7SENG001 Enterprise Application Development

MVC Architecture

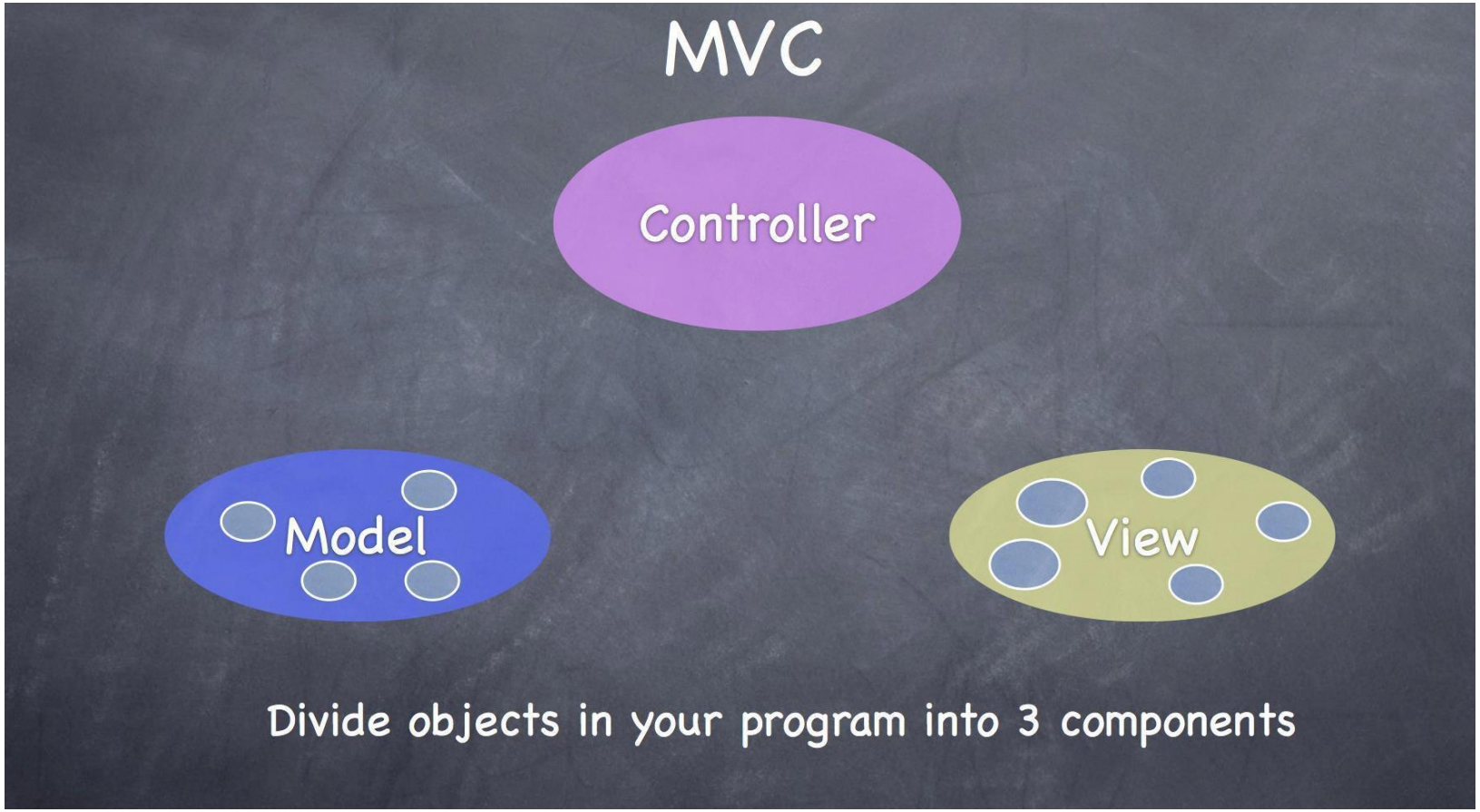
Week 2

Learning Outcomes

- MVC
- UML
- The Model
- Design
- UML introduction
- UML tools
- Domain Modelling

