Modelling II

Class diagrams

Ketil Stølen

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

Overview of lecture

- What is modelling?
- Why do we model?
- Modelling languages
- UML – Unified Modelling Language
- Classes and objects
- UML class modelling

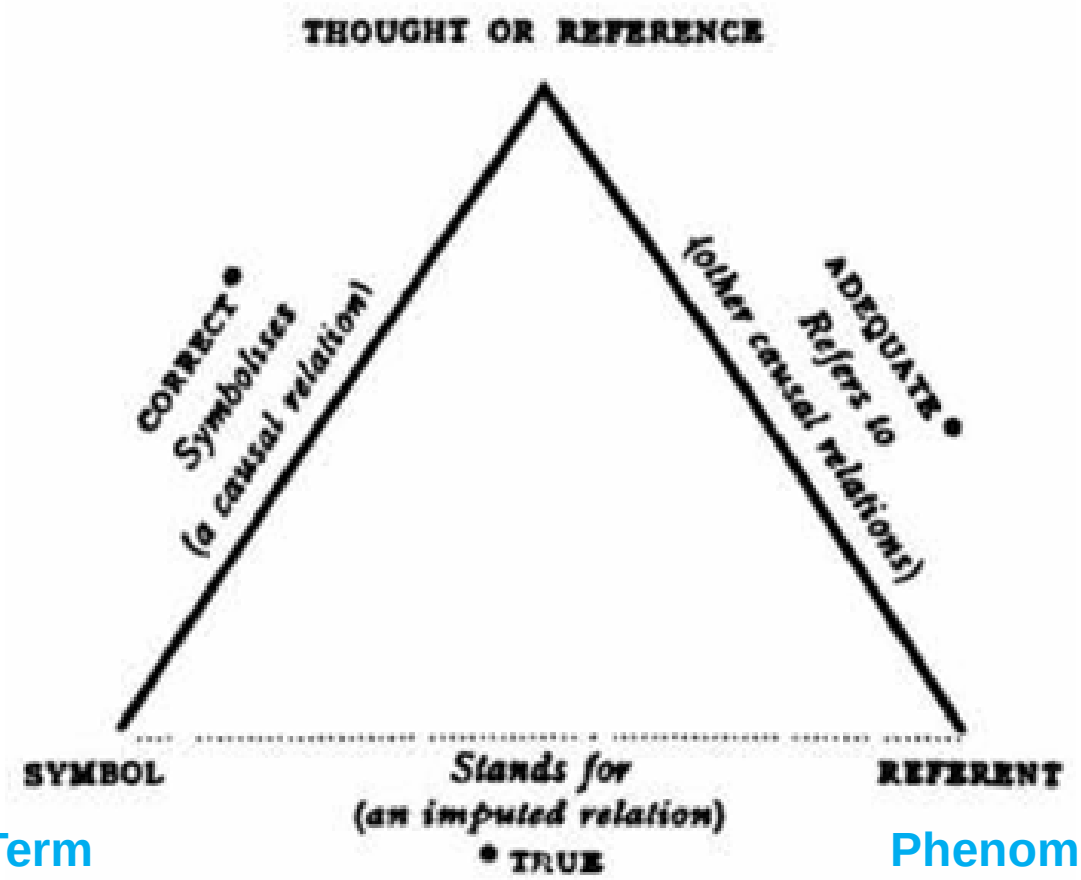
What is modelling?

A model is a representation of an idea, an object, a process or a system that is used to describe some phenomena

- modelling involves abstraction
- abstraction means leaving out information about the phenomena that is not relevant for the aspects we want to study

