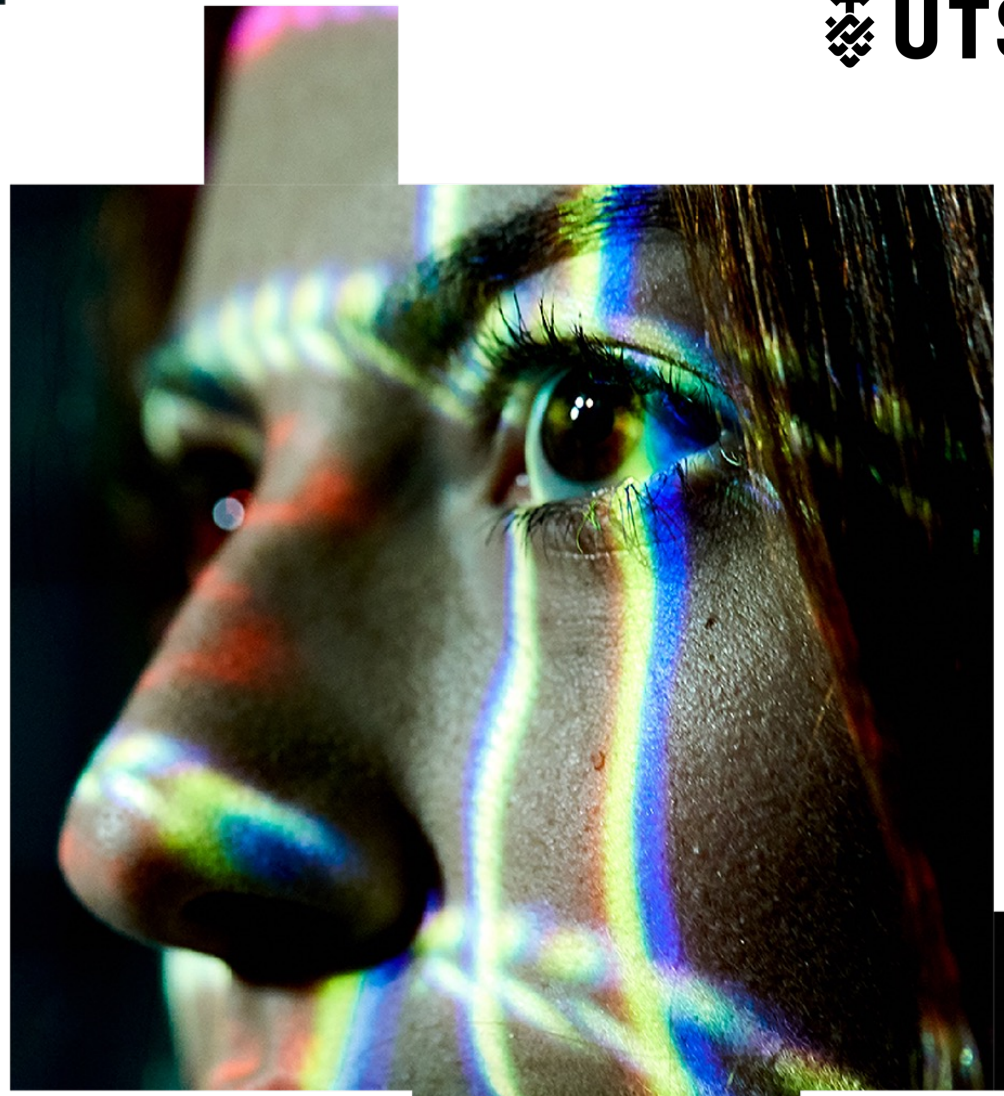
Module 4:
OO Modelling & UML



SSTC 2022
Module 4 – Overview

Dr. Salvatore Flavio Pileggi

SalvatoreFlavio.Pileggi@uts.edu.au
https://www.uts.edu.au/staff/salvatoreflavio.pileggi

School of Computer Science, Faculty of Engineering and IT
University of Technology Sydney (Australia)

OO Modelling

Module 4 goals

OO Modelling, focus on Requirements

- Appreciate how Object-Oriented Modelling techniques can help to **understand the working of business systems**
- How OO Object-Oriented Modeling can be used to **specify systems and user requirements**

Object-oriented Modelling (OOM)

A simple definition …

Object-oriented modelling (**OOM**) is an approach to **modelling an application** that is used at the beginning of the software life cycle when using an **object-oriented approach** to software development.

https://en.wikipedia.org/wiki/Object-oriented_modeling

OO Modelling

Key concepts

By definition, OO Modelling is based on:

- **Objects**
  - Represent **real-world or abstracted things**
- Normally, Objects contain
  - **Methods** (or processes)
  - **Attributes** (or data)

OO Modelling

OO vs Structured Modelling

OO Approach:

- Applications are understood as a **collection of interacting objects**
- Objects may interact with elements external to the application (e.g. people or other systems) and each other
- Objects send and respond to messages/methods

Structured approach:

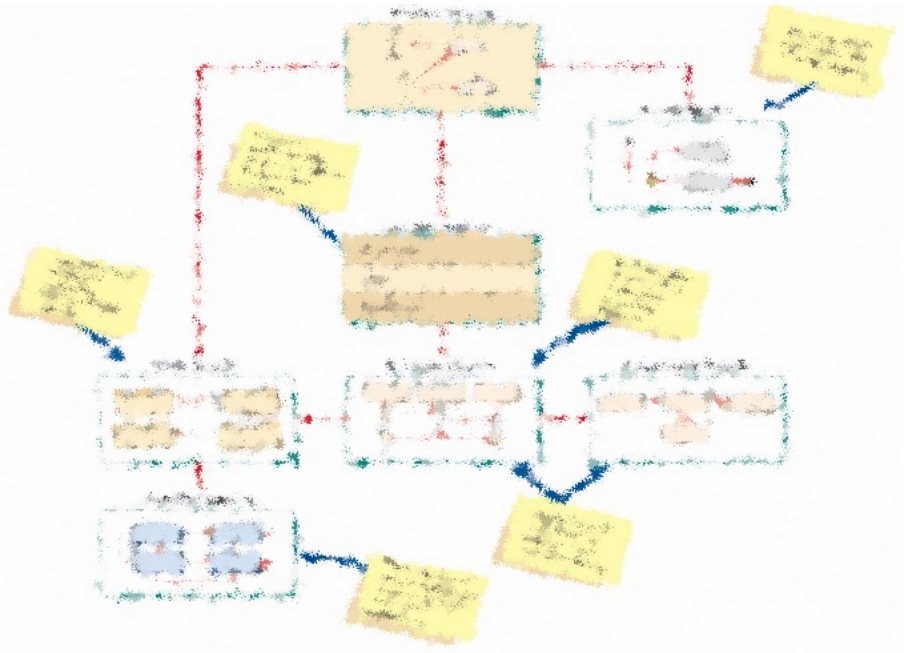
- Applications are modelled as a collection of **processes organized into a system**
- Processes interact via flows of data
- Processes accept inputs and produce outputs

Unified Modeling Language (UML)

Unified Modelling Language (UML)

Overview

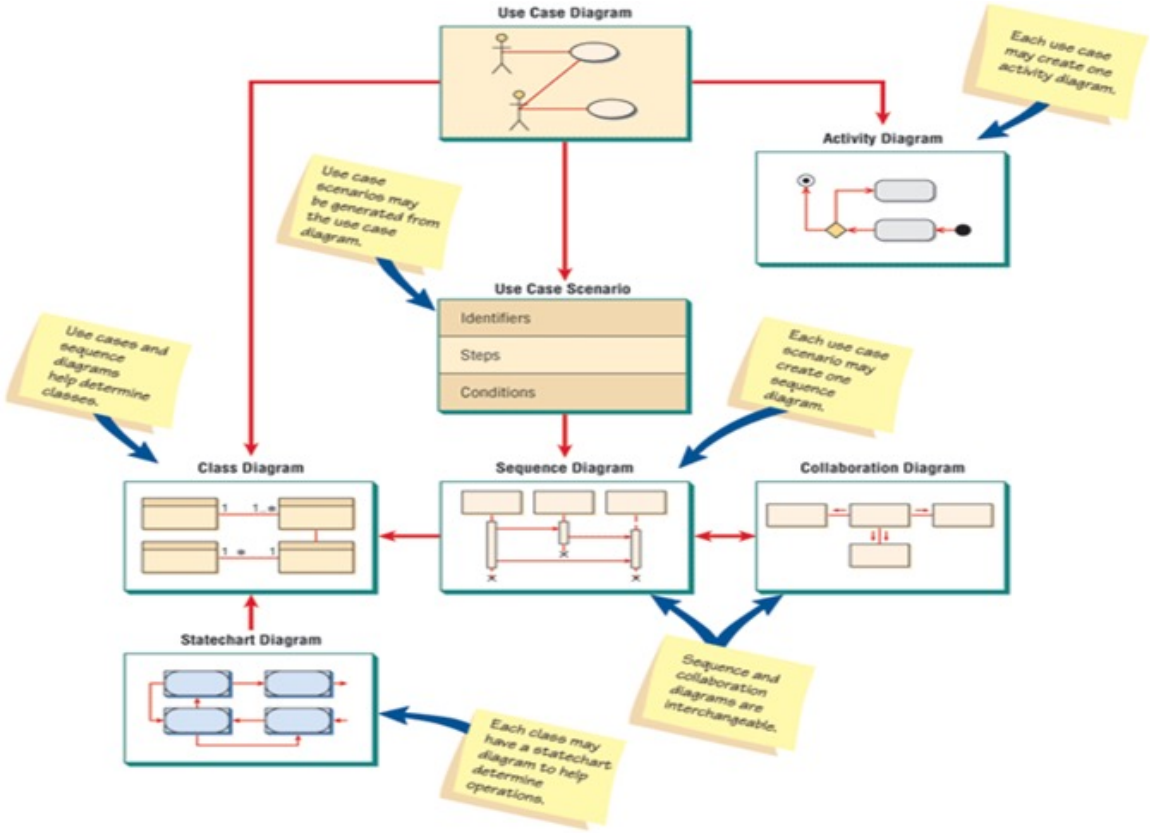
- UML is a **standard language** for specifying, visualizing, constructing, and documenting the artefacts of software systems.
- It is an industry standard developed **to support Object-Oriented analysis and design** (currently version 2.X)
- UML is a **visual language** based on diagrams used to make software blue prints.
- UML diagrams are not only made for **developers** but also for **business users** and anybody interested to understand the system.
- Specializations of UML (e.g. SoaML for Service-Oriented Architectures)



