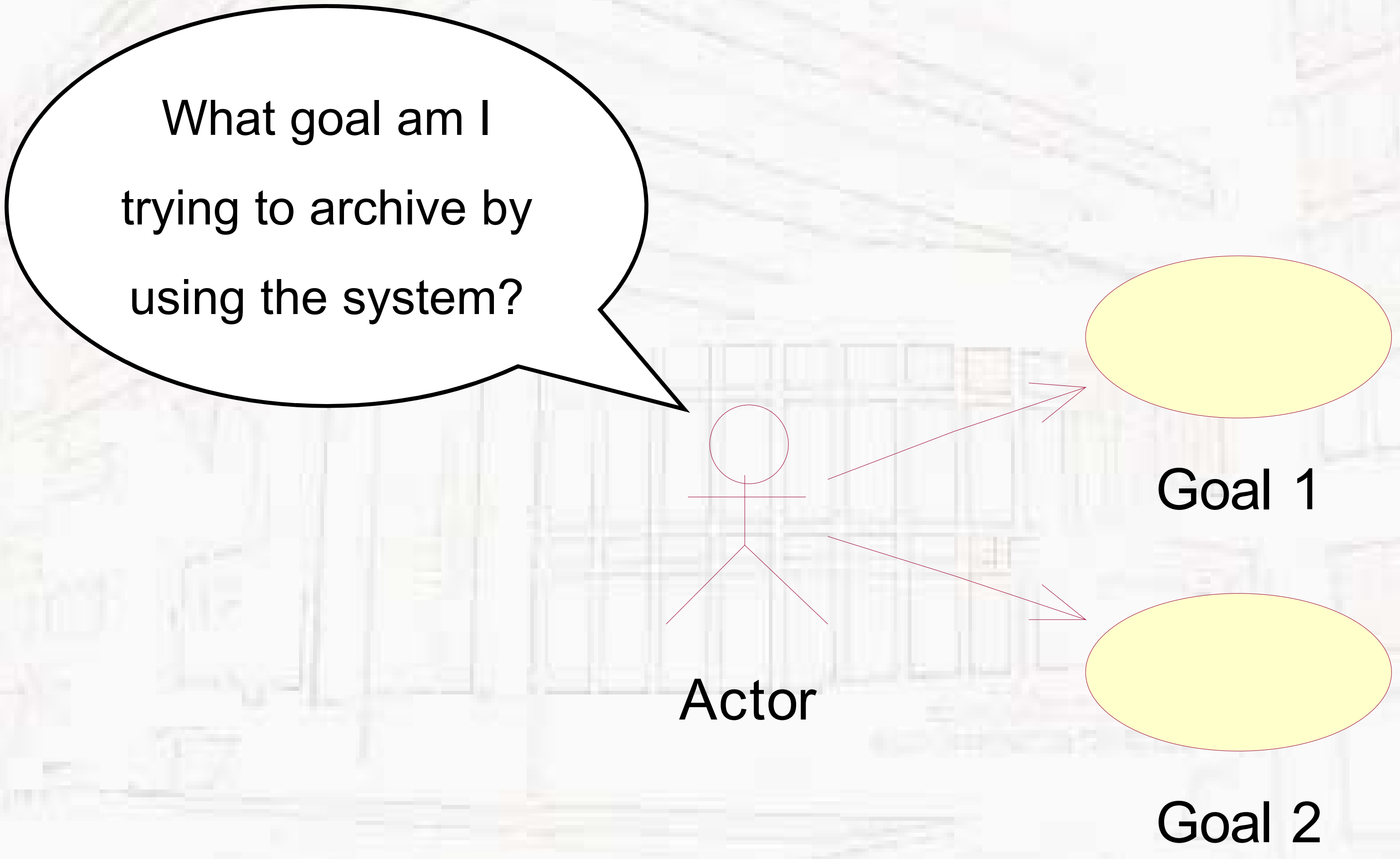
- Who/what uses the system?
- Who/what gets information from this system?
- Who/what provides information to the system?
- Where in the company is the system used?
- Who/what supports and maintains the system?
- What other systems use this system?