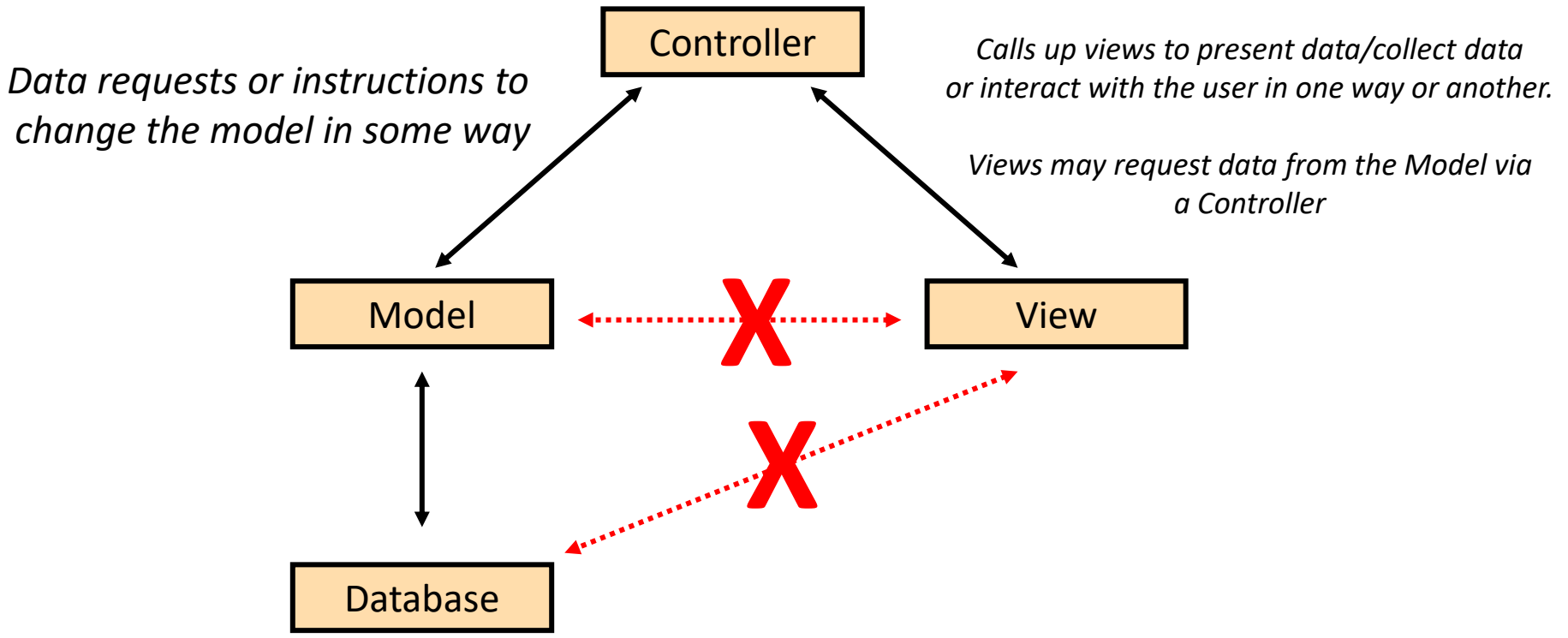
Object Oriented Design Patterns

The Model View Controller



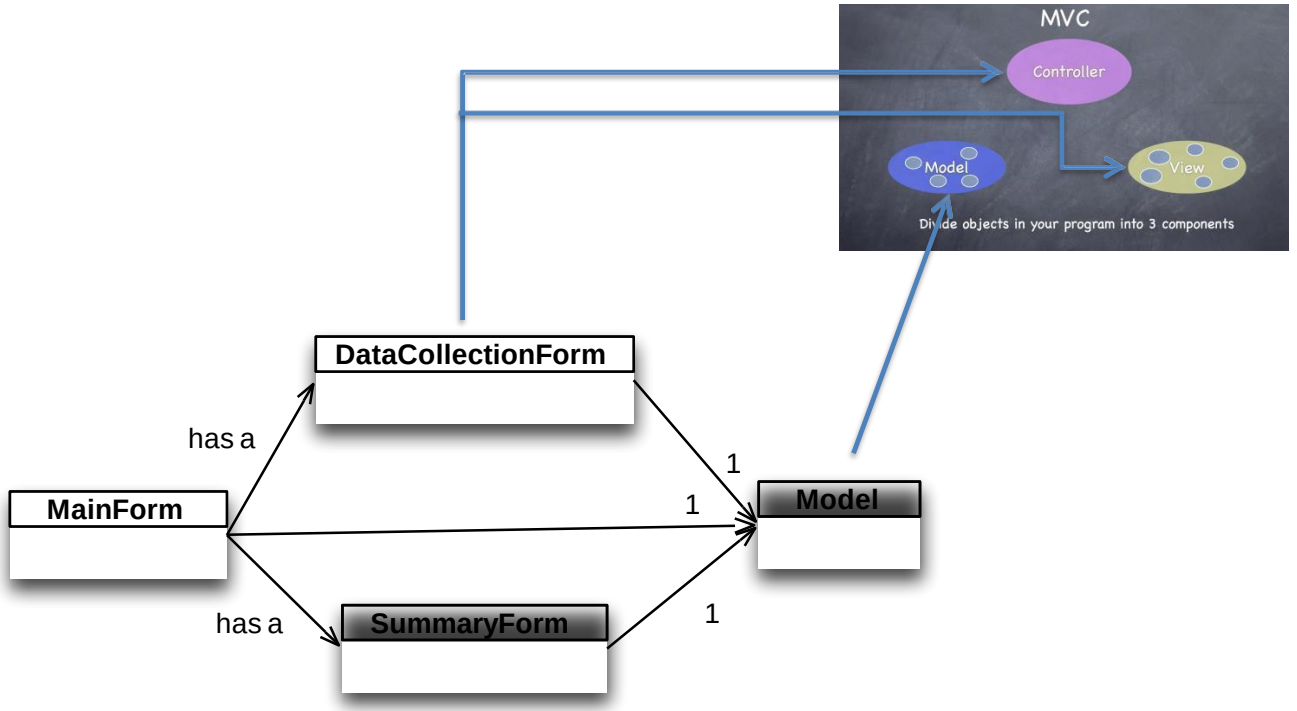
Object Oriented Design Patterns

The Model View Controller



MVC Pattern – Windows Forms

In Windows Forms the View and Controller generally reside in the same class so you need to be aware of this



Module summary so far

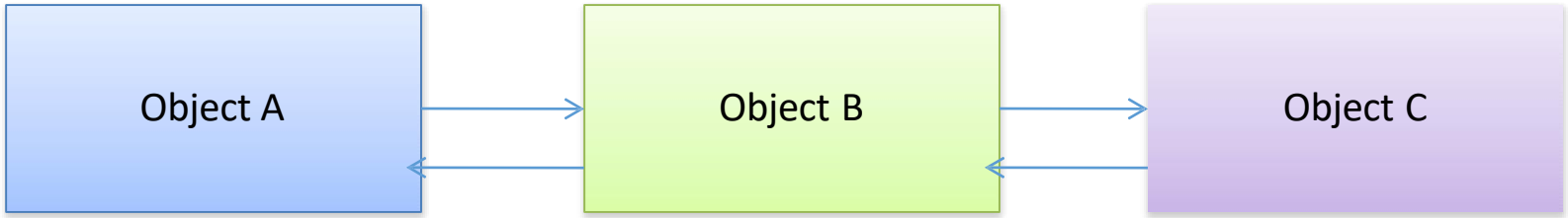
- We have largely concentrated on OOP
- But what about software design?

