

Introduction to coursework 1.1 and Use Case Analysis

Lecture 5

Requirements

- Can be functional
 - What the system should do
 - e.g. provide a login facility that uses a username and password
- Can be non-functional
 - Constrains on how the functions are provided
 - Username must be longer than six characters
 - Software must be written in Java

Use cases and user stories

- In this module we will be focusing on the application of use cases for requirements analysis
- User stories are less formal and simpler to define than use cases and are widely used in Agile development

User story

- As a **returning user**, I want to log in using my username and password to access my account. The login page should display clear labels for username and password fields.
- History Kent Beck 1997

Use cases

(Jacobson 1986)

- Use → How you use the system
- Case → An example scenario
- Functional modelling of requirements
- Show the external view of the system
 - (Does not model internal processes)
- Show the system relative to different users of the system
- Users of the system are termed **Actors**

Use Cases

- **Use-Cases** are a scenario based technique in the Unified Modeling Language (UML) which identify the **actors** in an interaction and which describe the interaction itself.
- A set of use-cases should describe all possible interactions with the system.
- **Sequence diagrams** may be used to add detail to use-cases by showing the sequence of event processing in the system (we shall study sequence diagrams later).

Use Cases

- In a **use-case diagram**, an **actor** is a user of the system (i.e. Something external to the system; can be human or non-human) acting in a particular role.
- A **use-case** is a task which the **actor** needs to perform with the help of the system, e.g., find details of a book or print a copy of a receipt in a bookshop.

Actors (players)

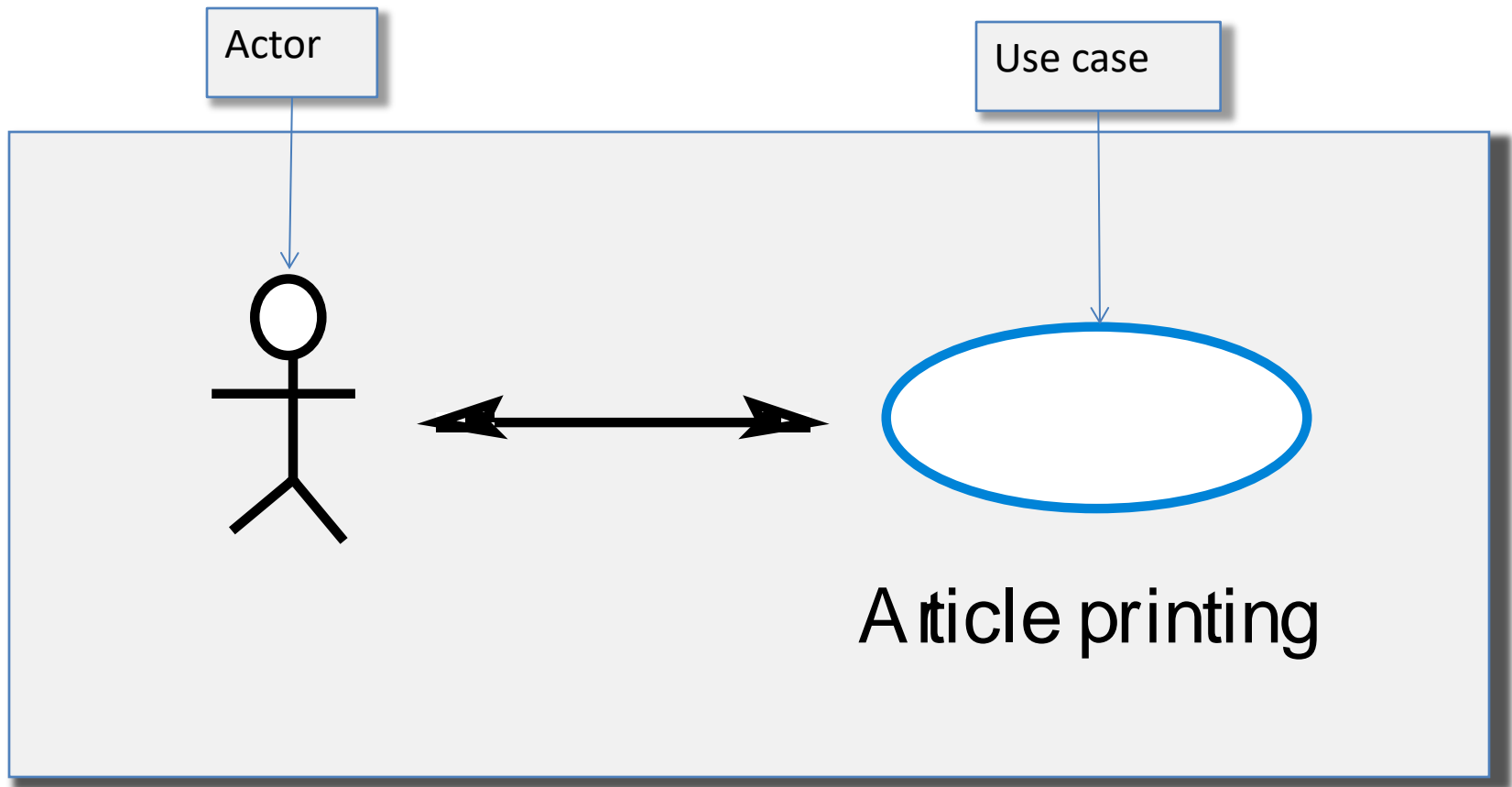
- Anything external to the system which the system interacts with
- Can be human
 - Customer
 - Player (game)
 - Driver
- Can be non-human
 - Sensor (smoke detector)
 - Payment service (credit cards)
 - Geo location
 - Robotic arm
 - Email server

