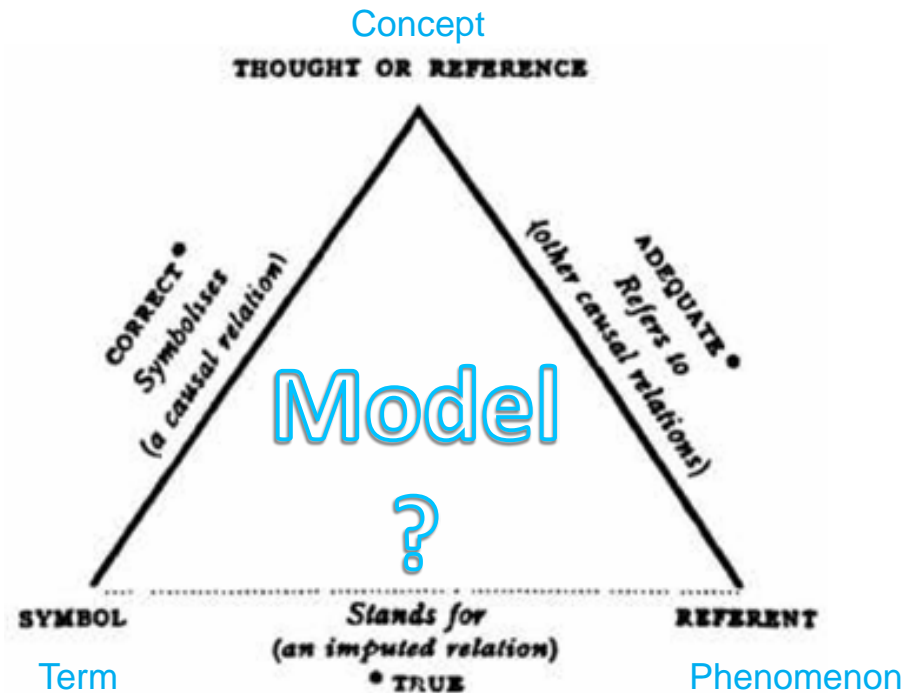# Modeling I

Class diagrams

Ketil Stølen

Partly based on slides prepared by Prof. Øystein Haugen, HiØ & SINTEF

What's a Model?

Exercise: Explain *class* in the setting of the previous slide

- In which corner does *class* belong?

- What would you put in the two other corners?

Artefacts in Informatics

Abstraction

Models   **Frameworks**   **Patterns**   **Algorithms**

Languages

Programming   **GPL**   **DSL**   **Formal**   **Visual ....**

Tools

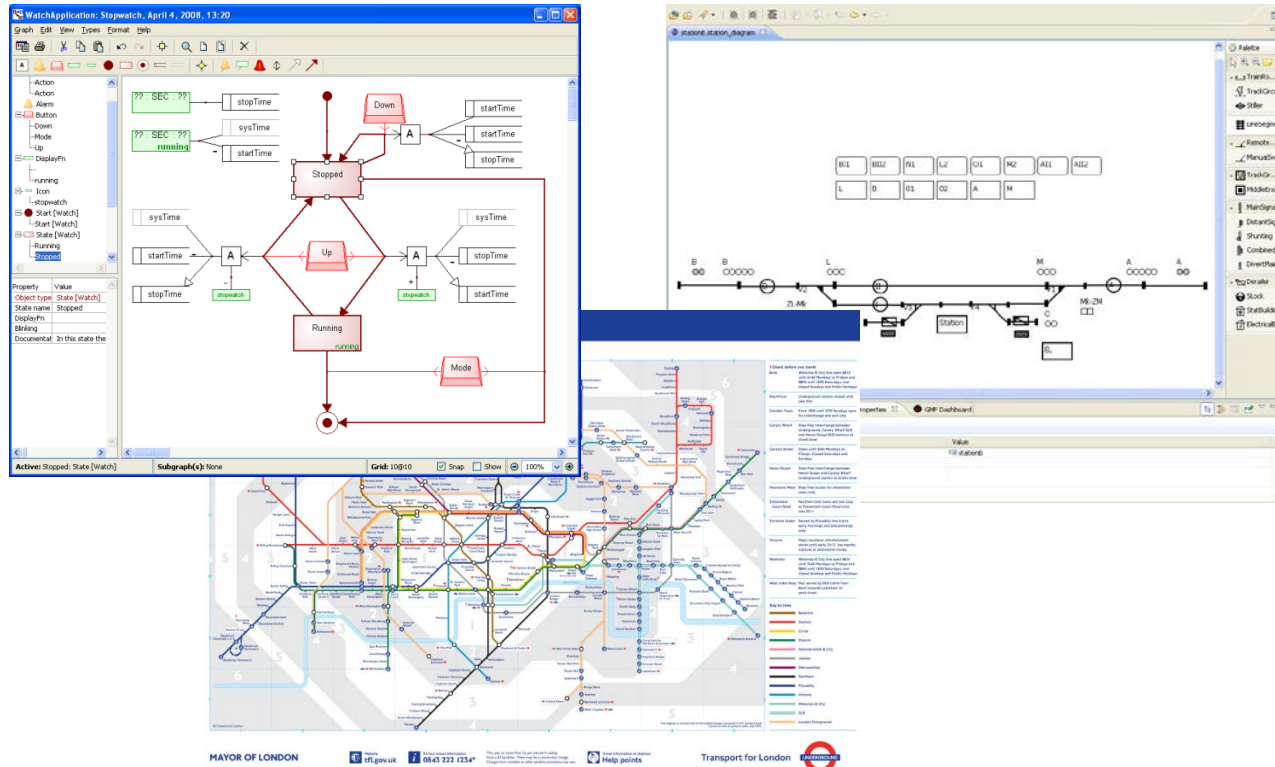**Editors**   **Compilers**   **Verifiers**   **Simulators**   **Apps ....**

