Conceptual modelling and Database modelling : UML revisited

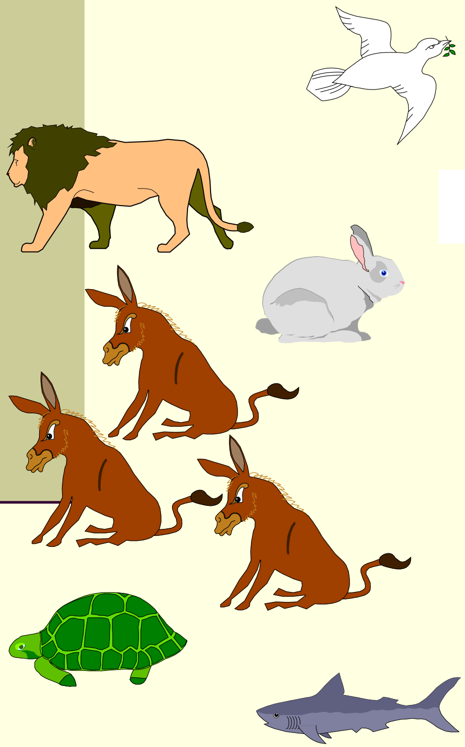
From reality to system

Conceptual modeling

Modelling patterns

Building the “right” system

Universe of Discourse



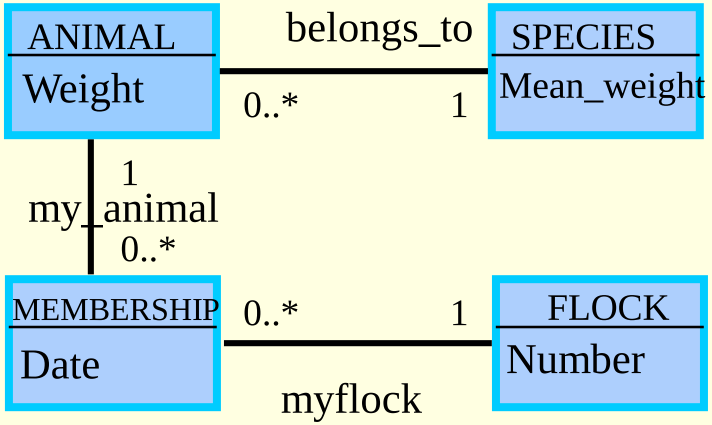
?

=

<

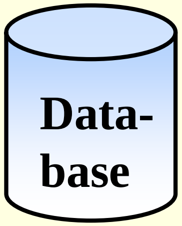
>

Conceptual Model (CM)



Use of CM

Contract



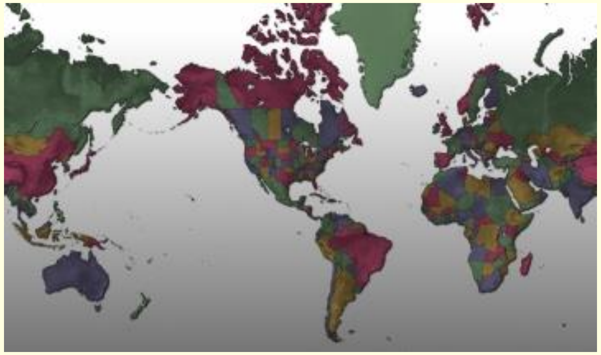
“System”

Education,
Reverse Engineering

Models – true?, false?, good?, problematic?

Model: A structure-map of some part of reality?

