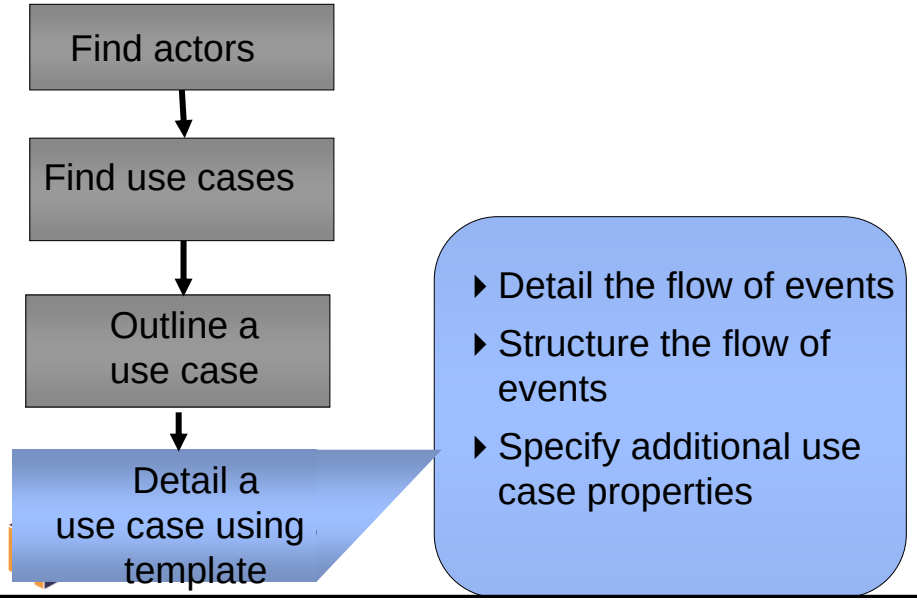
Use Case Documentation

Il-Yeol Song, Ph.D.
Professor
College of Information Science and Technology
Drexel University
Philadelphia, PA 19104
Song@drexel.edu
http://www.ischool.drexel.edu/faculty/song/

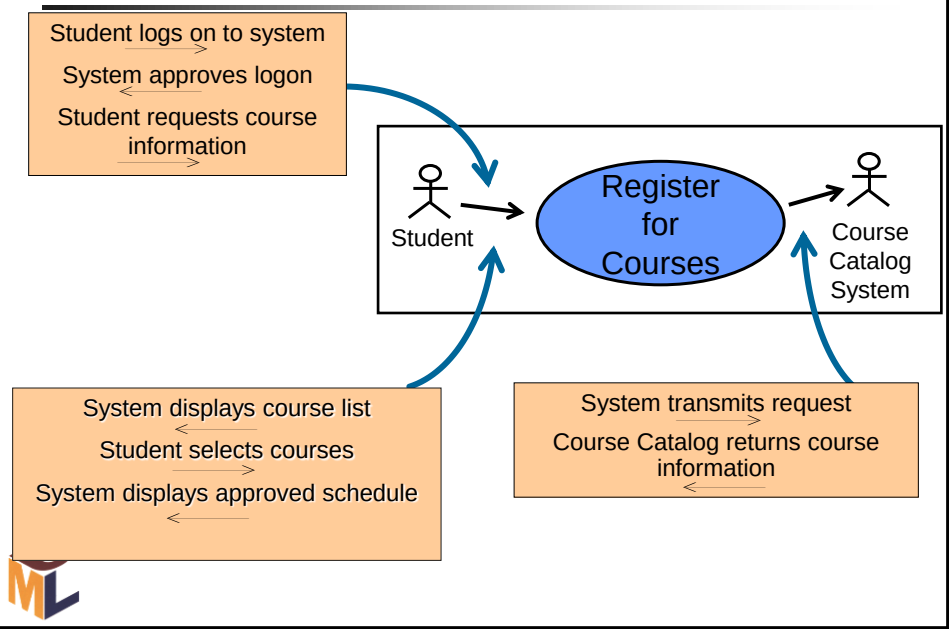
Objectives

- Use case documentation
  - Using use case template
  - Components of the use case template
  - Important rules for writing use case scenarios
  - Iterative development of use case descriptions
- Supporting use case template with activity diagrams

