MUNI
FI

PB007 Week 03

Samuel Sabo

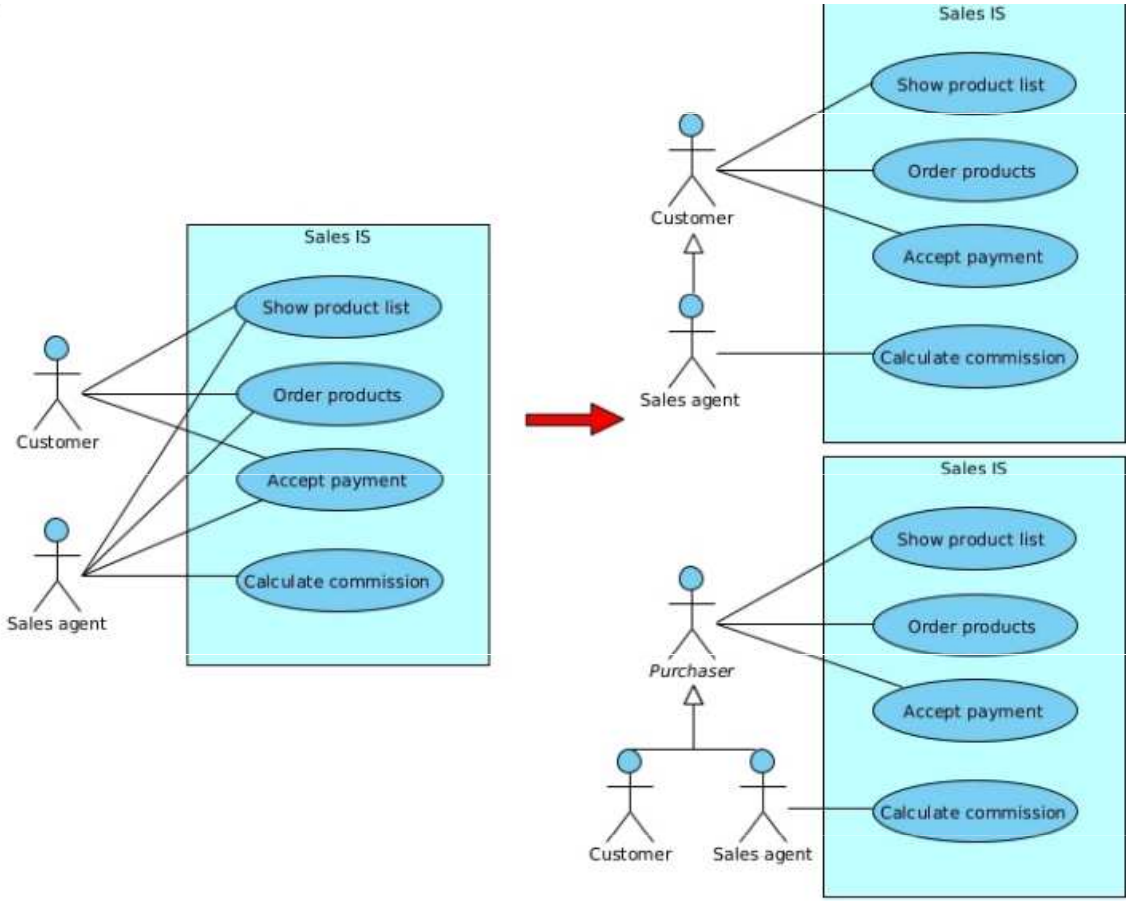
Relationships in use case diagrams

- In addition to the communication association between actors and use cases, there are other types of possible relationships:
  - actor generalization
  - use case generalization
  - <<include>>
  - <<extend>>

Actor generalization

- Actor generalization (inheritance) is the relationship between the general and specialized actors
  - General actors are often abstract
  - Specialized actor inherits all roles and relationships of the parent actor
  - Each expected occurrence of the parent actor can be substituted by any of its descendant actors
  - It is usually suitable when multiple actors share some of the use cases

