Bournemouth University

## Overview

SoSyM

- Use case description
- Example descriptions.
- Include and extends
- Writing use cases
- Components of the description
- An example: car park

Bournemouth University

## Recap: Use Cases

SoSyM

- A use case is a technique used to describe **what** a new system should do or what an existing system already does from the user's point of view.
- An (important) aim of use cases is to describe the functional requirements of the system.
- The functionality of the system is represented by a complete set of *use cases*.
- Each use case specifies a "complete functionality": one general usage of the system.

## Example Use Case Description 1.

SoSyM

Use Case Title: New Equipment Request

Actors: Academic

Context: Academic wants to request a new printer and software package in order to conduct an experiment.

Pre-condition: The University network is working.

Bournemouth University

## Example 2.

SoSyM

Main flow of events:

1. Academic accesses the system by entering password *<<include>> (Validate Password) use case*.

1.1. If password correct, continue to 2. If not, repeat 1.

2. Academic goes to Requests Page.

3. Academic selects hardware or software options.

4. Academic completes data fields.

5. Academic clicks “Send Request”

6. End of use case: MessageBox says “Request Sent.”

## Example 3

Exceptional flows of events:

2. Academic goes to the wrong page and must return to the Requests Page.
2. Academic cannot access the Requests Page. The Academic reports the happening <<*extend*>> *(Report Errors) use case*.
3. Academic chooses wrong option, clicks Clear Form and chooses the correct option.

## Example 4

5. Academic sends incomplete fields - the system refuses to send and indicates what needs to be completed.
6. Nothing is returned to the Academic. The Academic reports the happening *<<extend>> (Report Errors) use case.*

## UML <<include>> use cases

- When we access general information that is used by many functions then we <<include>> that use case. If we use the <<include>> relationship we **have** to go to this use case **every time**.
- Remember: the <<include>> use case describes general functionality that is used by at least 2 other use cases.

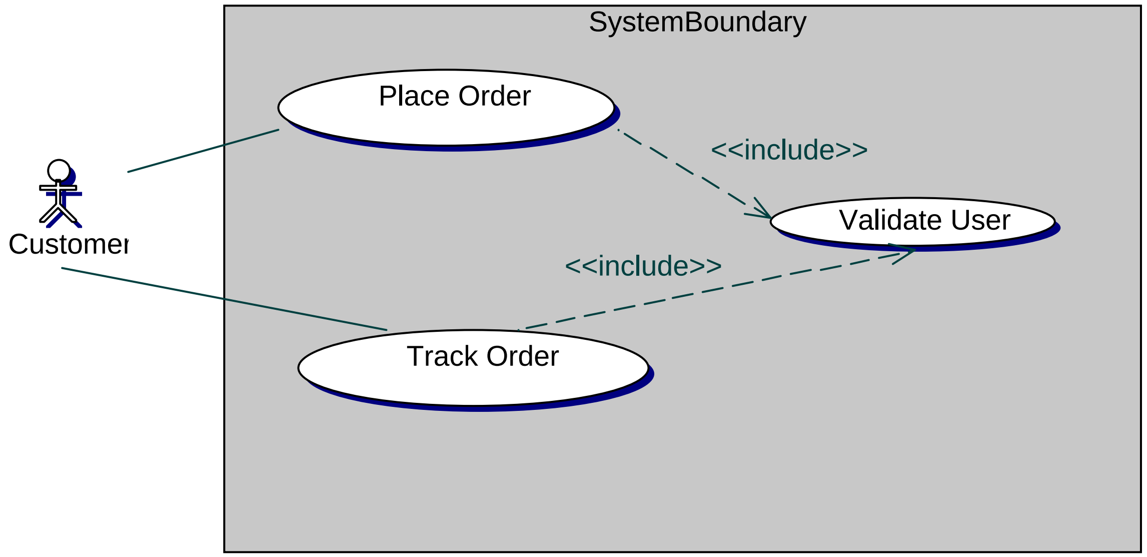
Bournemouth University

## Use Cases for reuse: <<includes>>

SoSyM

- An <<includes>> use case is created by factoring out common behaviour from two or more original use cases.
- The <<includes>> use case(s) never stand alone - they are always instantiated as part of some other use case that includes or ‘uses’ it.

# Example <<includes>>


SystemBoundary
Place Order
<<include>>
Validate User
Customer
<<include>>
Track Order

# Include within description

| | |
|---|---|
| **Use Case:** | **Place Order** |
| **Actors:** | Customer |
| **Purpose:** | Capture an order from a regular customer |
| **Precondition:** | The customer has logged in to the on-line system and selects Place Order |
| **Flow of events:** | 1. The use case begins when a Customer selects place order.<br>2. The Customer enters their name and e-mail address.<br>3. **Use Validate User**<br>4. The Customer enters product codes for the products to be ordered.<br>5. The system provides a product description and a price for each item.<br>6. The system provides user payment details for confirmation.<br>7. The system provides delivery address for confirmation.<br>8. The Customer presses submit.<br>9. The system gives the customer an order number and confirms the order. |
| **Postcondition:** | The order is posted to the inventory and accounting systems |

