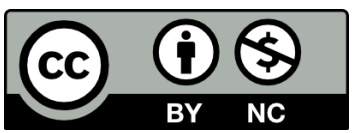
6 MODELLING

course "software requirements and architecture"

Lasy Update: Feb.2024

Created: Sep.2015

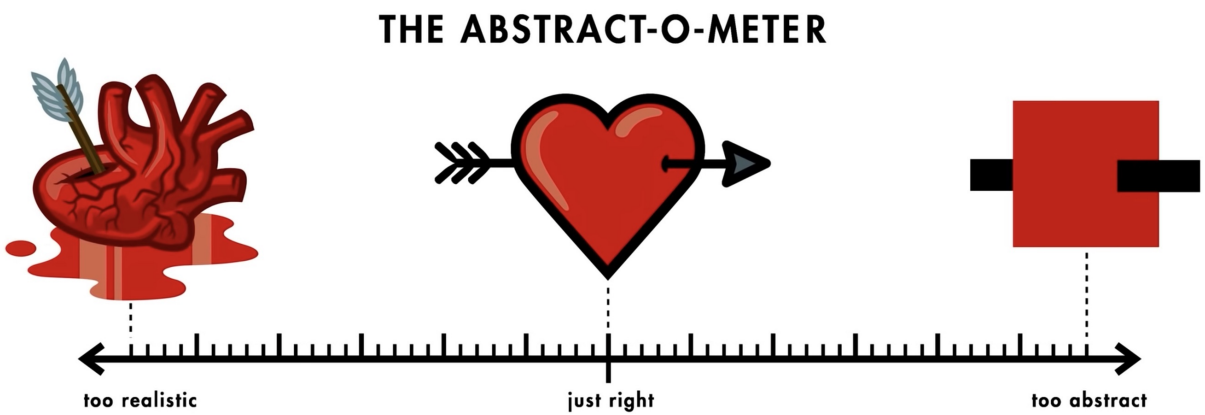


contents

contents

modelling

- Modelling is an essential ingredient in all engineering fields.
- It is a highly creative task.
- Modelling is the process of identifying adequate concepts and selecting abstractions to construct a model that appropriately reflects a given universe of discourse.
- Modelling permits the cost-effective use of the model in place of the real-world object or process for some purpose.
- Modelling is related to abstraction, simplification, and formalisation.
- Abstraction is the idea of getting rid of everything that is not essential to make a point.



purpose of modelling

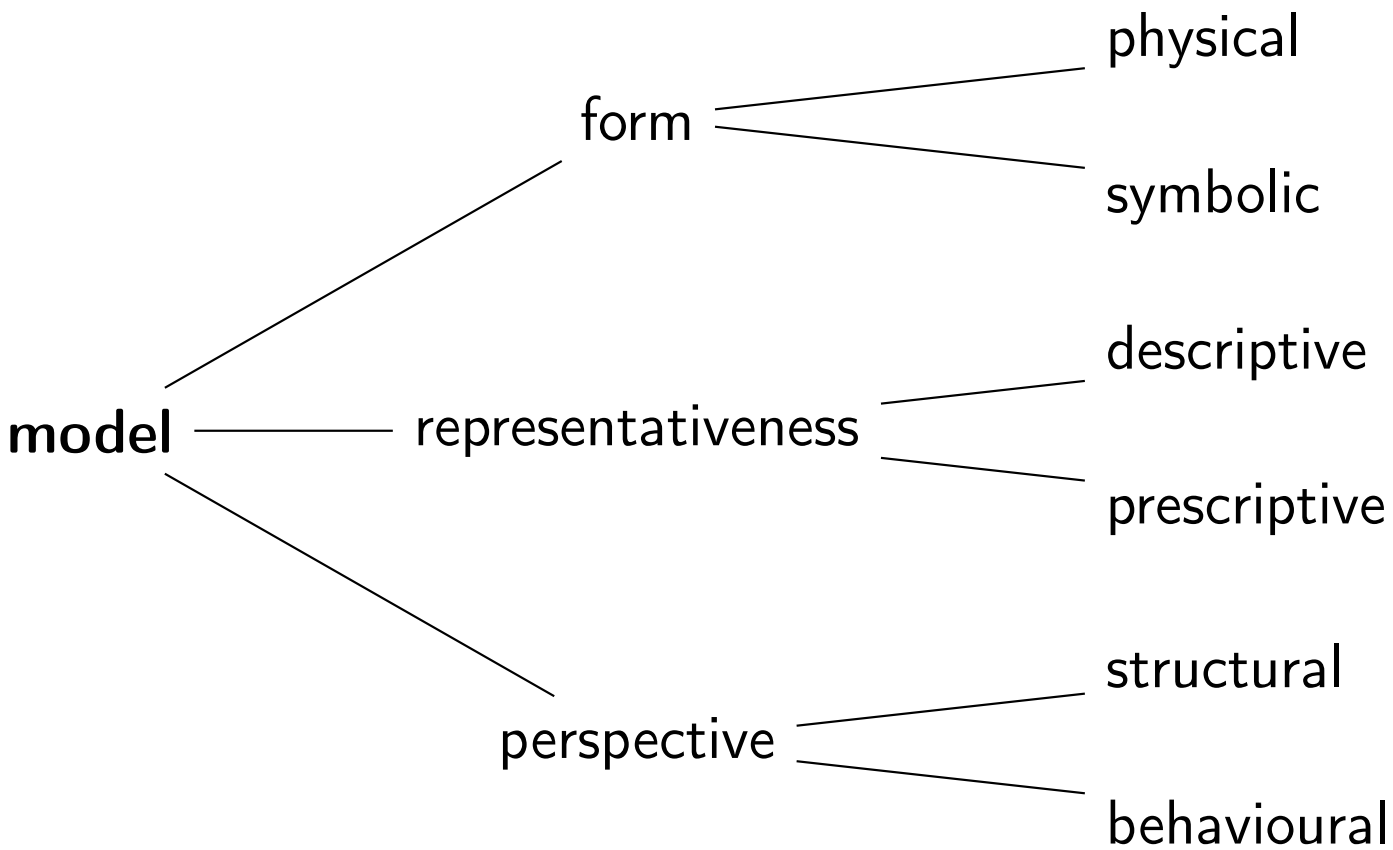
- A model represents the reality for a given purpose.
- It is a simplification of reality in the sense that the model cannot represent all its aspects.
- To be useful, a model must not represent all aspects of reality.
- The purpose of the model, related to the intention of the modeller, determines which aspects to represent and which aspects not to.
- It is not a replication but an intentional selective construction of a new thing meant to stand for something else.

characteristics of models

- **Abstraction**. A model is a reduced description of the system.
- **Understandability**. By removing detail that is irrelevant for a given viewpoint, models, if specified in an intuitive way, allows one to more easily understand some of the system properties.
- **Accuracy**. For the properties of interest, a model provides a true-to-life representation of the system.
- **Reasoning**. A model helps in correctly analysing and reasoning about the interesting but non-obvious properties of the system, either through experimentation (e.g., by simulating the model on a computer) or some type of formal analysis.
- **Inexpensiveness**. A model must be drastically cheaper to construct and analyse than the system.

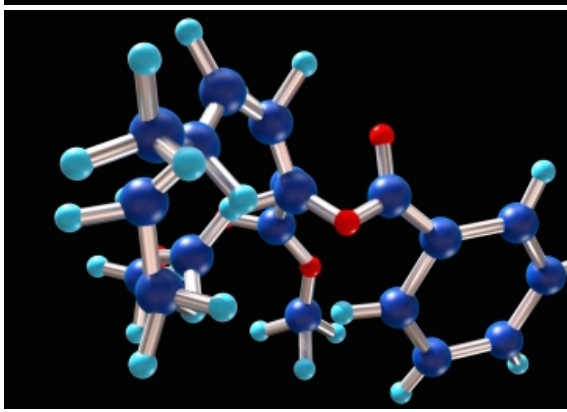
contents

dimensions


model
form
physical
symbolic
representativeness
descriptive
prescriptive
perspective
structural
behavioural

physical model

- A physical model (or iconic) is a reproduction at a reduced scale of a process arising in Nature.
- The word 'physical' means literally materialised or concrete, since one can touch the model.
  - An iconic model is perceived as imitating the system, being similar with respect to some of its properties.
  - A physical model is typically a smaller representation of the original object.
  - It can be larger if the original object is too small for humans.
  - Usually, a physical model is not as accurate or complete as reality.