Example of a model: - a MAP



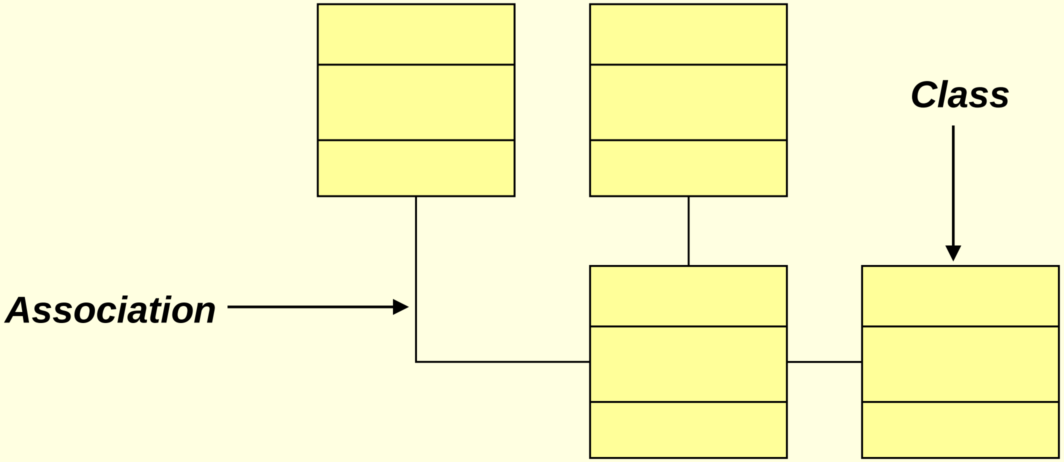
Models simplify

Models distorts - Greenland, Africa

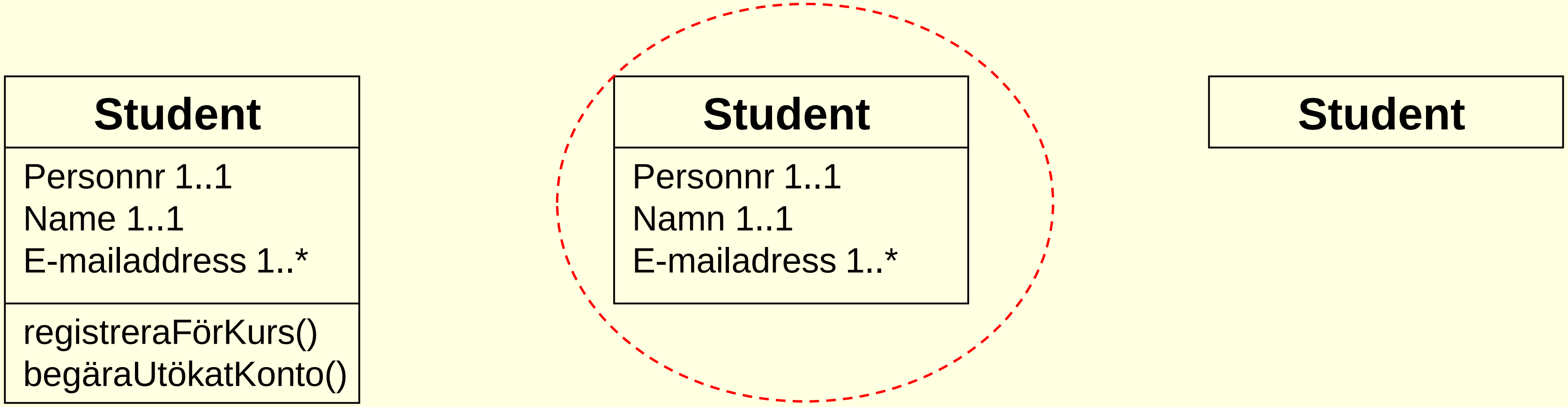
Models focuses and highlights - topograpically, politically etc. etc.

Modelling languages: Complex languages with many symbols VS simpler languages that are, perhaps?, easier to understand and validate for domain-knowledgeable stakeholders who know the domain but not the modeling-language.