Use Cases

- **The details of each use case should also be documented by a use case description: E.g.,**
 - **Print receipt** – A customer has paid for an item via a valid payment method. The till should print a receipt indicating the current date and time, the price, the payment type and the member of staff who dealt with the sale.
 - **[Alternate Case]** – No print paper available – Print out “Please enter new till paper” to the cashier’s terminal. Try to print again after 10 seconds.

An alternate case here is something that could potentially go wrong and denotes a different course of action.

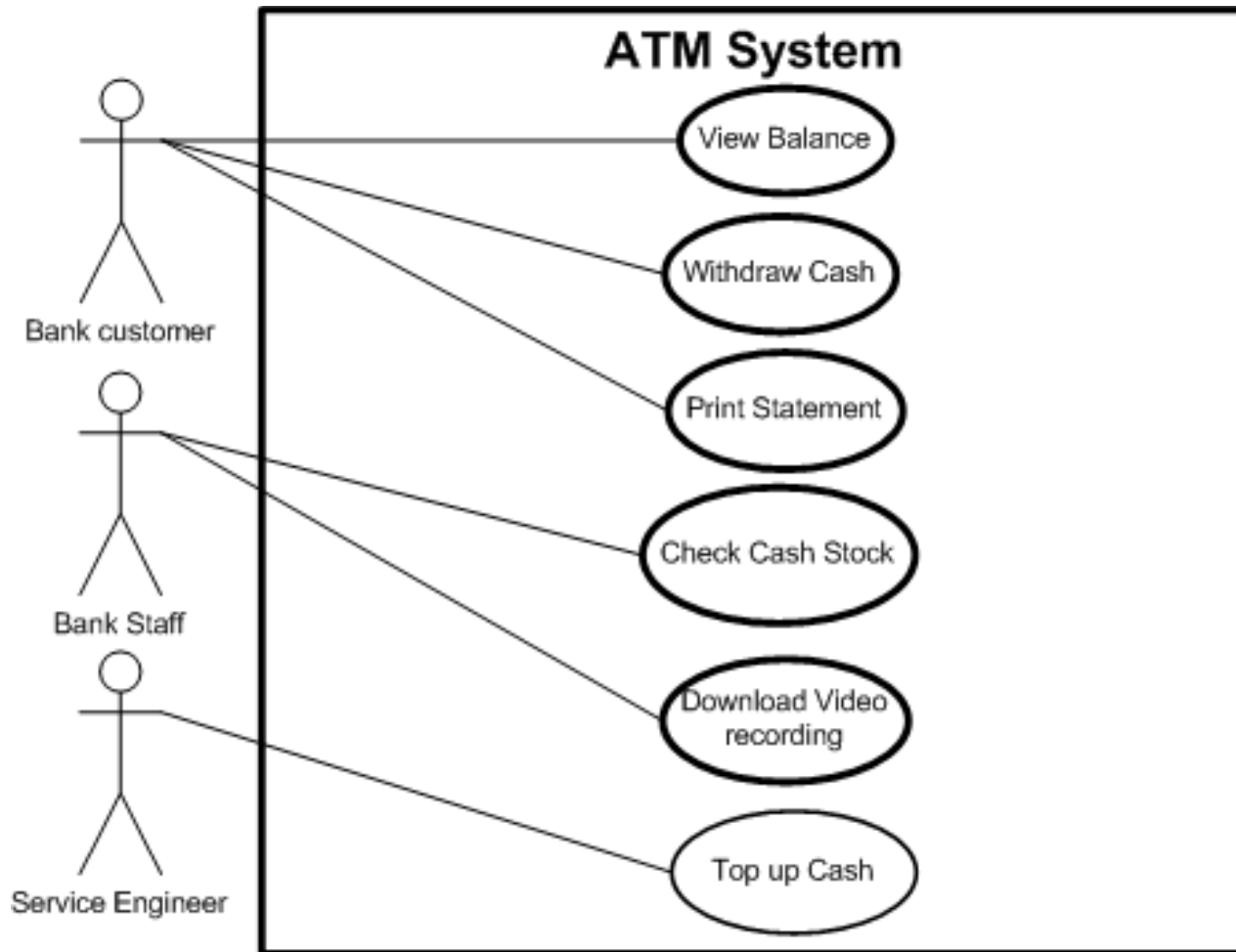
Example - Article Printing Use-Case



ATM machine

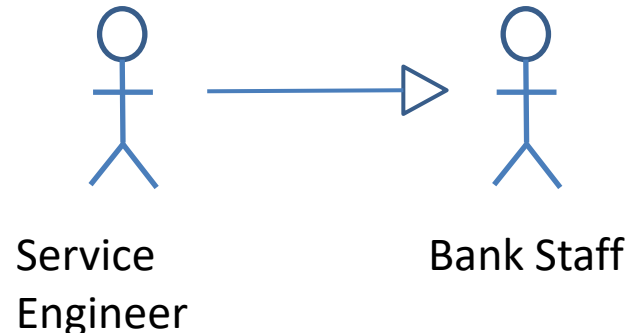
- Actors
 - Customers
 - Bank staff
 - ATM service engineer
- Use cases
 - Withdraw cash
 - Check balance
 - Add cash to machine
 - Check security video recording

