MANCHESTER 1824

The University of Manchester

COMP23420 Lecture 3

Realising Use Cases

Kung-Kiu Lau

kung-kiu@cs.man.ac.uk

Office: Kilburn 2.68

Overview

- Active learning sheets (during lectures)
- Generic software development process (recap)
- ATM case study
- Realising use cases
- Workshop 2 (Functional Modelling: Use Cases)

MANCHESTER 1824

The University of Manchester

COMP23420 Active Learning Sheet for Lecture 3

The 4 main phases of the generic software development process are:

The functional model consists of:

The structural model consists of:

The behavioural model consists of:

A use case is realised by .

A domain class can participate in the realisation of use cases.

A domain class that participates in different use cases realisations play different in these collaborations, and have different in these collaborations.

A communication diagram for a group of collaborating objects describes the between the objects.

Software Development Process

User's requirements

Software Development Process

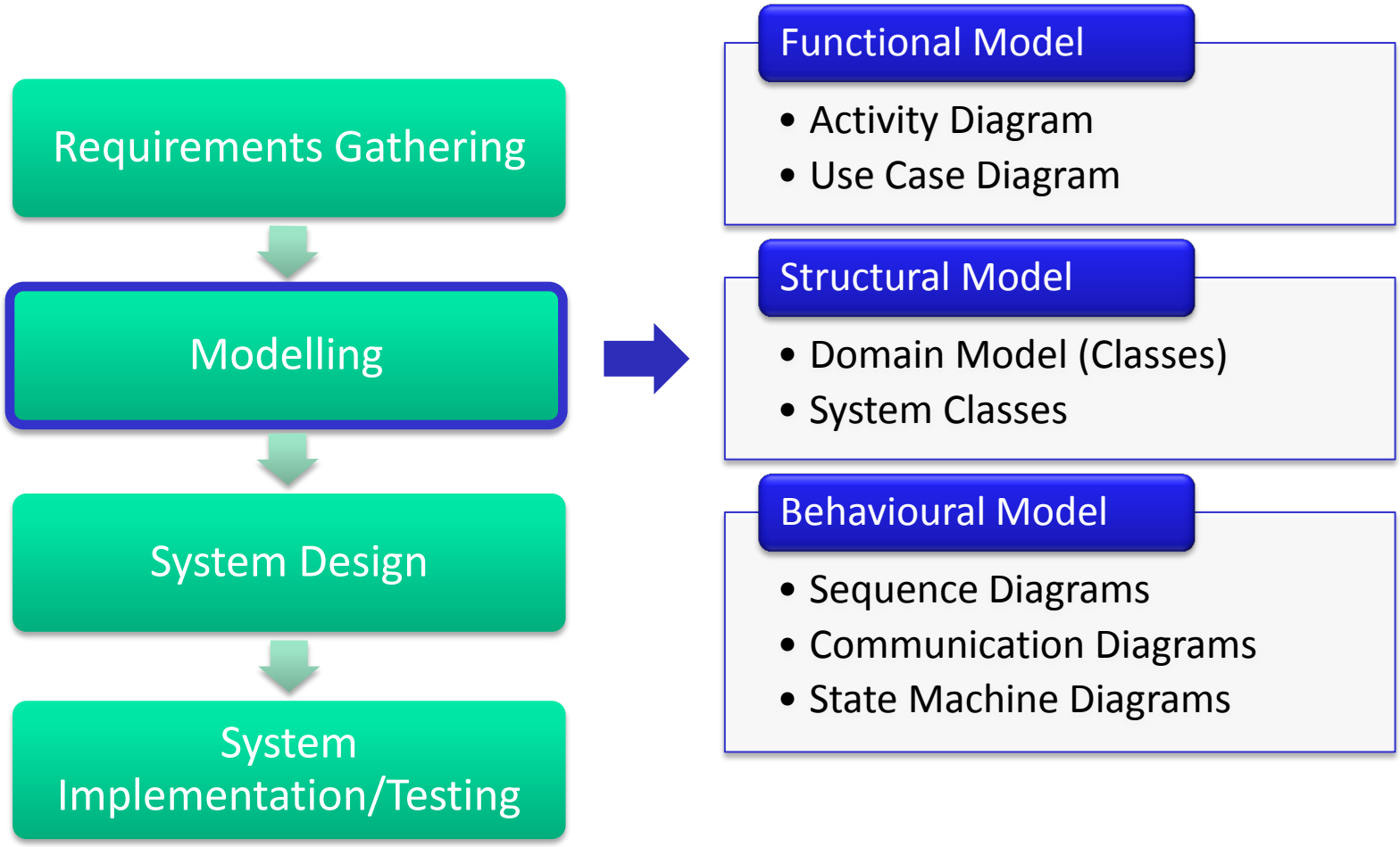
Software System

www.bigstock.com · 29436170

www.shutterstock.com · 57909892

gg58120616 www.gograph.com

A Generic Software Development Process



MANCHESTER 1824

The University of Manchester

ATM Case Study

Complete example with UML modelling

- From requirements to behavioural model

Available on Moodle

COMP23420: Software Engineering

The ATM Example

Kung-Kiu Lau and Robert Stevens
kung-kiu@cs.man.ac.uk
robert.stevens@manchester.ac.uk