UML klassdiagram revisited



Classes – variations in syntax



Student

Personnr 1..1
Name 1..1
E-mailaddress 1..*

registreraFörKurs()
begäraUtökatKonto()

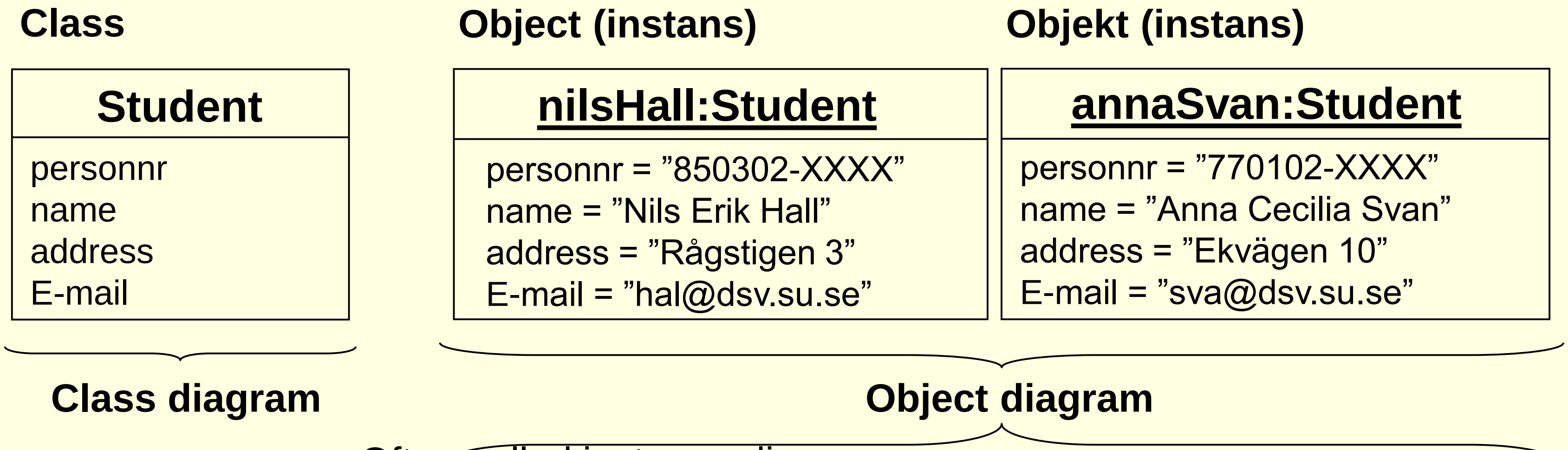
Student

Personnr 1..1
Namn 1..1
E-mailadress 1..*

Student

This syntax (a bit simplified) is commonly used when analysing the
Information need at an organization, institution, system etc.

Class diagram vs object diagram



Class

Student

personnr
name
address
E-mail

Object (instans)

nilsHall:Student

personnr = ”850302-XXXX”
name = ”Nils Erik Hall”
address = ”Rågstigen 3”
E-mail = ”hal@dsv.su.se”

Objekt (instans)

annaSvan:Student

personnr = ”770102-XXXX”
name = ”Anna Cecilia Svan”
address = ”Ekvägen 10”
E-mail = ”sva@dsv.su.se”

Class diagram

Object diagram

Often called instance-diagram.

Extended syntax to identify the instance.

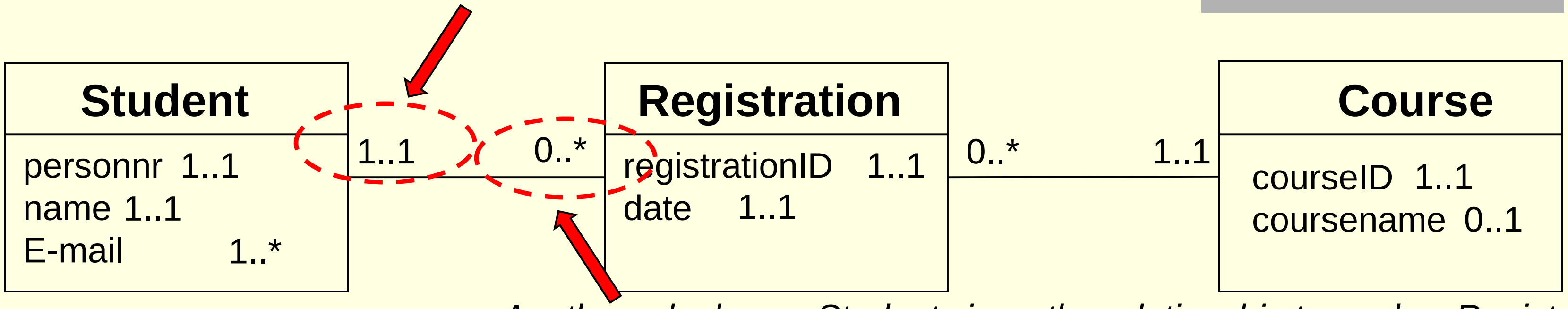
The attributes are given actual values.