What language(s) to use?

> Must have good mechanisms for abstraction

> Must have adequate tooling

> Must scale to "real systems"

Why make a language?

UML Class modelling

- Concepts
- Inheritance
- Generation
- Meta
- Aggregate

SINTEF

Concepts

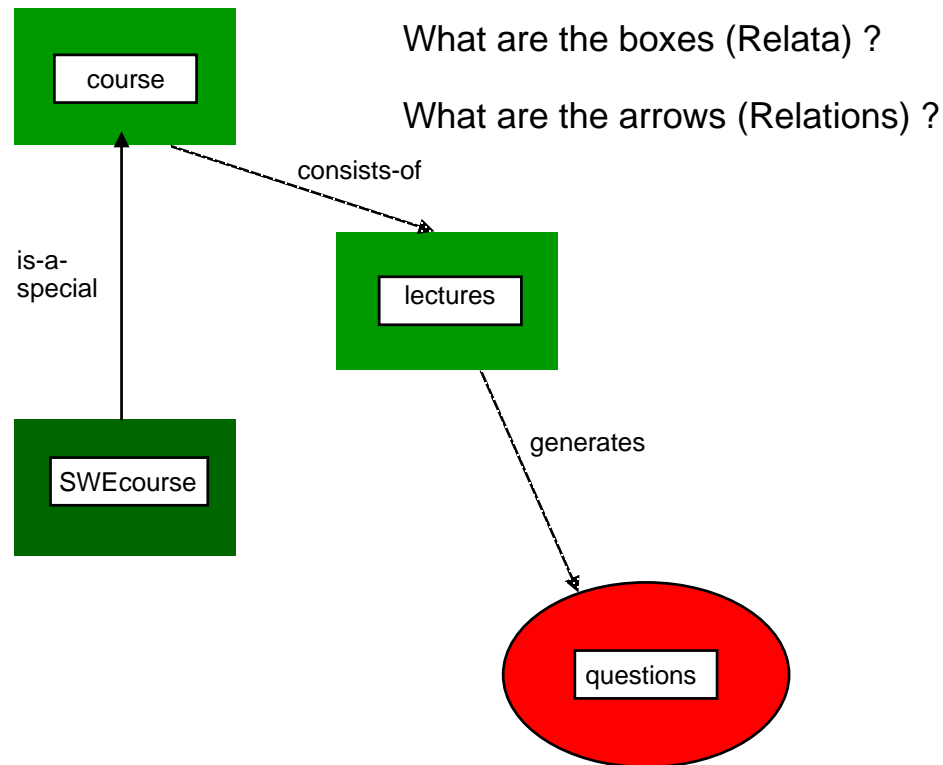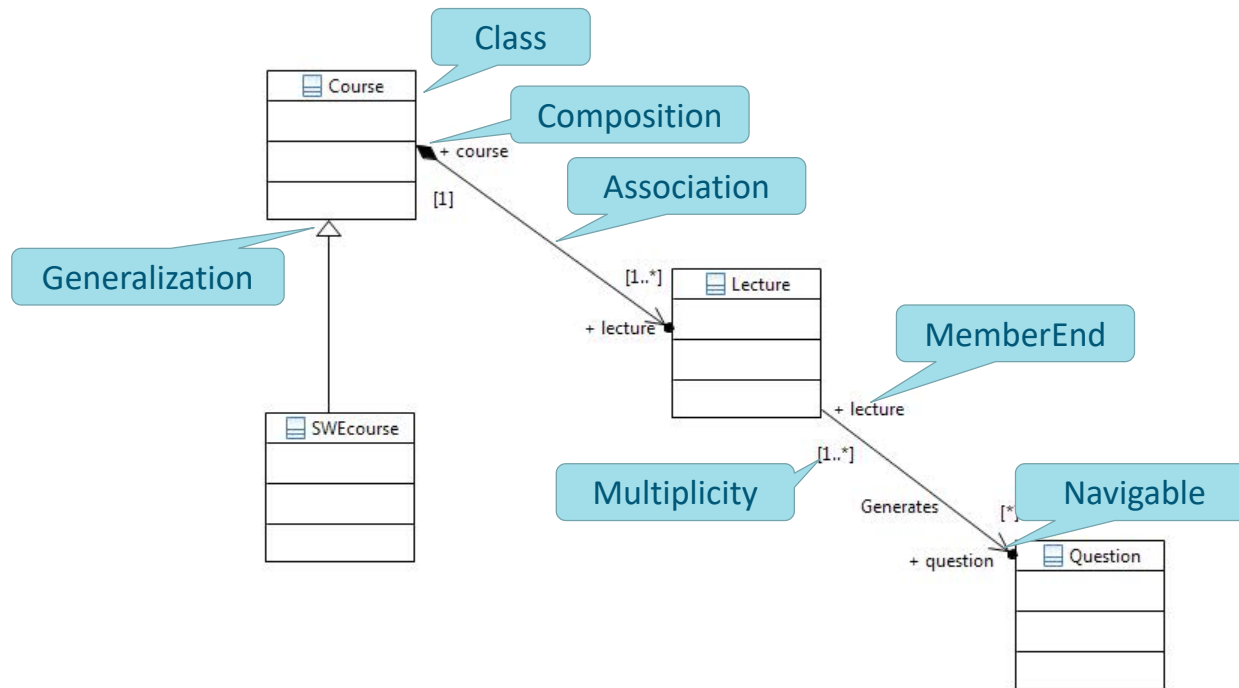|		
Class		Object
Type		Instance