Contents

http://moodle.cs.man.ac.uk/file.php/238/ImportantDocuments/Semester1/atm.pdf

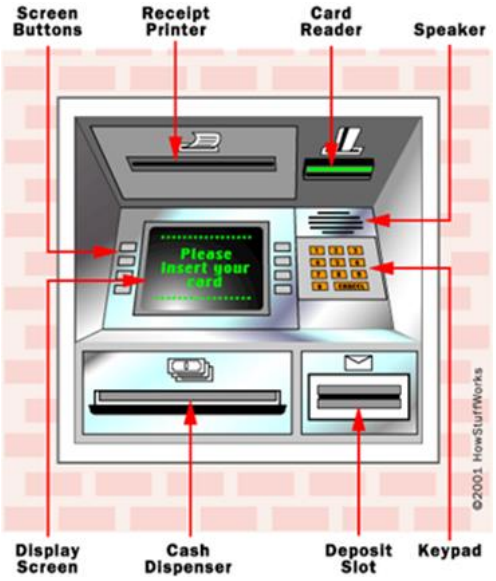
Realising Use Cases

The software development process is all about realising use cases.

All use cases must be realised (to meet all user requirements).

Use cases are realised by domain classes

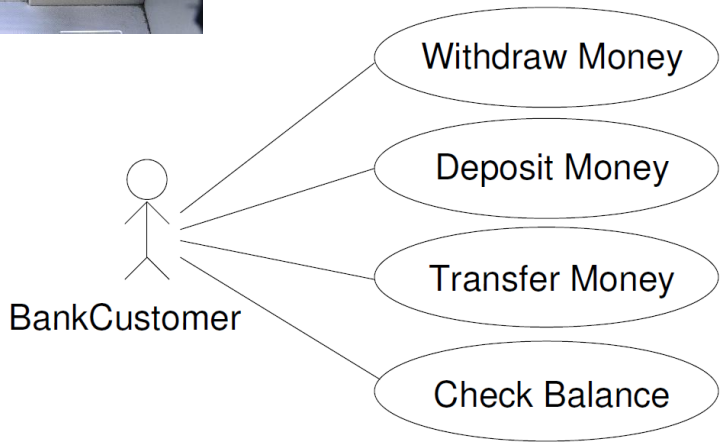
- conceptual classes that represent entities in the problem domain