The model

- An abstract view of the system
- Documents the system architecture
- Means of communicating design
- Different Models
  - **Analysis models**
    - Presents information about a given application area **(the problem space)**
    - Used to gain understanding of the application
  - **Design Models**
    - Presents information about the software being developed **(the solution space)**
    - Defines the structure of the software and guides implementation
  - Object Oriented Concepts can be used to construct both types

The Object Model

- 1. The basic architecture of a system is assumed to be a set of classes
- 2. At run-time, instances of these objects interact (pass messages) to perform various tasks



UML The Unified Modelling Language

- UML is the most widely used OO design language
- Originally developed at the Rational Corporation, now maintained by the OMG (Object Management Group) as a non-propriety standard
- **Not a methodology**
- Defines a language for documenting software systems

The nature of UML

- **Graphical** language not textual
- Different diagrams used to represent a facet of the system being designed
- **Views** - Systems can be described in a number of views (for different stakeholders)

UML Terminology

- **Models** present the information relevant in a particular view
- **Model Elements** are the atoms out of which models are built (classes, objects, methods and so on)
- **Diagrams** are graphical representations of collections of model elements
- Typically **each view** will have associated with it a number of diagram types

UML Diagram Types

- **Use Case Diagrams (not strictly part of UML but very useful)**

- **Class diagrams**

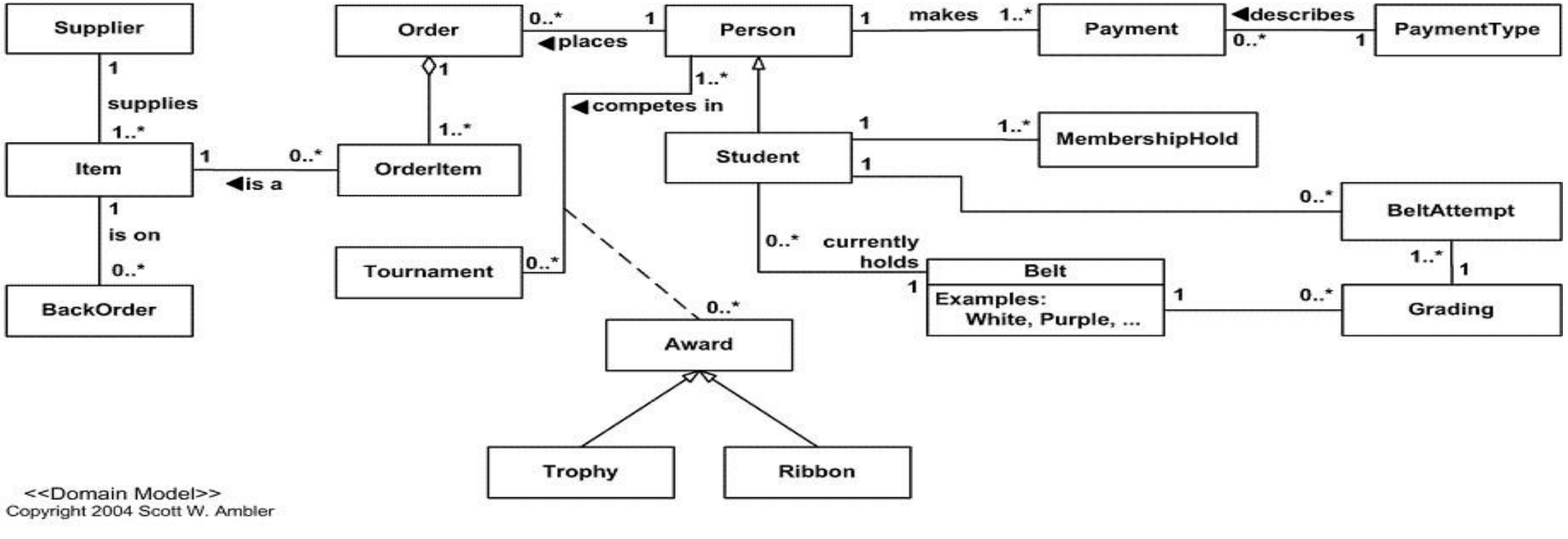
- **Interaction diagrams (e.g. sequence)**
  - show how object communicate when system is running

- **State diagrams (activity and state)**
  - show the state of objects over history

- **Component diagrams**
  - show the physical components (source code, libraries) that make up the system

- **Deployment Diagrams**
  - Show how the components are physically distributed over the intended target delivery platform