Unified Modelling Language (UML)

UML diagrams

- No UML diagram alone can describe/model all aspects of a system
- The key idea is to have **multiple models** to address the different aspects (views)

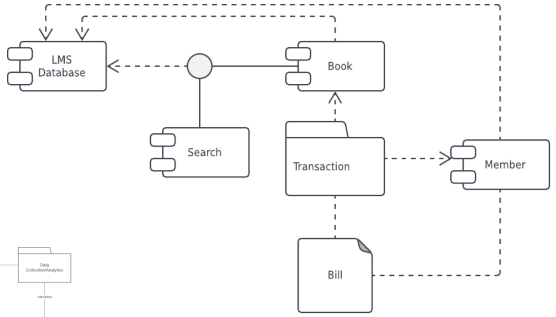


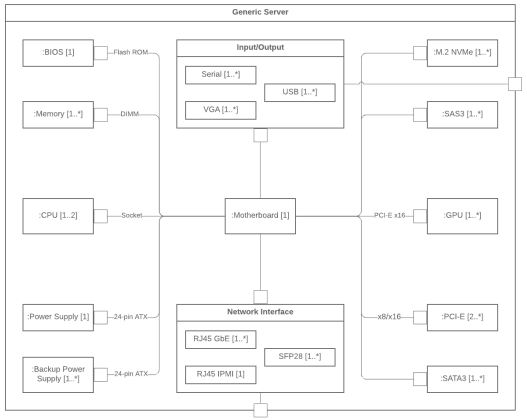
Unified Modelling Language (UML)

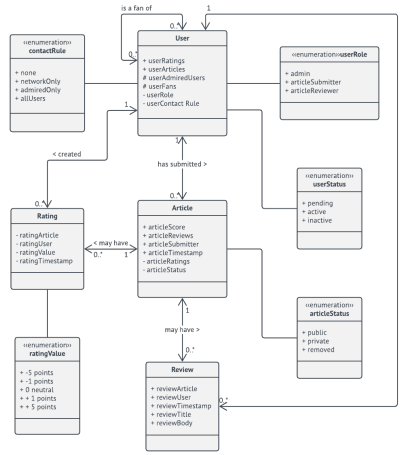
UML diagrams

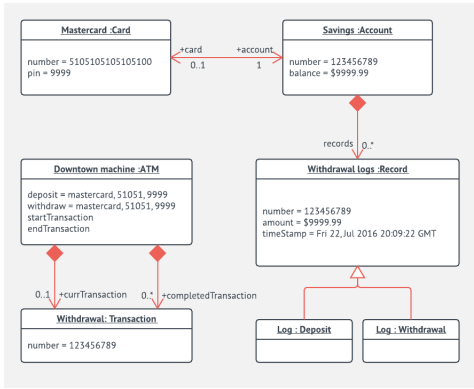
Structural diagrams

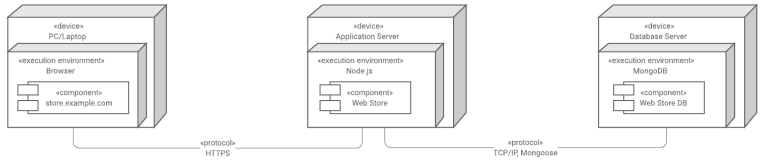
- **Class diagram**
- **Component diagram**
- **Deployment diagram**
- **Composite structure diagram**
- **Object diagram**
- **Package diagram**