Images courtesy of njaj and dream designs at FreeDigitalPhotos.net

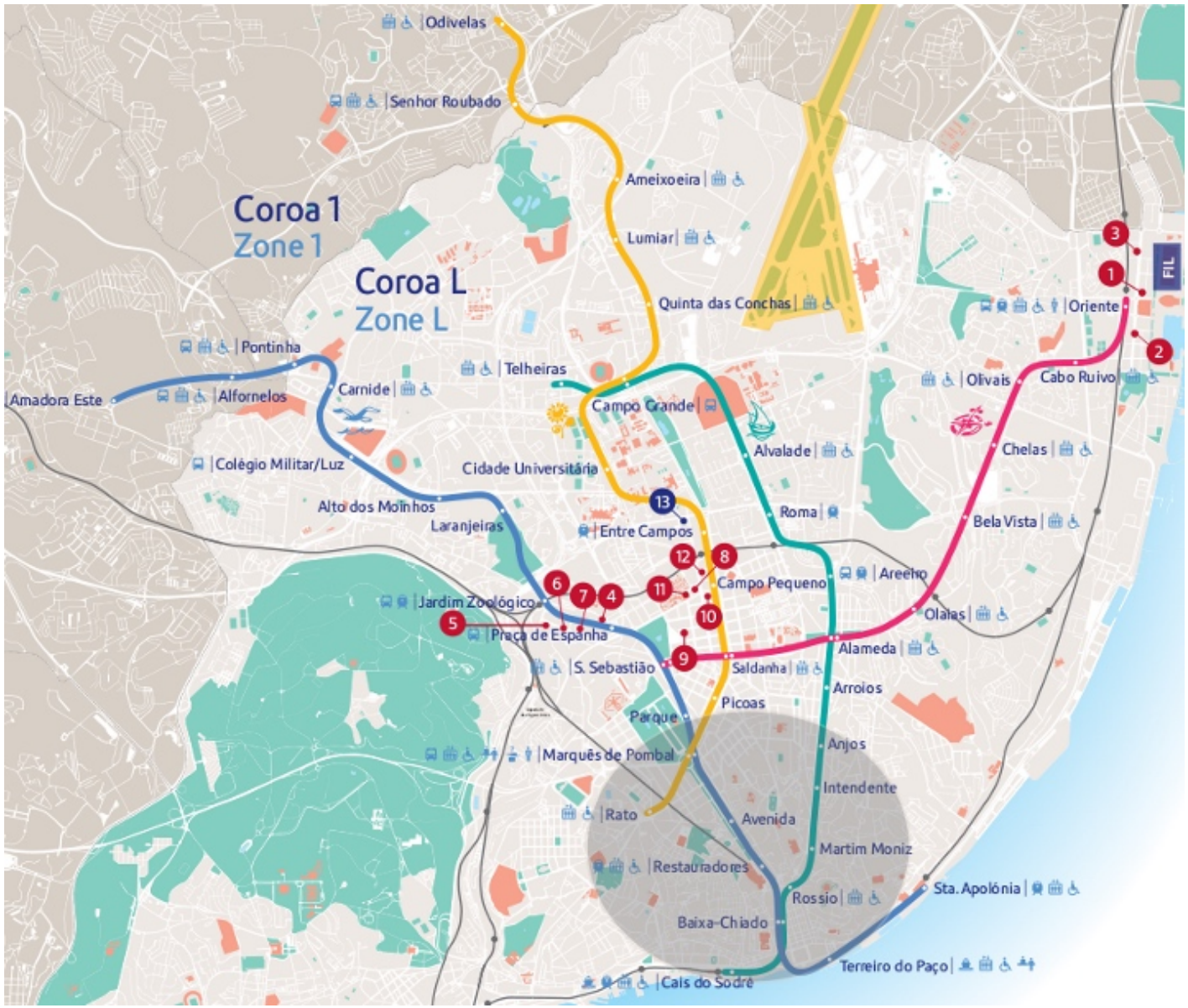
symbolic model

- A symbolic model (or mathematical) uses logical and quantitative relationships involving the dimensions of the system.
- In electrical engineering, models are a set of equations that represent the electrical circuits by applying some laws.
- An example is Ohm's law: $V = R \times I$.
- Typically, a symbolic model does not resemble the system it represents, but is fundamentally arbitrary or conventional.
- Arbitrariness is the absence of any necessary connection between a linguistic form and its meaning.
- A class is represented in UML by a rectangle, but this is a pure convention.
- The word 'strawberry' has no relation to the fruit itself; it is a pure convention.

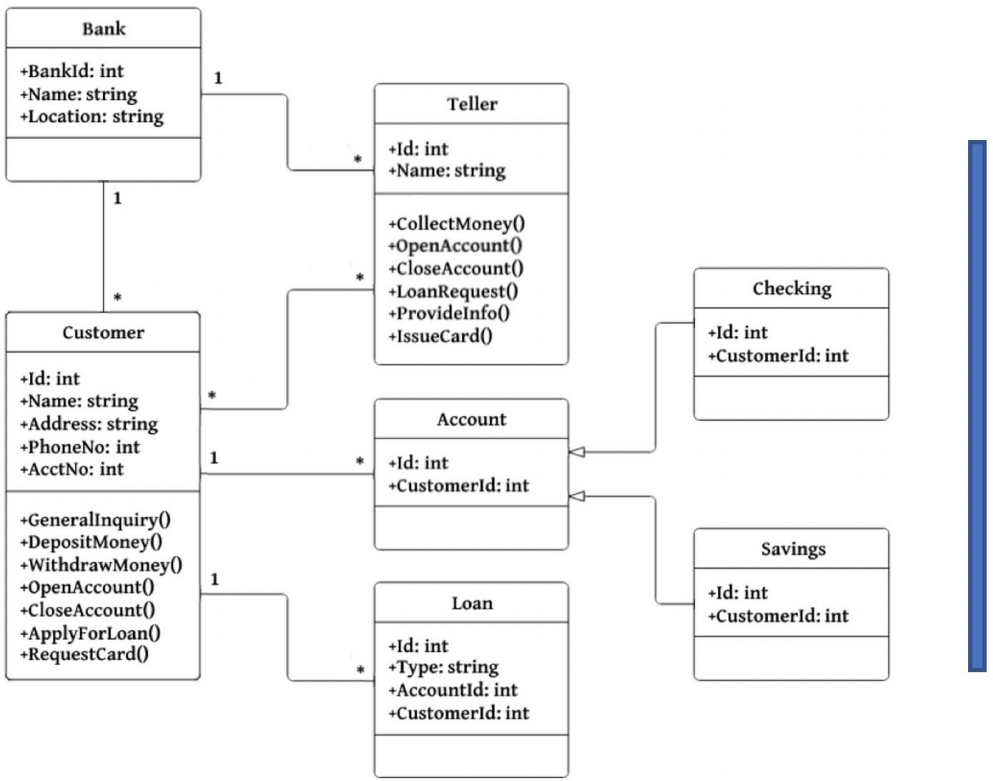
Image courtesy of antpkr at FreeDigitalPhotos.net

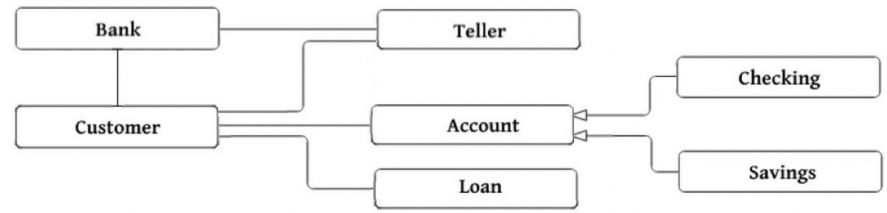
topological map

topographic map



class diagrams





too complex diagrams

models of a house

symbolic model

physical model

reality

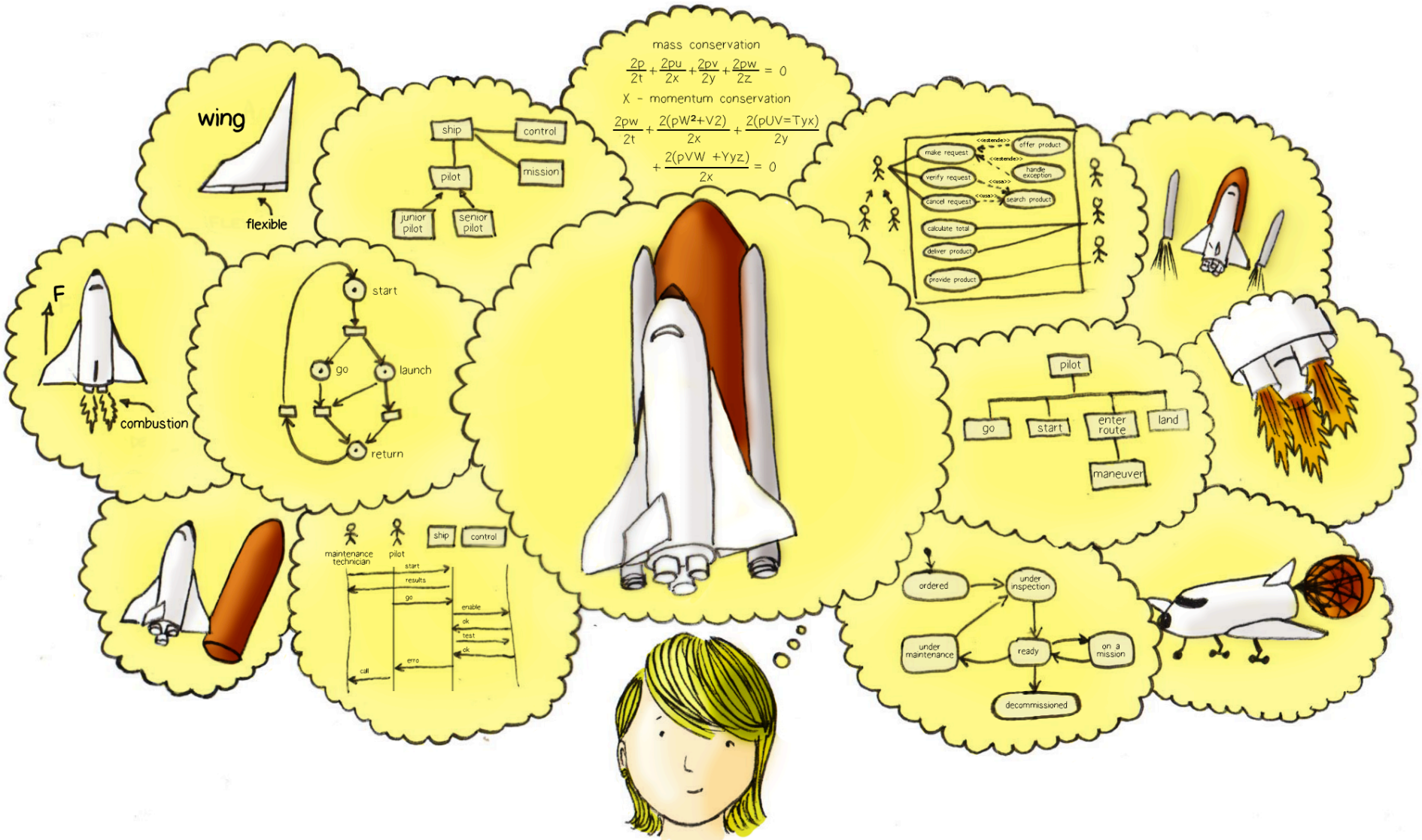
descriptive model

- A descriptive model is used to describe or mimic a real-world phenomenon or system.
- With a descriptive model, one can reason about the properties or the behaviour of the system.
- An example is a model of the weather that allows meteorologists to forecast it.
- In almost all natural sciences, models are descriptive as scientists try to understand how the natural world behaves.
- In engineering, descriptive models are used in reverse engineering when one wants to reason about an existing system without directly affecting it.