May be seen as an observation of one or several objects at a given point in time, e.g attribute values may, for instance, change during the life-cycle of an object.

NOT used in this course, our instances will be rows in a relational table in a relational database management system. More on this in the next lecture on the relational model and relational DBMS.

Multiplicities – minimum..maximum

A role: how a Registration views the relationship toward a Student!



Student

personnr 1..1
name 1..1
E-mail 1..*

1..1

0..*

Registration

registrationID 1..1
date 1..1

0..*

1..1

Course

courseID 1..1
coursename 0..1

Another role: how a Student views the relationship towards a Registration.

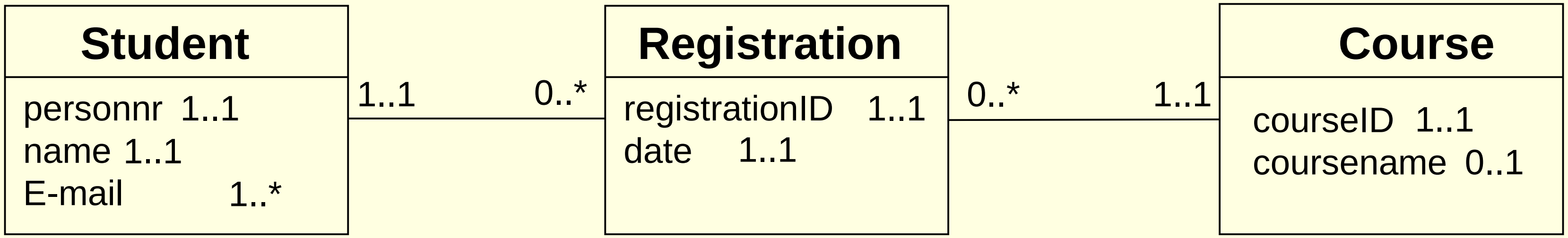
Multiplicities **(or cardinalities)** for an **attribute** ***or a*** **role in an association** shows how many objects the assocation can/must refer to or how many valued the attribut can or must have.

Values for multiplicities:

- The minumum value is given first
- The maximum value last

-Two full stop separate minimum and maxiumum values: **Min.. Max**

Multiplicity – common combinations



1..1 At least 1 and at most one

Example: A student has one and only one value on personnr.

1..* At least 1 and possibly many

Example: A student always have an e-mail address but can possibly have very many.

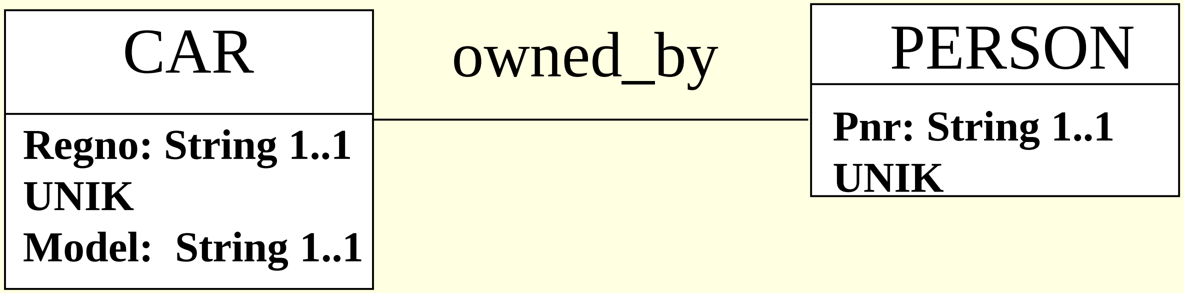
0..* Not necessarily any but possibly many.

Example: A student may not be registred at any course but can be registred at many courses.