Description of an Actor

- Text
  - Name
  - Brief description
  - Relationships with use cases
- Example
  - Student
  - A person who signs up for a course



Find Use Cases

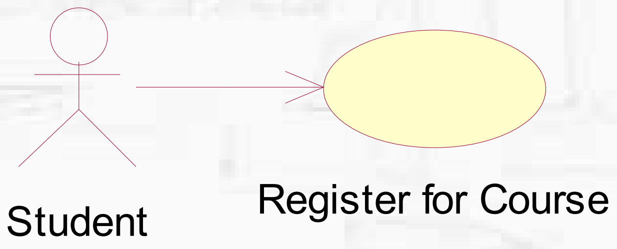


Identify Use Cases

- What are the goals of each actor?
  - Why does the actor want to use the system?
  - Will the actor create, store, change, remove, or read data in the system? if so, why?
  - Will the actor need to inform the system about external events or changes?
  - Will the actor need to be informed about certain occurrences in the system?
- Does the system supply the business with all of the correct behavior?

Describe of a Use Case

- Text description of a use case
  - Name
  - Brief description
  - Relationship with actors
- Example
  - Register for course
  - The student registers for courses. The student obtains course information prior to registering.



Functional Decomposition

- Is the breaking down of a problem into small isolated parts.
  - The parts:
    - Work together to provide the functionality of the system.
    - Often do not make sense in isolation.
- Use cases:
  - Are NOT functional decomposition.
  - Keep the functionality together to describe a complete use of the system.
  - Provide context.

Avoid Functional Decomposition

Symptoms

- very small use cases
- Too many use cases
- Use cases with no result of value
- Names with low-level operations
  - "Operation" + "object"
  - "Function" + "Data"
  - Example: "Insert Card"
- Difficult understanding the overall model

Corrective Actions

- Search or larger context

  "why are you building this system?"
- Put yourself in user's role

  "What does the user want to achieve?"

  "Whose goal does this use case satisfy?"

  "What value does this use case add?"

  "What is the story behind this use case?"

Checkpoints for Use Cases

- The use-case model clearly presents the behavior of the system; it is easy to understand what the system does by reviewing the model.
- All use cases have been identified; the use cases collectively account for all require behavior.
- All functional requirements are mapped to at least one use case.
- The use-case model contains no superfluous behavior; all use cases can be justified by tracing them back to a functional requirement.

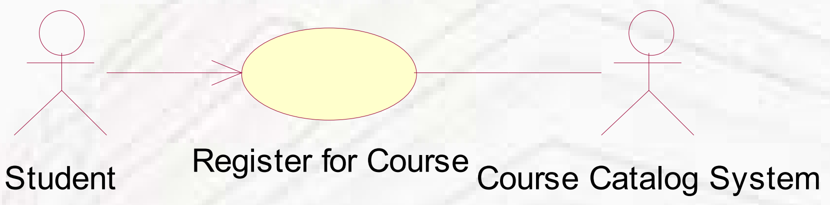
Checkpoints for Use Cases (cont.)

- Do the use cases have unique, intuitive, and explanatory names so that they cannot be mixed up at a later stage? If not, change their names.
- Do customers and users alike understand the names and description of the use cases?
- Does the brief description give a true picture of the use case?
- Is each use case involved with at least one actor?
- Do any use cases have very similar behaviors or flows of events?

Checkpoints for Actors

- Have you found all the actors? That is, have you accounted for and modeled all roles in the system’s environment?
- Is each actor involved with a least one use case?
- Can you name at least two people who would be able to perform as a particular actor?
- Do any actors play similar roles in relation to the system? If so, you should merge them into a single actor.