Example - ATM Use Case Diagram



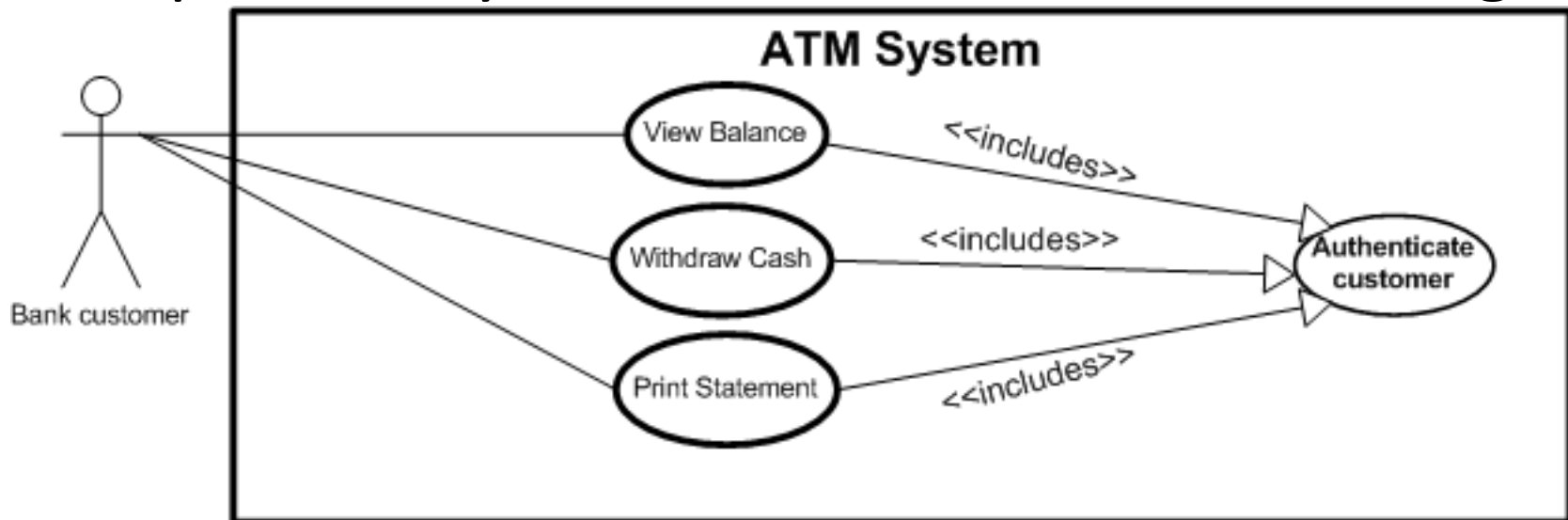
Advanced Use Case Diagrams

- We can draw a box (with a label) around a set of use cases to denote the system boundary, as on the previous slide (“ATM system”).
- **Inheritance** can be used between actors to show that all use cases of one actor are available to the other:



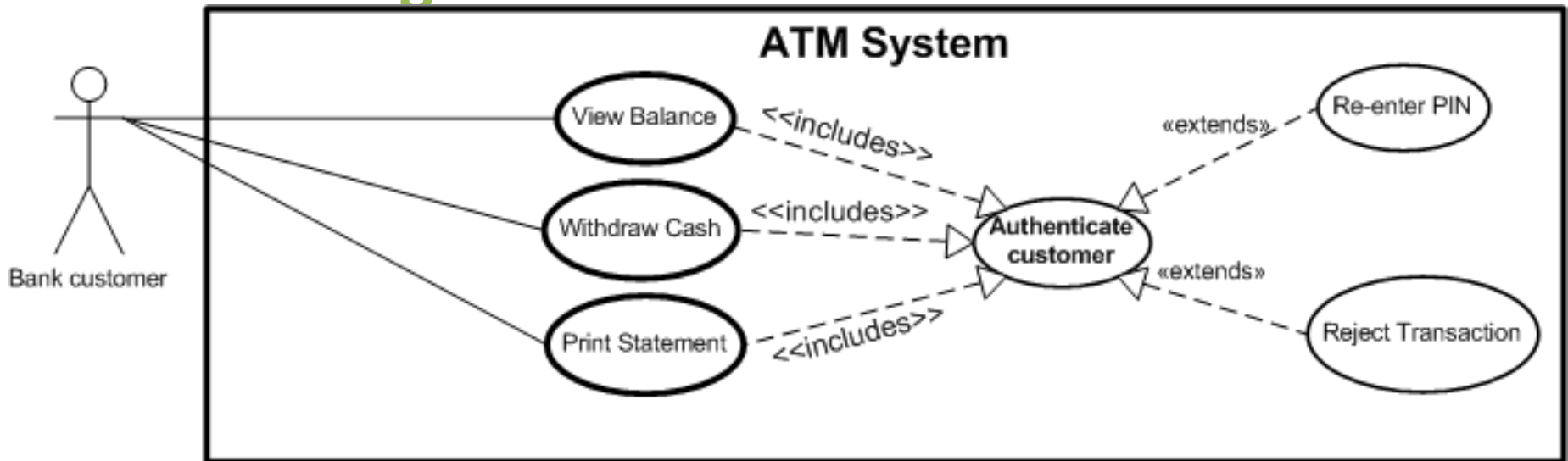
Include Relations

- If several use cases include, as part of their functionality, another use case, we have a special way to show this in a use-case diagram



Extend Relations

- If a use-case has two or more significantly different outcomes, we can show this by **extending** the use case to a **main use case** and



In summary

- Include
 - When the other use case is always part of the main use case
- Extend
 - When the other use case, is needed as an optional outcome due to certain circumstances/scenarios

A Word on Extend/Include

- Note the directions of the arrows in the previous two slides, they are different for each (according to whether a use case “includes” another, or “extends” it).
- One of the benefits of UML diagrams is their simplicity and that they can be shown to and worked through with, customers.
- This is to some extent lost by using more advanced features like “include” and “extend” relations; they should thus be used with care.

Full use case template

- **ID**
 - Short ID (useful for diagrams and reference) Example UC1
- **Name**
 - Full name Login
- **Description**
 - Full description User login
- **Pre-condition**
 - What must be true before the use case can proceed System is online
- **Event flow**
 - Flow of behaviour that makes up this use case, can include other use cases
user enters username + password, if credentials correct user is authenticated
- **Post-condition**
 - Possible change of internal state (e.g. authenticated=true if username, password correct, user account locked if wrong password 3 times)
- **Includes**
 - What other use cases are used
- **Extensions**
 - Optional behaviour
- **Triggers**
 - What makes this use case happen

Notes about use cases

- They do NOT describe internal behaviour
- Must describe behaviour with external Actors
- But external Actor can be
 - External system (e.g. Paypal)