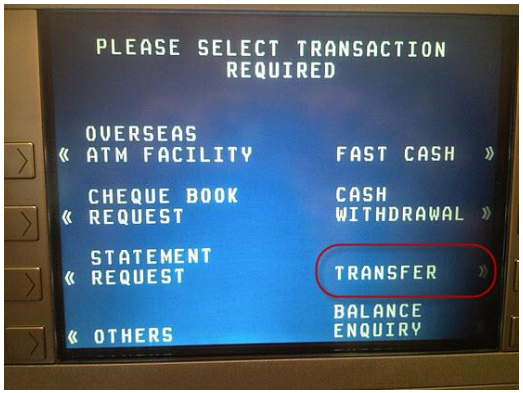


MANCHESTER 1824

The University of Manchester

Realising Use Cases: the ATM Example



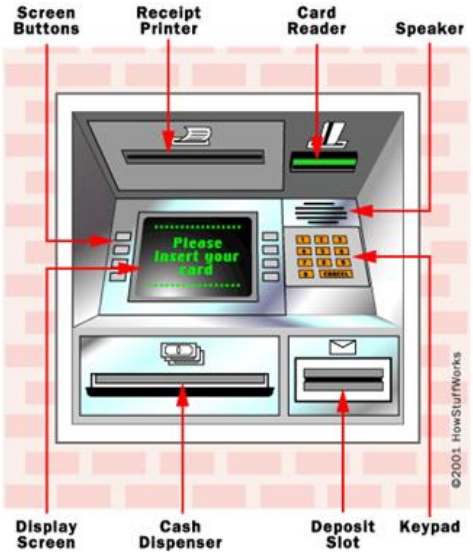


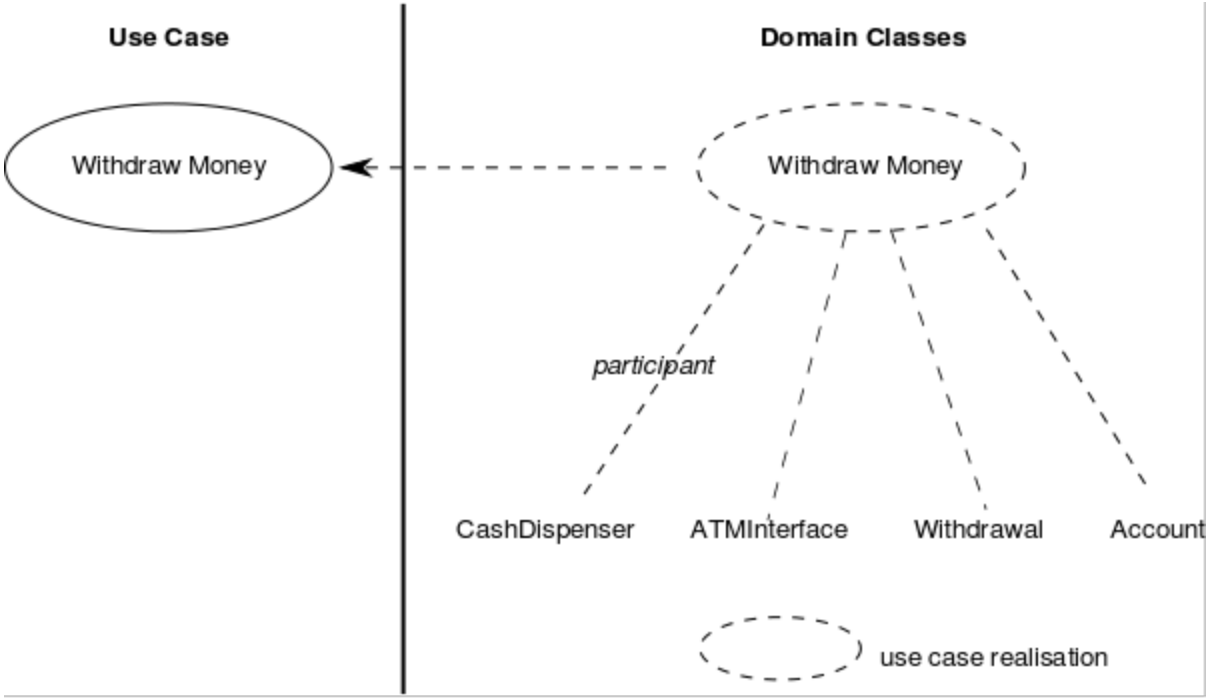
Use cases for the ATM example

Acknowledgement: This example is taken from, and its associated diagrams are based on corresponding ones in, Jacobson, Booch and Rumbaugh, *The Unified Software Development Process,* Addison Wesley, 1999.

Realising Use Cases: the ATM Example

For each use case, domain classes that realise the use case must be identified.