prescriptive model

- A prescriptive model is used to define how a yet-to-be-built system is supposed to be.
- Prescriptive models are adopted in the so-called forward engineering.
- In software engineering, models created during the analysis stage describe the problem at hand.
- When prescriptive models are used, the target system does not exist yet! It must be engineered.
- Most of the models used in engineering are prescriptive.

perspectives



structural model

- A structural model is focused on the static aspects of a system.
- These models are used for describing the components or modules that are part of the system, so they serve for conceptualising the system architecture.
- UML class, component, and deployment diagrams are used for representing structural models.

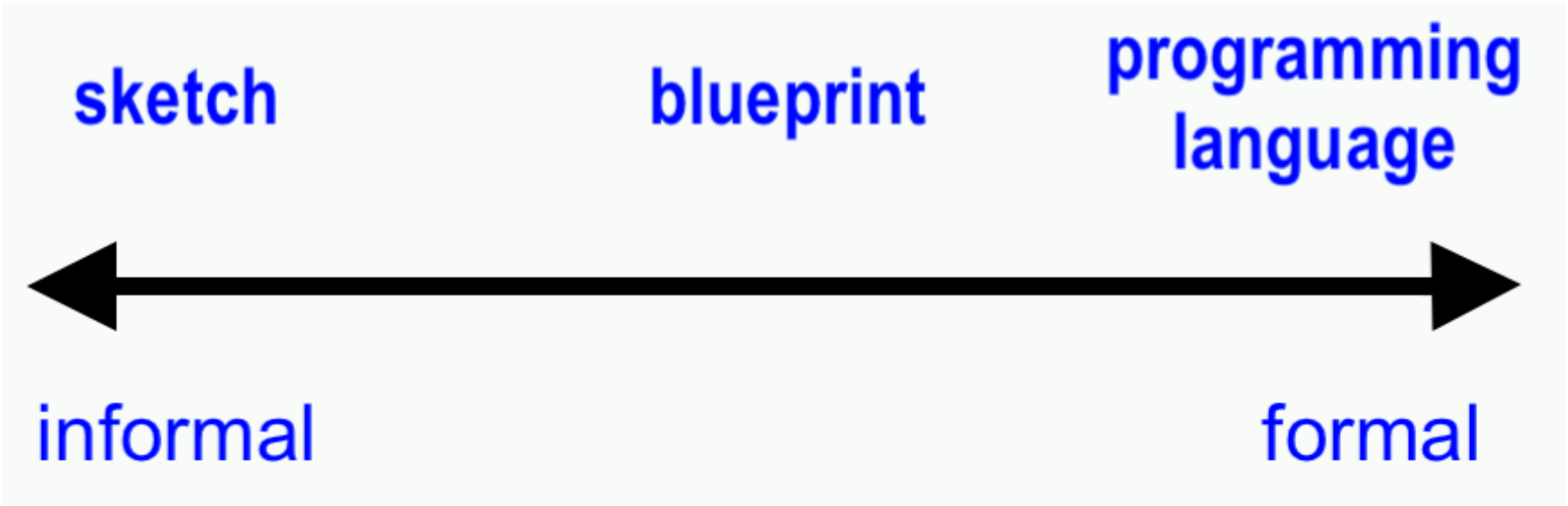
behavioural model

- A behavioural model emphasises the dynamic, functional, and temporal aspects of the system.
- This type of models address the behaviour of the system, being thus especially relevant in the analysis phase.
- Examples: finite state machines, Petri nets, and data flow diagrams (DFDs).

contents

UML models

- Modelling, done during the analysis phase, aims to specify the requirements of the systems.
- Most software models are, in industrial contexts, represented in UML.
- This language presents many types of diagrams.
- Most developers use in practice only a subset of those diagrams and from those also resort just to part of their available constructors.
- This situation is easily framed in the Pareto law (or 80/20 rule).



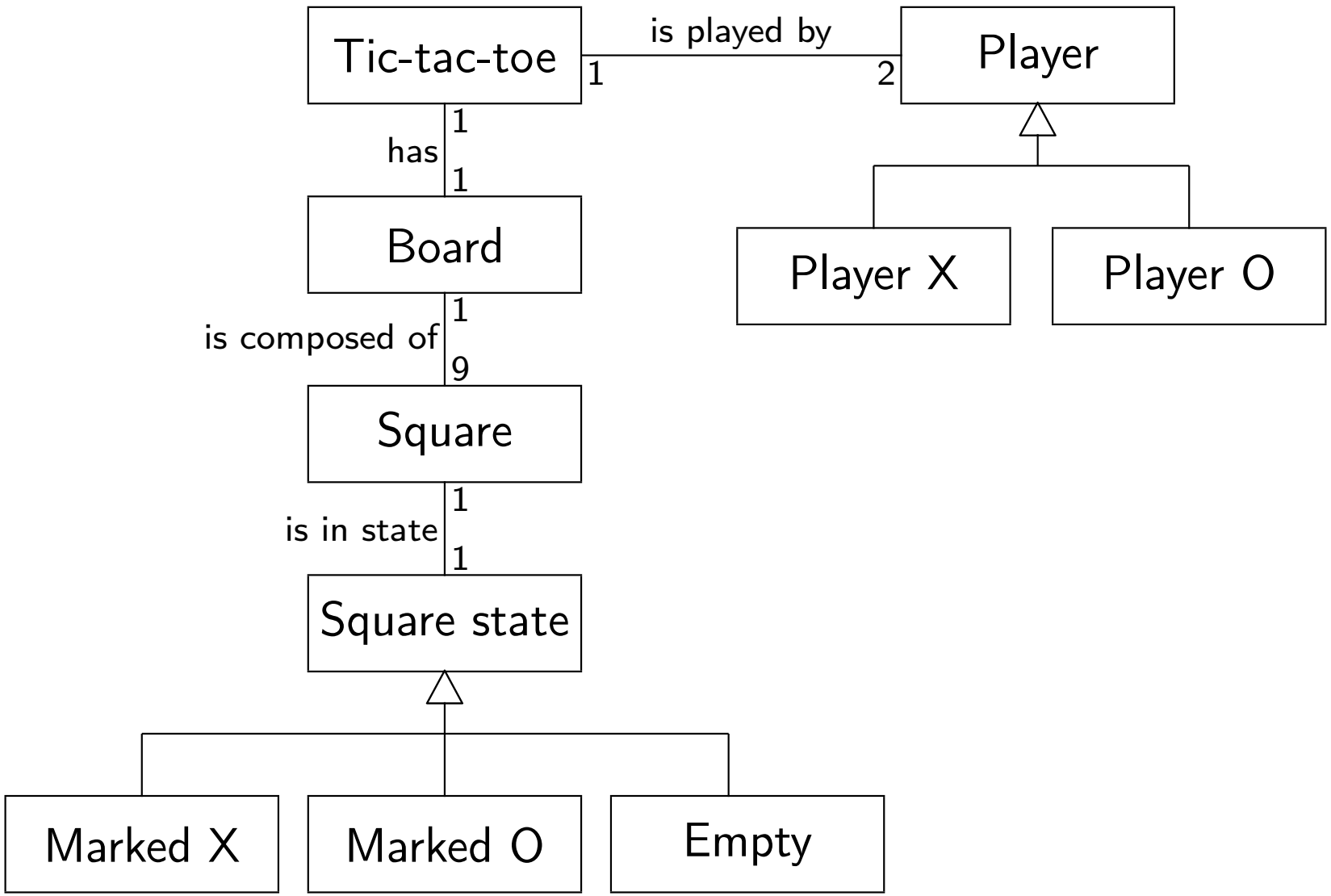
essential UML diagrams

model	purpose
domain	describe the vocabulary, concepts of the domain and characteristics of the systems that can be developed for that domain
use case	describes the proposed functionalities of a given system
interaction	show how the various objects or entities collaborate, emphasising the flow of control and data among them
class	present a set of concepts, types and classes and the respective relations
state	specify the behaviour of an entity or indicate the various states (or modes) through which it transits throughout its life
activity	show the control flow among the activities of a process

domain model

- Many software systems are focused on a given domain.
- In those cases, it is relevant to use domain models for capturing the common elements of the systems to be developed.
- A *domain model* is a description of the common properties and variables of the domain related to the system that is being developed.
- The domain model expresses enduring truths about the universe that is relevant to the system at hand.
- That description must include:
  1. a definition of the scope of that domain, providing examples of systems or generic rules of inclusion,
  2. the vocabulary of the domain (i.e., the glossary with the principal terms),
  3. a model of concepts that identifies and relates the concepts of that domain.