Actor generalization - example

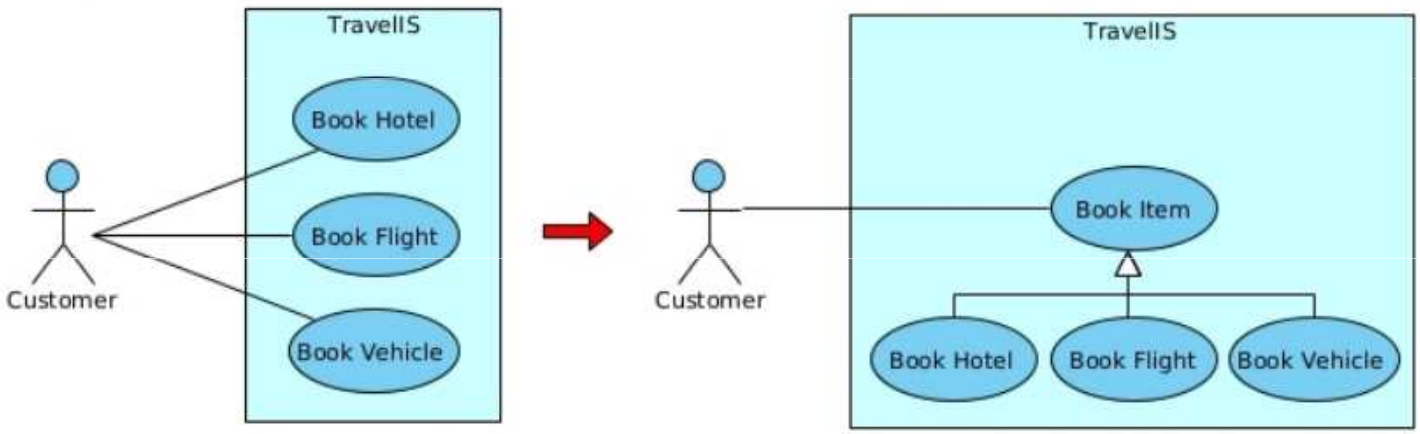


MUNI FI

Use case generalization

- Use case generalization (inheritance) is the relationship between the general and specialized use cases
  - Specialized use case inherits the properties of its parent and it can add new properties or override inherited ones. Note.: it cannot overload parent's extension points.
  - The text specification of inherited use cases should mark the changes from the parent use case.
  - Parent use case can be abstract (recommended), i.e. it has either no specification or incomplete specification of the event flow

Use case generalization - example

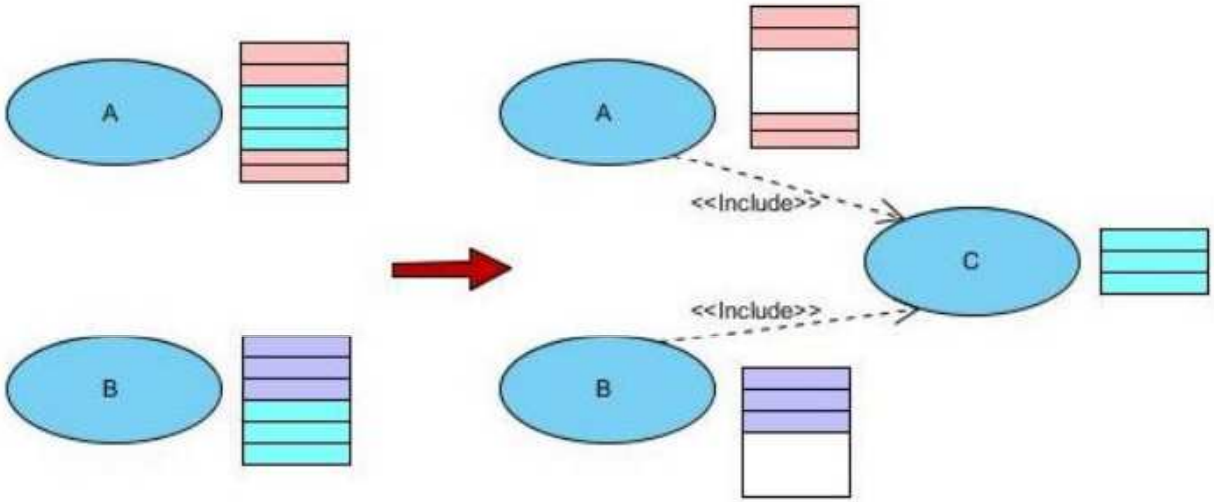


MUNI FI

<<include>>

- <<include>> allows to extract steps that are repeated in multiple use cases into separate use case.
  - The base use case is incomplete without all the included use cases.
  - The included use case may or may not be complete.
  - It should not be misused for functional decomposition