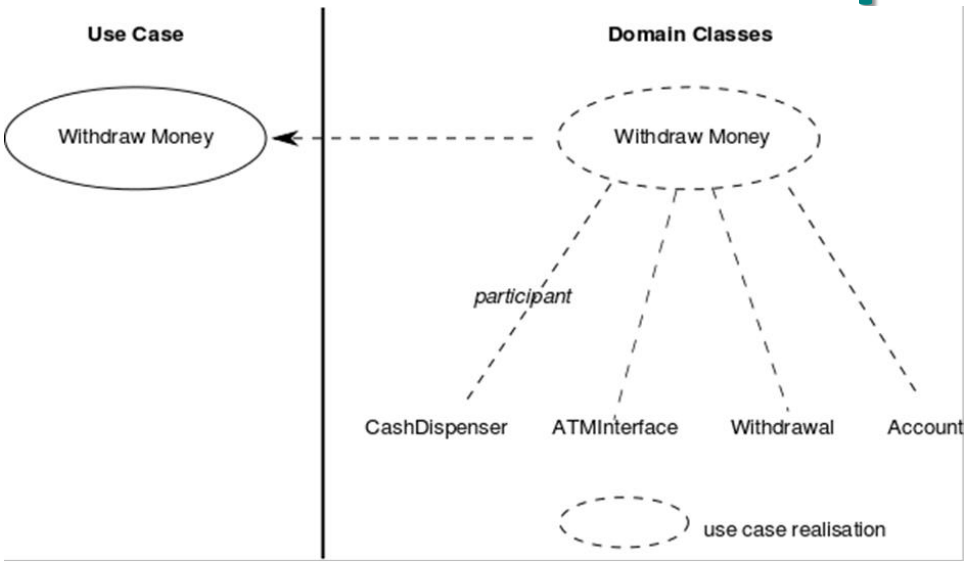




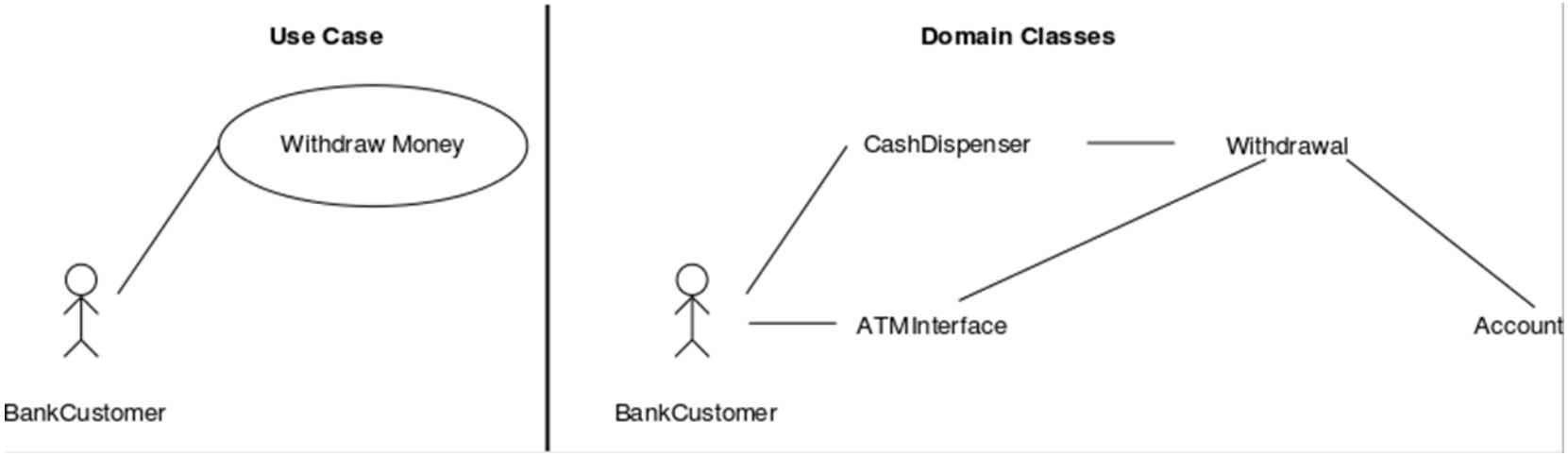
Realising the Withdraw Money use case by domain classes

Realising Use Cases: the ATM Example

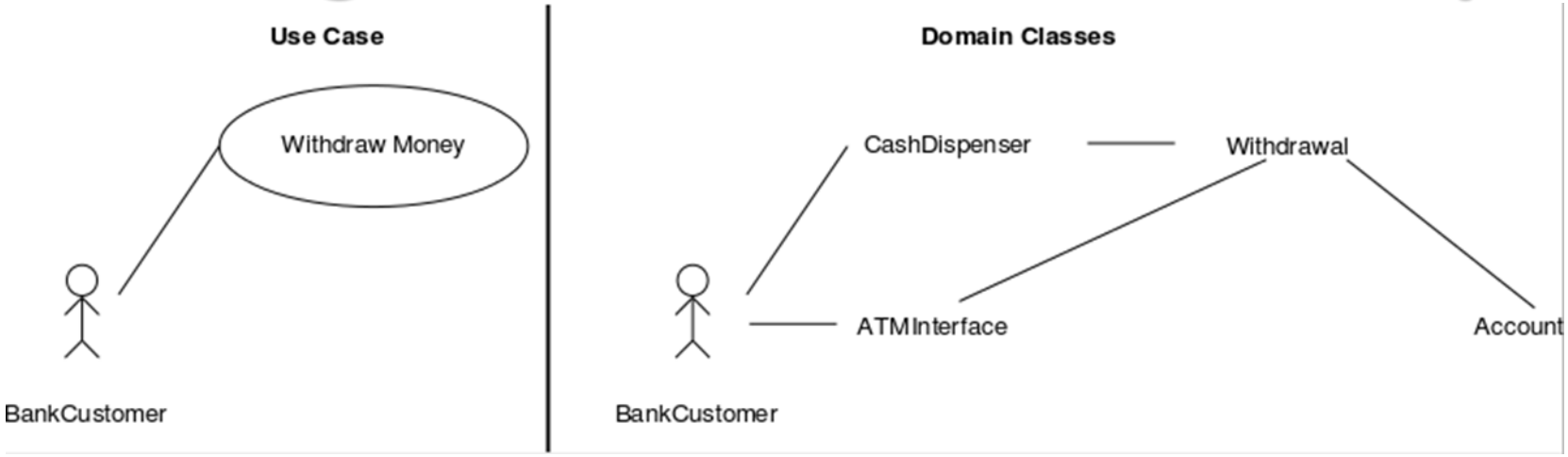
Typically a use case is realised by several collaborating domain classes.



