Object Oriented Analysis and Design
CS 6359.001: Reading Assignment #3

Due on Sunday September 22, 2019 at noon

Instructor: Prof.Mehra Nouroz Borazjany

Shyam Patharla **(sxp178231)**

1. Define what is actor?

An actor is a business role played on behalf of entities which are external to the system and interacts with the system.

- **What are the three kinds of actors?**
  Primary actors: These actors directly have their business task fulfilled by the use case (eg. person withdrawing cash from an ATM system).
  Supporting actors: These actors provide specific services to system (eg. bank database for an ATM system)
  Offstage actors: These actors are not directly involved in the use case but have an interest in the proper functioning of the system and execution of use cases. (eg. the bank whose ATM machine it is)
- **What is an actor goal?**
  An actor goal is a business task which the actor wants to accomplish using the system or use case. An actor can have many actor goals.

2. Define what is a use case?

A use case is a business process which

- starts with an actor
- ends with an actor
- does something which is useful for the actor

- **Are they diagrams?**
  A use case is not necessarily a diagram but can be represented by a use case diagram. Other ways of representation include just abstract use case (just use case name along with actor name and subsystem name) , high level use case (only begin and end actions) and extended use case (detailed description).

- **Are they functional requirements?**
  Functional requirements are goals which must be met for the system to be operational. Use cases are drawn from functional requirements and represent business processes involving actors and system/subsystems.

3. How we select a name for a use case? What is an abstract use case?

While we deduct use cases from the requirements specification, we look for **verb-noun** phrases which represent ”do something”. We also look for **nouns** which represents actors/systems/subsystems. Once all these are identified, we check the following conditions:

- Is it a business process?
- Is it started by an actor?
- Does it end when the actor performs an action?
- Does it do something useful for the actor?

If all the conditions are satisfied, we conclude that it is a use case and choose the verb-noun phrase as the name (eg. **withdraw cash** in case of ATM machine).

An **abstract use case** specifies only the name of the use case, generally a verb-noun phrase. We could include the name of the actor and system name in brackets).

4. What is the relation between a use case and an actor goal?
A use case is the business process through which an actor satisfies an actor goal. In a use case diagram, we draw a link (**association** relationship) between the actor and the use case to indicate which use case is used to satisfy a particular actor goal.

5. What is a common use case format?
A use case can be represented as an

- **Abstract Use Case**: Specifies only the name of the use case, generally a verb-noun phrase. We could include the name of the actor and system name in brackets.
- **High Level Use Case**: Specifies the scope of the use case i.e. when and where the use case begins (TUCBW) and when the use case ends (TUCEW)
- **Expanded Use Case**: Outlines the steps involved in how the actor interacts with the system and how the use case

6. Define what are use case scenarios?
A use case is a **business process** consists of many **operations**, which in turn are composed of several **actions** or steps. At any step/operation in the process, there could be multiple ways of the use case proceeding further (eg. failed input, processing error, different branch in flow, successfully processed, etc). Use case scenarios are multiple paths which can occur during the steps of a use case are executed.

7. What is a requirement use case traceability matrix?
It is a matrix where we specify which use case is derived from which requirement. Each requirement has a priority associated with it. The priority of a use case is the maximum of the priorities of all the requirements it is associated with.

- It is useful to see which use cases are not required i.e. they do not satisfy any requirement and eliminate them
- Whether each requirement is satisfied by atleast one use case i.e. no empty rows
- Assign higher priority use cases to initial iterations and developed faster

Source: Object Oriented Software Engineering: An Agile Unified Methodology, David C. Kung, McGrawHill