<<include>> - example

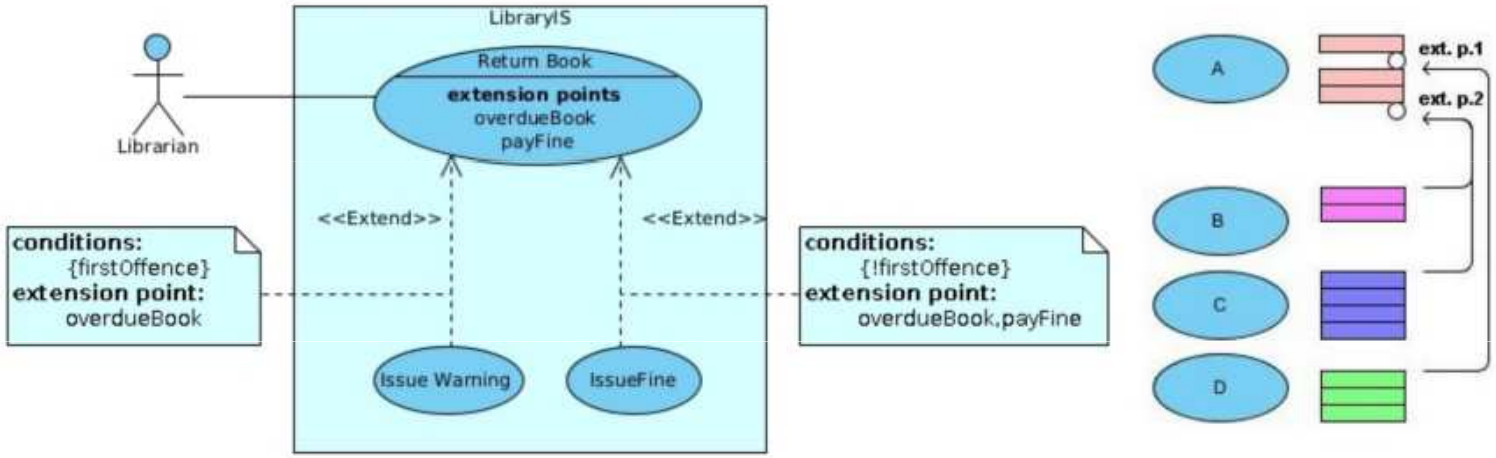


MUNI FI

<<extend>>

- The <<extend>> relationship allows to add a new behaviour into an existing use case.
  - The base use case contains extension points within the flow of events.
  - The extension use case specifies to which extension points it is attached.
  - The base use case does not know about its extension use cases.
  - The extension use case is usually incomplete. It can contain multiple segments, each specifying their extension points.
  - Multiple extension use cases can share the same extension points. In such case, it is appropriate to define conditions to determine which of them should be used in given situation.

<<extend>> - example



MUNI FI

Textual specification of use cases

- Textual specification of use cases should contain:
  - Name (ID)
  - A brief description
  - The primary actors
  - The secondary actors
  - Preconditions
  - The main flow of events
  - Post-conditions
  - An alternative flow of events

Flow of events

- Main flow(Primary Scenario) of events is a sequence of steps in the interaction between actors and the system in the ideal case
  - It always starts with an action from the primary actor. Use the form:
    1. The use case begins when, <actor><function>
  - The individual steps should be short, precise a understandable. Use the form:
    <step id><actor/system><action>
    You can modify the flow with IF, FOR, WHILE, GOTO etc.
- The alternative flow represents deviations from the main flow caused by the errors and interruptions. It can be also used to capture more complex branching.

Flow of events - example

Update Ticket Status

Use Case ID	UC9
Brief Description	Support Managers updates the information or status of the issue ticket to reflect the work progress on the issue
Primary Actors	Support Technician
Secondary Actors	Issue Reporter
Preconditions	Issue Reporter is logged in the system
Main Flow of Events	1. Use case starts when **Support Technician** selects "Update Tickets" option in the menu 2. *INCLUDE*(Find Tickets) 3. *IF* at least one ticket assigned to the Issue Technician is found 3.1. **Support Technician** selects one of the tickets from the displayed list 3.2. **System** shows detailed information about the issue ticket 3.3. **Support** Technician selects the "Edit Status" option. 3.4. **System** displays an editable form for the ticket details. 3.5. **Support Technician** updates the ticket information or its status. *EXTENSION POINT*(TicketisClosed) 3.6. **System** stores the changes in the database
Alternative Flows	Issue Reporter can log out at any time
Post-conditions	The status of the issue ticket was updated.

MUNI FI

Task for this week

- Review the initial use case diagram from the previous session. Fix any problems and consider the use of advanced relationships
- Briefly document all use cases. This will help to better understand the project
- Choose 3 use cases (ideally the more complex ones) for detailed textual specification. The selection should be approved by the seminar tutor. The chosen use cases should use different colour in VP
- Submit pdf report to homework vault