Pattern		Entity
Method		Method call
Function		Function call
Datatype		Variable
	Prototype	
	Clone	

A small story about Courses

> The Software Engineering Course is a special Course

> Courses contain Lectures

> The lectures may generate questions
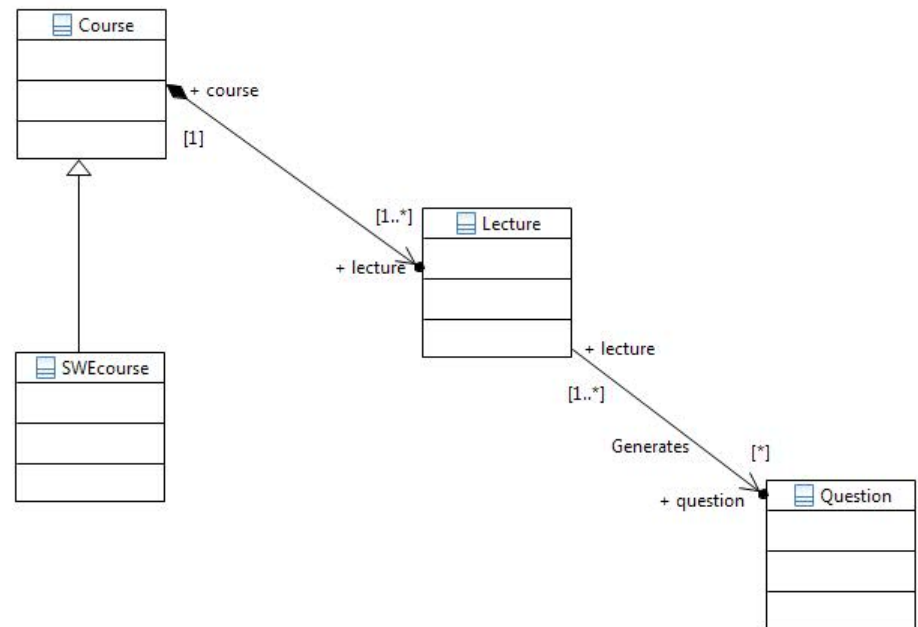
A small Story with Boxes and Arrows

course

What are the boxes (Relata) ?

