DEVELOPING USE CASE DIAGRAMS

MR. ANIKET MISHRA

CONTENTS

- What is a Use-Case diagram?
- When Use-Case diagrams are useful?
- Various components of a Use-Case diagram.
- Relationship between actors and Use-Cases
  - Includes
  - Extends
  - Generalization
- Indentifying actors and Use-Cases
- Guidelines for drawing Use-Case diagram

WHAT IS A USE-CASE DIAGRAM?

- A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved.

- They belong to the category of behavioral diagrams in UML diagrams.

HOW USE-CASE DIAGRAMS ARE USEFUL?

- Provide a common understanding for the end-users, developers and the domain experts.

- It is used to capture the basic functionality i.e. use cases, and the users of those available functionality, i.e. actors, from a given problem statement.

COMPONENTS OF A USE-CASE DIAGRAM

Actor

- An object or set of objects, external to the system, which interacts with the system to get some meaningful work done.
- Actors could be human, devices, or even other systems.
- E.g. consider the case where the customer withdraws cash from ATM. Here customer is an actor

TYPES OF ACTORS

- Primary actor: They are principal users of the system, who fulfill their goal by availing some service from the system.
- For example, a customer uses an ATM to withdraw cash when he needs it. A customer is the primary actor here.
- Supporting actor: They render some kind of service to the system. "Bank representatives", who replenishes the stock of cash, is such an example.
- It may be noted that replenishing stock of cash in an ATM is not the prime functionality of an ATM.

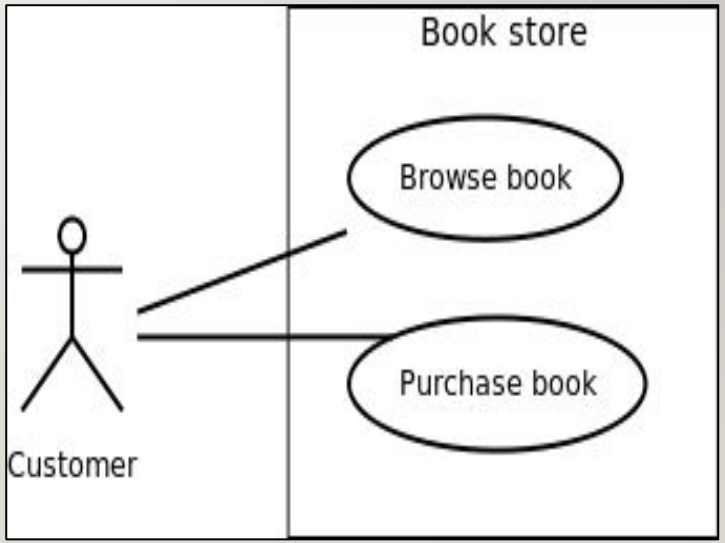
COMPONENTS OF A USE-CASE DIAGRAM

Use-Case

- Are simply the functionality provided by a system.
- Eg. Withdraw cash, deposit cash, check balance in case of an ATM

GRAPHICAL REPRESENTATION

An actor is represented by a stick figure and name of the actor is written below it. A use case is depicted by an ellipse and name of the use case is written inside it. The subject is shown by drawing a rectangle. Label for the system could be put inside it. Use cases are drawn inside the rectangle, and actors are drawn outside the rectangle, as shown in figure.



HOW ARE ACTORS AND USE-CASES ASSOCIATED?

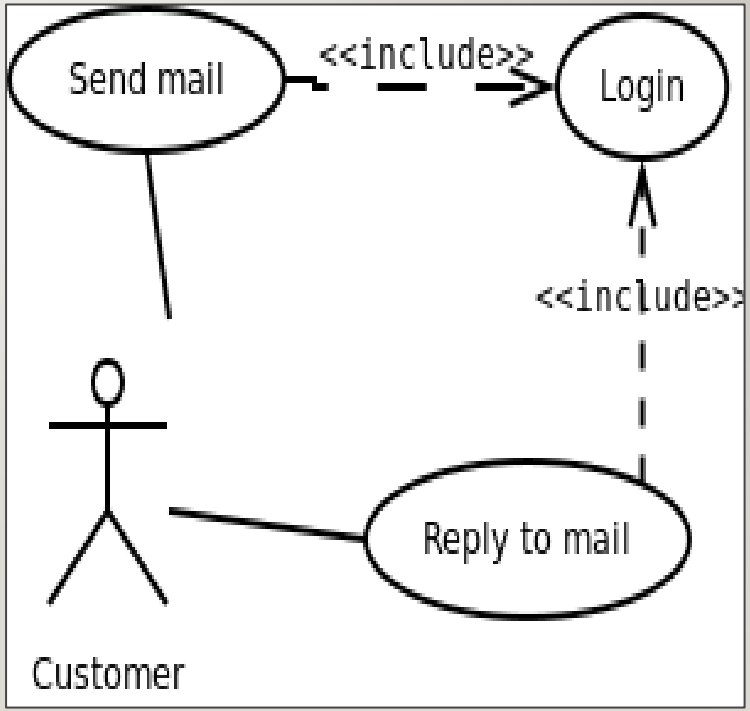
- Actors and Use-Cases communicate throy message passing and have a binary association between them.
- An actor must be associated with atleast one Use-Case.
- A Use-Case must be associated with atleast one actor.
- Actors don't have association between them, but can have a hierarchy between them.
- Three types of relationship exist :
  - Include
  - Extends
  - Use-Case generalization

INCLUDES RELATIONSHIP

- depict common behaviour that are shared by multiple use cases.
- Whether sending a mail or replying to a mail, the user has to login first.