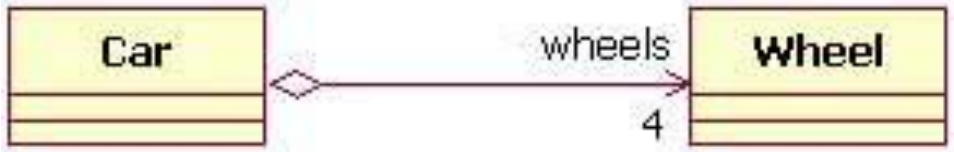
A Domain Model Example



A Domain Model Example

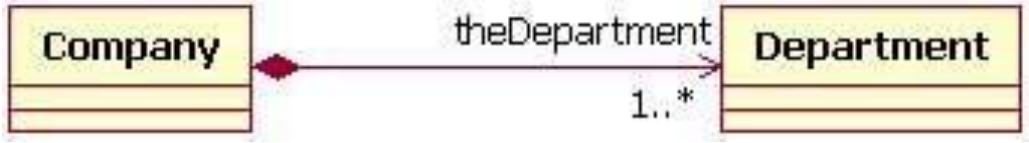
Basic aggregation

An association with an aggregation relationship indicates that one class is a part of another class. In an aggregation relationship, the child class instance can outlive its parent class.



Composition relationship

Because the relationship is a composition relationship, when the Company instance is removed/destroyed, the Department instance is automatically removed/destroyed as well.

