

Software Design and Architecture

Use Case Diagrams

- ▶ A use case is a set of scenarios that describe an interaction between a user and a system.
- ▶ Use case model is a representation of sequence of transactions initiated by the user (actor) from outside the system.
- ▶ In the process, the transaction uses internal objects to take the transaction to completion.



-
- ▶ Use case defines and describes what happens in the system in logical order, termed as system behaviour.
 - ▶ They represent the flows of events that the user triggers. The user/actor is anything that initiates or triggers the action in the system.
-



Example: Ticket Reservation System

- ▶ The passenger has a prior knowledge of the reservation and ticketing system.
 - ▶ The passenger arrives at the ticket counter and interacts with the clerk first through an enquiry
 - ▶ Passenger then follows the process of form filling, tendering, payment and collecting the tickets.
 - ▶ Passenger accepts the ticket or leaves the counter.
-



-
- ▶ • Use cases within an Airline Reservation system might include
 - ▶ - Checking in for a flight
 - ▶ - Assigning a seat
 - ▶ - Checking baggage
-



Basic Concepts

- ▶ Use cases are not inherently object-oriented
 - ▶ An external (user) view of the system
 - ▶ Intended for modelling the dialog between the users and the system
 - ▶ The main concepts in use cases are
 - ▶ Actor
 - ▶ Use Case
 - ▶ Associations
 - ▶ <<includes>>
 - ▶ <<extends>>
-



Actor

- ▶ An Actor is a role of an object or objects outside of a system that interacts directly with it as part of a coherent work unit (a use case)
- ▶ One physical object (or class) may play several different roles and be modeled by several actors
- ▶ Notation

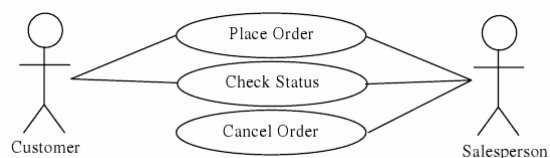


- ▶ Example actors for an Airline Reservation system
 - ▶ Airline administrators (fare/schedule setting)
 - ▶ Travel Agent
 - ▶ Airline Reservations Agent
 - ▶ Check-in Agents at Airport
 - ▶ Gate Agent at Airport



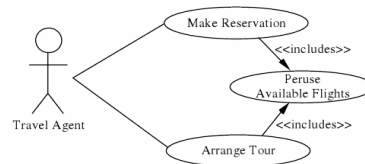
Use Case

- ▶ A Use Case captures some actor-visible function
- ▶ Achieves some discrete (business-level) goal for that actor
- ▶ May be read, write, or read-modify-write in nature
- ▶ Notation



Associations: <<includes>>

- ▶ When a use case is depicted as using the functionality of another use case in a diagram.
- ▶ An include relationship is depicted with a directed arrow having a dotted shaft. The tip of the arrowhead points to the parent use case and the child use case is connected at the base of the arrow.
- ▶ Example



- ▶ Make Reservation and Arrange Tour both depend on Peruse /examine Available Flights
- ▶ Note that the arrows go from the dependent use cases
- ▶ Typically used when the same unit of functionality is part of more than one use case
- ▶ The base use cases are, in a sense, incomplete without the included use case

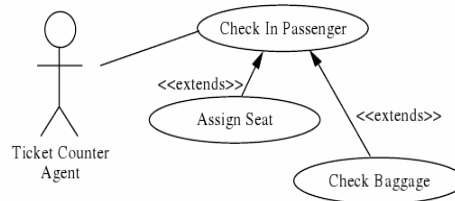


Associations: <<extends>>

- ▶ In an *extend* relationship between two use cases, the child use case adds to the existing functionality and characteristics of the parent use case.
- ▶ An extend relationship is depicted with a directed arrow having a dotted shaft, similar to the include relationship.
- ▶ The tip of the arrowhead points to the parent use case and the child use case is connected at the base of the arrow. The stereotype "<<extend>>" identifies the relationship as an extend relationship



► Example:



- Assign Seat and Check Baggage both depend on Check In Passenger
- Note that the arrows go from the dependent use cases
- Typically used when there are important, *optional* variations on the basic theme of the base use case
- The base use case is complete in and of itself

►

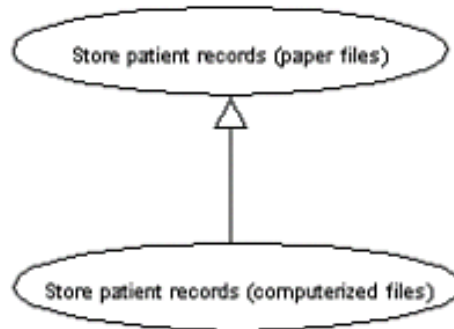
Associations: Generalizations:

- A *generalization* relationship is also a parent-child relationship between use cases.
- The child use case in the generalization relationship has the underlying business process meaning, but is an enhancement of the parent use case.
- In a use case diagram, generalization is shown as a directed arrow with a triangle arrowhead.
- The child use case is connected at the base of the arrow. The tip of the arrow is connected to the parent use case.

►

Associations

- ▶ Generalization Relationship



- ▶ Arrow directed from parent use case to child use case



Difference between Extend Relationship and Generalization Relationship

- ▶ On the face of it, both generalizations and extends appear to be more or less similar. But there is a subtle difference between a generalization relationship and an extend relationship.
 - ▶ a generalization relationship between use cases:
 - ▶ the parent use case can be replaced by the child use case without breaking the business flow.
 - ▶ an extend relationship between use cases:
 - ▶ that the child use case enhances the functionality of the parent use case into a specialized functionality. The parent use case in an extend relationship cannot be replaced by the child use case.