Process of writing use cases



Each communicates-association is a whole, 2-way dialog



Different Ways of Describing Use Cases

- Plain text
- Activity diagram: supporting use case descriptions
- ***Use case template***
  - A Use case diagram serves as a support for text but not vice versa
    - A bubble does not tell us the story

Levels of Use Case Descriptions

- Iterative development of Use case descriptions
  - Brief description
    - Summary statement
  - Intermediate description
    - Expands brief description with internal flow of activities
  - Fully Developed Description
    - Expands intermediate description for richer view

Iteration 1: Brief Statements

***Create new order* description**

When the customer calls to order, the order clerk and system verify customer information, create a new order, add items to the order, verify payment, create the order transaction, and finalize the order.

Brief Description of ***Create New Order*** Use Case

Iteration 2: Intermediate Description of *Telephone Order Scenario* for *Create New Order* Use Case

Flow of activities for scenario of *Order Clerk creates telephone order*

Main Flow:

1. Customer calls RMO and gets order clerk.
2. Order clerk verifies customer information. If a new customer, invoke *Maintain customer account information* use case to add a new customer.
3. Clerk initiates the creation of a new order.
4. Customer requests an item be added to the order.
5. Clerk verifies the item and adds it to the order.
6. Repeat steps 4 and 5 until all items are added to the order.
7. Customer indicates end of order; clerk enters end of order; system computes totals.
8. Customer submits payment; clerk enters amount; system verifies payment.
9. System finalizes order.

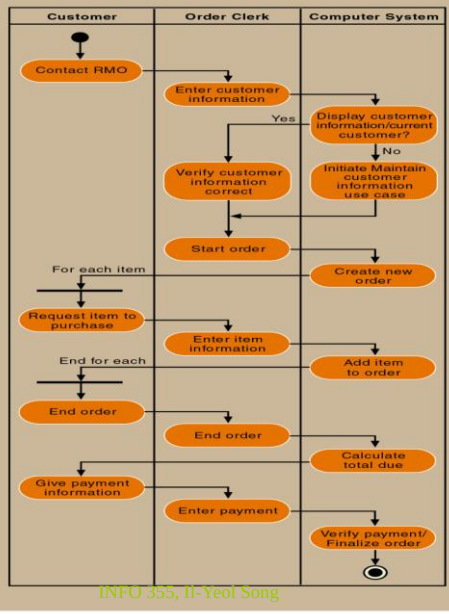
Exception Conditions:

1. If an item is not in stock, then customer can
   a. choose not to purchase item, or
   b. request item be added as a back-ordered item.
2. If customer payment is rejected due to bad-credit verification, then
   a. order is canceled, or
   b. order is put on hold until check is received.

Iteration 3: Fully Developed Description of *Telephone Order Scenario* for *Create New Order* Use Case

Use Case Name:	Create new order
Scenario:	Create new telephone order
Triggering Event:	Customer telephones RMO to purchase items from the catalog.
Brief Description:	When customer calls to order, the order clerk and system verify customer information, create a new order, add items to the order, verify payment, create the order transaction, and finalize the order.
Actors:	Telephone sales clerk
Related Use Cases:	Includes: *Check item availability*
Stakeholders:	Sales department: to provide primary definition Shipping department: to verify that information content is adequate for fulfillment Marketing department: to collect customer statistics for studies of buying patterns
Preconditions:	Customer must exist. Catalog, Products, and Inventory items must exist for requested items.
Postconditions:	Order and order line items must be created. Order transaction must be created for the order payment. Inventory items must have the quantity on hand updated. The order must be related (associated) to a customer.
Flow of Events:	**Actor** / **System**
	1. Sales clerk answers telephone and connects to a customer.
	2. Clerk verifies customer information.
	3. Clerk initiates the creation of a new order. — 3.1 Create a new order.
	4. Customer requests an item be added to the order.
	5. Clerk verifies the item (*Check item availability* use case). — 5.1 Display item information.
	6. Clerk adds item to the order. — 6.1 Add an order item.
	7. Repeat steps 4, 5, and 6 until all items are added to the order.
	8. Customer indicates end of order; clerk enters end of order. — 8.1 Complete order. 8.2 Compute totals.
	9. Customer submits payment; clerk enters amount. — 9.1 Verify payment. 9.2 Create order transaction. 9.3 Finalize order.
Exception Conditions:	2.1 If customer does not exist, then the clerk pauses this use case and invokes *Maintain customer information* use case. 2.2 If customer has a credit hold, then clerk transfers the customer to a customer service representative. 4.1 If an item is not in stock, then customer can a. choose not to purchase item, or b. request item be added as a back-ordered item. 9.1 If customer payment is rejected due to bad-credit verification, then a. order is canceled/or b. order is put on hold until check is received.