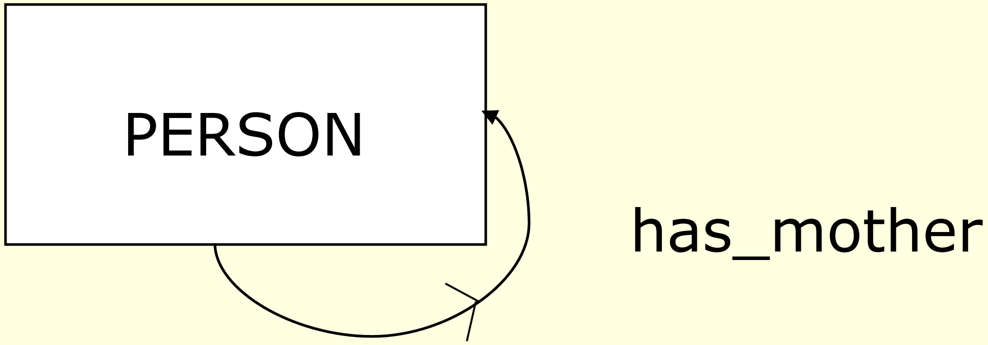
0..1 Not necessariy any but at most oen

Exempel: A course need not have a course name and can have at most one

Multiplicities – rules in the system vs ’common sense’

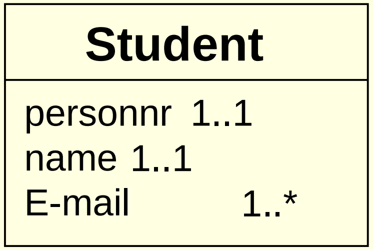


Multiplicities – an other exercise



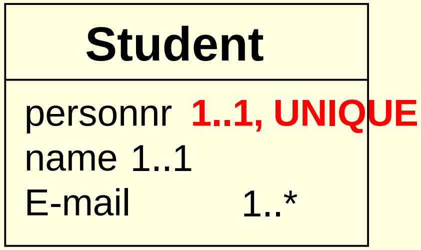
What multiplicities hold for the assocation ’has_mother’ ?

Multiplicities for ATTRIBUTES in UML – What identifies a class?



Student

personnr 1..1
name 1..1
E-mail 1..*



Student

personnr **1..1, UNIQUE**
name 1..1
E-mail 1..*

So how do we start to model…?

- In a real situation users are interviewed, manuals read, legacy systems studied, oranizations analyzed etc.…
- Sometimes we only have access to textual descriptions of the system-to-be-modelled
- How to analyze large walls of text? Verbs, nouns, adjectives? Modelling patterns, so called data-abstractions, rules of thumb, etc...

Same word – different concepts (homonyms)

- How is the word ”bok”used in the sentences below?
  - Jules Verne skrev många böcker, en har boktiteln ’Kapten Grants barn’
  - Biblioteket i Vällingby har många böcker
  - Om ’Kapten Grants barn’ trycks i A4-format blir antalet sidor i en tryckt bok 100.
  - Bokhandlaren i Jönköping sålde många böcker

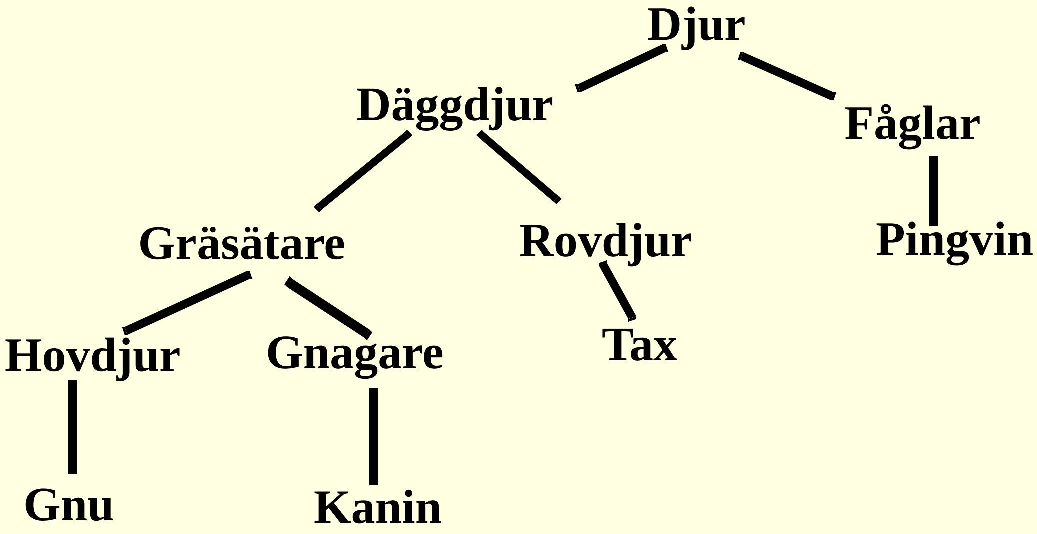
The same concept – different words/terms (synonyms)