## UML <<extend>> use cases

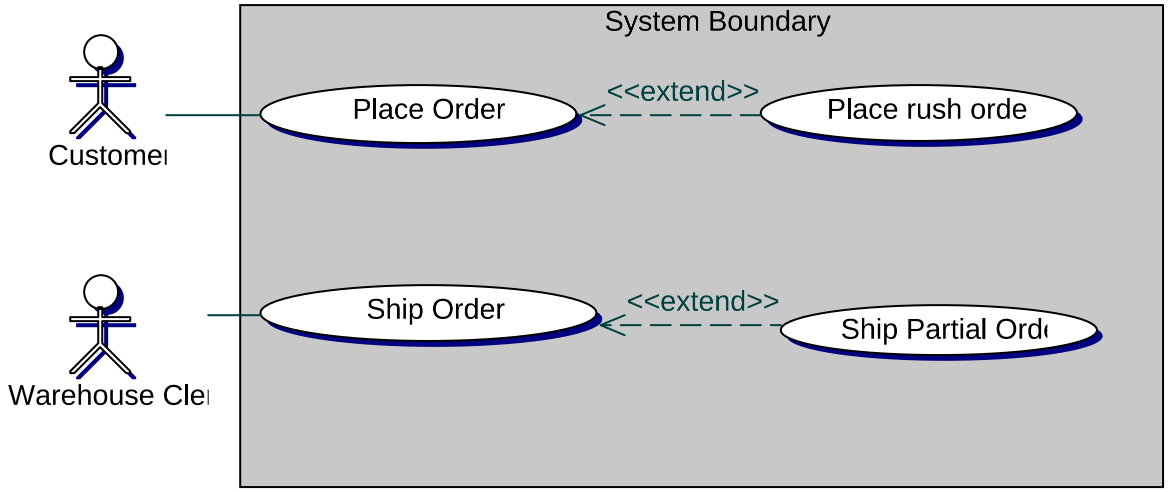
- When we use <<extend>> use cases we are showing that some rare behaviour is occurring because the main use case has reached an unusual condition forcing the <<extend>> use case upon us.
- <<extend>> use cases show occasional and exceptional functionality.
- *Though some argue a more casual usage..*

## When to use <<extends>>

SoSyM

- To model the part of a use case that the user sees as optional system behaviour (*if rare*).
- To model a separate sub-flow that is executed only under certain (*unusual*) conditions
- To model several flows that may be inserted at a certain point, governed by explicit interaction with an actor. [not convinced – more like alternates]
- *Implicit that these are not the norm (unusual).*

# Example <<extends>>


System Boundary
Customer
Place Order
<<extend>>
Place rush orde
Warehouse Cle
Ship Order
<<extend>>
Ship Partial Ord

## Use-Case Description Format

- Title (which describes the UC goal)
- Actor(s)
- Context
- Pre-condition and Post-condition
- Main Flow
- 1.
- 2. etc
- Alternative flow(s)
- Exceptional flow(s)

## Templates and Guidelines

SoSyM

- Much debate over the best templates for use case descriptions.
- Some suggest templates, others content or structure guidelines.
- Debate over general structure.
  - E.g., Some suggest allowing alternatives (like ifs) within the main flow, most adamant that alternatives come at the end.
  - Correct length (up to a page?)
- Clearly written, comprehensible?

# Extend in Description

| | |
|---|---|
| **Use Case:** | **Place Order** |
| **Actors:** | Customer |
| **Purpose:** | Capture an order from a regular customer |
| **Precondition:** | The customer has logged in to the on-line system and selects Place Order |
| **Flow of events:** | 1. The use case begins when a Customer selects place order.<br>2. The Customer enters their name and e-mail address.<br>3. **Use Validate User**<br>4. The Customer enters product codes for the products to be ordered.<br>5. The system provides a product description and a price for each item.<br>6. The system provides user payment details for confirmation.<br>7. The system provides delivery details for confirmation.<br>8. **If the customer selects next day delivery then Place rush order**.<br>9. The Customer presses submit.<br>10. The system gives the customer an order number and confirms the order. |
| **Postcondition:** | The order is posted to the inventory and accounting systems |

## Title and Actors

- Title is vital. It should be a verb-noun phrase that represents the goal of the requirement it is describing. E.g.,
  - Make Book Order
  - Update Records
- Actors are really roles so make their names generic. E.g.,
  - Line manager
  - Assistant Cashier

## Context and pre/post

SoSyM

- Context: a short paragraph that explains why the use case is needed now.
  - The context must be concise and its first sentence MUST give the reason why the actor wants to do the use case.
- Pre-condition & Post-condition: announces what the system will ensure is *true* before letting the use case start and what is true afterwards.
  - Since it is enforced by the system and known to be true, it will not be checked again. E.g.
    - The User has already logged on and has been validated
    - The account has been updated for the Customer

## Main Flow

SoSyM

- Give a meaningful first event - this shapes the rest of the UC from the reader's perspective.
  - 1. The clerk enters the customer's bank account number
  - NOT
  - 1. The clerk turns on the application OR
  - 1. The user logs on to the system
- Number the events. E.g.,
  - 1.
  - 2.

Bournemouth University

## Alternatives and Exceptions

SoSyM

- Alternative Flows of Events
  - These are events that can occur at any point in the use case but don’t  - they are the options not chosen by the actor in this instance.
- Exceptional Flows of Events
  - These describe alternative events that occur only a small percentage of the time and are considered not normal behaviour for the actor or system. They are not exceptions in the programming sense that deal with error handling, though they are often treated that way.

## Summary

- Use case diagrams versus description
  - Diagrams useful for overview
    - Although can be arbitrary partitioning.
  - Descriptions VITAL.
- Include and extends
- Writing use cases
- Components of the description
- Guidelines (of which more to come)

Bournemouth University

## A use case for you to describe

SoSyM

- Imagine that you want to park your car in a car park in Bournemouth so that you can go shopping in the sales.
- On your own write a short use case description of how you would do this.