Example of Activity Diagram of the Telephone Order Scenario



Typical Use Case Specification

- Use case name
- Scenario
- Trigger
- Actor
- Goal (Purpose)
- **Overview & Scope**
- Level
- Stakeholders
- Preconditions
- **Postconditions**
- List of included use cases
- List of Extending use cases
- **Main Success Scenario (Happy Scenario)**
- Alternative Success Scenarios
- Exceptions (Erroneous scenarios)
- Others:
  - Priority
  - Frequency
  - **Business rules and data logic**
  - **Non-functional requirements**
  - Superordinates
  - Developer
  - Creation date &Last modificationd ate
  - **Glossary**
  - Comments

Use Case Description Details (I)

- **Use case name**
  - Verb-noun
- **Scenario** (if needed)
  - A use case can have more than one scenario (special case or more specific path)
- **Triggering event**
  - The event that initiates the use case
- **Actors**
  - One or more users who directly initiates the use case
- **Goal:** One phrase or sentence summary statement
- **Overview & Scope**
  - 3-4 sentences that summarizes the use case
- **Level:** Base use case, included, or extending use case
- **Stakeholders**
  - Anyone with an interest in the use case

Use Case Description Details (II)

- **Preconditions**
  - What must be true before the use case begins
- **Post conditions**
  - The conditions that must be true when the use case is completed
  - Use for planning test case expected results
- **Main Successful Scenario (The Happy Path)**
  - The main flow activities that go on between actor and the system
- **Other Successful Scenario**
  - The successful flow activities that go on between actor and the system
- **Exception conditions**
  - Where and what can go wrong

Use Case Description Details (III)

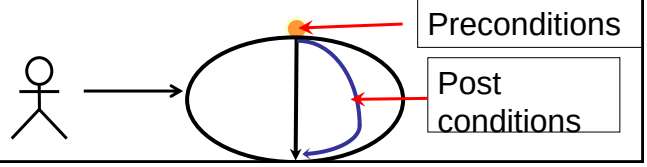
- **Priority:** Ranking or classification of use cases into first, second, optional
- **Frequency:** How often the use case will be executed
- **Business rules and data logic**
  - Business rules and data logic related to the use case such as Technical constraints (e.g., credit card authorization equipment)
  - Data variations (e.g., RFID, UPCs bar code technology)
- **Other NFRs**: Other NFRs (performance, usabilty, security, reliability) related to the use case
- **Superordinates**: If this use case is an included/extending, which primary use cases call it
- **Developer:** Developer of the use case description
- **Creation date & Last modification date**
- **Glossary:** Any definition of terminology used in this use case description
- **Other comments:** any other important assumptions or concern to be addressed later

Some Components of Use Case Descriptions

- **Precondition**
  - Conditions that need to be true **before** interactions can occur
  - *Conditions that have been set up by another use case*
    - *E.g, Log in*
  - Preconditions are not tested within the use case
  - Include only noteworthy assumptions (e.g., not *the system has power; the website is up*)

Some Components of Use Case Descriptions

- **Post-conditions**
  - The conditions that must be true when the use case is completed
  - Rules on writing post-conditions at the system level
    - What to write?
      - **Object creation/deletion**
      - **Attribute value changes**
      - **Linking objects**
  - How to write
    - Write in ***passive*** and ***past*** tense
    - Note: different from the book; uses only passive

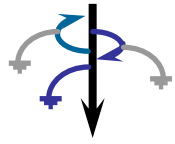


Some Components of Use Case Descriptions

- Examples of post-conditions
  - Use case: Create a new order
  - Post conditions
    - Order and OrderLineItem were created
    - Payment object was created
    - Order quantity was set
    - Order total price was computed
    - Order object was linked to Customer object
    - Payment object was linked to Order object