What are the arrows (Relations) ?

consists-of

is-a-special

lectures

SWEcourse

generates

questions

A small Story with UML class diagram

Exercise:

- Can Software Engineering course (SWEcourse) be held without lectures?

- Can there be lectures without questions asked?

- Can the very same lecture be given in two different courses?

- Can the very same question be posed to several lectures?

- If a course is cancelled, will all remaining lectures also be cancelled? (or "terminated")

Identity

Identity modifiers:	Languages:
Generalization	UML
Subclass	Simula
Derived Classes	C++
Extension	Java
Interface	UML, Java
Parameters	FORTRAN, Pascal, Algol, ...
Overloading	C++, Java
Redefined operations	UML
Virtual procedures	Simula, Smalltalk
Virtual functions	C++
Overriding methods	Java
Pointers to functions	C, C++

Subclassing or Inheritance
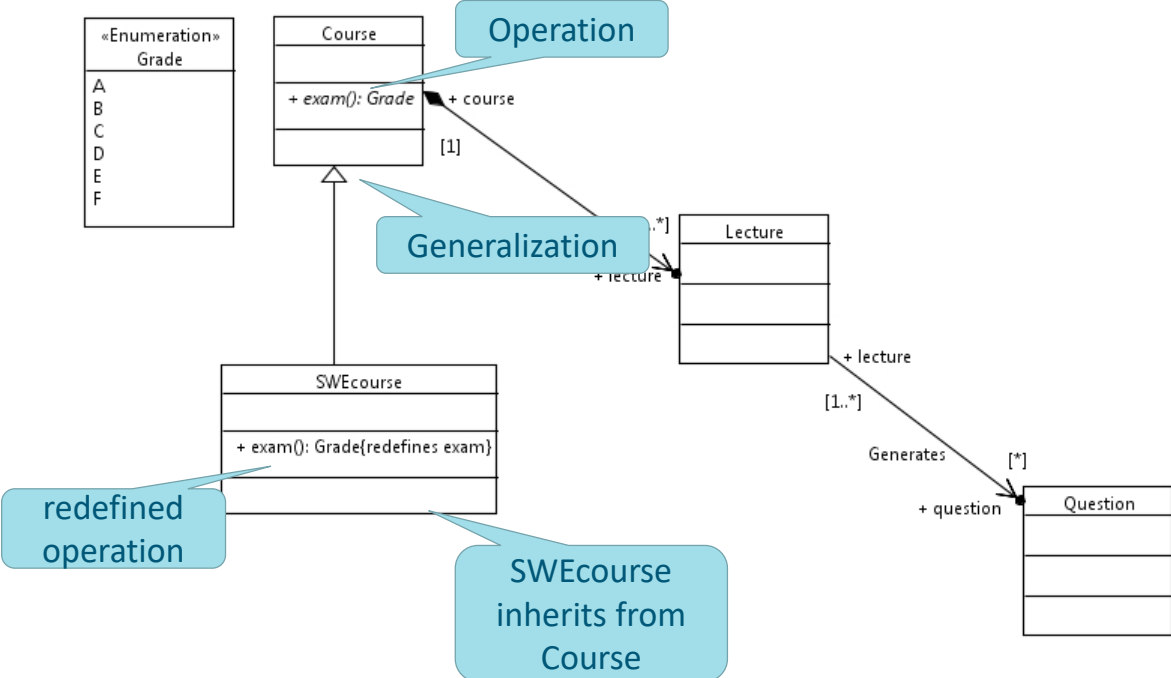
Generation

Aggregation

Meta

Metamodel

Defines

Modeling Language

describes

Model

defines

Program Execution

Modeling Environment

- Editor
- Compiler
- Interpreter
- Debugger
- Model database
- Simulator
- Validator
- Text generation