Diagram -> Outline -> Detail



Register for Course

Outline

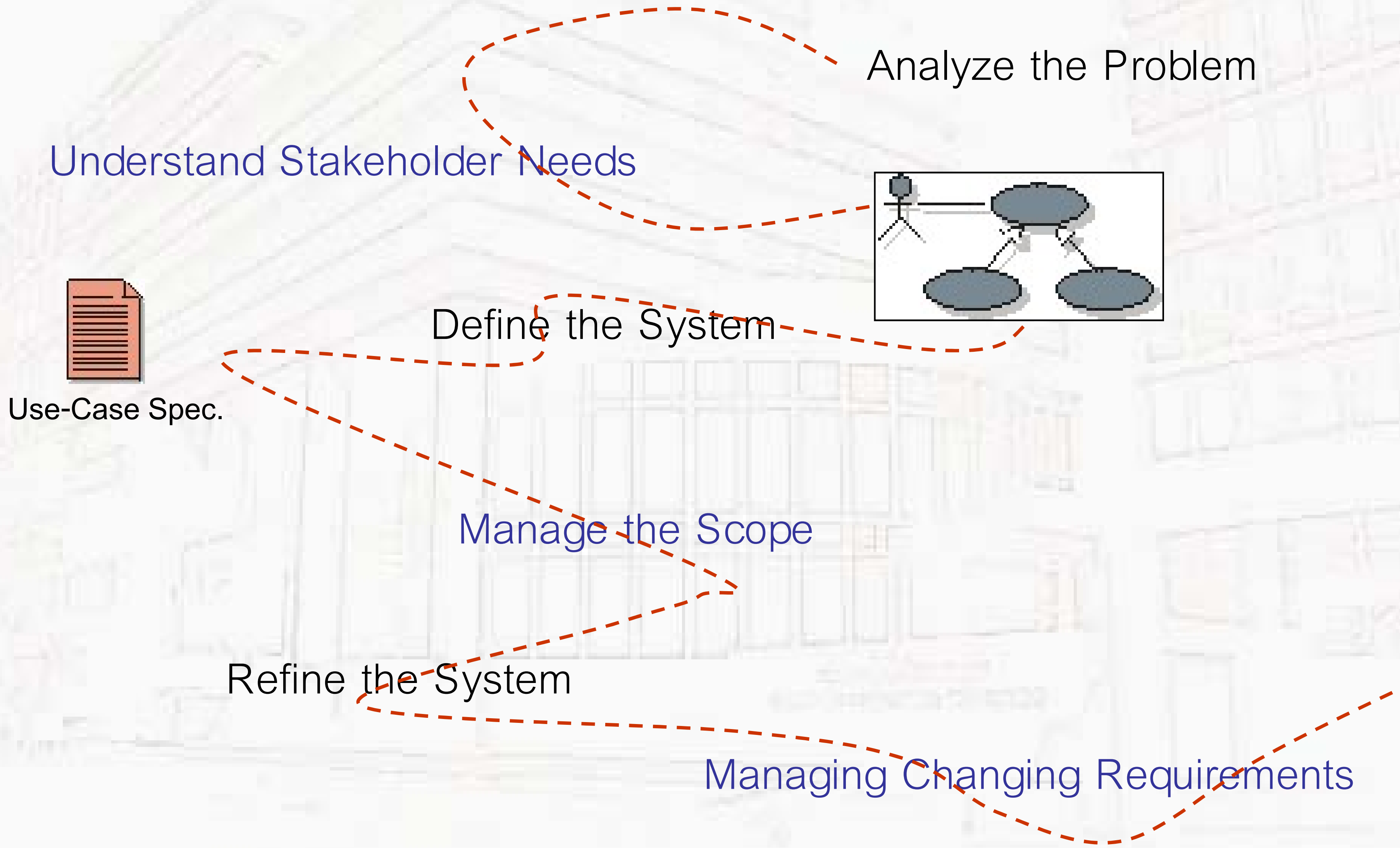
- Brief description
- Flow of events
  - Step-by-Step

Register for Courses

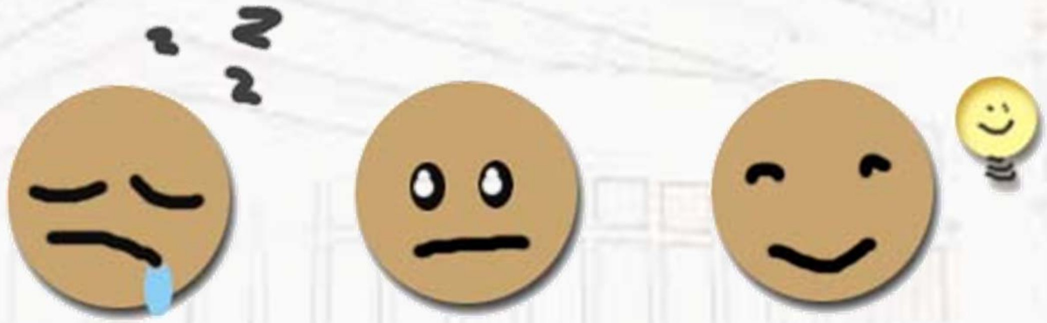
Use-Case Specification

- Brief description
- Flow of events
- Special Requirements
- Pre/Post Conditions

Where Do Use Case Fit into the RM Process?



Question ? ? ?



Review

1. What are the benefits of use-case modeling?
2. What is included in a use-case model?
3. How do you identify actors and use cases?
4. What is functional decomposition?
5. Why do we want to avoid functional decomposition?
6. What are some questions you can ask to test the quality of your model?