Some Components of Use Case Descriptions

- **Basic (Main Success) Scenario (Happy Path):**
  - Essential Main success scenario
  - Defer all conditional and branching statements to the alternate success scenarios
  - What event starts the use case?
  - How does the use case end?
- **Alternative flows**
  - Are there optional situations in the use case?
  - What variants might happen?
  - What may go wrong?
- **Unsuccessful Scenarios**
  - Two parts: the condition and the handling



Example of Flow Outline: Register for Courses

Basic Flow (Happy path)

1. Student logs on.
2. Student chooses to create a schedule.
3. Student obtains course information.
4. Student selects courses.
5. Student submits schedule.
6. System displays completed schedule .

Alternative Flows

A1. Unidentified student.
A2. Cancel.
A3. Cannot enroll because prerequisite not satisfied.
A4. Course Catalog System unavailable.
A5. Course registration closed.

Use case template

- There are many variations
- Adopt one model that suits your needs
- Example: use the attached template for your assignments and projects

Use Case Template: Header Part

USE CASE #	<< An arbitrary use case number for reference purpose>>
USE CASE Name	<< "Use Verb + Noun" form as the use case name>>
Scenario	<<Specific scenarios of the use case>>
Trigger	<< Which event starts the use case?>>
ACTOR	<< The actor who will be using this use case. You may list more than one actor here>>
Goal (1 phrase)	<< Write a phrase for the goal of the use case in the form of "To + verb + ….">>
Overview and scope	<< Write 2-4 sentences for the overview and main scope >>
Level	<< Primary or included or extended >>
Stakeholders	<<People who are concerned with the use case>>
Preconditions	<< Non-trivial pre-conditions to execute this use case>>
Postconditions in words (write in passive and past tense)	<< Successful end conditions in business terms. Write them in passive and past tense. Write 3 ideas: (1) What objects were created and deleted in this use case? (2) What attributes changed their values? (3) Which objects were connected/disconnected with which objects? >>
Included Use Cases	<< List of included use case names >>
Extending Use Cases	<< List of extending use case names >>

Use case template: The Main Success Scenario

USE CASE DESCRIPTION in numbered sequence (main scenario or the happy path) Reference "included use cases" in this section using INCLUDE *ius_name*	**Actor Action**	**System Action**
	1 Actor does this	2 System does that
	3 Actor does another	4 System does another
		5 INCLUDE <<included use case name>>
	6	7
	8 Customer pays by credit card	9 Process the credit card
		<< Add more rows if you need>>

ius_name included_use_case_name

Use Case Template: Alternative Success and Exception Scenarios

OTHER SUCCESSFUL SCENARIOS (Specify any ***successful*** variations of the *normal* execution path, including any extension points using EXTEND *eus_name*)	**Step**	**Branching Action**
	2a	<<Branching action of step 2 above>>
	2b	<< Another branching action of step 2 above >>
	8a Customer pays by cash	9a. System indicates cash and enter amount
	8b Customer pays by check	9b. System indicates check and enter amount
	<< Add more rows if you need>>	
UNSUCCESSFUL SCENARIOS (*erroneous* situations)	**Conditions**	**Actions**
	*a	Abort the transaction. <,Here * means this can be executed at any step>>
	8c. Customer is short of money;	Abort the transaction
	9c. Credit card validation fails	Ask to pay by another means or abort
	<< Add more rows if you	

Use case template: Other Non-Functional Requirements

Priority in scheduling	<<Classify the use case into first, second, third, depending on the importance. Or Primary/Secondary/Optional>>
Frequency	<<How often will this use case be performed?>>
Business rules and data logic	<< Any integrity constraints on data and business rules related to this use case>>
Other non-functional requirements	<< Optional, non-functional requirements such as performance, reliability, usability, and other design considerations etc.>>
Superordinates	<<If this is an included or extending use case, which primary use cases call this use case? If it is a primary use case, leave this row blank>>
Developer	<<Who developed this use case?>>
Creation date and last modified date	<< The version dates as stated >>
Glossary	<<*Italicized terms* are defined here or in the main Glossary>>
Other Comments	<<Any extra comments, time constraints, etc.>>

Use case template example

- VRS: Process Rents

Leave the user interface out of the use case

- Use cases are independent of the user interface

How much detail in a use case?
Enough to satisfy all stakeholders that their interests (requirements) will be satisfied in the delivered system.