Terms, concepts and phenomenons

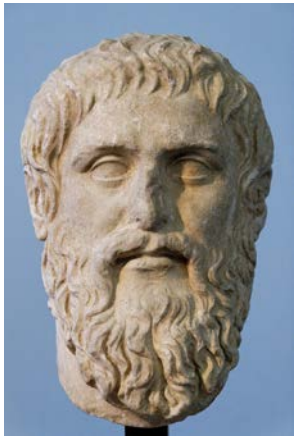
Concept



Term

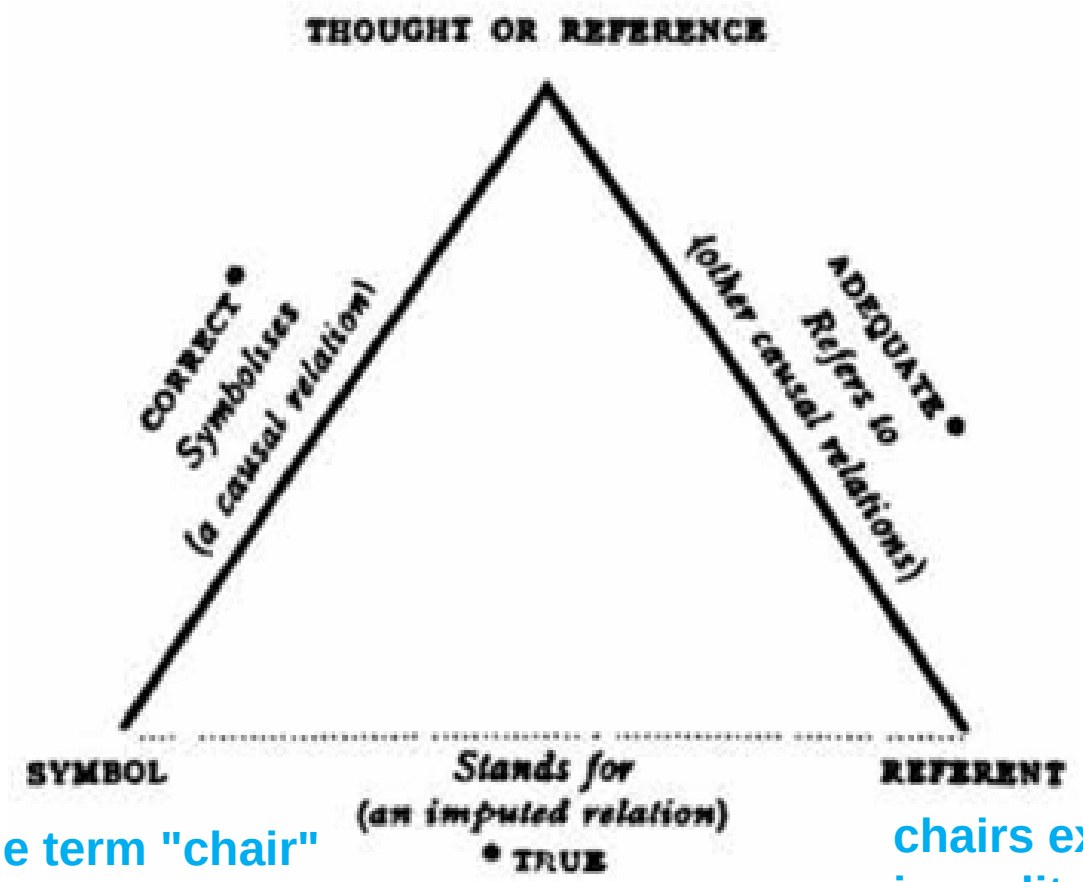
Phenomenon

The ideas of Plato



https://en.wikipedia.org/wiki/Plato

idea of chair

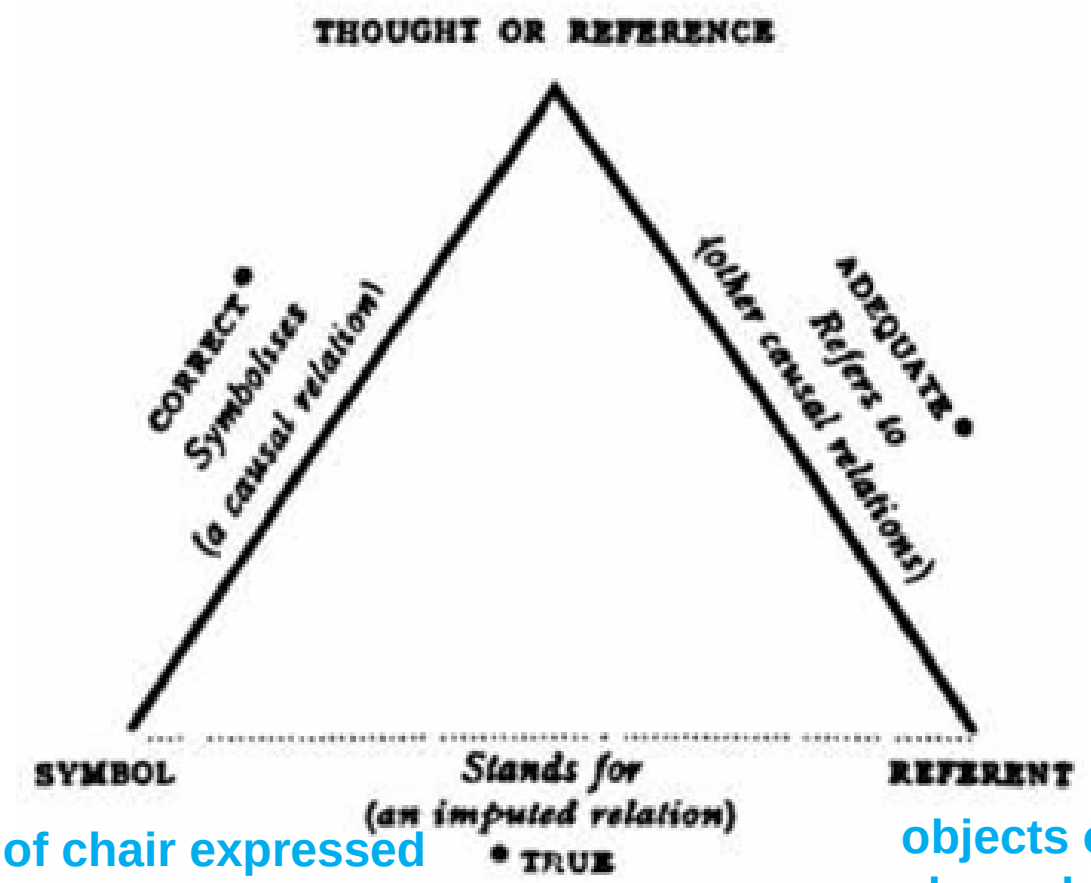


e.g. the term "chair" in English

chairs existing in reality

Computer science exemplification

class of chair

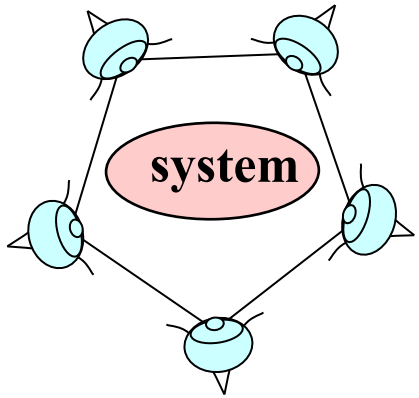


class of chair expressed in some OO-language

objects of class chair

Why do we model?