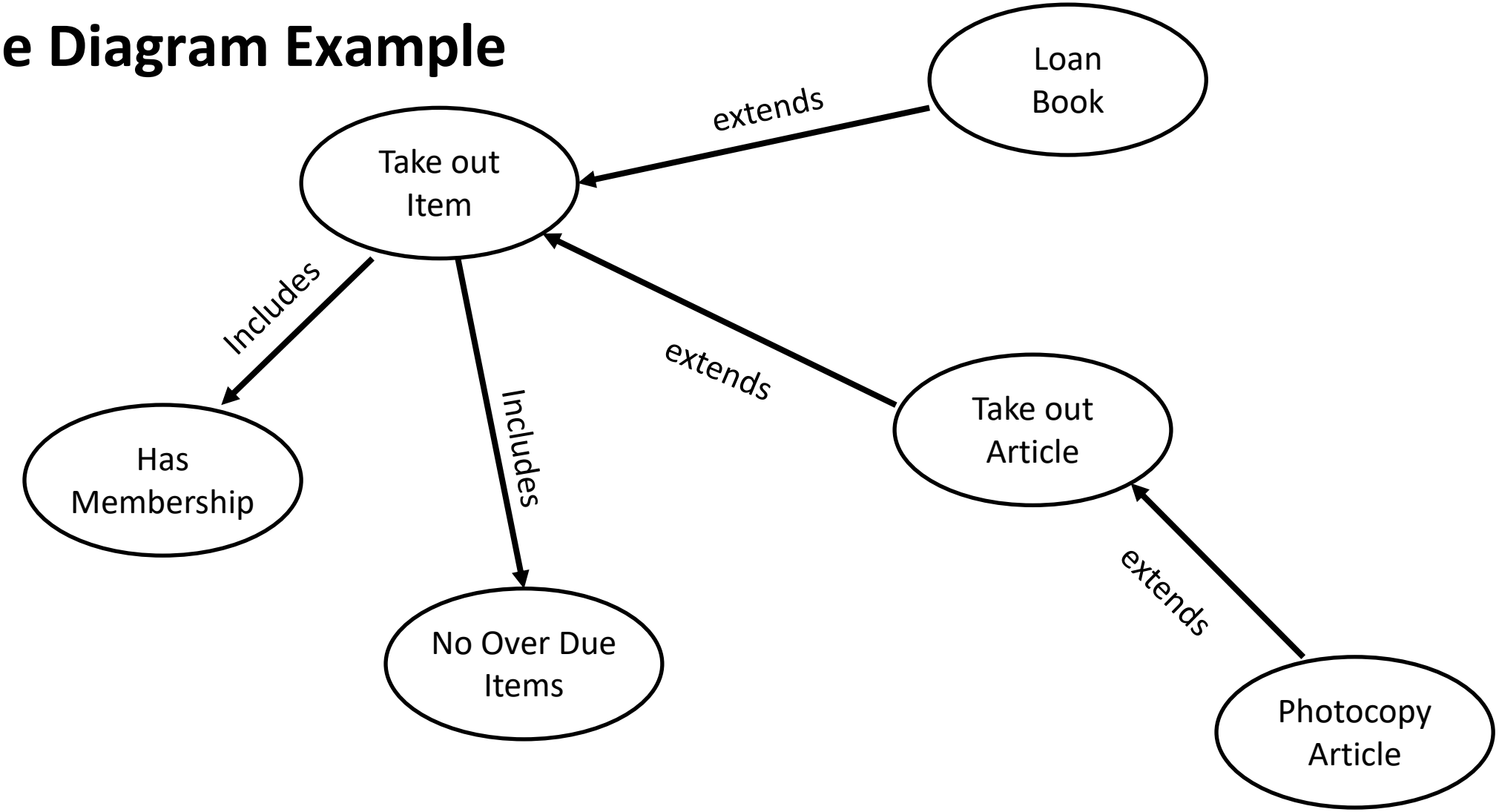


A Domain Model Example

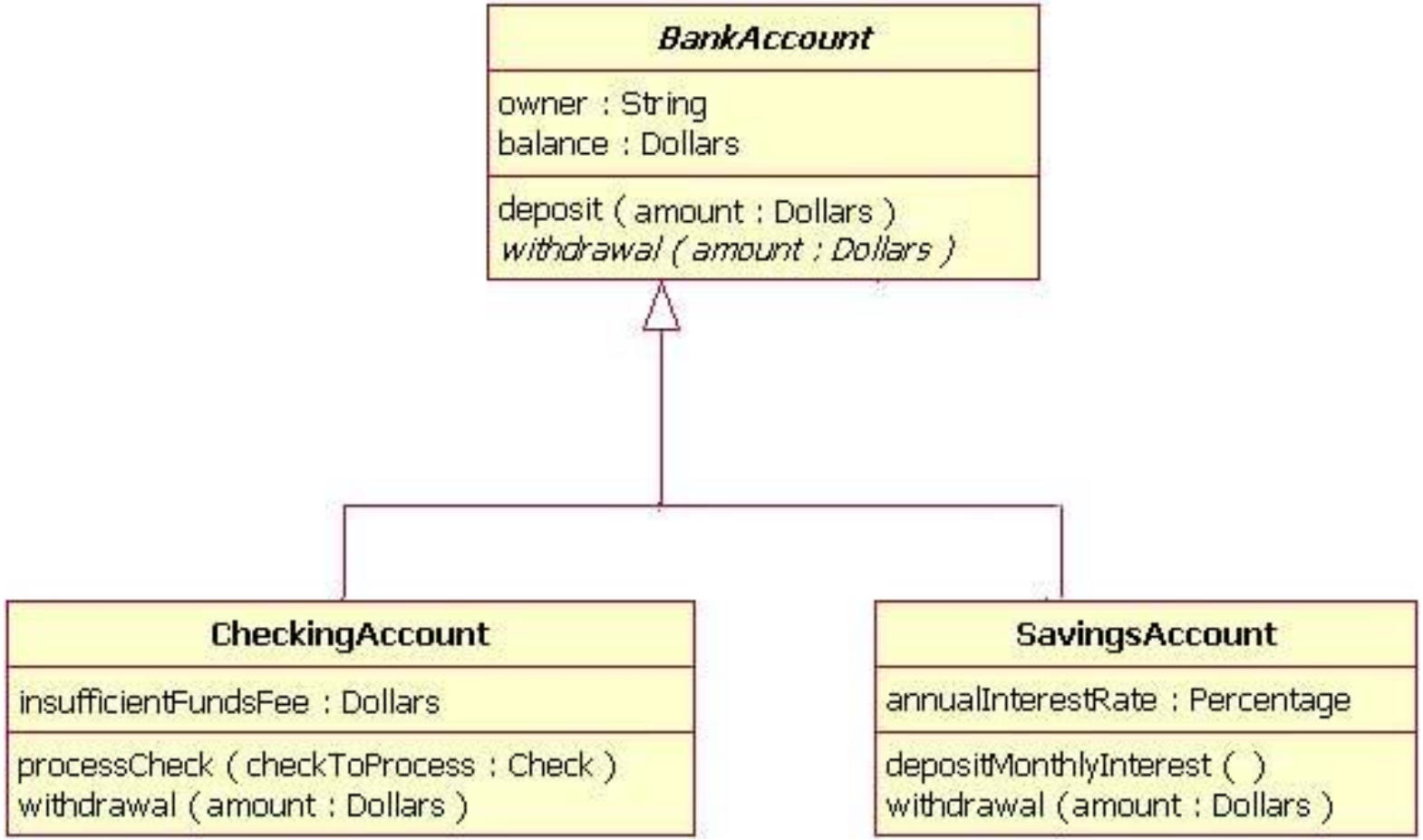
Multiplicity values and their indicators

Indicator	Meaning
0..1	Zero or one
1	One only
0..*	Zero or more
*	Zero or more
1..*	One or more
3	Three only
0..5	Zero to Five
5..15	Five to Fifteen