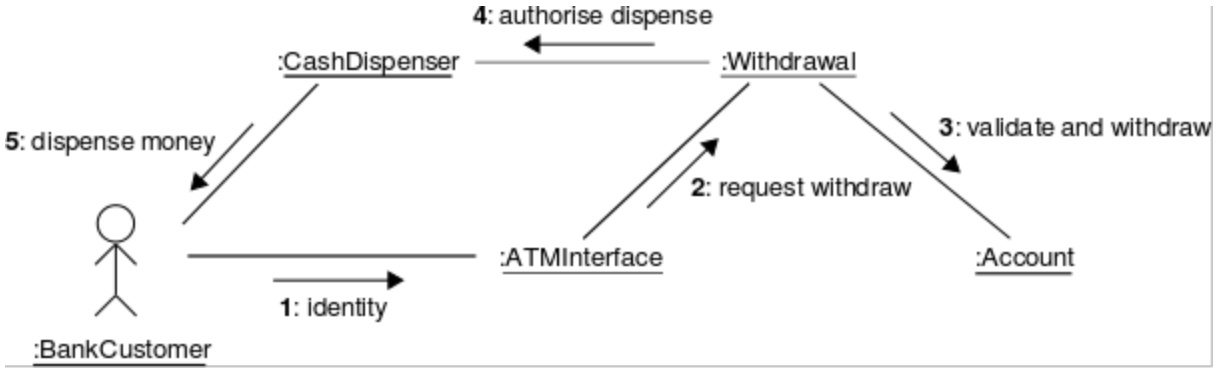
The realisation of a use case can also be modelled as follows:



Realising Use Cases: the ATM Example

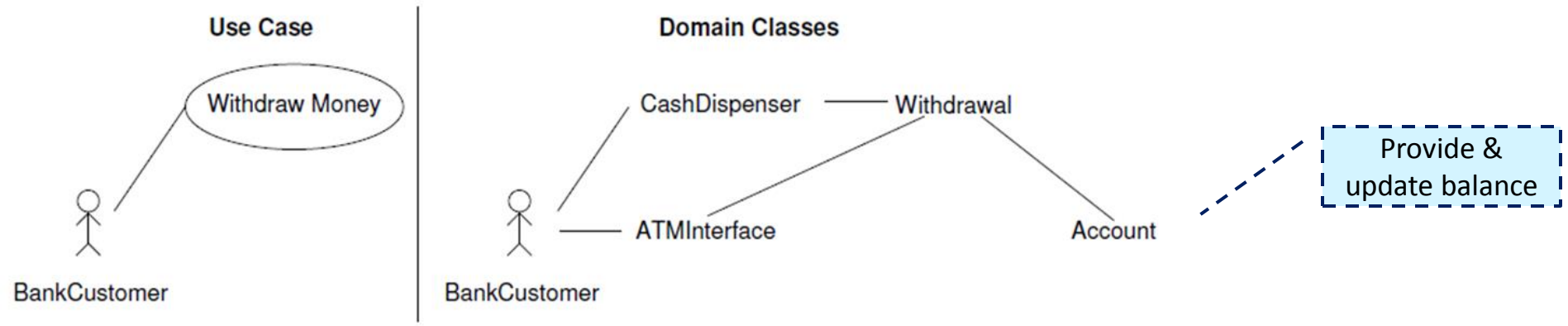


The details of a collaboration (of domain classes) can be described by a communication diagram.

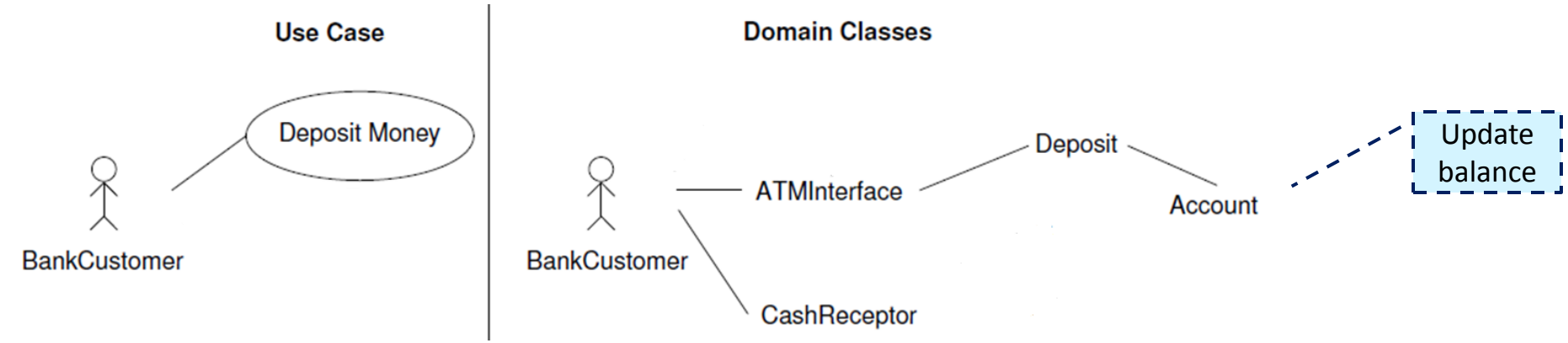


Roles and Responsibilities

A domain class can participate in several use case realisations.