Systems of today are large and complex



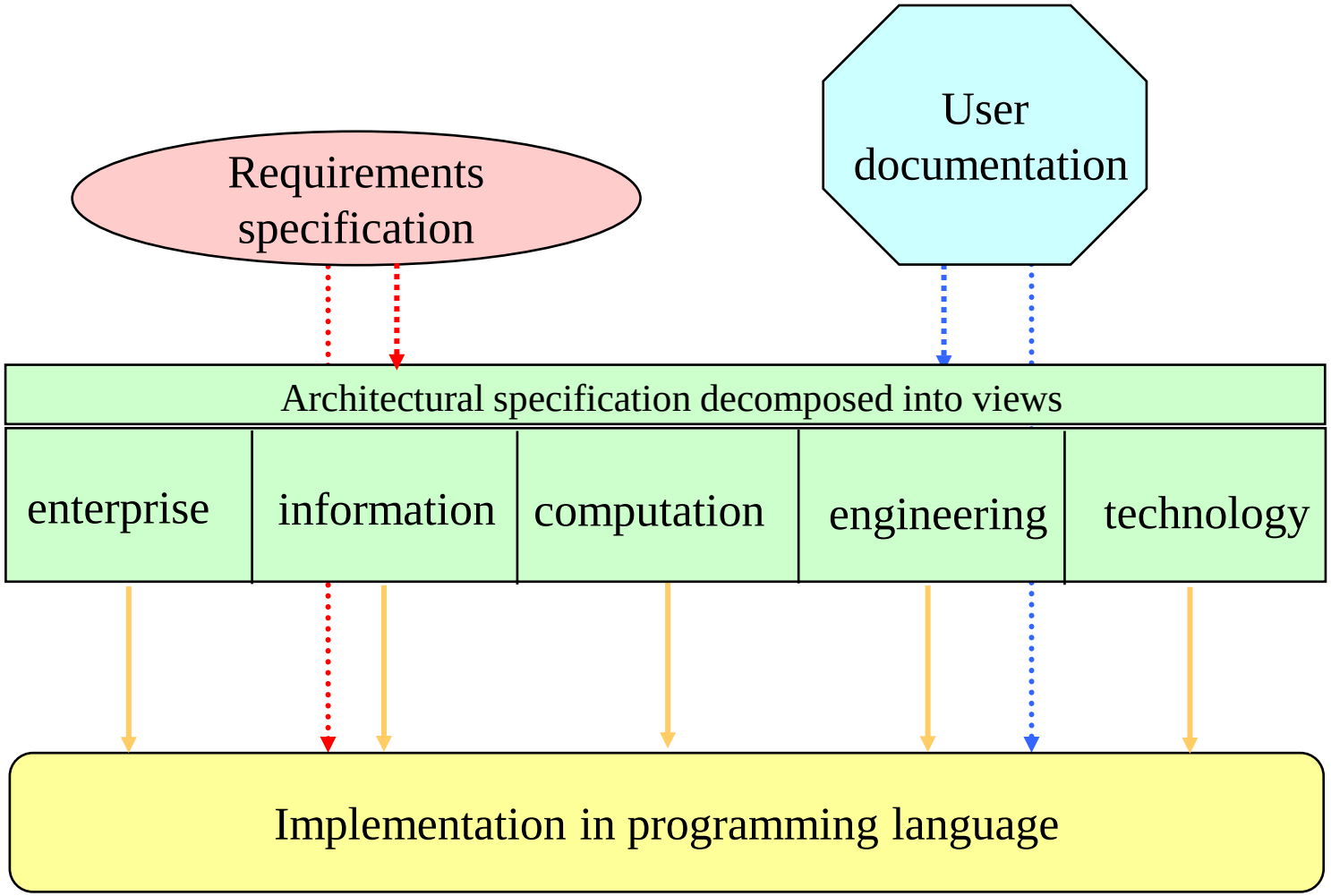
Abstraction is a necessary means to

- explain what the system does
- explain how the system is built
- distinguish the essentials from the inessential
- decompose large and complicated aspects into small more easily understandable entities
- extract specialized system views

Examples of abstraction

- Models: system drawings, designs, specifications
- Frameworks: generic functionality that can be selectively changed by additional user-written code
- Patterns: generic descriptions providing solutions to often occurring problems
- Algorithms: high-level description of computational procedures

Various models



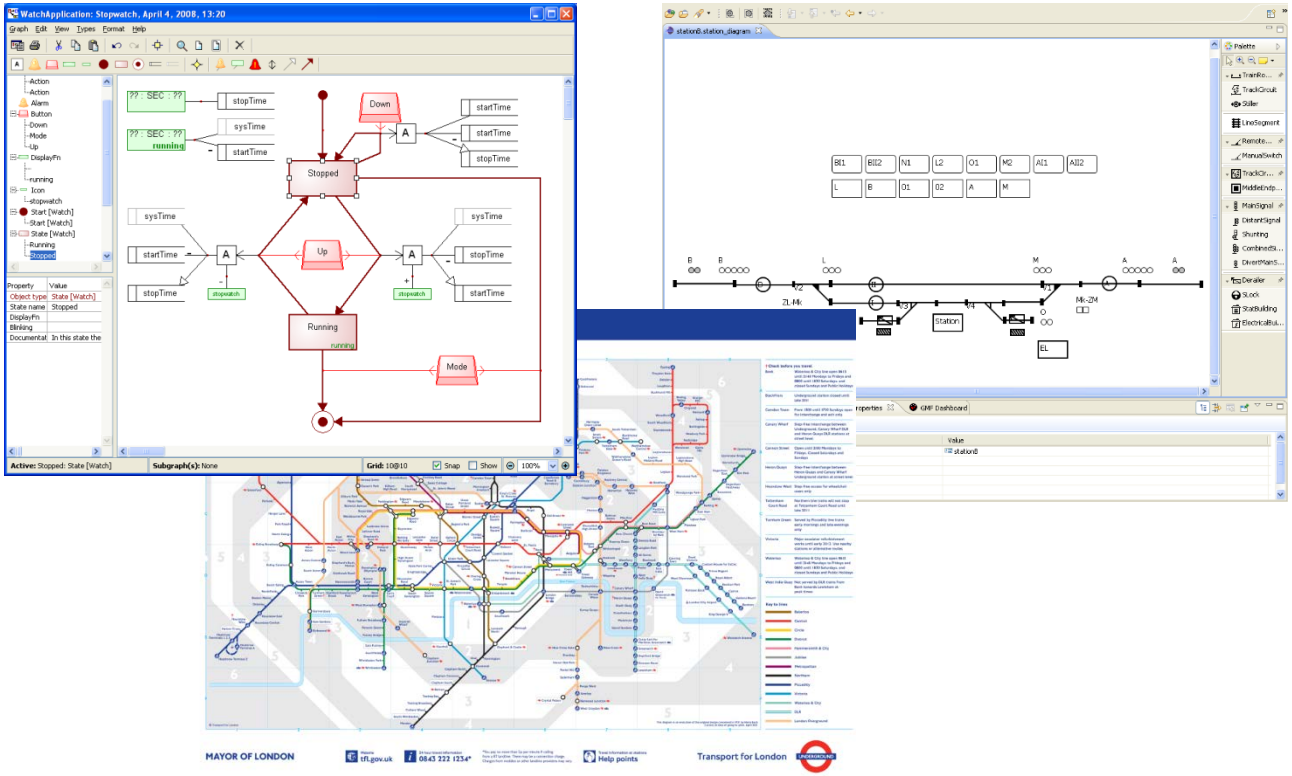
Modelling is a prerequisite for

- Analysis
- Testing
- Validation
- Security risk assessment
- Certification
- Verification

Modelling languages

- Programming languages
- GPL – General Purpose Languages
- DSL – Domain Specific Languages
- Mathematical languges
- Graphical languages

Why make all these languages?



Three main concepts of language theory

Syntax: The relationship between symbols or groups of symbols independent of content, usage and interpretation

Semantics: The rules and conventions that are necessary to interpret and understand the content of language constructs

Pragmatics: The study of the relationship between symbols or groups of symbols and their interpretation and usage

As presented by
https://en.wikipedia.org/wiki/Unified_Modeling_Language

UML – Unified Modelling Language

- The Unified Modeling Language (UML) is a general-purpose, developmental, modeling language in the field of software engineering that is intended to provide a standard way to visualize the design of a system.

- The creation of UML was originally motivated by the desire to standardize the disparate notational systems and approaches to software design. It was developed by Grady Booch, Ivar Jacobson and James Rumbaugh at Rational Software in 1994–1995, with further development led by them through 1996.

- In 1997, UML was adopted as a standard by the Object Management Group (OMG), and has been managed by this organization ever since. In 2005, UML was also published by the International Organization for Standardization (ISO) as an approved ISO standard.