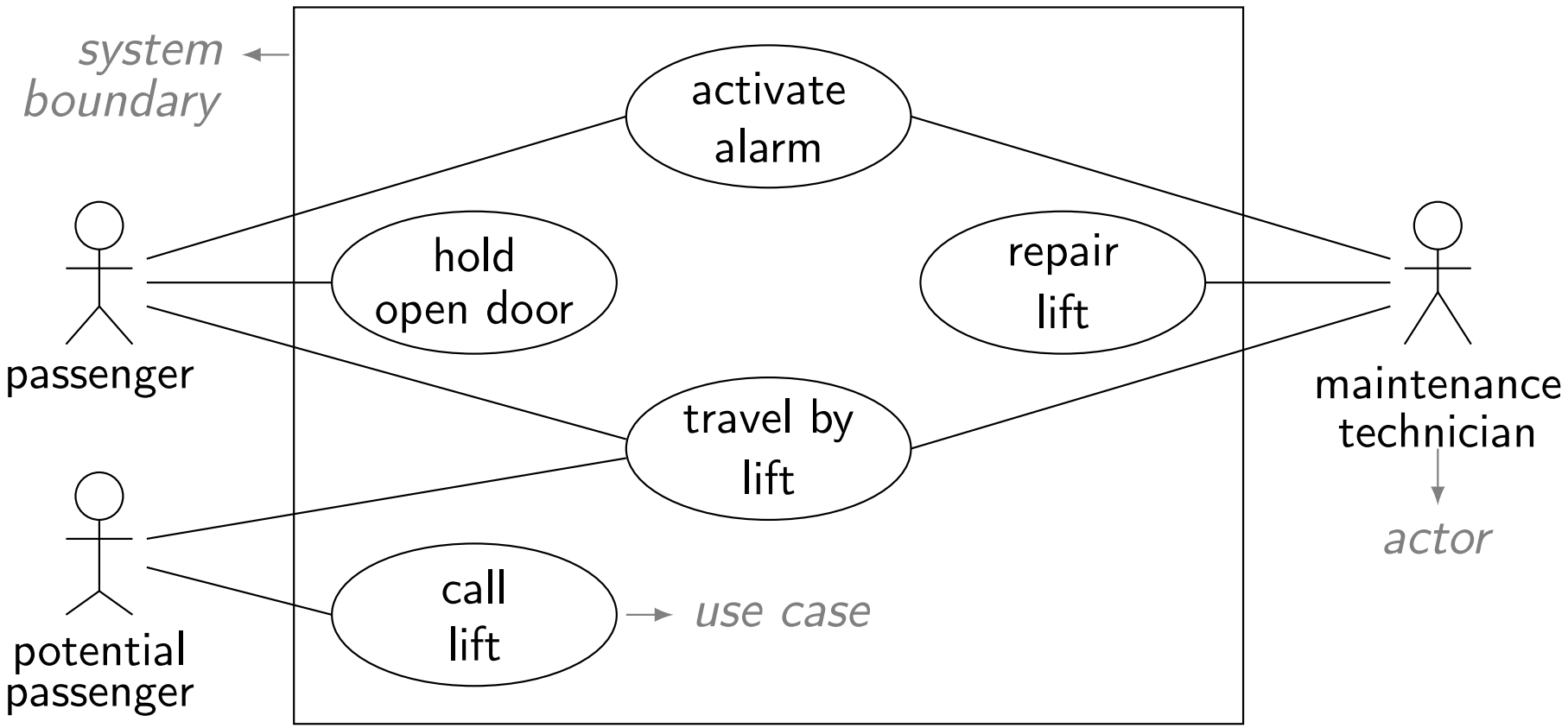
domain model for tic-tac-toe



use case models

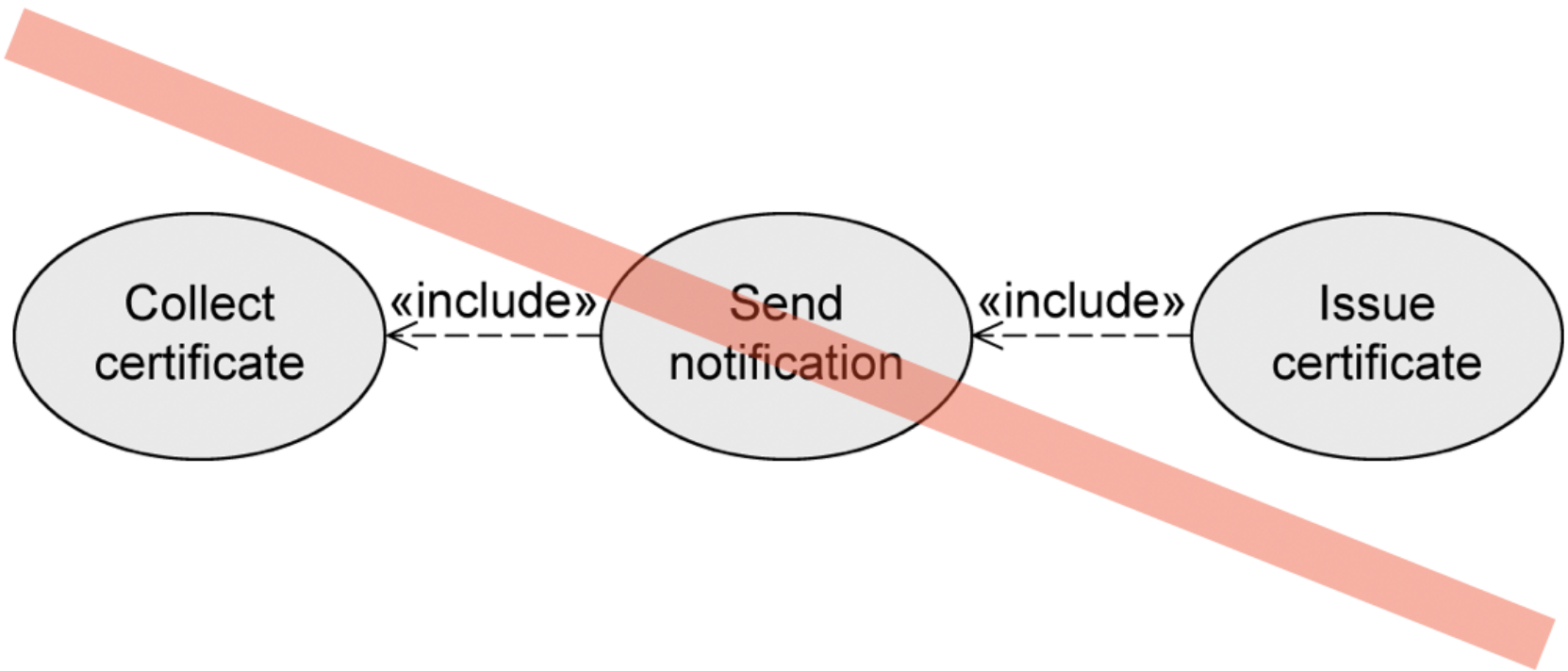
- The utilisation of use case models serves essentially two purposes:
  1. defining the frontier of the system with the environment,
  2. specifying the functionalities that the system makes available to its users.
- A *use case diagram* resorts to, as basic elements, use cases and actors.
- The usage of verbs is recommended to characterise the use cases, thus enhancing their functional nature.