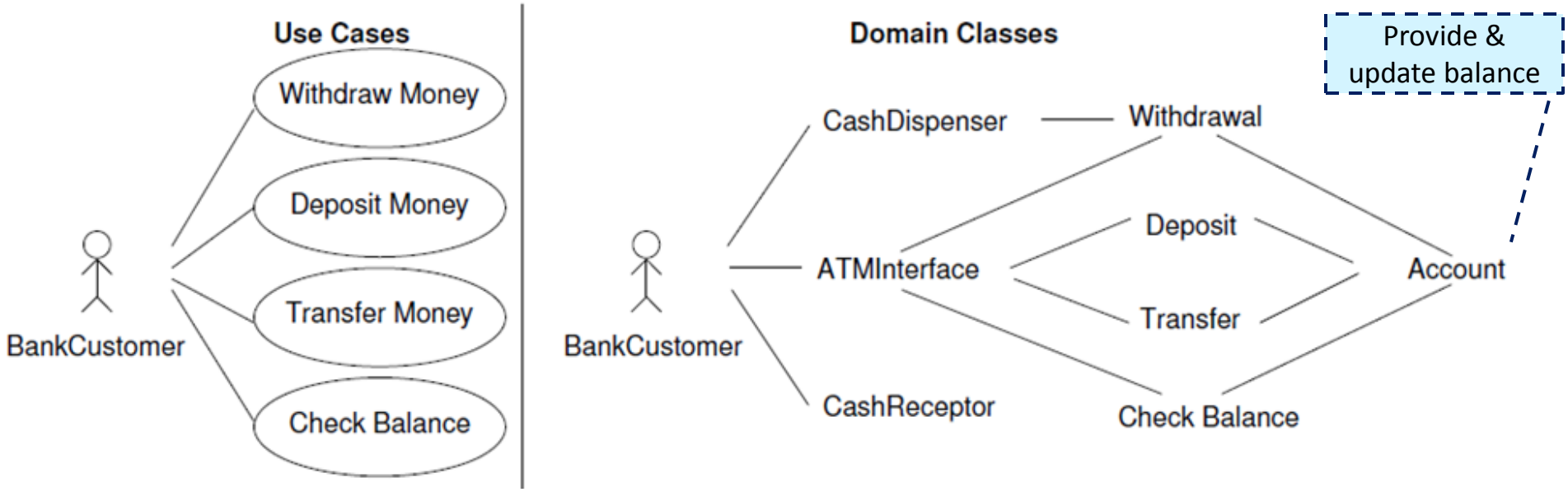


A domain class plays different roles in different use case realisations, i.e. it has different responsibilities.



Realising All Use Cases

All use cases must be realised.

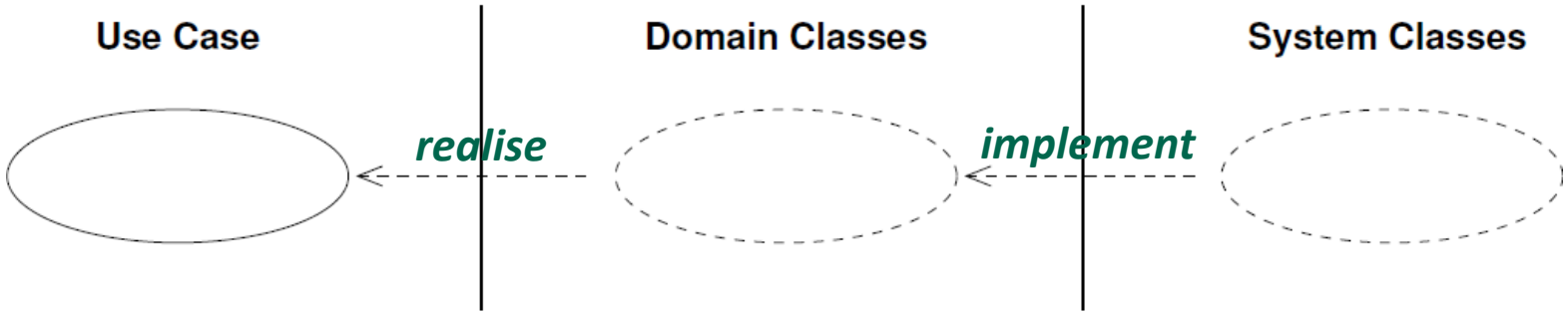


The responsibilities of a domain class are the sum total of its responsibilities in all the use case realisations in which it plays a role.