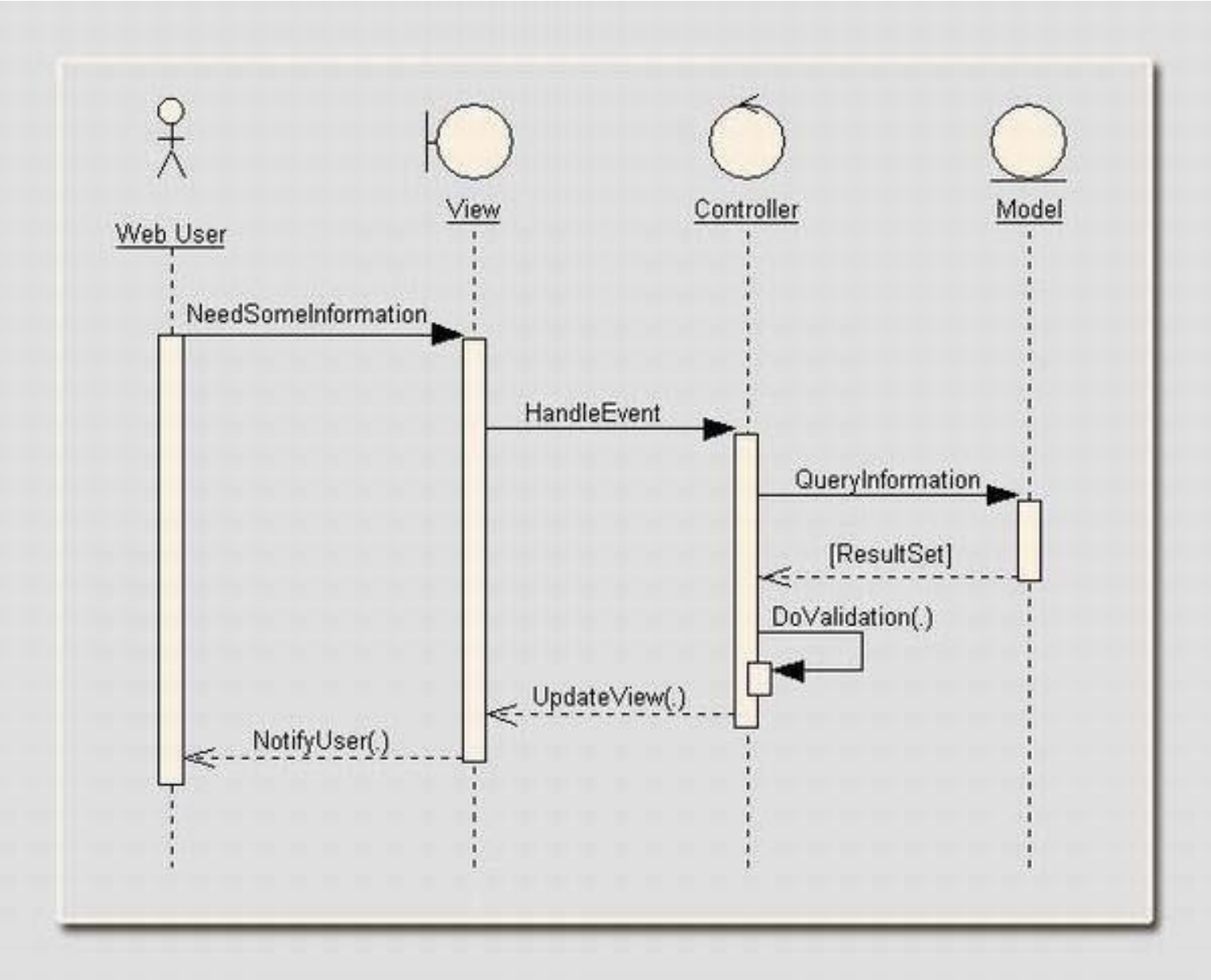
Use Case Diagram Example



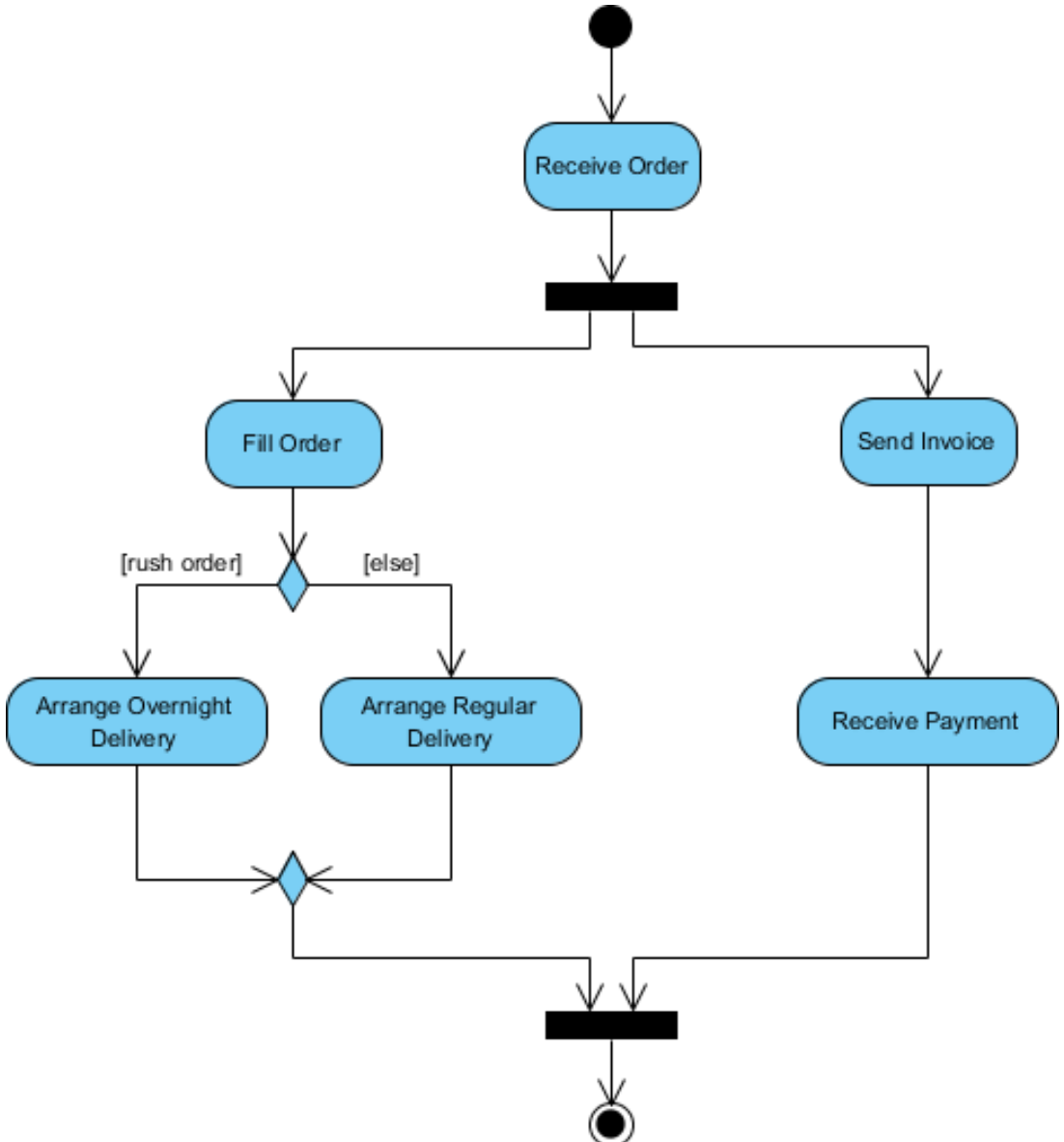
UML Example Class Diagram



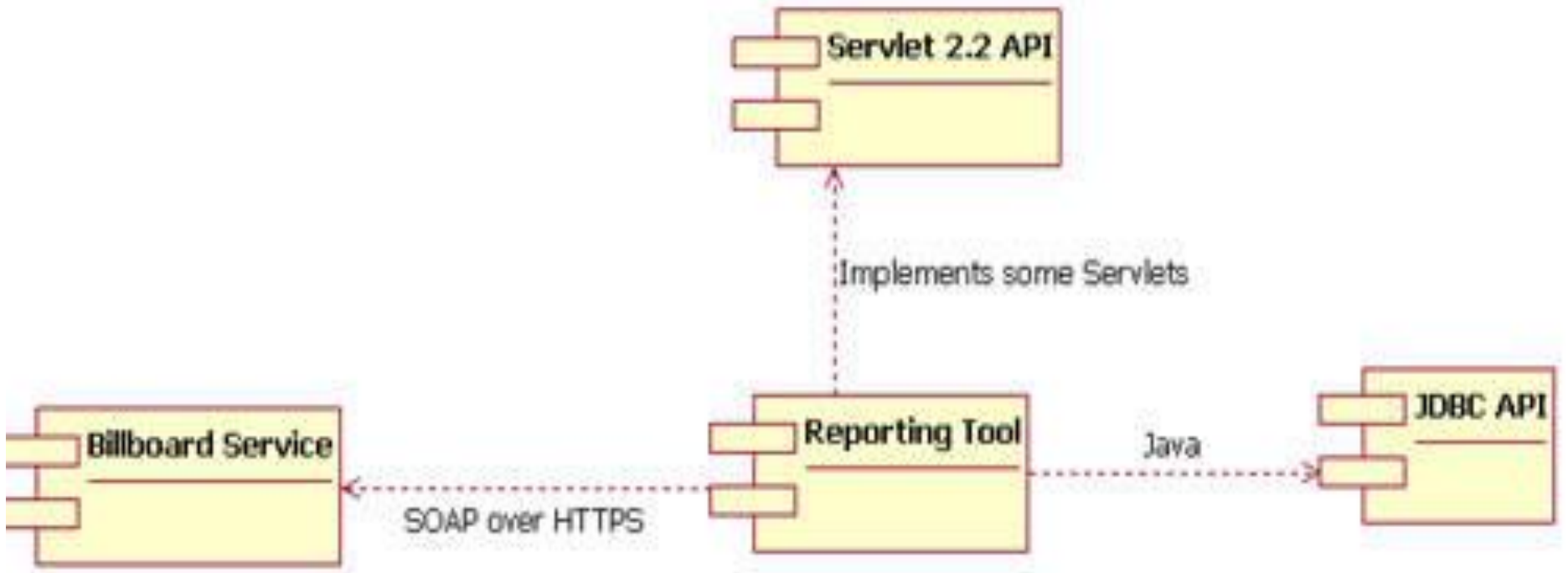
UML Sequence Diagram



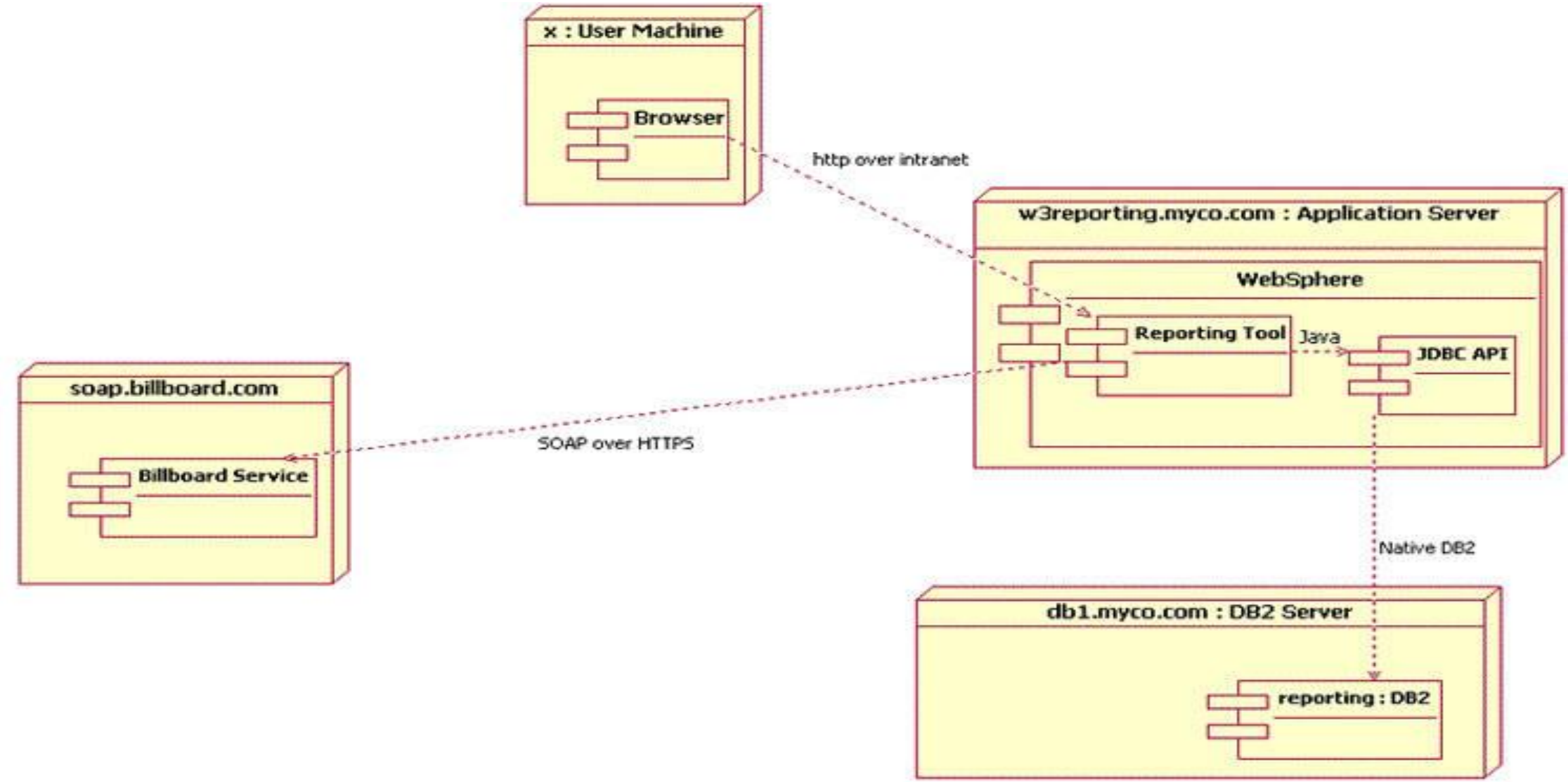
UML Activity Diagram



UML Component Diagram



UML Deployment Diagram



UML Tools

- Astah professional (available in PC labs)
  - http://astah.net/
  - http://astah.net/student-license-request
- Rational Rose