use case diagram



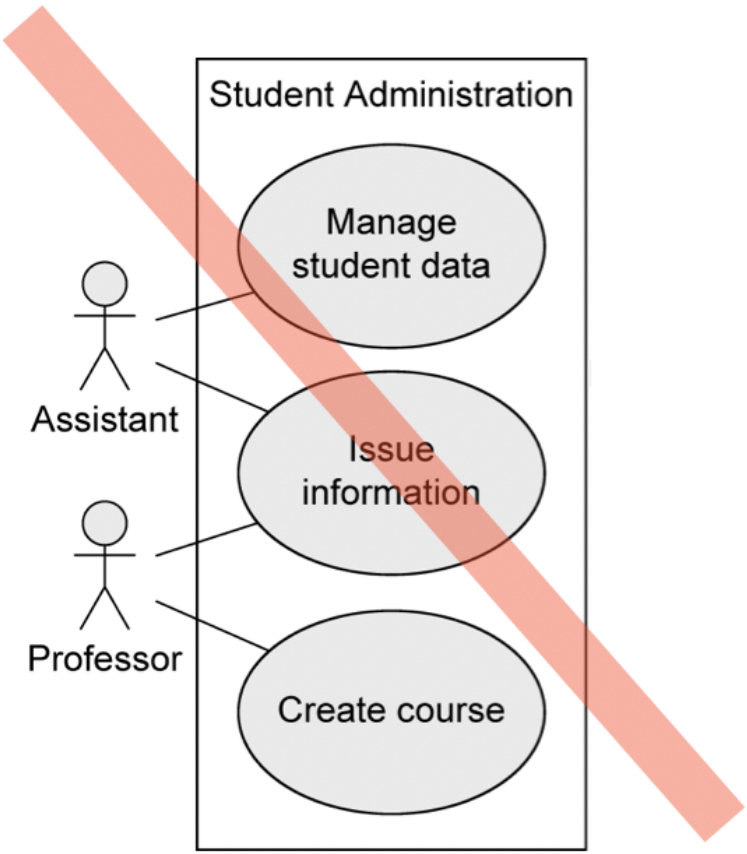
use case diagram

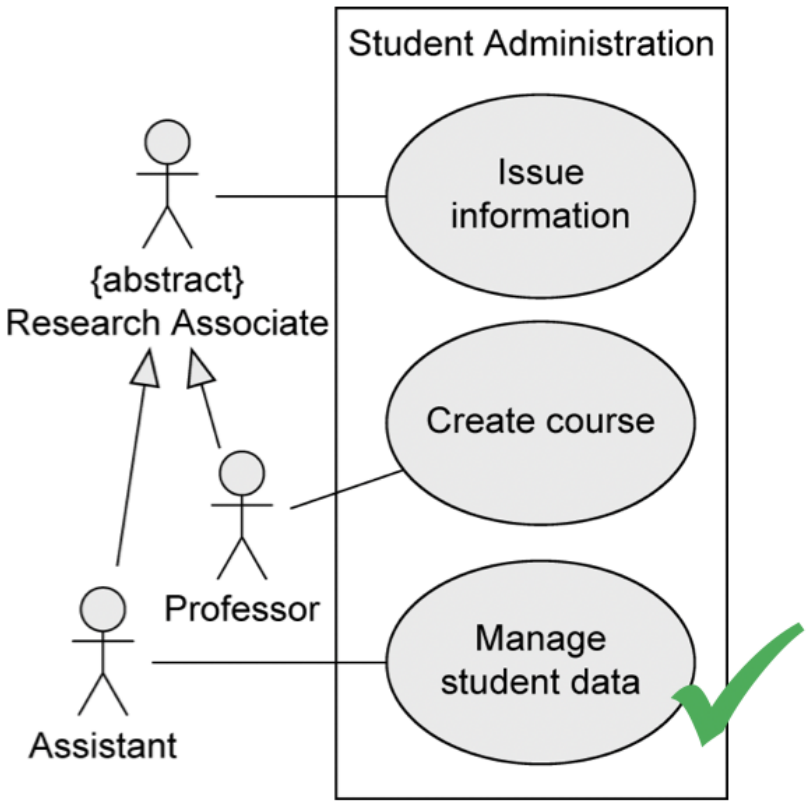
Use cases do not model processes/workflows



use case diagram

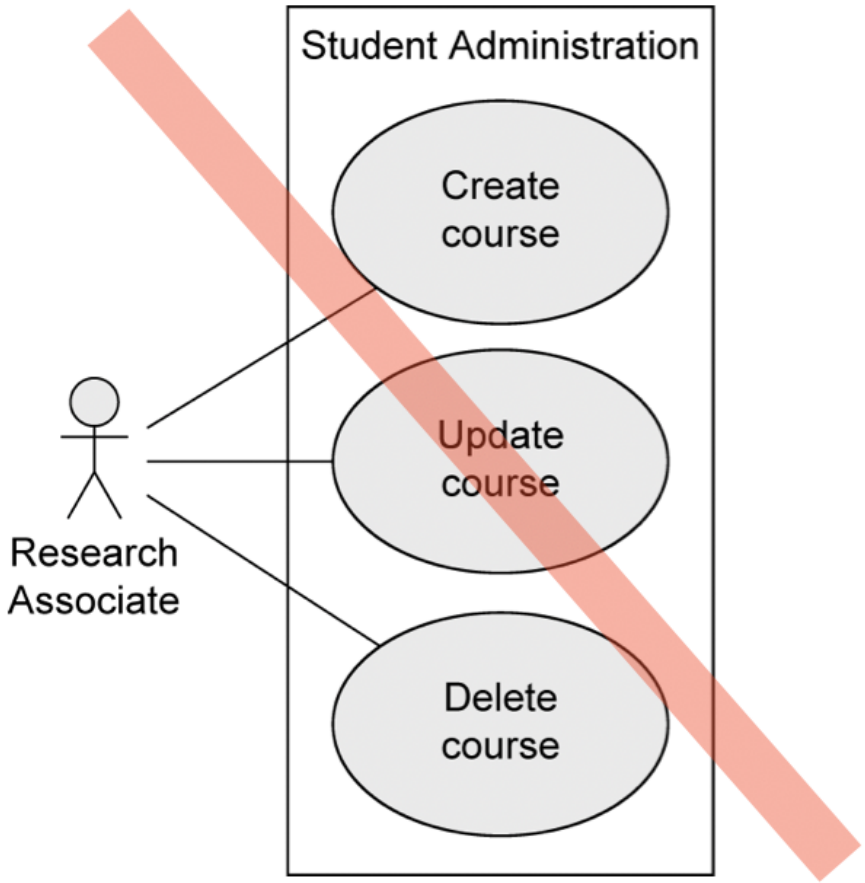
Issue information needs EITHER **Assistant** OR **Professor** for execution

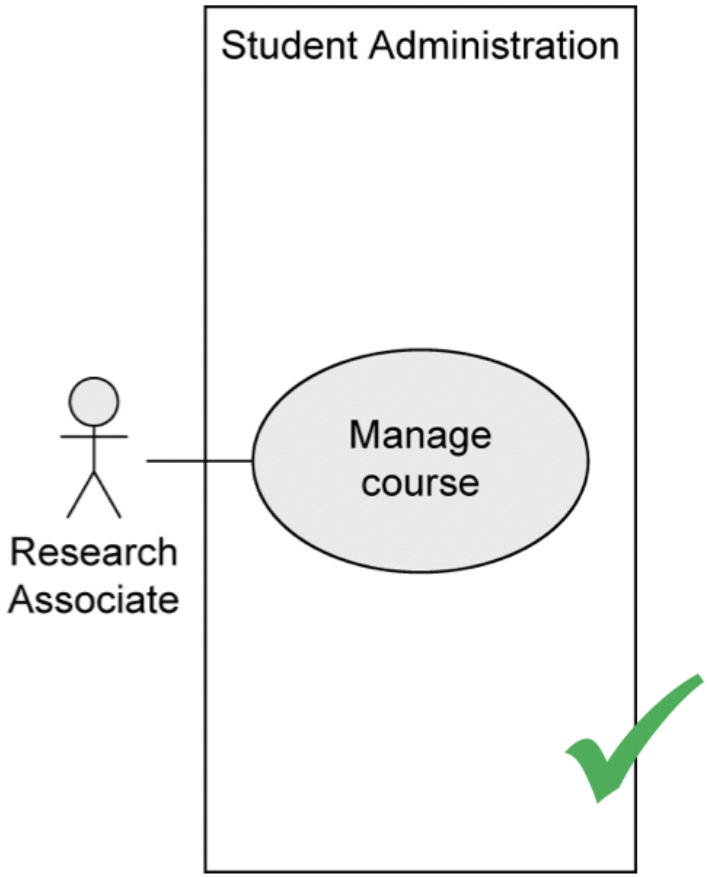




use case diagram

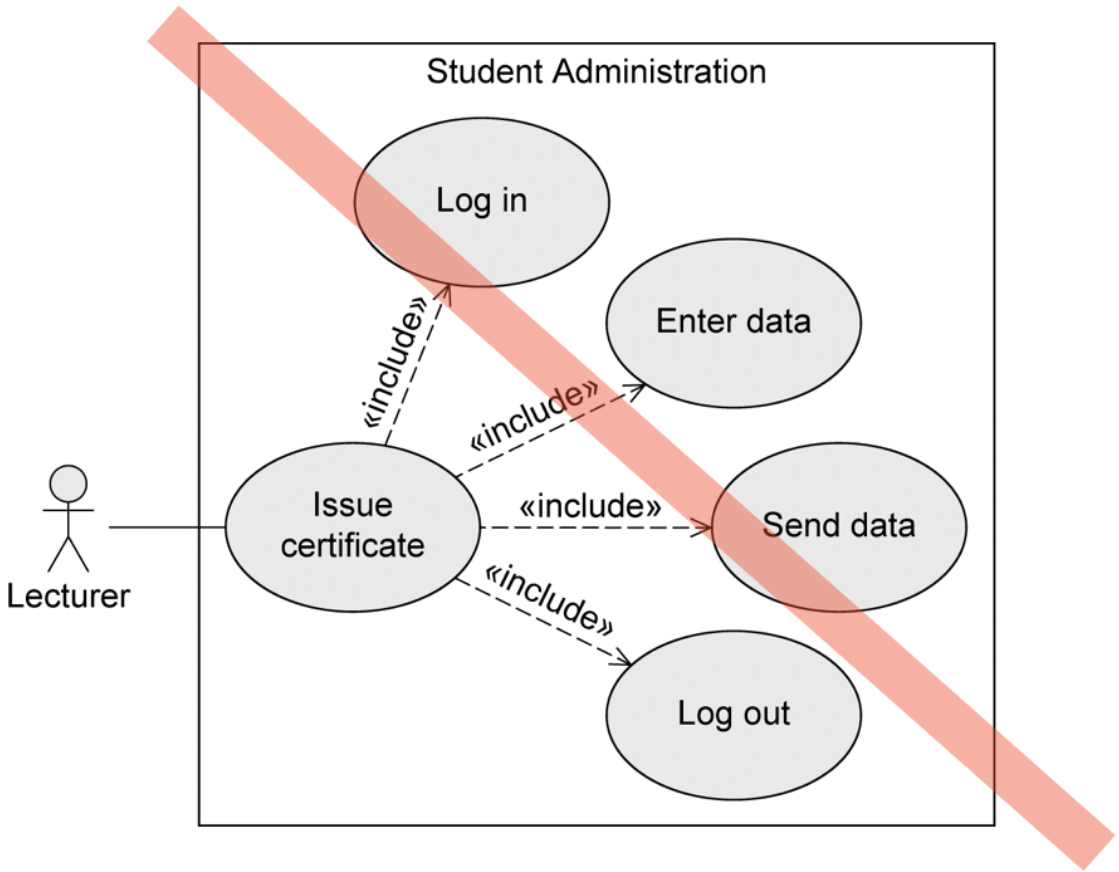
Many small use cases with the same objective may be grouped to form one (more abstract) use case