Refining Domain Classes

Domain classes are refined into system classes

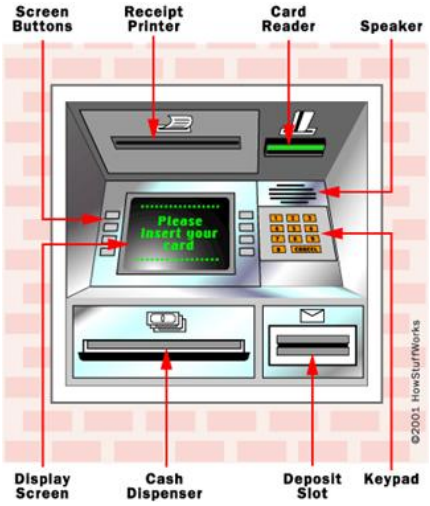
- real classes used for implementing the system

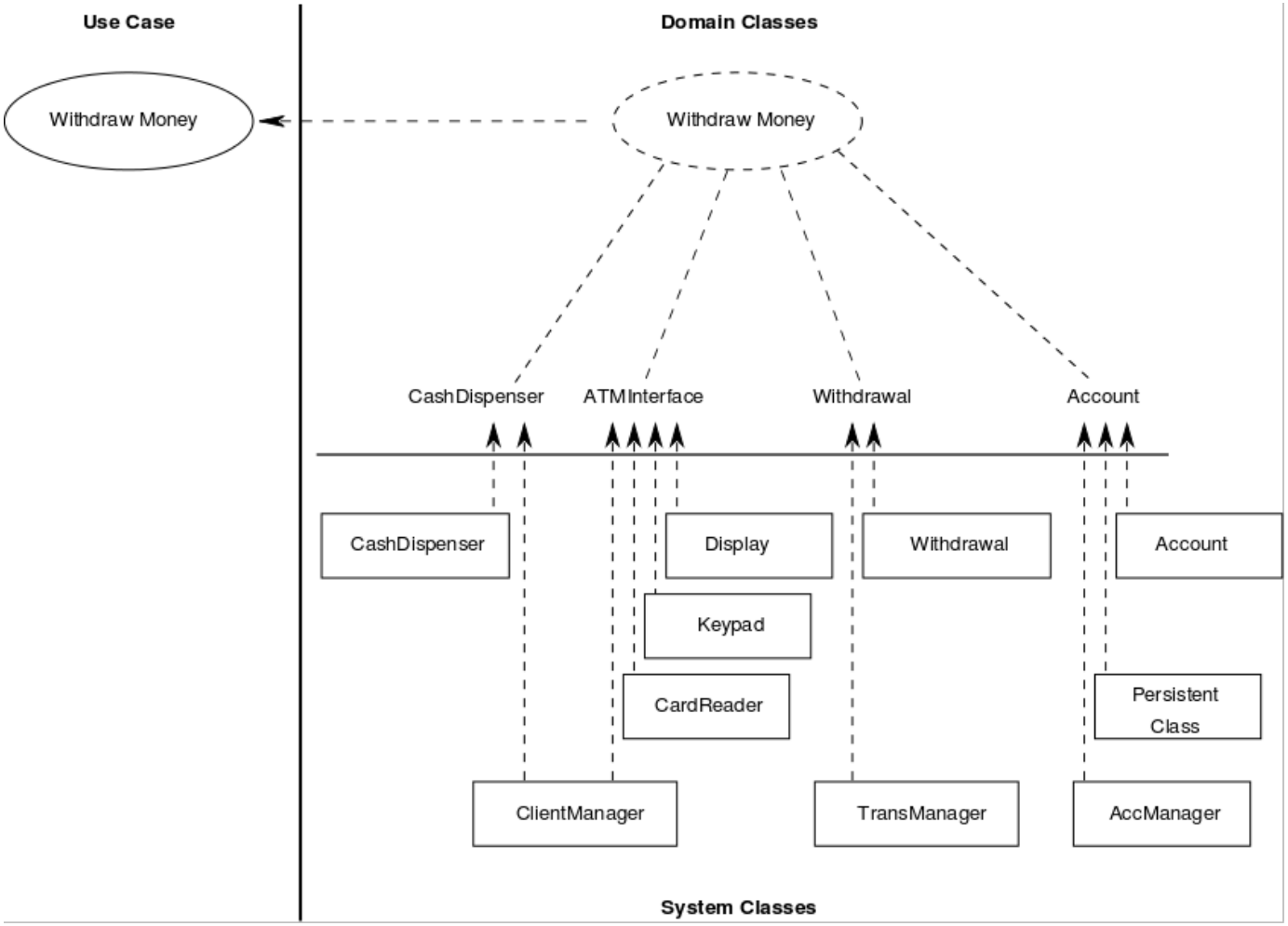


Use case realisation by domain classes and system classes.

