Large Scale Systems Design
G52LSS

Lecture 13 – Use Case Analysis

- Refine Requirements
- Use Case Diagrams and Use Cases
- Steps of Use Case Analysis
- Example: University Registration System

Learning outcomes: understand the importance of continuously refine requirements; identify some tools for use case analysis; apply the steps for use case analysis; demonstrate use case analysis for a small case.

Refine Requirements

To better understand processes and data within the system and to help refining system requirements the following can help:

- Summary of business activities
- Use-case analysis: user stories, use case diagrams, use cases

The requirements specification phase in the SDLC has a clear purpose:

to define requirements that match the real users’ needs

Summary of Business Activities

Short and easy to understand description of all the business activities that should be incorporated into the system to be developed.

The key difference between the summary of business activities and a user story is in their scope.

The summary of business activities describes all business processes within the system.

A user story refers to a specific business process or function within the system.

User Stories

Helpful technique from extreme programming (XP) to identify valuable business user requirements.

Narrates from the users’ perspective, the way in which business processes are performed.

Short and easy to understand so that developers get an overall picture that is clear enough to estimate what it takes to complete the project.

Several user stories may be needed to describe the various processes in the system to be developed.

User stories help developers to understand what happens between the users and the system.

Users and developers have different backgrounds and points of view. Therefore, user stories is a good way of communication between developers and users.

Purpose and Characteristics of Good User Stories

- Common language
- Simple sentences
- Starting even and ending
- Notions from known vocabulary

- Interactions between roles and system
- Notions are transformable to code and constructs
- Changes in story are easy to trace in the code

Reading good stories is easy but writing them is not straightforward.
Good user stories can follow the SVO (subject-verb-object) format plus some explanation of the terms used in the story.

For example:
- Student enters the ID
- System verifies the ID is valid (where ID is a 7-digit number starting with 4 and unique for each student)
- System shows list of available modules
- Student selects modules to register
- System validates that chosen modules are allowed

The development cycle of user stories
1. Users write initial stories
2. Users meet with developers and together write clarified user stories with notions
3. Developers translate the clarified stories into code.
   Stories are further clarified with users when required.
   Test stories are also written
4. Developers make sure that the test stories are fulfilled and hand system to users
5. Users verify system and possibly write corrected stories and new stories

Use Case Diagrams and Use Cases

The use case diagram shows the system’s behavior together with the key actors for a specific scenario.

The elements of use case diagrams are:
- Actors
- Use cases
- System boundary
- Connections
- Extend relationships
- Include relationships

Example 13.1 The following is the use case diagram for an appointments system in a surgery.

A use case shows the behavior of a specific functionality of the system and consists of a set of possible sequences of interactions between the user and the system.

Basic information
- Name, number and brief description
- Trigger – event that causes the use case to be
   - External trigger – some from outside the system
   - Temporal triggers – time-based occurrences
- Viewpoint in a use case should be consistent (from same actor)

Major inputs and outputs
- Sources and destinations
- Goal is to be all inclusive

Example 13.2 The following is one use case for the appointments system in a surgery in which the main actor is the patient.
**Steps of Use Case Analysis**

Following the creation of a use case diagram, then cycle through the steps below in an iterative manner.

1. Identify the **major use cases**
   - Use one use case form for each use case
   - If more than nine, group into packages
   - Ask **who, what, and where** about the tasks and their inputs and outputs

2. Identify the **major steps** within each use case
   - For each use case, fill in the major steps needed to process the inputs and produce the outputs
   - Ask **how** about each use case

3. Identify **elements within steps**
   - For each step, identify its triggers, its inputs and outputs
   - Ask **how** about each step

4. Confirm the use case.
   - For each use case, validate that it is correct and complete
   - Ask the user to execute the process using the written steps in the use case, that is, have the user role-play the use case

---

**Example: University Registration System**

Exercise D, (Dennis et al. 2006, chapter 5). A University Registration System should enable staff of each academic department to examine the modules offered by their department, add and remove modules, and change the information about them (e.g. the maximum number of students permitted).

It should permit students to examine currently available modules, add and drop modules to and from their schedules, and examine the modules for which they are enrolled. Department staff should be able to print a variety of reports about the modules and the students enrolled in them. The system should ensure that no student takes too many modules and that students who have any unpaid fees are not permitted to register (Note: assume that a fees data store is maintained by the university’s financial office; this data store is accessed by the registration system but the fees data store is not modified by the registration system).

---

**Step 1. Identify Major Use Cases:**
- Maintain information about available modules
- Enrol student in module
- Produce reports about modules and student enrolments

For each use case identify the basic information, inputs and outputs

**Step 2. Identify the major steps within the use case**

**Step 3. Identify elements within the steps**

**Step 4. Confirm each use case with the user**
A simple **user story** for the case in which a student enrols in a module:

**Student access the system.** If the student has a valid username and password, allow the student to sign in, otherwise display an explanatory message.

**Student requests list of available modules.** Search the fees file to determine whether the student is permitted to register or not.

**System shows list of available modules.** Displays a list of available modules according to the fee paying status of the student.

**Student request enrolment in selected module.** Check if the enrolment is permitted according to the number of modules in which the student is already registered.

**System confirms enrolment.** Ask the student to confirm the enrolment. Display confirmation of enrolment and update the enrolment file. Provide the student with a confirmation code for the enrolment.

**Student requests dropping a module.** Display updated information and ask the student to confirm the drop. Update the enrolment file. Provide the student with a confirmation code.

**Student continues with more enrolments and drops.** If the number of modules in which the student is enrolled has not reached the maximum permitted, give the student the option to enrol in other modules. If there are at least one module in which the student is enrolled, give the student the option to drop modules.

**System shows current student’s schedule.** Display student’s schedule for modules enrolment. Give student the options of printing, saving and emailing the current schedule.
Students are permitted to access the list of available modules if they do not have unpaid fees.

Students make request for enrolment in a module of their choice. If the student has not reached the maximum number of modules permitted then the enrolment is processed. Students are notified of the enrolment.

Students make request for dropping a module of their choice. Students are notified of the change in their enrolment record.

Students can access their schedule and they can print, email or save it in a file.

**Additional Reading**

Chapter 5 of (Dennis et al., 2006)

Chapter 7 of (Hoffer et al., 2005) pages 225-233.