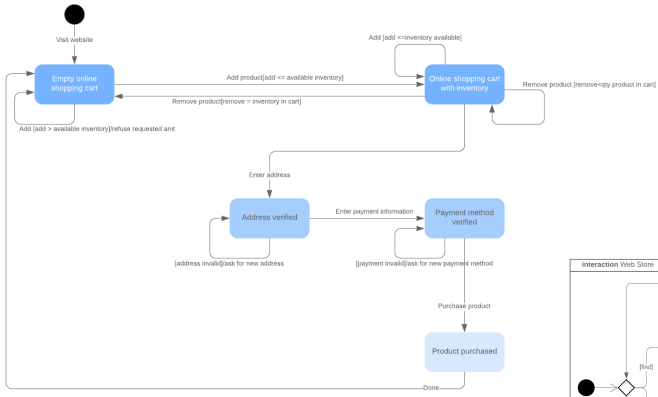
https://www.lucidchart.com/blog/types-of-UML-diagrams

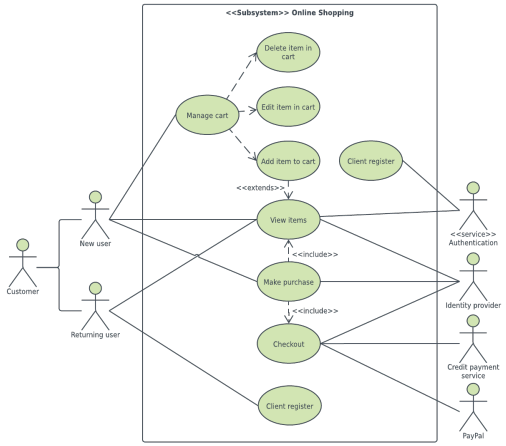
Unified Modelling Language (UML)

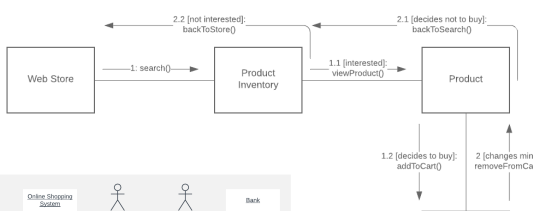
UML diagrams

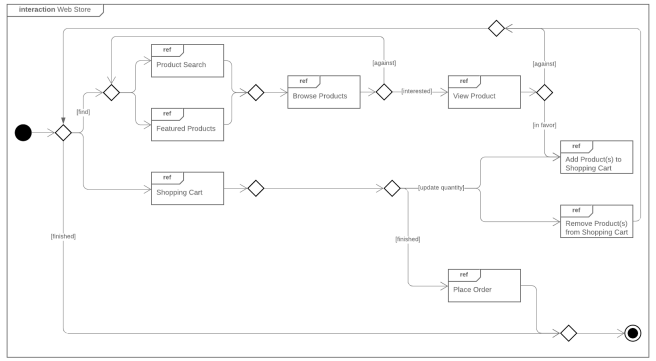
Behavioural diagrams

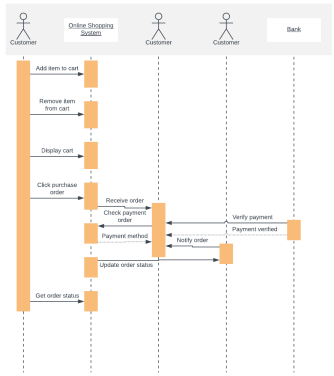
- **Interaction overview diagram**
- **Communication diagram**
- **State diagram**
- **Use case diagram**
- **Sequence diagram**

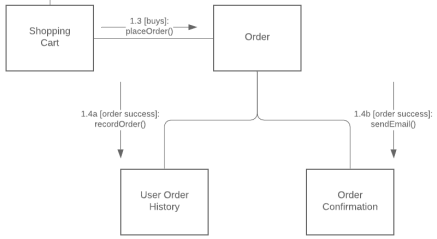












https://www.lucidchart.com/blog/types-of-UML-diagrams


Thank You!