

USE CASE DESCRIPTION

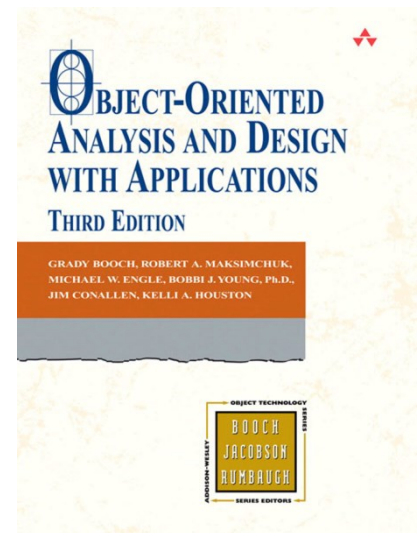
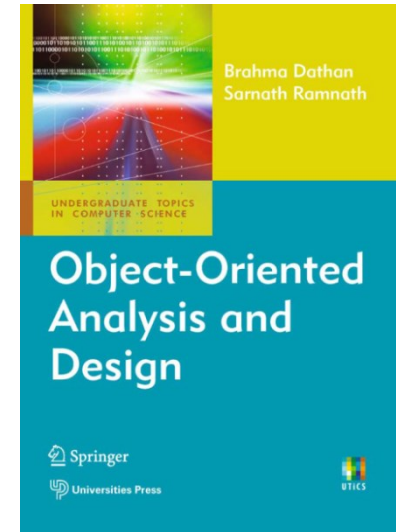
Danang Wahyu Utomo

danang.wu@dsn.dinus.ac.id

+6285 740 955 623

Reference

- ▶ Brahma Dathan, Sarnath Ramnath – **Object-Oriented Analysis and Design** (2011)
- ▶ Grady Brooch, Robert A Maksimchuk, Michael W. Engle, Robbi J. Young, Jim Conallen, Kelli A. Houston – **Object-Oriented Analysis and Design With Applications** Third Edition (2007)



Use Case Description

- ▶ Describe basic functions of the system using words
 1. What the user can do
 2. How the system responds



Use Case Description

- ▶ Each use case describes one and only one function
- ▶ But may have several paths that the user can take to accomplish that single function



Type Of Use Cases

Overview Vs Detail

▶ Overview

- High level overview of requirement
- Allows users and analysts to agree on major requirements
- Created early, documents only basic info
name, ID, primary actor, type, brief description

▶ Detail

- Once user and analyst agree
- Documents all information needed for the use case



Type Of Use Cases

Essential Vs Real

▶ Essential

- Describes only essential issues needed to understand the required functionality (e.g. make appointment)

▶ Real

- goes further and describes a specific set of steps (e.g. make entry into outlook database)



Elements of Use Case Description

- ▶ Contains all information needed to build use case diagram
- ▶ But expresses it less formally
- ▶ Three basic parts :
 1. **Overview** information
 2. **Relationships**
 3. Flow of **events**

Element of Use-Case Description

Use Case Name:	ID:	Importance Level:
Primary Actor:	Use Case Type:	
Stakeholders and Interests:		
Brief Description:		
Trigger:		
Relationships: (Association, Include, Extend, Generalization)		
Normal Flow of Events:		
Subflows:		
Alternate/Exceptional Flows:		



Use Case Name: Make appointment	ID: 2	Importance Level: High
Primary Actor: Patient	Use Case Type: Detail, essential	
Stakeholders and Interests: Patient - wants to make, change, or cancel an appointment Doctor - wants to ensure patient's needs are met in a timely manner		
Brief Description: This use case describes how we make an appointment as well as changing or canceling an appointment.		
Trigger: Patient calls and asks for a new appointment or asks to cancel or change an existing appointment.		
Type: External		
Relationships: Association: Patient Include: Make Payment Arrangements Extend: Create New Patient Generalization:		
Normal Flow of Events: <ol style="list-style-type: none"> The Patient contacts the office regarding an appointment. The Patient provides the Receptionist with their name and address. The Receptionist validates that the Patient exists in the Patient database. The Receptionist executes the Make Payment Arrangements use case. The Receptionist asks Patient if he or she would like to make a new appointment, cancel an existing appointment, or change an existing appointment. <ul style="list-style-type: none"> If the patient wants to make a new appointment, the S-1: new appointment subflow is performed. If the patient wants to cancel an existing appointment, the S-2: cancel appointment subflow is performed. If the patient wants to change an existing appointment, the S-3: change appointment subflow is performed. The Receptionist provides the results of the transaction to the Patient. 		
Subflows: <ul style="list-style-type: none"> S-1: New Appointment <ol style="list-style-type: none"> The Receptionist asks the Patient for possible appointment times. The Receptionist matches the Patient's desired appointment times with available dates and times and schedules the new appointment. S-2: Cancel Appointment <ol style="list-style-type: none"> The Receptionist asks the Patient for the old appointment time. The Receptionist finds the current appointment in the appointment file and cancels it. S-3: Change Appointment <ol style="list-style-type: none"> The Receptionist performs the S-2: cancel appointment subflow. The Receptionist performs the S-1: new appointment subflow. 		
Alternate/Exceptional Flows: <ul style="list-style-type: none"> 3a: The Receptionist executes the Create New Patient use case. S-1, 2a1: The Receptionist proposes some alternative appointment times based on what is available in the appointment schedule. S-1, 2a2: The Patient chooses one of the proposed times or decides not to make an appointment. 		

Exercise : Case Study Library System

- ▶ Create a set of use cases for a university library borrowing system. The system will record the books owned by the library and will record who has borrowed what books.
- ▶ Before someone can borrow a book, he or she must show a valid ID card that is checked to ensure that it is still valid against the student database maintained by the registrar's office (for student borrowers), the faculty/staff database maintained by the personnel office (for faculty/staff borrowers), or against the library's own guest database (for individuals issued a "guest" card by the library).
- ▶ The system must also check to ensure that the borrower does not have any overdue books or unpaid fines before he or she can borrow another book. Every Monday, the library prints and mails postcards to those people with overdue books.
- ▶ If a book is overdue by more than two weeks, a fine will be imposed and a librarian will telephone the borrower to remind him or her to return the book(s). Sometimes books are lost or are returned in damaged condition. The manager must then remove them from the database and will sometimes impose a fine on the borrower.

Case Study : University System

- ▶ Create a set of use cases for an online university registration system. The system should enable the staff of each academic department to examine the courses offered by their department, add and remove courses, and change the information about them (e.g., the maximum number of students permitted). It should permit students to examine currently available courses, add and drop courses to and from their schedules, and examine the courses for which they are enrolled. Department staff should be able to print a variety of reports about the courses and the students enrolled in them. The system should ensure that no student takes too many courses and that students who have any unpaid fees are not permitted to register. (Assume that a fees data store is maintained by the university's financial office, which the registration system accesses but does not change).

TERIMA KASIH