Structural UML diagrams

- Class Diagram – class, features and relationships
- Object Diagram – example configurations of instances
- Component Diagram – structure and connections
- Composite Structure Diagram – runtime decomposition of class
- Deployment Diagram – deployment of artifacts to nodes
- Package Diagram – compile time hierarchic structure

Structural UML diagrams

- **Class Diagram – class, features and relationships**
- Object Diagram – example configurations of instances
- Component Diagram – structure and connections
- **Composite Structure Diagram – runtime decomposition of class**
- Deployment Diagram – deployment of artifacts to nodes
- Package Diagram – compile time hierarchic structure

Behavioral UML diagrams

- Activity Diagram – Work process behaviour
- Use Case Diagram – Human-system interaction
- Interaction Overview Diagram – Dataflow
- Timing Diagram – Timed interactions
- State Machine Diagram – Algorithmic behavior of communicating objects
- Communication Diagram – Interaction between objects, focus on links
- Sequence Diagram – Interaction between objects, focus on sequences

Behavioral UML diagrams

- Activity Diagram – Work process behaviour
- Use Case Diagram – Human-system interaction
- **Interaction Overview Diagram – Flow of interaction**
- Timing Diagram – Timed interactions
- **State Machine Diagram – Algorithmic behavior of communicating objects**
- Communication Diagram – Interaction between objects, focus on links
- **Sequence Diagram – Interaction between objects, focus on sequences**

UML Class diagrams

Agenda

- Concepts
- Example
- Basics
- Inheritance
- Generation
- Composition
- Meta modelling
- Summary

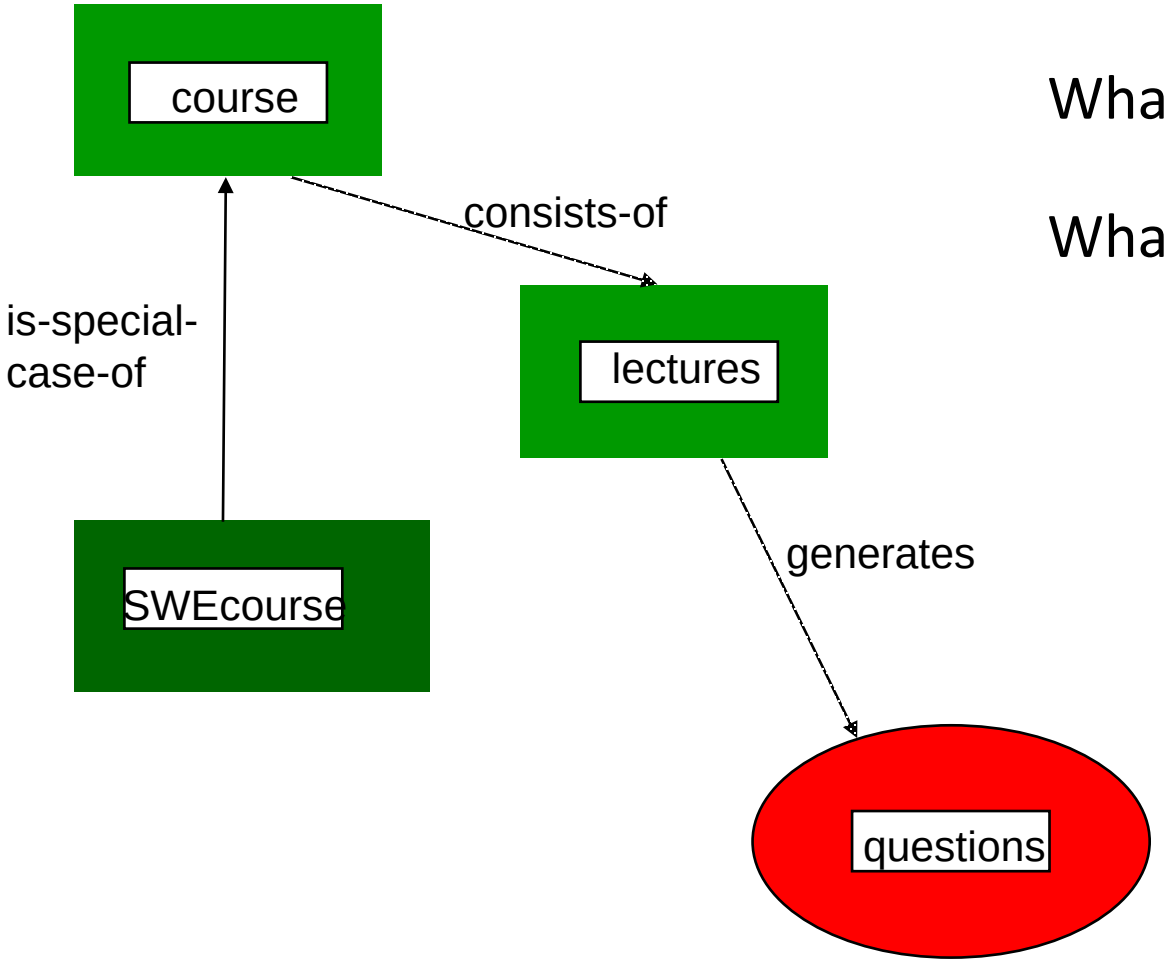
Concepts

Class	Object
Type	Instance
Pattern	Entity
Method	Method call
Function	Function call
Datatype	Variable

A small story about Courses

- The Software Engineering Course is a special Course
- Courses contain Lectures
- The Lectures may generate Questions