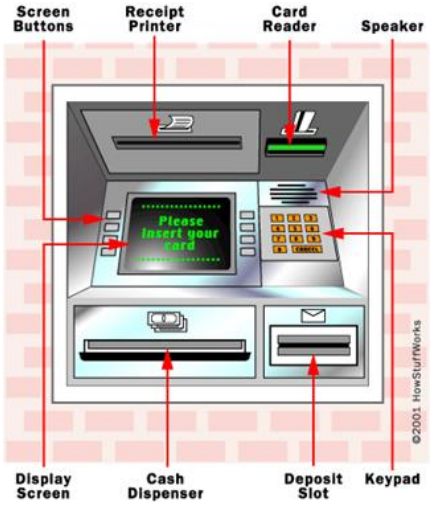
During the system design phase (of the development process), system classes may be further refined into classes used in the final implementation.

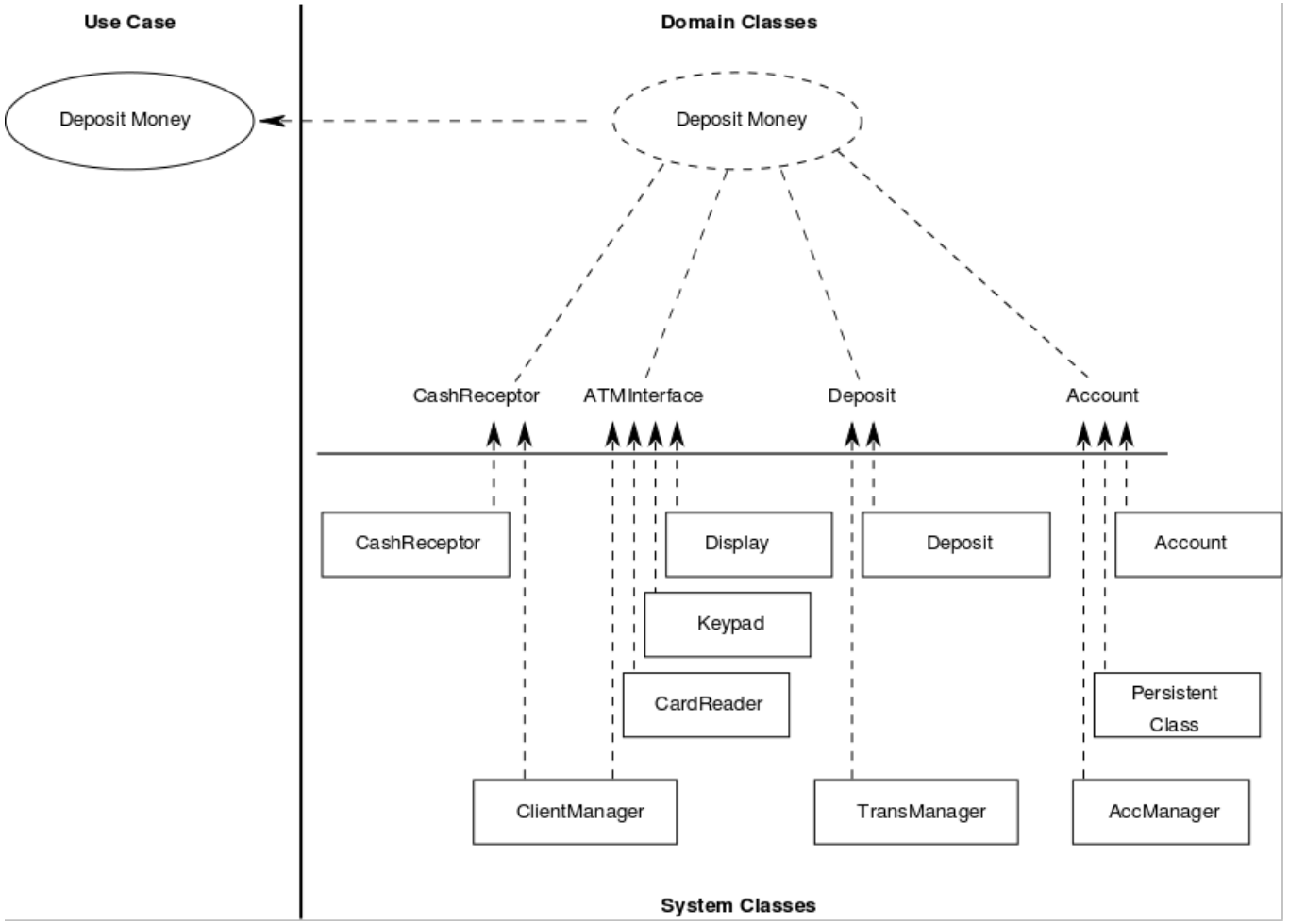
Refining Domain Classes: the ATM Example





Refining Domain Classes: the ATM Example





Summary

- Realising use cases is what it's all about
- Use cases are realised by domain classes (and system classes)
- A domain class can participate in the realisation of several use cases
- Domain classes are refined into system classes
- System classes may be further refined into classes used in the final system implementation

Workshop 2
Functional Modelling: Use Cases

- Identify use cases
- Draw use case diagrams
- Write use case descriptions