Notation

- Include relationship is depicted by a dashed arrow with a «include» stereotype from the including use case to the included use case.

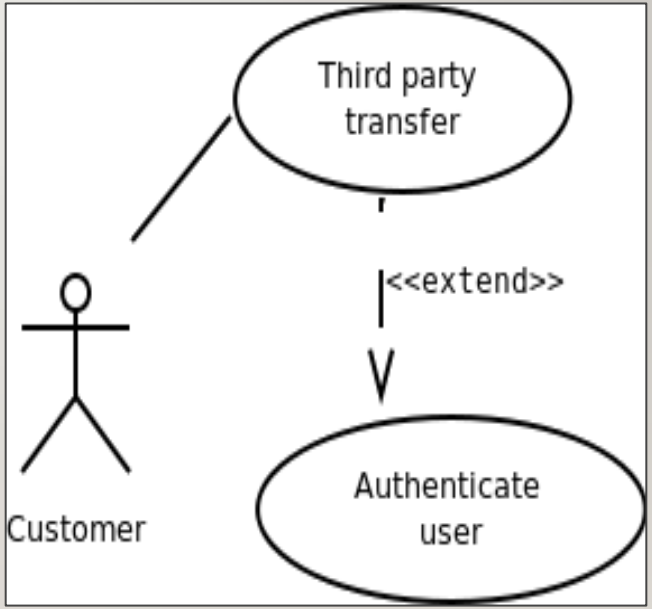


EXTENDS RELATIONSHIP

- Use case extensions are used used to depict any variation to an existing use case.
- They are used to the specify the changes required when any assumption made by the existing use case becomes false.

Notation

- Extend relationship is depicted by a dashed arrow with a «extend» stereotype from the extending use case to the extended use case.



MORE POINTS ABOUT EXTENDS RELATIONSHIP

- While the base use case is defined independently and is meaningful by itself, the extension use case is not meaningful on its own.

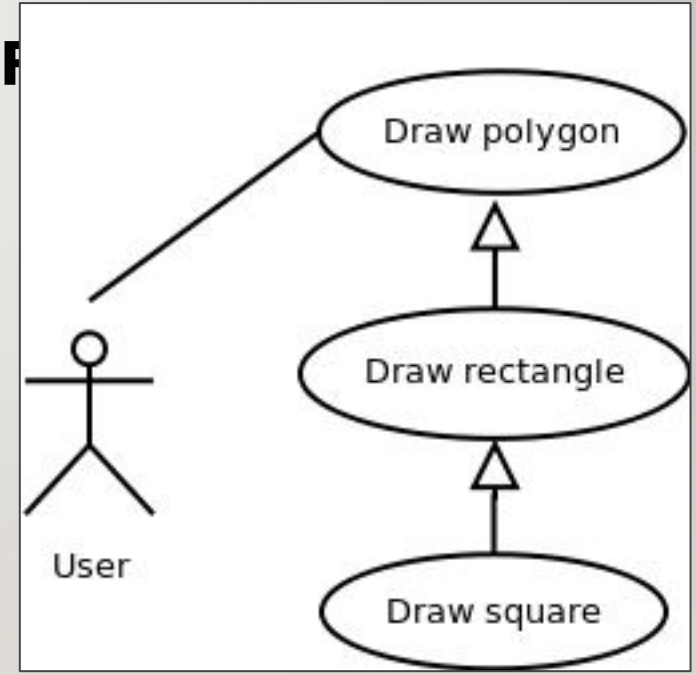
- The extension use case can access and modify the attributes of the base use case; however, the base use case is not aware of the extension use case and, therefore, cannot access or modify the attributes and operations of the extension use case.

GENERALIZATION RELATIONSHIP

- Used to represent the inheritance between use cases.
- A derived use case specializes some functionality it has already inherited from the base use case.

Notation

- Generalization relationship is depicted by a solid arrow from the specialized (derived) use case to the more generalized (base) use case.



IDENTIFYING ACTORS

Given a problem statement, the actors could be identified by asking the following questions:

- Who gets most of the benefits from the system? (The answer would lead to the identification of the primary actor)
- Who keeps the system working? (This will help to identify a list of potential users)
- What other software / hardware does the system interact with?
- Any interface (interaction) between the concerned system and any other system?

IDENTIFYING USE-CASE

Once the primary and secondary actors have been identified, we have to find out their goals i.e. what are the functionality they can obtain from the system. Any use case name should start with a verb like, "Check balance".

GUIDELINES FOR DRAWING USE CASE DIAGRAMS

- Determine the system boundary
- Ensure that individual actors have well-defined purpose
- Use cases identified should let some meaningful work done by the actors
- Associate the actors and use cases -- there shouldn't be any actor or use case floating without any connection
- Use include relationship to encapsulate common behaviour among use cases , if any

THANK YOU!