  - www-03.ibm.com/software/products/en/rosemod
- Visual Paradigm
  - www.visual-paradigm.com
- Tools (to an extent) ensure correctness of diagrams
  - Avoid doing UML in applications like Visio - use a tool that can check your diagrams such as Astah
- **Pen and paper is still the best place to start your analysis and design**

AstahUML Tool

Quick Access Menu
Frequently-used-commands provided as icons

Tool Bar
A palette to provide a set of models per diagram. Choose a model you want to create, and click on the diagram canvas.

Structure View
Edit models in a tree view. Alias set-up, search & replace are also available

Map
Overview of currently-open diagram. Easier to access area you want in canvas

Properties
Enter details of each model

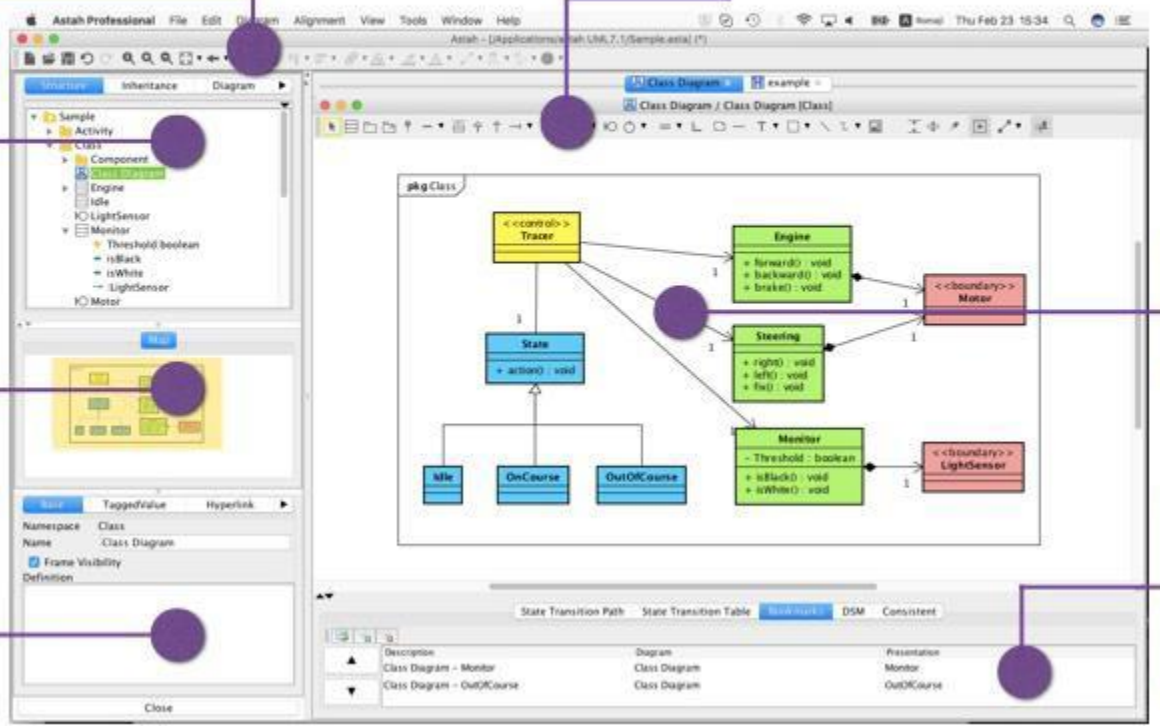


Diagram Editor
Canvas where you draw diagrams.

Extended View
Additional view to analyze models added by Plug-ins. This pane can be hidden.