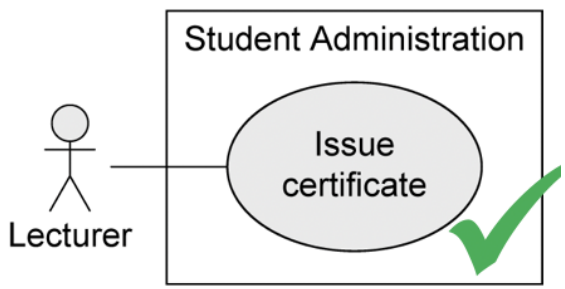




use case diagram

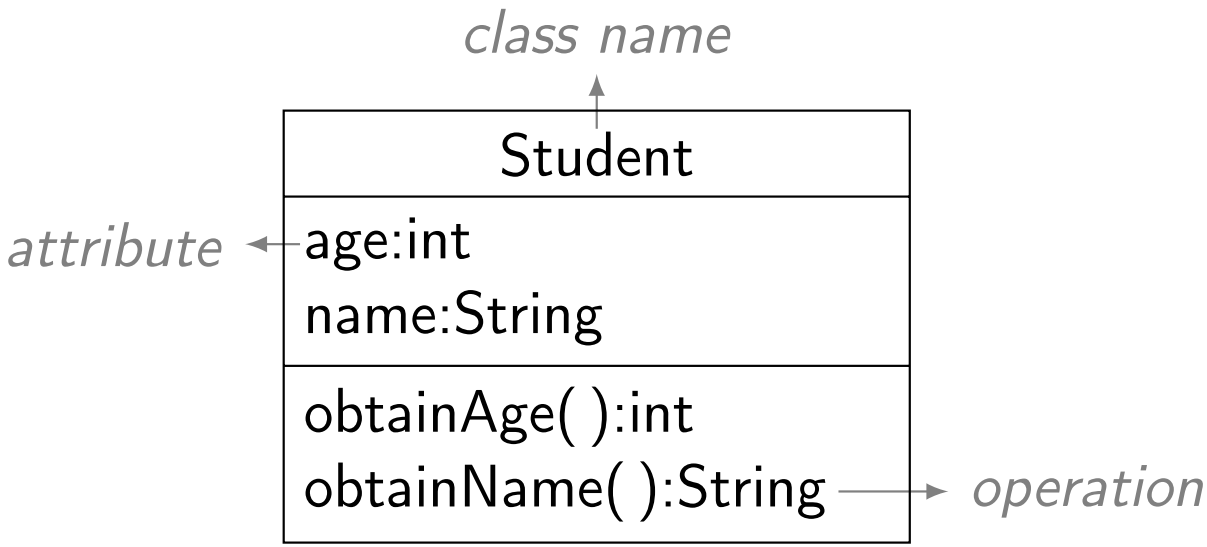
The various steps of a use case are not separate use cases themselves!
NO functional decomposition



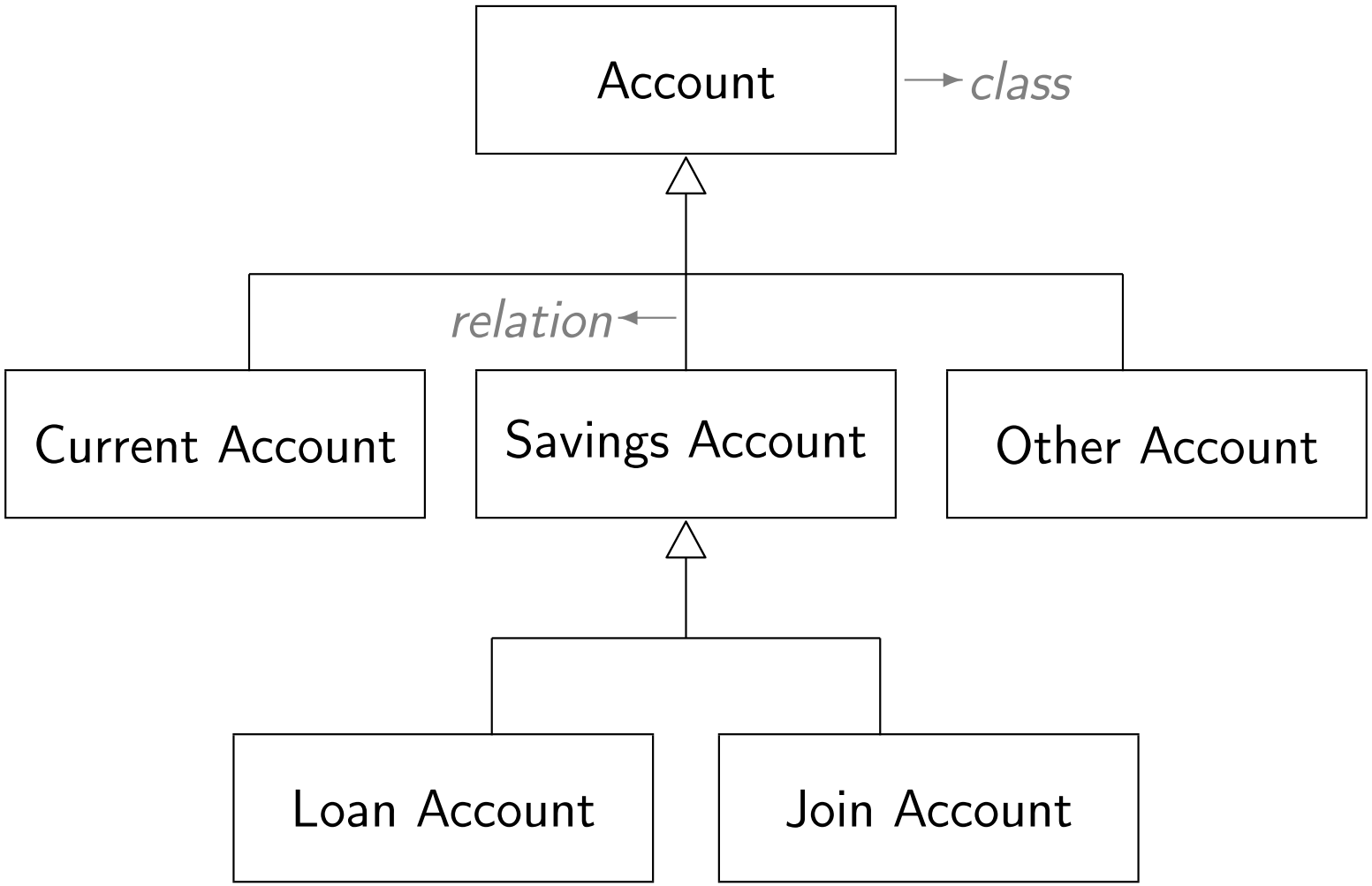


class models

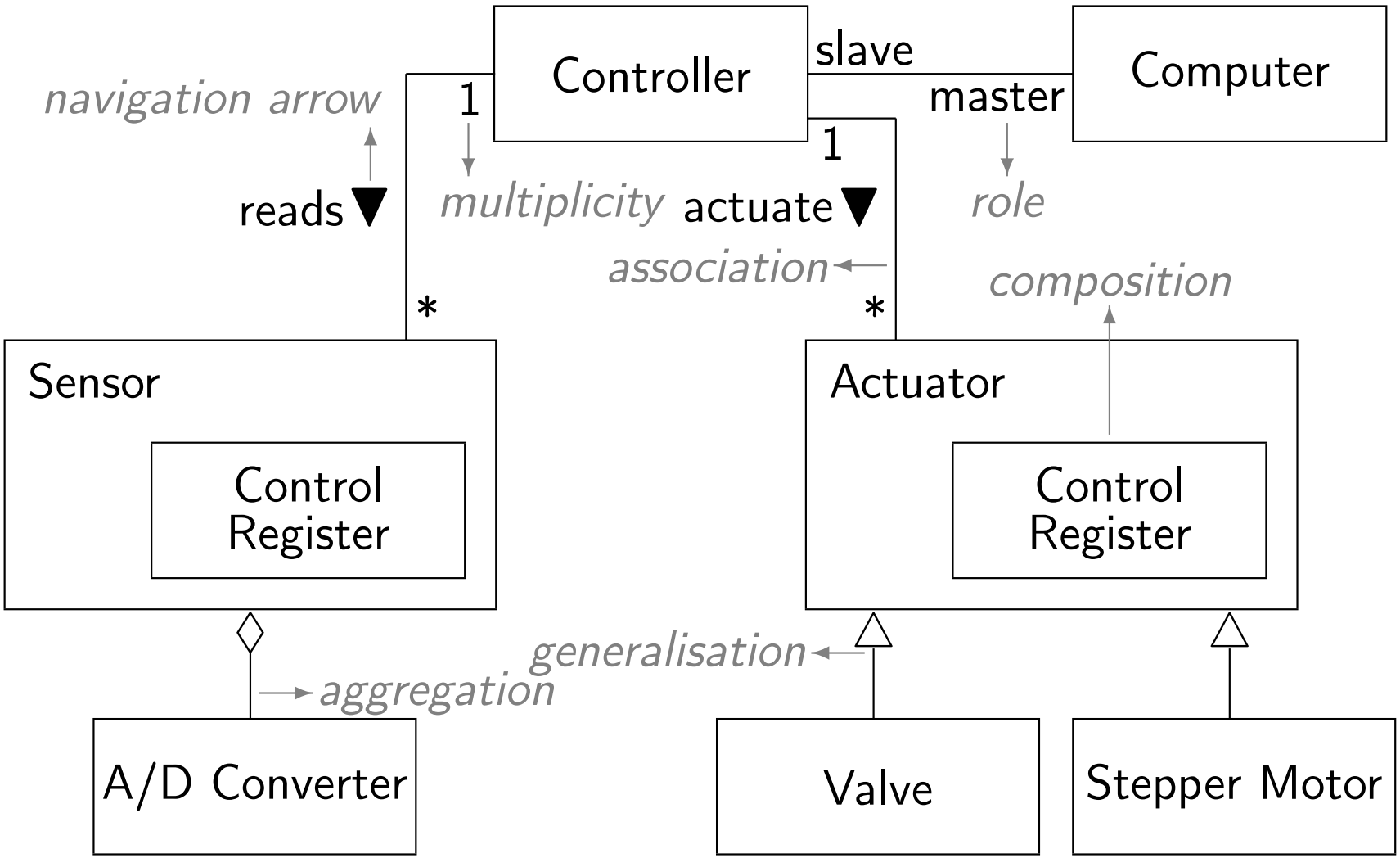
- *Class models* are necessary to indicate the existing classes and their relations.
- These models are contemplated by all object-oriented software development methods.