Words to Avoid		
Click	Drag	Open
Close	Drop	
Button	Drop-down	
Pop-up	Scroll	
Browse	Record	

Words to Use	
Prompts	Chooses
Initiates	Specifies
Submits	Selects
Starts	Displays
Informs	Create

Writing Steps

- **No ambiguous adjectives**. Don't use any adverbs or adjectives like *appropriate, required, relevant or sufficient*. Using these adjectives make the sentence ambiguous.
- **Use *active* sentences**
  - Say: "The System validates …"
  - Instead of: "The choice is validated …"
- ***Avoid compound sentences***: Instead, use simpler grammar
- ***Each step must be a goal-driven movement***, *not interface level*:
  - Say: "user enters name and address"
  - Instead of: "system asks for name, user enters name, system ask for address, user enters address"

Seven Step Method for Developing UCD

Each step should be *only one logical step* towards the use case goal

- State one logical step at a time. Do not combine two different steps that require a system interaction in one step.
- For example: "**Search products and select the item**" are two steps and should be documented in two steps because the system needs to display the output before the second action.

Use the glossary effectively

Use Case

5. Enter Customer Information

The system prompts the Customer to enter their *Customer Details*.

The Customer enters the *Customer Details*.

The Customer creates the account.

Glossary

Customer Details
Information that uniquely identifies and provides contact information for a customer located in the U.S.A. The information consists of **Name, two address lines, city, state, ZIP code,** and **daytime phone number**.

Implementation

Exercise: Write preconditions and postconditions for use case "process payment" in ROS

The *Fancy Eight* is a newly opened five star restaurant in Walnut St. in Philadelphia. We would like to develop an automated restaurant order system (ROS) for *Fancy Eight*. Initially, we will focus on only reservation, seating, orders, payments, and menus. That is, a customer is assigned to a table by a hostess. A hostess may reserve tables for customers. A waiter or waitress will bring a menu, then the customer picks any number of courses, which may include any combination of appetizer, salad, soup, wine, main dish, drinks, and desserts. After the meal, the waiter or waitress brings a bill. The customer can pay by cash, check or a credit card. The owner may add/delete menus and change prices occasionally. The current system will not include any functionalities on inventory, purchase of supplies, or kitchen work yet.

Exercise for Use case description: ROS: Process payments

Levels of Use Case Descriptions

- Business level
  - Describe the internal flow in English at Business level
- System level
  - Describe the detailed interactions between the actor and the system using the vocabulary of the class model
  - After developing a class diagram, rewrite the business level into the system level

Check Lists for Use Case Specification

- Does each primary use case have a use case description?
- Does each use case have one primary scenario, one or more variations of successful scenarios, and failure scenarios identified?
- Are scenarios detailed enough to visualize the steps of the scenario?
- Is each step specific? (e.g., 'add customer data' is not specific) unless the term is defined in glossary.
- Is each step of a use case description conceptually *only one step* that can be logically done? (e.g., '*search customer* and *add their data*' must be split.)
- Does each *included* use case or *extending* use case have its own use case template?

Check Lists for Use Case Modeling

- **USE CASE DESCRPTION**
  - Are preconditions those that are need to be true before the execution of the use case?
  - Are post-conditions written in past and passive form?
  - Are all post-conditions identified?
    - Identify *all the objects* that need to be created or deleted?
    - Identify *all the attributes* that need to change their values?
    - Identify all the *connection/disconnection* between objects?
  - Are there any duplicate use cases or relationships?

A Summary of Use cases

- A use case is a functionality that achieves a goal of an actor
- The set of use cases specifies the functional requirements of the system.
- Use case description describes the detailed interaction between a user and the system.
- Walk the use case descriptions
- Rank the use cases and tackle high-ranking use cases first
- Provides a basis for performing system tests
  - Remember *use case driven* system development