- **Avskeda - Friställa**
- **Arbetsgivare – Arbetsköpare**
- **Student – Elev?**

Do synonyms always mean the same? For instance across organisational or institutional borders?

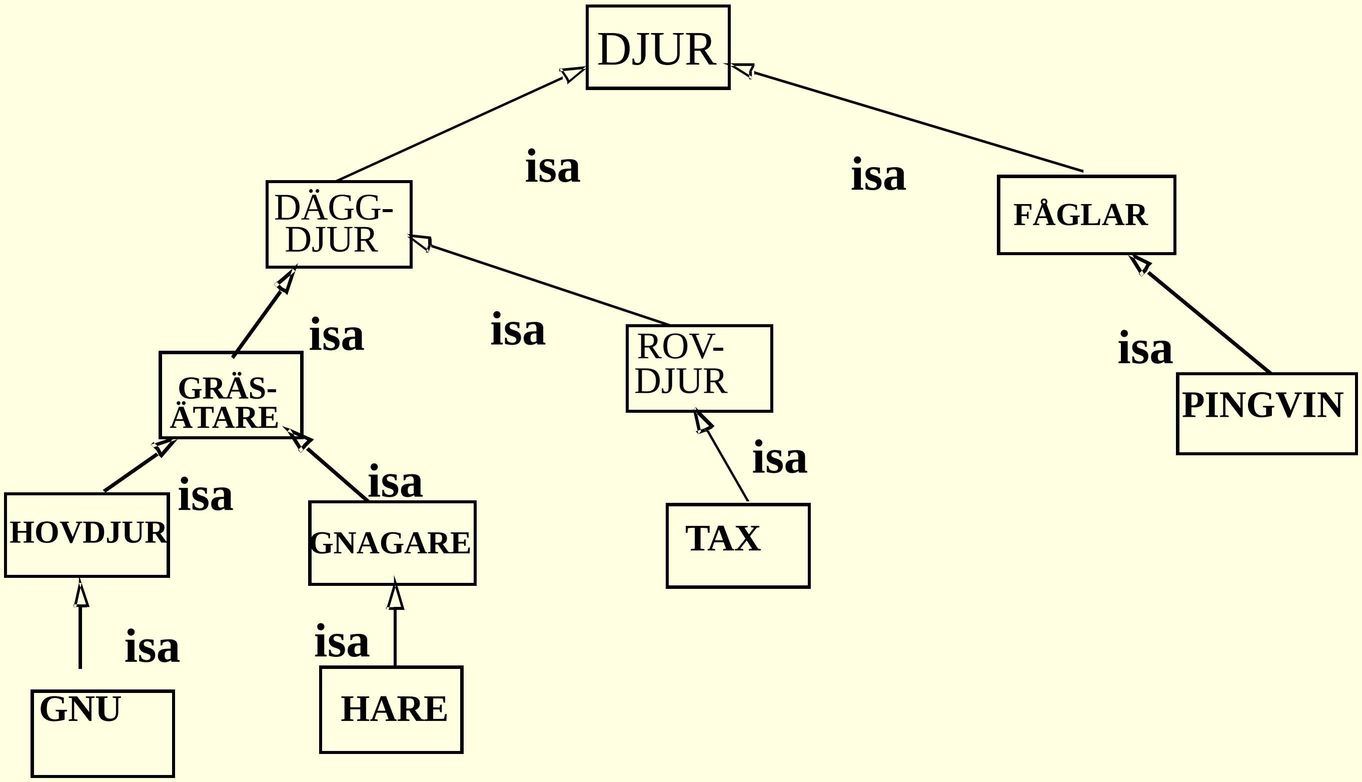
Modelling patterns – Analysis patterns - Data abstractions