class models



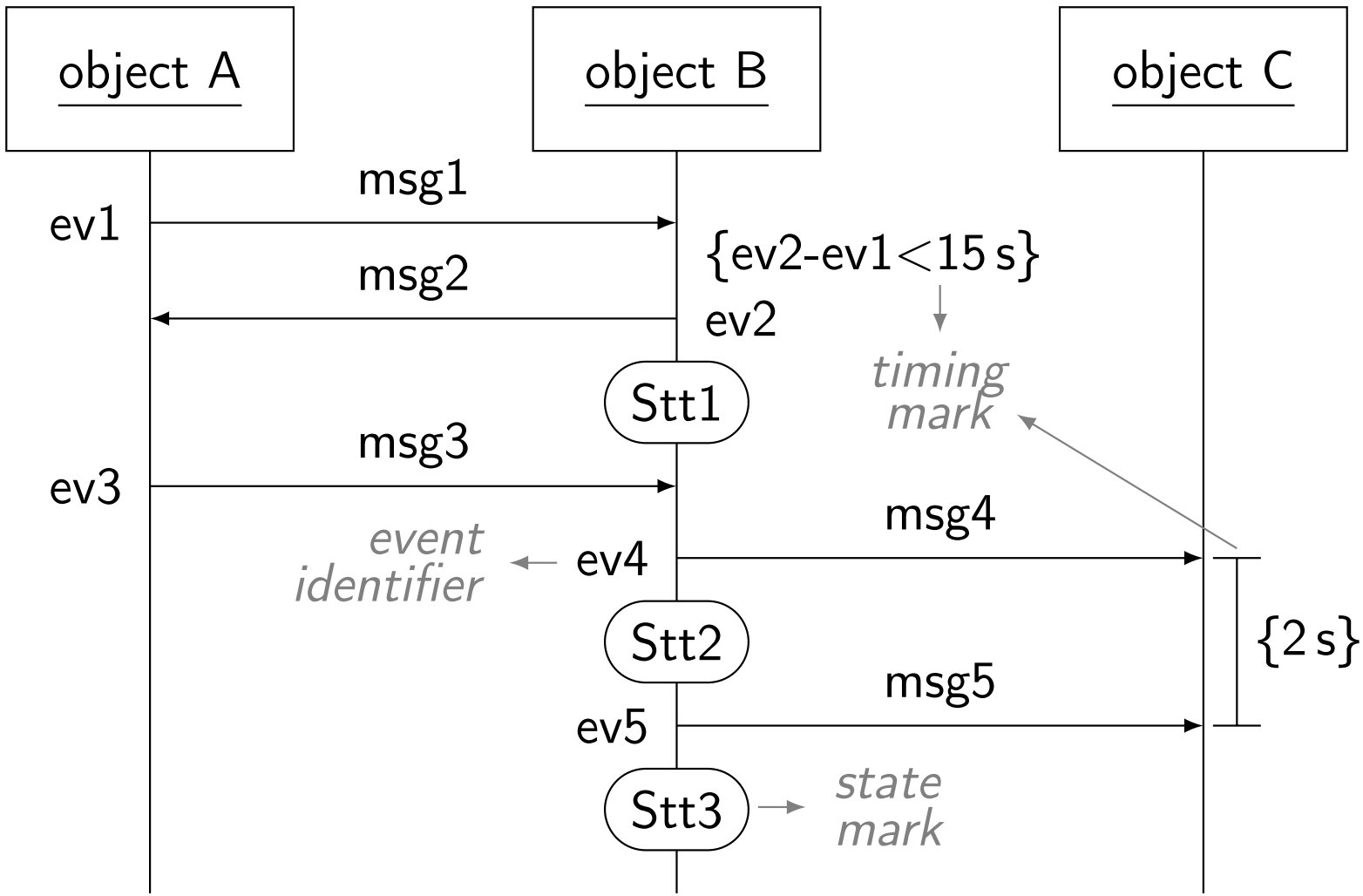
class models



sequence models

- In same cases, it is necessary to model the dynamic aspects related to the exchange of messages between objects.
- Those models can be represented by *interaction diagrams*.
- An interaction model can be used for representing an instance of a use case.
- They describe how a group of objects communicate amongst them.
- In UML (version 2.2), there are four different types of diagrams that allow interaction models to be represented.
- We present only sequence diagrams.