Change Observe

Data Structure

SINTEF
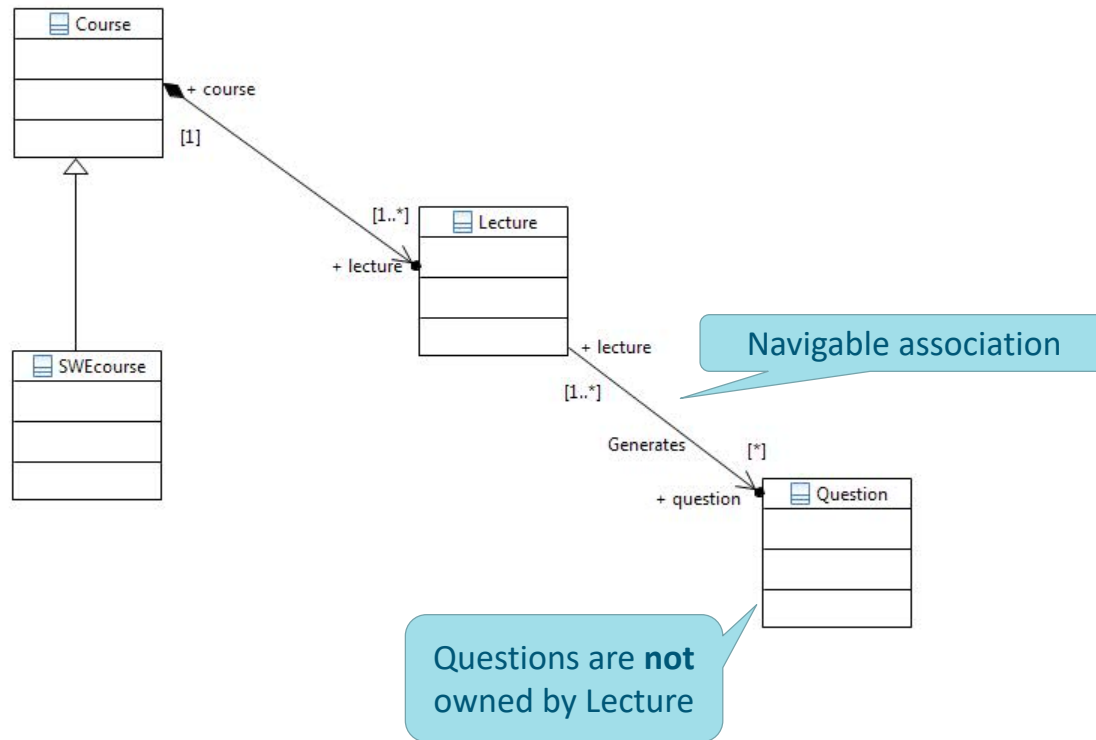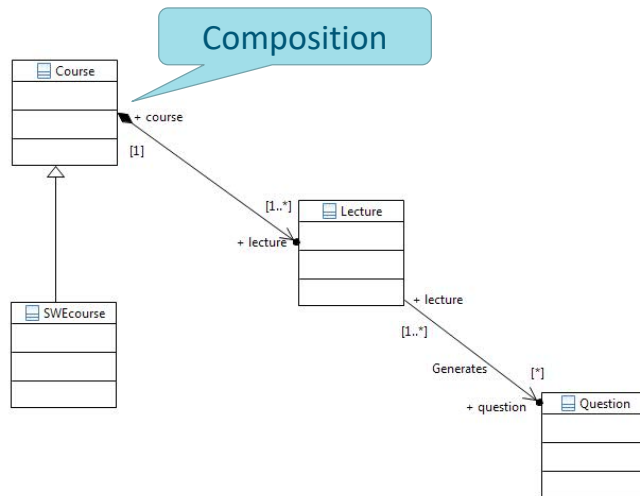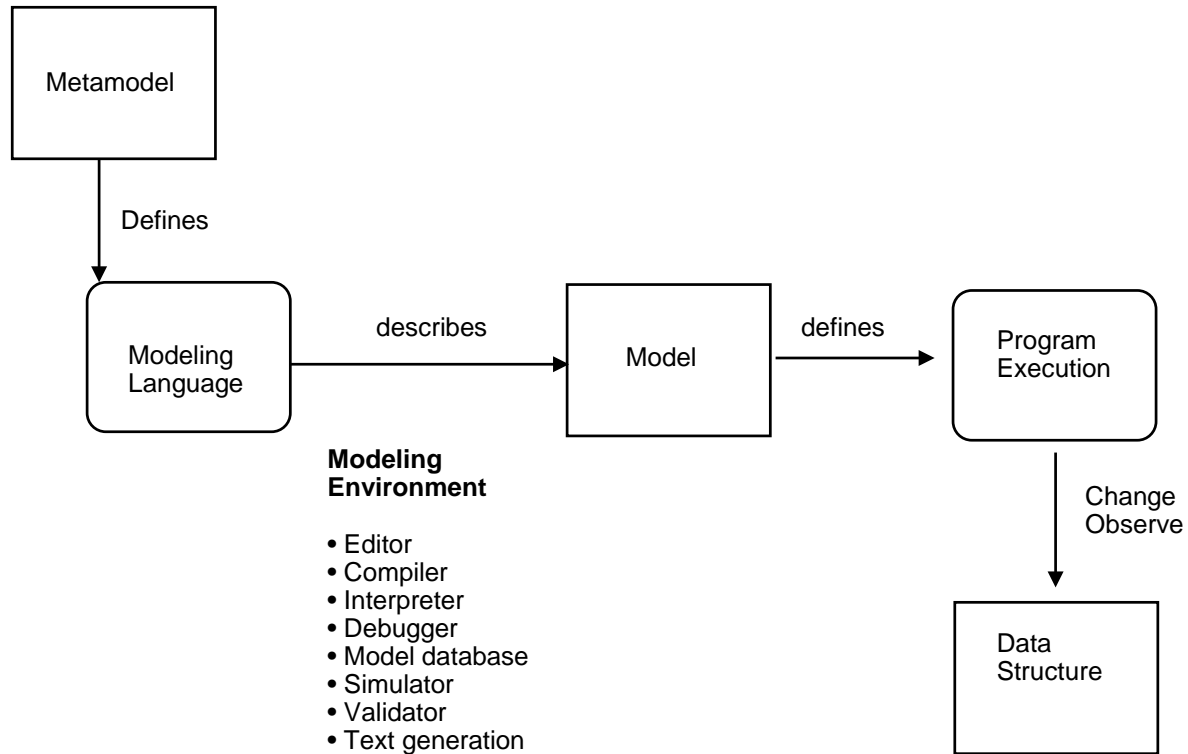
Exercise: Explain the previous slide wrt the language *English*