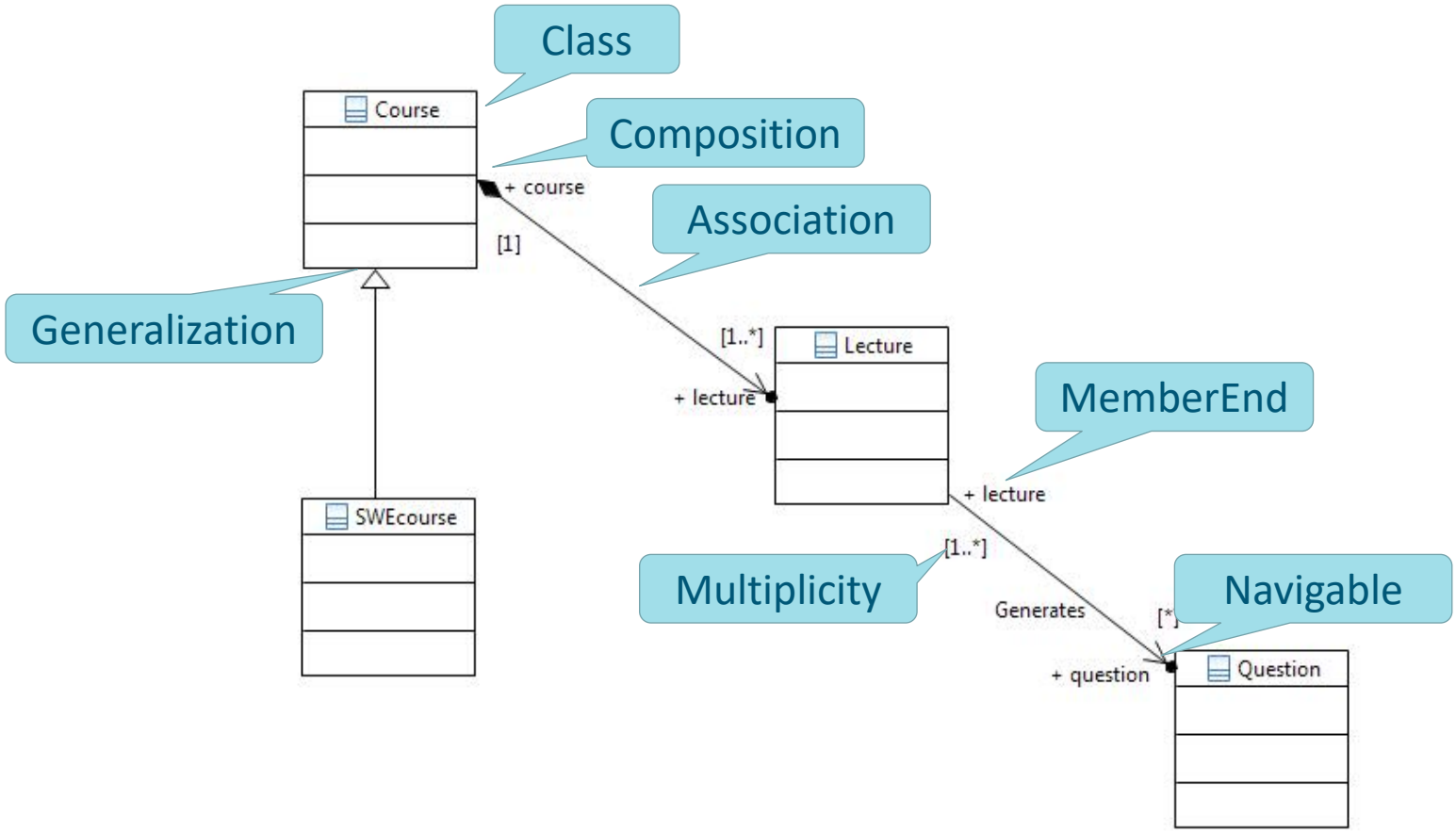
The story with Boxes and Arrows



What are the boxes?

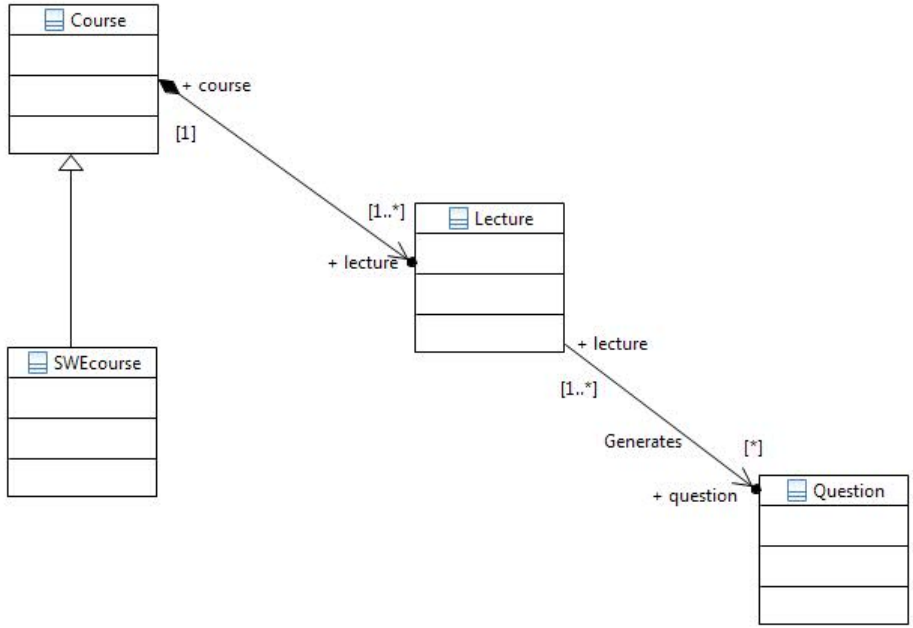
What are the arrows?