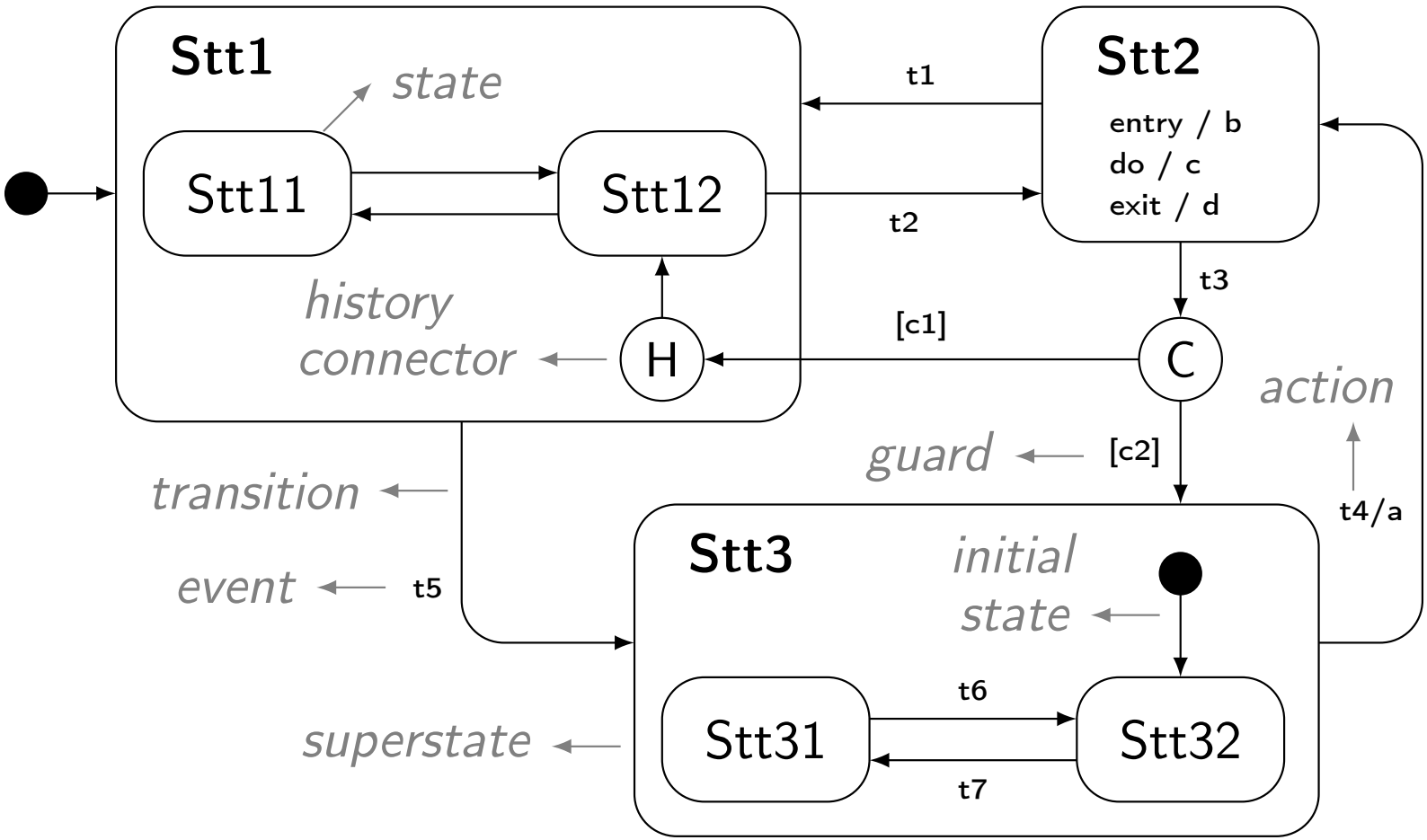
sequence models



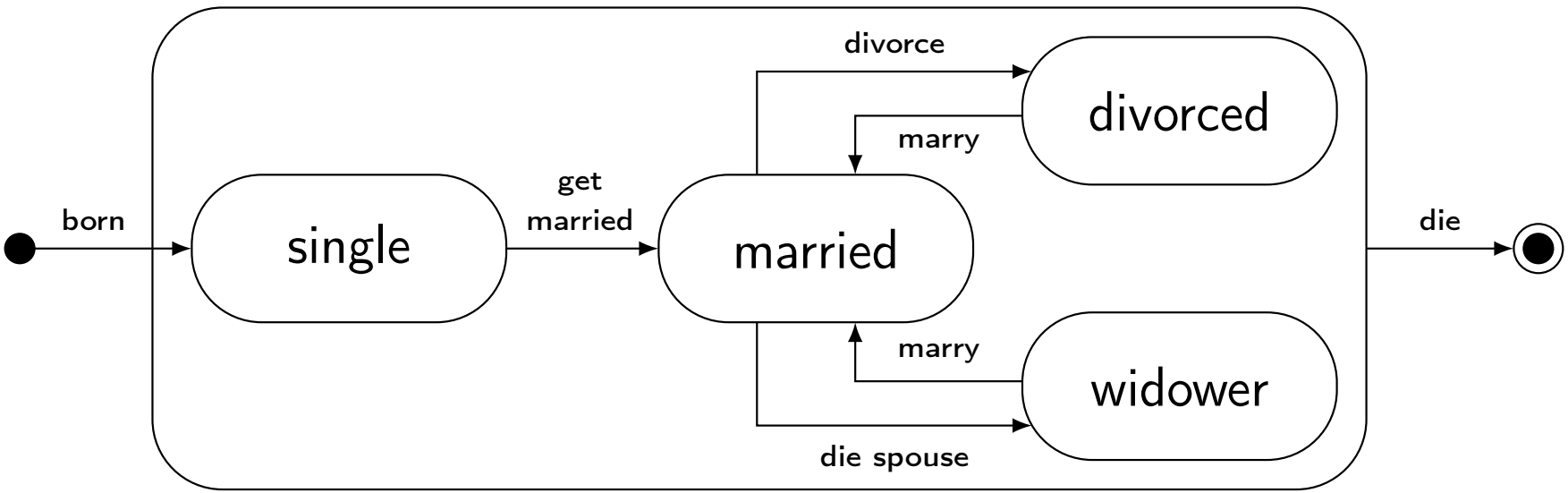
state models

- Class models do not allow the dynamic behaviour of the instances of the classes to be determined.
- The use of state diagrams has become popularised in the hardware domain, but its modelling capacity has proved to be useful in diverse computing areas.
- State diagrams can be used for defining the (dynamic, temporal) behaviour of a class.
- In a conventional state diagram, one and only one state is active in each instant.
- A state is an ontological condition that persists for a significant period of time that is distinguishable and disjoint from other similar conditions.

state models



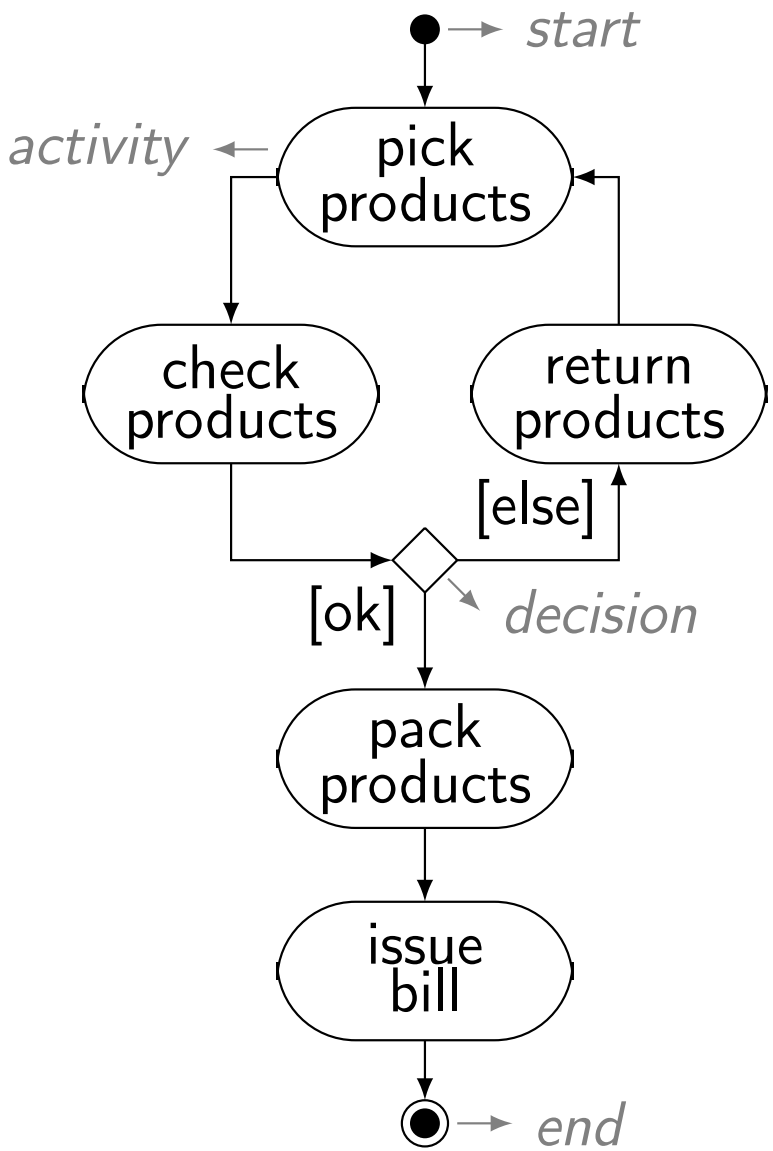
state models



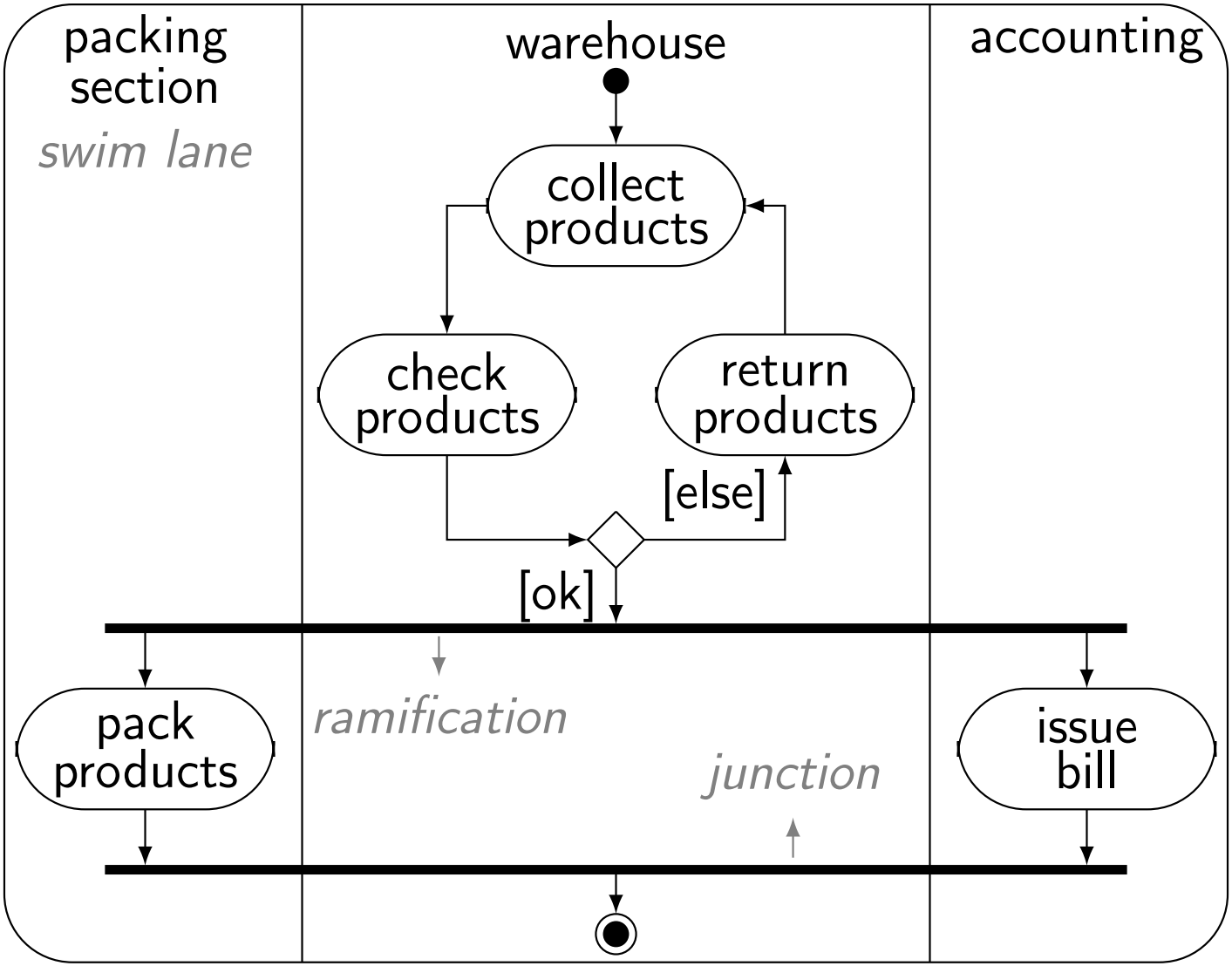
activity models

- Activity models are useful to relate the control flow among the activities of a given business process.
- These models address behavioural aspects of the systems or entities under consideration.
- These models are appropriate when the behaviour change occurs, mainly due to the end of the action/activity executed.

activity models



activity models



Summary

- Using models is essential in all engineering branches.
- A model represents in a simplified way the reality for a given purpose, emphasising some elements and ignoring others.
- The models to be effectively useful, must possess to a sufficient level the following characteristics: abstract, comprehensible, accurate, predictable, and inexpensive.
- models can be characterised according to three dimensions: form (symbolic, physical), representativeness (prescriptive, descriptive), and perspective (behavioural, structural).
- Some UML models for documenting the requirements of a system were presented.
- For requirements engineering, domain, use case, interaction, class, state, and activity models are deemed crucial.

bibliography

- Fernandes JM and Machado RJ; *Requirements in engineering projects*, Springer, Lecture Notes in Management and Industrial Engineering series, ISBN 978-3-319-18596-5, 2016. [chapter 8]

  `http://www.springer.com/978-3-319-18596-5`