- What is the meta-model?

- What is the modeling-language?

- What is a model?
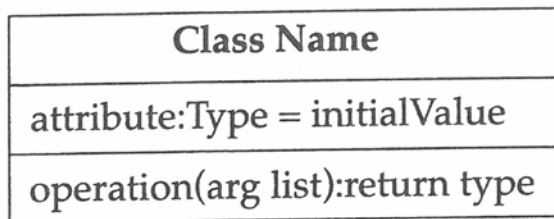
- What is program execution?
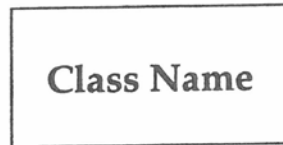
- What is the data structure?

The 4-level meta hierarchy

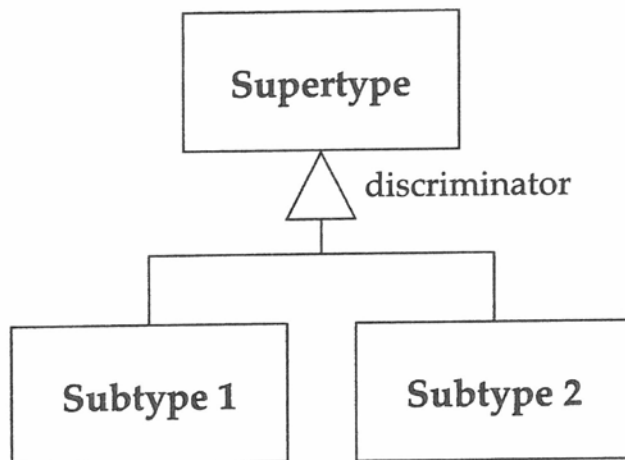| Lev	UML model	Language	Programming	Language
M3	MOF metamodel	MOF	Grammar of BNF	BNF?
M2	UML metamodel	MOF	Grammar of Java	BNF
M1	UML user model	UML	Java user program	Java
M0	Execution of user model		Execution of java program	