A story with UML class diagram

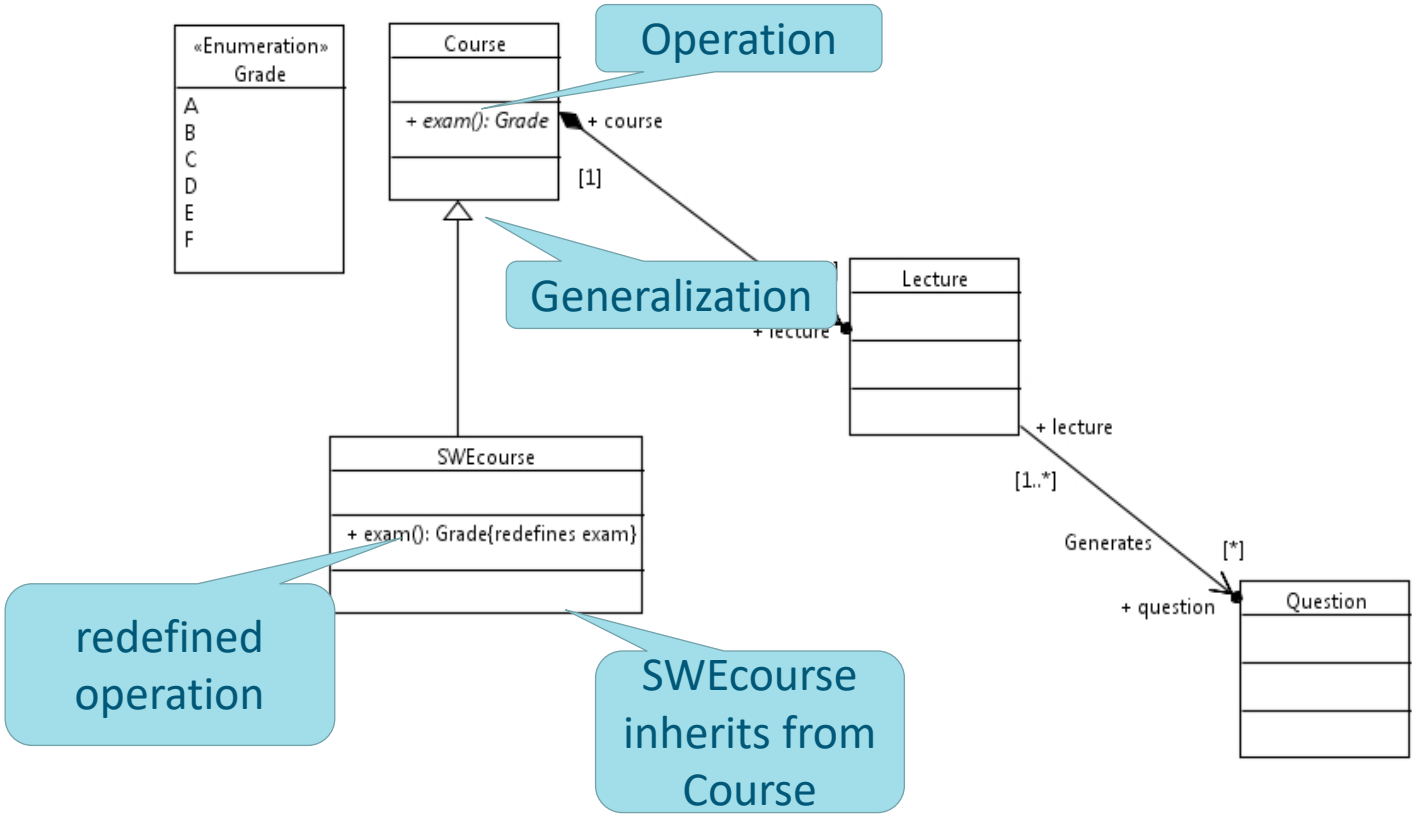


Questions that we may now answer

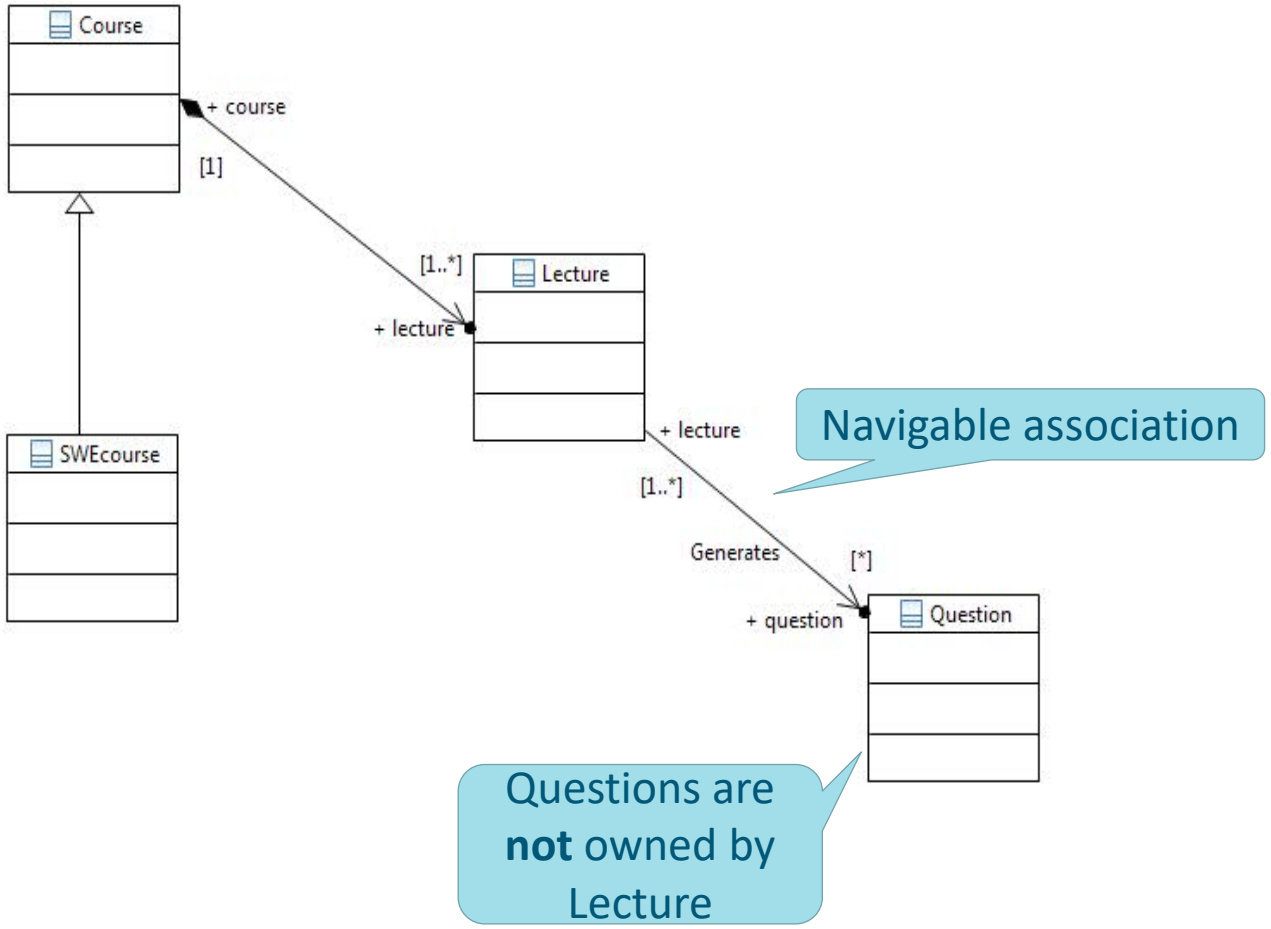
- Can Software Engineering course be held without lectures?
- Can there be lectures without questions asked?
- Can the very same lecture be given in different courses?
- Can the very same question be posed to several lectures?
- If a course is cancelled, will all remaining lectures also be cancelled?



