

OO Analysis: Use Case Description

Introduction

Use cases model the user's view of the functionality of a system, i.e. what it does that is of value to the user. The use case model structures the system into user's view of its main tasks. Use cases are normally presented in a graphical form as use case diagram, which is supported by textual descriptions (use case and actor descriptions, and scenarios).

A use case describes a cohesive piece of the system's functionality as the user perceives it. The user may see it as a task that he uses the system to achieve, one of the jobs that make up his daily workload, or it may produce a list or a report that he gets from the computer. A use case is a complete end-to-end use of the computer, a complete path through the system. A use case must deliver some benefit to the actor associated with it.

Each use case will have several scenarios associated with it. Some will be successful, some will not. The software developer needs to be aware of all possible scenarios because the system must be able to cope with them all and respond appropriately. Scenarios can be produced to illustrate what happens typically and interesting exceptions. A use case description is used to describe the use case in general terms and document the main variations from the norm.

Use Case Descriptions

The use case description is a narrative document that describes, in general terms, the required functionality of the use case. Typically, it describes goal of the use case and gives a generic description of what usually happens, the normal course of events, adding a brief description of any minor variations. Use case should be written in such a way that it encompasses every sequence of events, every scenario, relating to the use case.

The description is written in terms of what the system should do, not how it should do it. A use case describes the system as the user sees it.

High Level Format: Normally in the early stages of software development, when no detailed decisions have been made about the design the use case is described in High Level format. In this situation it is enough to have short unstructured descriptions. These descriptions need only document the purpose of the use case, the actors involved and give a general overview of what happens. Even in the high-level format, the description must name the use case, main actor and goal of the user case in unambiguous terms. For example, in "Hire-n-Ride-Bike" case study, the main use related to bike issue activity, high level format description is listed below:

Use case: Issue bike
Actors: Receptionist
Goal: To hire out a bike

Description:

When a customer comes into the shop they choose a bike to hire. The Receptionist looks up the bike on the system and tells the customer how much it will cost to hire for a specified period. The customer pays, is issued with a receipt, then leaves with the bike.

From the following description of a library system, we first identify use case and actors and then write high level description of the main use case.

A library lending information system is a simple example to record books issued to and returned by the users. It is used at the gate of a library. It includes a computer, a bar code scanner, and software to run the system. The focus is on the issues relevant to software development

Actor: Book Borrower, Library Assistant

Use Cases: Borrow Books, Return Books

High level description of Borrow Books use case:

Use Case: Borrow Books

Actors: User, Library Assistant

Type: Primary

Description: A User arrives at the lending counter with books to borrow. The Library Assistant records the books in the User's name. The User leaves the counter with the books and gate pass.

Expanded Format: In early design situation it is useful to have more detailed structured descriptions known as expanded format. In this situation we write only the most critical use cases in expanded format, so as to judge the complexity of the task. Write real use cases if concrete descriptions are considered necessary to fully comprehend the system. This description is more detailed and structured than the high-level use case description. It should document:

- what happens to initiate the use case.
- which actors are involved.
- what data has to be input.
- the use case output.
- what stored data is needed by the use case.
- what happens to signal the completion of the use case.
- minor variations in the sequences of events.

In expanded description, you may include sections for preconditions and post conditions. For expanded description of use cases, designers have created templates like the one listed below:

Name: Use case name

Actors: that communicate with the use case

Brief Description/Summary

Precondition: prerequisite for successful execution

Postcondition: system state after successful execution

Error situations: errors relevant to the problem domain

System state: on the occurrence of an error

Trigger: events which initiate/start the use case

Standard process: individual steps to be taken

Alternative processes: deviations from the standard process

Here we mention and use vending machine example (in lecture notes "OO Analysis Basics - Use Case Model/Diagram Creation") for which we have identified following use cases and actors:

Actors: Customer, Agent/Operator

Use Cases: Buy Drink, Restock

Now we describe Buy Drink in expanded format as below:

| | |
|-----------------------------------|---|
| Use Case: | Buy Drink |
| Primary Actor: | Customer |
| Goal in context: | Customer buys a drink |
| Stakeholders and Interests | |
| Customer : | wants to buy a drink |
| Vendor: | wants to get paid for any goods dispensed |
| Preconditions: | Machine is ready & awaiting interaction |
| Minimum guarantee: | Customer gets his coins back if no drink dispensed |
| Success guarantees: | Customer has the drink; vending machine has the money for the drink; Vending machine provides no change |

Main Success Scenario

1. Customer inserts sufficient money into the coin slot
2. Customer pushes one of the three select buttons
3. The selected drink is dispensed
4. The pushed select button lights up for 5 seconds
5. Coins are taken by the machine

Extensions

- 1a. Customer inserts no or insufficient money. Pushing a selection button has no effect
- 2a. Customer fails to push a button within 1 minute from insertion of the last coin. Coins are returned
- 2b. Customer selects beverage out of stock. Coins are returned

Now we use another example (in lecture notes “OO Analysis - Creating a Use-Case Diagram (with Generalization, Include, and Extend relationships) for which we have identified use cases and actors and created use case diagram. In the following we describe two use cases “Reserve lecture hall” and “Print certificate” in expanded form:

| | |
|---|--|
| Name: | Reserve lecture hall |
| Actors: | Employee |
| Short Description: | An employee reserves a lecture hall at the university for an event |
| Precondition: | The employee is authorized to reserve lecture halls |
| Postcondition: | A lecture hall is reserved |
| Error situations: | There is no free lecture hall |
| System state in the event of an error: | The employee has not reserved a lecture hall |
| Trigger: | Employee requires a lecture hall |
| Standard process: | |
| | 1) Employee logs in to the system |
| | 2) Employee selects the lecture hall |
| | 3) Employee selects the date |
| | 4) System confirms that the lecture hall is free |
| | 5) Employee confirms the reservation |
| Alternative processes: | |
| | 4') Lecture hall is not free |
| | 5') System proposes an alternative lecture hall |
| | 6') Employee selects alternative lecture hall and confirms the reservation |

Name: Print certificate
Actors: AdminEmp
Short description: On request from a student, an employee prints the student's certificate for a course on paper
Precondition: All data relevant for the certificate has been sent and the student has been graded
Postcondition: Certificate is available to the student in printed form
Error situations: Printer is not working
System state in the event of an error: Certificate is not printed
Trigger: Student requests printed certificate
Standard process:
1) Student enters the student office and requests a certificate
2) AdminEmp enters the student's registration number
3) AdminEmp selects the certificate
4) AdminEmp enters the print command
5) System confirms that the certificate was printed
6) Certificate is handed over to the student
Alternative processes:
1') Student requests certificate via e-mail
2-5) As above
6') Certificate is sent by post

Actor Descriptions

An actor description briefly describes the actor in terms of role and job title. An actor represents one particular way of using the system; an actor represents the role someone plays in the use case – e.g. the Receptionist issues the bike. It may be that several people can play this role. For our case study “Hire-n-Ride-Bike” we describe two actors as listed below:

The Receptionist uses the system to answer queries about bike availability and cost, to issue a bike for hire and to register a bike return. The Receptionist can be the Shop Manager (Aslam), any of the mechanics or the owner (Baber).

The Administrator uses the system to maintain lists of customers and bikes. The administrator can be the head mechanic, shop manager or shop owner.