



Software Re-Engineering

Lecture: 05

Sequence [Today's Agenda]

Content of Lecture

- Reverse Engineering and its Techniques

Reverse Engineering

- Reverse engineering uncovers the design and functionality of a product or system by working backward from its final form.

Use Case Modeling in Reverse Engineering

- Use case modeling plays a significant role in software reverse engineering by helping to understand and document the functionality of an existing software system, often without access to the original source code or detailed documentation.
- Use case models can serve as a bridge between understanding the software's functionality and its internal implementation.

- Here's how use case modeling helps in this context:

- Capturing Functional Behavior
- Understanding Use Cases
- Identifying Core Features
- Clarifying System Interactions
- Refining System Structure
- Identifying Gaps
- Mapping Old and New Systems

Introduction: Use Cases

- Use cases are text stories, widely used mechanism to discover and record requirements (especially functional).
- They influence many aspects of a project including OOA/D.
- Writing use cases — stories of using a system — is an excellent technique to understand and describe requirements.
- The UP defines the Use-Case Model within the Requirements discipline.
- Use-Case model is the set of all use cases; it is a model of the system's functionality and environment.

Goals and Stories: Use Cases

- Customers and end users have goals (also known as needs in the UP) and want computer systems to help meet them.
- Use cases are text stories of some actor using a system to meet goals, i.e. cases of use
- Example: Process Sale
 1. A customer arrives at a checkout with items to purchase.
 2. The cashier uses the POS system to record each purchased item.
 3. The system presents a running total and line-item details.
 4. The customer enters payment information, which the system validates and records.
 5. The system updates inventory.
 6. The customer receives a receipt from the system and then leaves with the items.

Use Cases by Ivar Jacobson

- “A use case is a specific way of using the system by performing some part of the functionality.
- Each use case constitutes a complete course of events initiated by an actor, and it specifies the interaction that takes place between an actor and the system.
- The collected use cases specify all the existing ways of using the system.”

Use Cases

- A use case is a typical sequence of actions that a user performs in order to complete a given task
- Use cases are functional requirements that indicate what system will do
- Use cases describe what a system does from the standpoint of an external observer
- The emphasis is on what a system does rather than how
- In terms of the FURPS+ requirements types, they emphasize the "F" (functional or behavioral)
- A functional requirement may be satisfied by one or more use cases
- The set of all use cases together describes the complete behavior of the system

-
- It is one of the key activities in requirements analysis
 - The objective of use case analysis is to model the system from the point of view of:
 - how users interact with this system, when trying to achieve their objectives
 - Use cases are text documents, not diagrams
 - Use-case modeling is primarily an act of writing text, not drawing
 - Text is important, diagram is optional
 - However, the UML defines a use case diagram to illustrate the names of use cases and actors, and their relationships

Reverse Engineering Tools

The focus is:

what functionality is involved and who will ask for the provided services

How the services are carried out by the system is not addressed

Use Cases are	Use Cases are NOT
What the system will do	How it will be done
Analysis Products	Design Products

Basic Definition

- Actor: Something with behavior, such as a person (identified by role), computer system, or organization OR A user or outside system that interacts with the system being designed in order to obtain some value from that interaction For example, a cashier
- Scenario: Specific sequence of actions and interactions between actors and the system Also called a use-case instance One particular story of using a system For example, the scenario of successfully purchasing items with cash, or the scenario of failing to purchase items because of a credit payment denial
- A use case is a collection of related success and failure scenarios that describe an actor using a system to support a goal.

Example

- Use case:
 - Register Course
 - Main Success Scenario: The system presents all possible courses for this student. The student select course and commits the entry by pressing the Enter button.
 - Alternate Scenario: The student can cancel a registration by pressing the Cancel button.

Use-Cases and Use-Case Model

- The UP defines the Use-Case Model within the Requirements discipline
- This is the set of all written use cases
 - It is a model of the system's functionality and environment
- The Use-Case Model may optionally include a UML use case diagram to show the names of use cases and actors, and their relationships
 - This gives a nice context diagram of a system and its environment
 - It also provides a quick way to list the use cases by name

Why Use-Cases

- Lack of user involvement in software projects is near the top of the list of reasons for project failure, so anything that can help keep them involved is truly desirable
 - Another value of use cases is that they emphasize the user goals and perspective
 - We ask the question "Who is using the system, what are their typical scenarios of use, and what are their goals?" This is a more user-centric emphasis compared to simply asking for a list of system features.

Black Box Use-Cases

- Most common and recommended type
- They do **not** describe the internal workings of the system, its components, or design
- During requirements analysis avoid making "how" decisions, and specify the external behavior for the system, as a black box

Black-box style	Not
The system records the sale.	The system writes the sale to a data-base. ...or (even worse): The system generates a SQL INSERT statement for the sale...

References

- Applying UML and Patterns
 - Chapter 6: Use Cases
- UML Distilled
 - Chapter 9: Use Cases

Thank You!