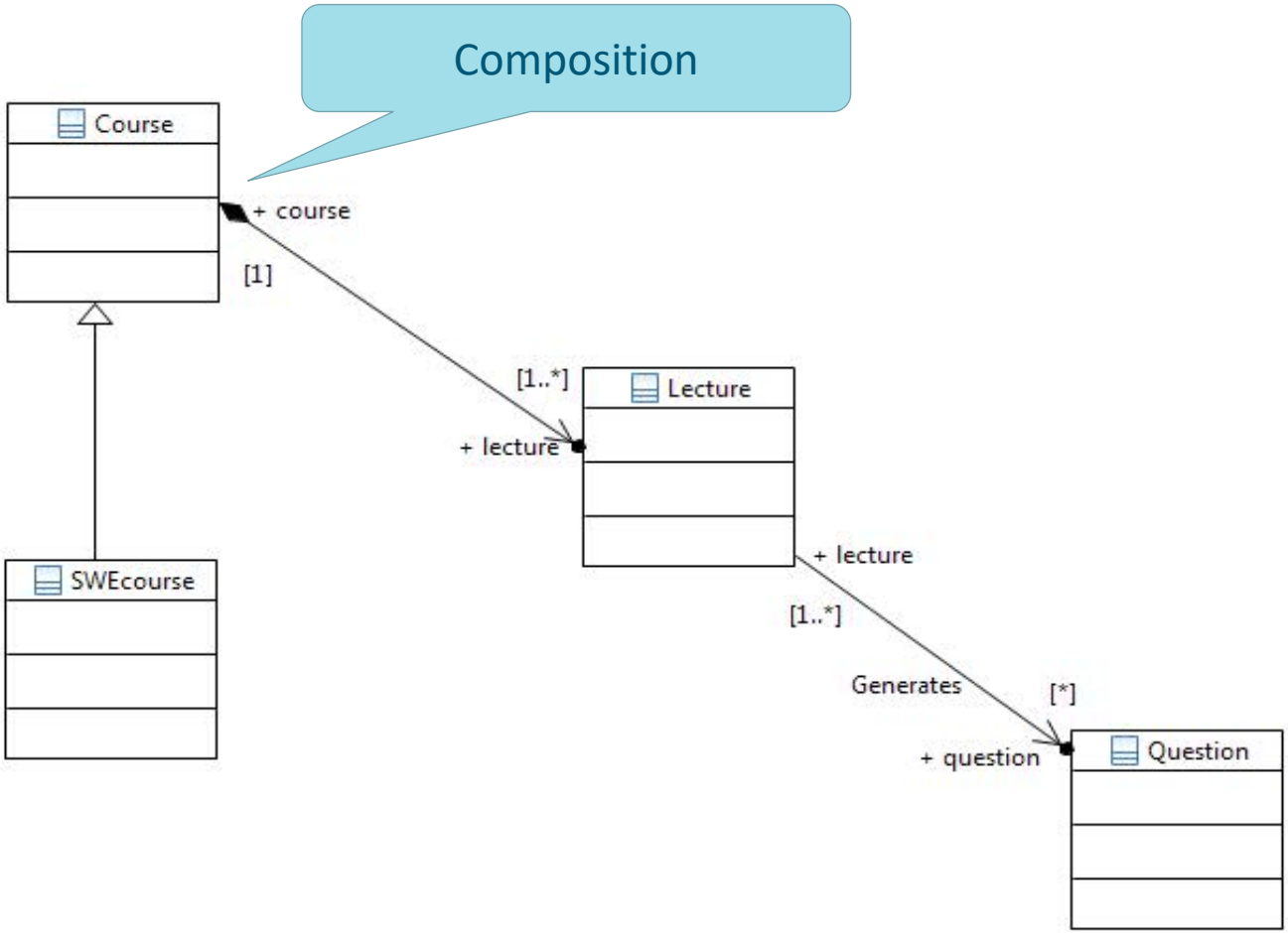
Inheritance



Generation


Course
+ course
[1]
[1..*]
Lecture
+ lecture
SWEcourse
+ lecture
[1..*]
Navigable association
Generates
[*]
+ question
Question
Questions are **not** owned by Lecture

Composition

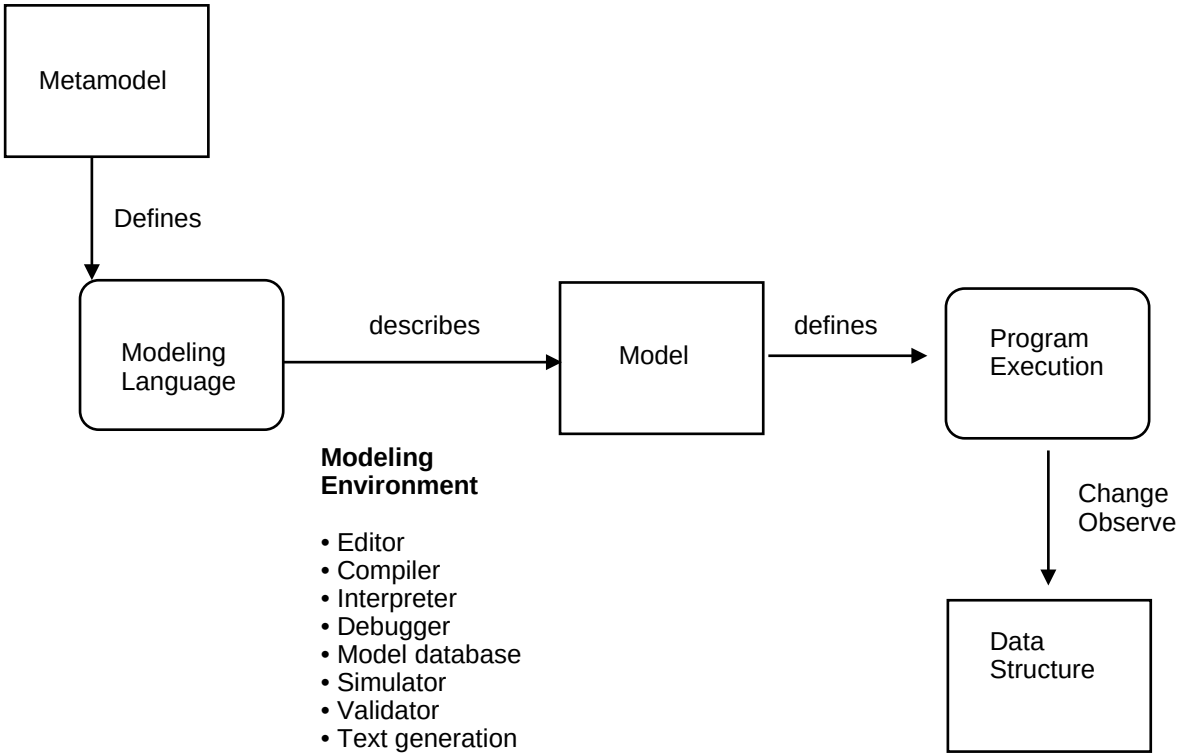


Meta language

- A meta language is a language used to describe another language

- In the UML community the meta language is known as MOF
  - MOF = Meta Object Facility

Meta modelling



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

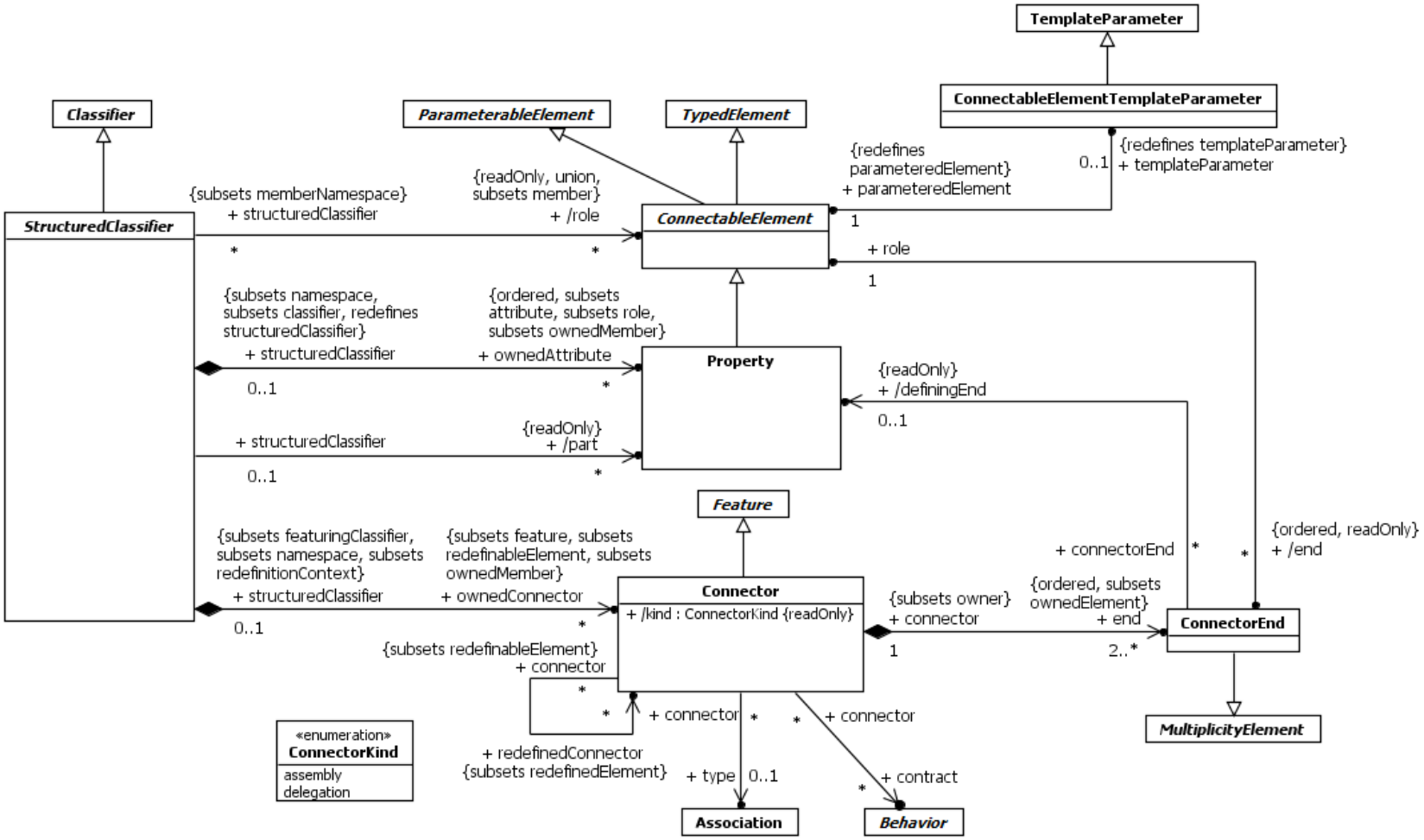
- What is the meta-model?
- What is the modeling-language?
- What is a model?
- What is program execution?
- What is the data structure?