 - External hardware (e.g. smoke detector fire alarm)
 - External agency (e.g. Police, fire brigade)
- So use cases are always involve external behaviour

ATM use case descriptions

ID	UC1
Name	Withdraw cash
Description	Bank customer withdraws cash from machine
Pre-condition	ATM in service
Pre-condition	ATM has sufficient cash stock
Event flow	<ol style="list-style-type: none">1. Include Use case 2 “Authenticate customer”2. Choose quick cash or enter exact amount3. Choose receipt option4. Take cash
Extension points	Use case 4 “Balance too low”
Triggers	Withdraw cash button pressed
Post condition	Transaction added to account

ATM use cases

ID	UC2
Name	Authenticate customer
Description	Bank customer proves their identity to the ATM.
Pre-condition	ATM in service
Pre-condition	User not authenticated
Event flow	<ol style="list-style-type: none">1. If user already authenticated exit from user case.2. User enters card and PIN number3. User re-enters PIN if PIN incorrect
Extension points	Use case 5 "Card stolen" Use case 6 "PIN entry failure"
Triggers	Authenticated service requested
Post-condition	User is authenticated if credentials correct

ATM use cases

ID	UC3
Name	Check balance
Description	Bank customer retrieves a balance for their account
Pre-condition	ATM in service
Event flow	<ol style="list-style-type: none">1. Include Use case 2 “Authenticate customer”2. Choose onscreen or paper balance
Extension points	None
Triggers	Check balanced requested
Post-condition	None

ATM use cases

ID	UC4
Name	Balance too low
Description	Bank customer cannot make cash withdrawal due to low balance
Pre-condition	None
Event flow	1. Customer chooses smaller amount or cancels transaction
Extension points	None
Triggers	Cash chosen greater than available balance
Post-condition	None

Next lecture...

- Please look at Hotel booking case study
- Please prepare this case study problem..