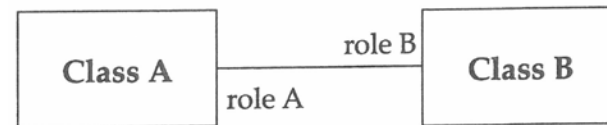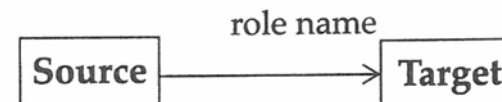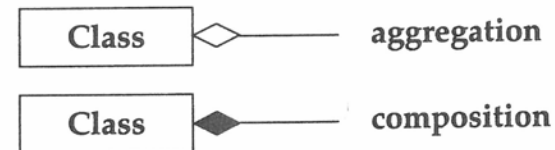# A piece of the UML Metamodel
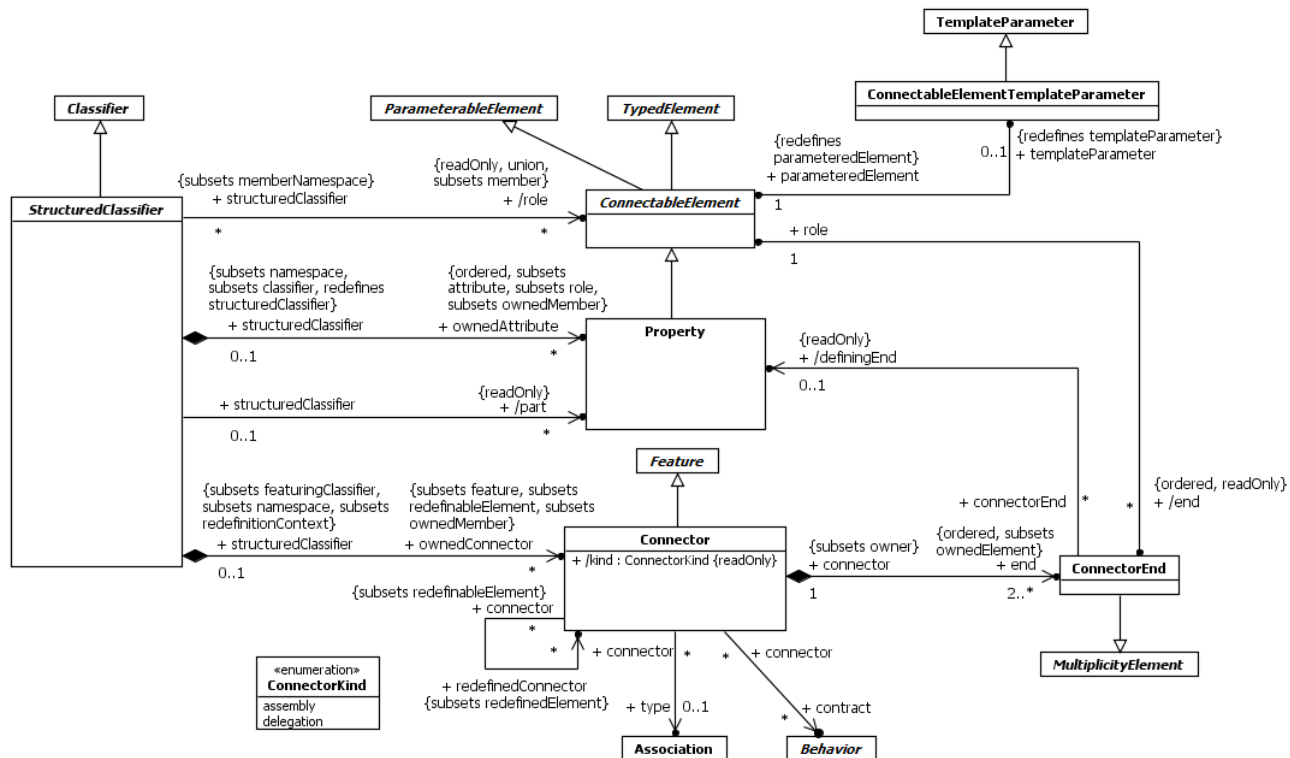
Class Diagram Summary
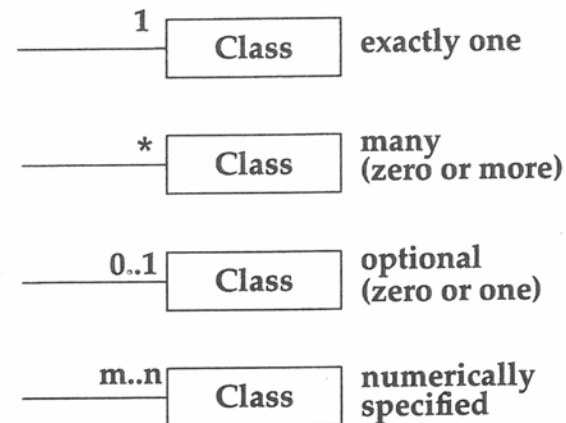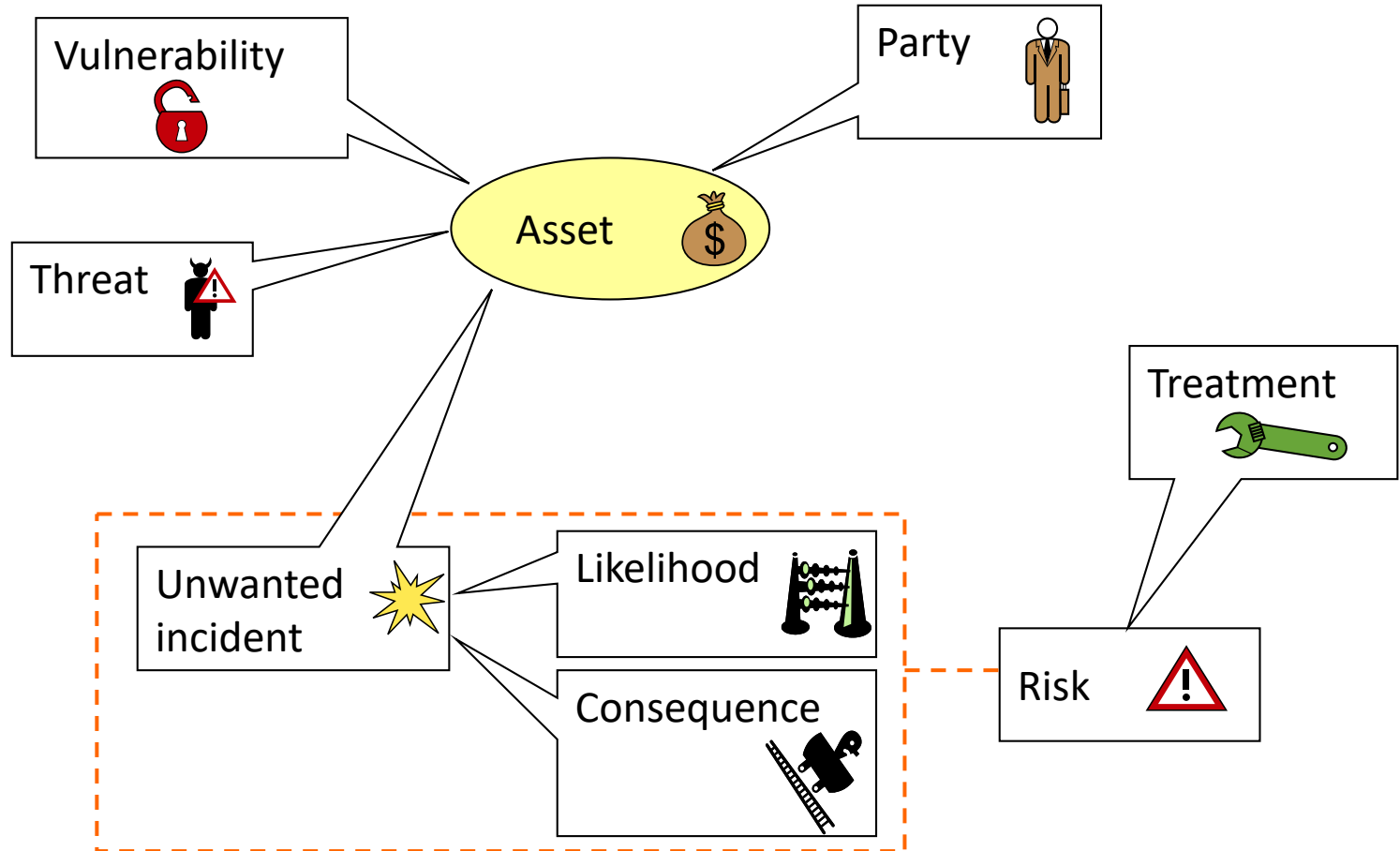
Exercise: Represent the drawing below in UML

Modeling tool used for the UML part of this course

You may use the tool of your preference

Some alternatives:

- https://www.eclipse.org/papyrus/ (powerful but involves a lot to install and use)
- https //www.draw.io app (light weight)