The 4-level meta hierarchy

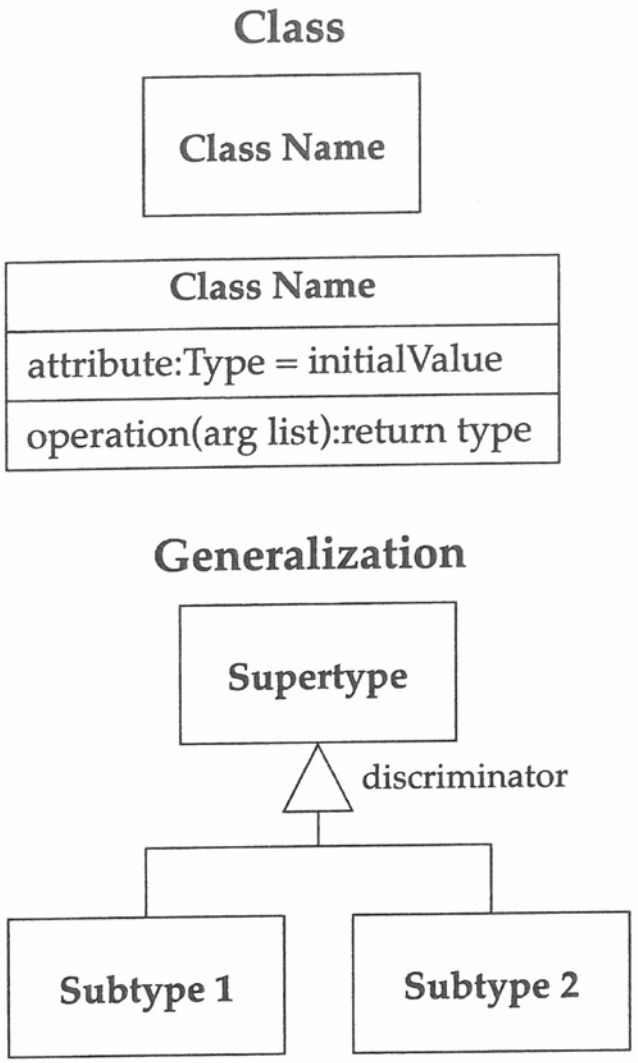
Lev	UML model	Language	Programming	Language
M3	MOF metamodel	MOF	Grammar of BNF	BNF?
M2	UML metamodel	MOF	Grammar of Java	BNF
M1	UML user model	UML	Java user program	Java
M0	Execution of user model		Execution of Java program	

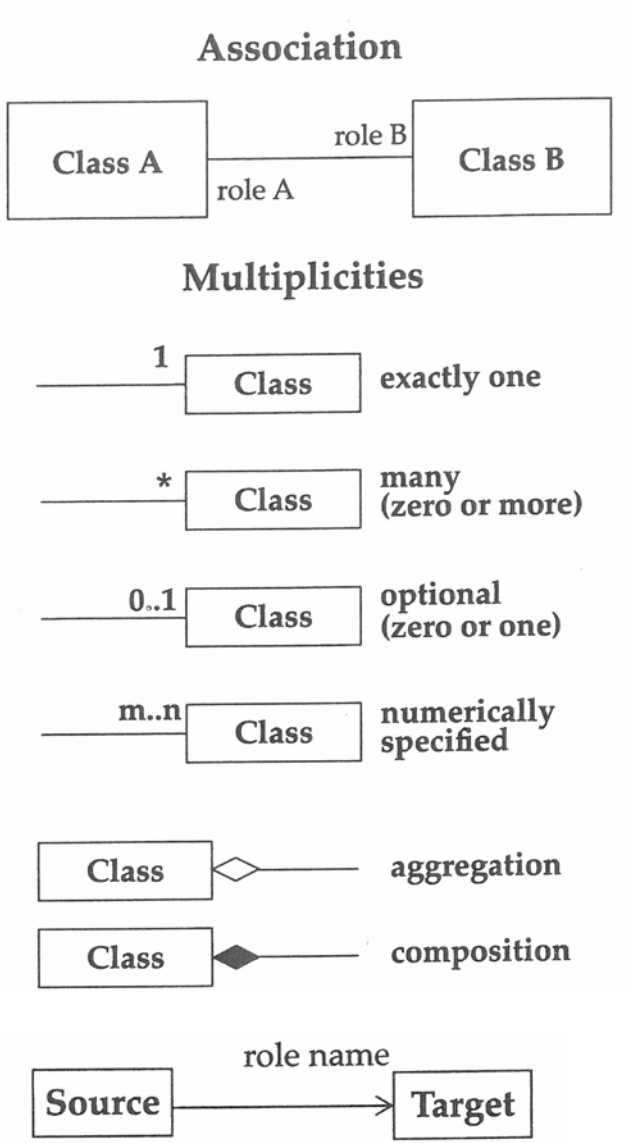
BNF = Baccus-Naur Form (famous notation to define context free grammars)

A piece of the UML metamodel



Class Diagram summary





Modelling 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 instal and use)
- https://www.draw.io (light weight)