Whatever reality one wish to model there will be hierarchical structures. This means we have to catch these structures in our model of the same reality.

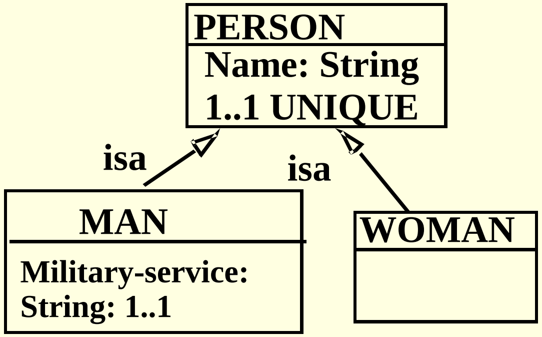


Inheritance cont.

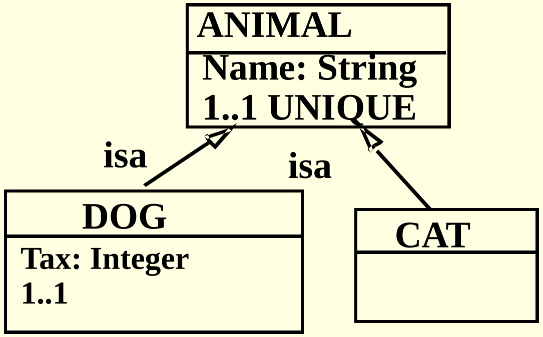
Class schema with so called ’isa’-associations:



Inheritance



MAN and WOMAN are mutually exclusive and exhaustive wrt PERSON.

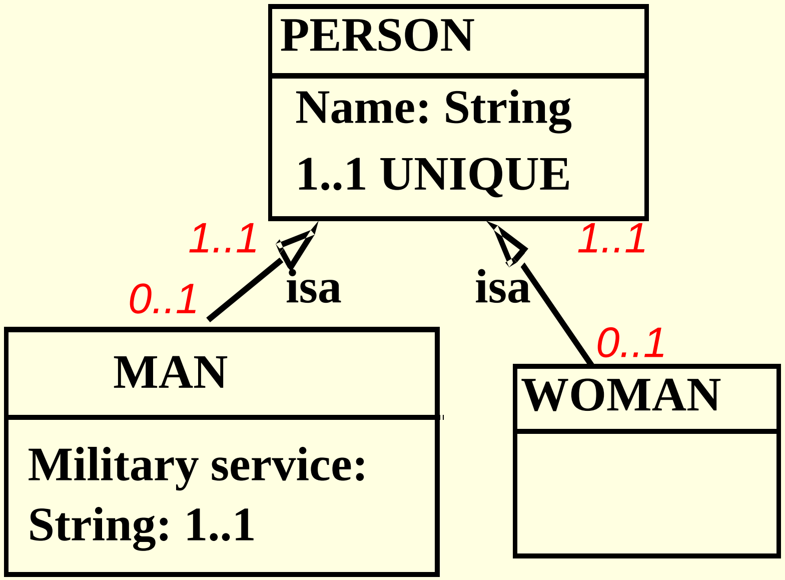


CAT and DOG are mutually exclusive but NOT exhaustive wrt PERSON.

An inheritance hierarchy consists of super and sub-types. The sub-type is a subset of the supertype. If the sub-types cover the entire super-type the subtypes are said to be exhaustive with respect to the super-type. If one and the same instance cannot be part of more than one sub-type the sub-types are said to be mutually exclusive.

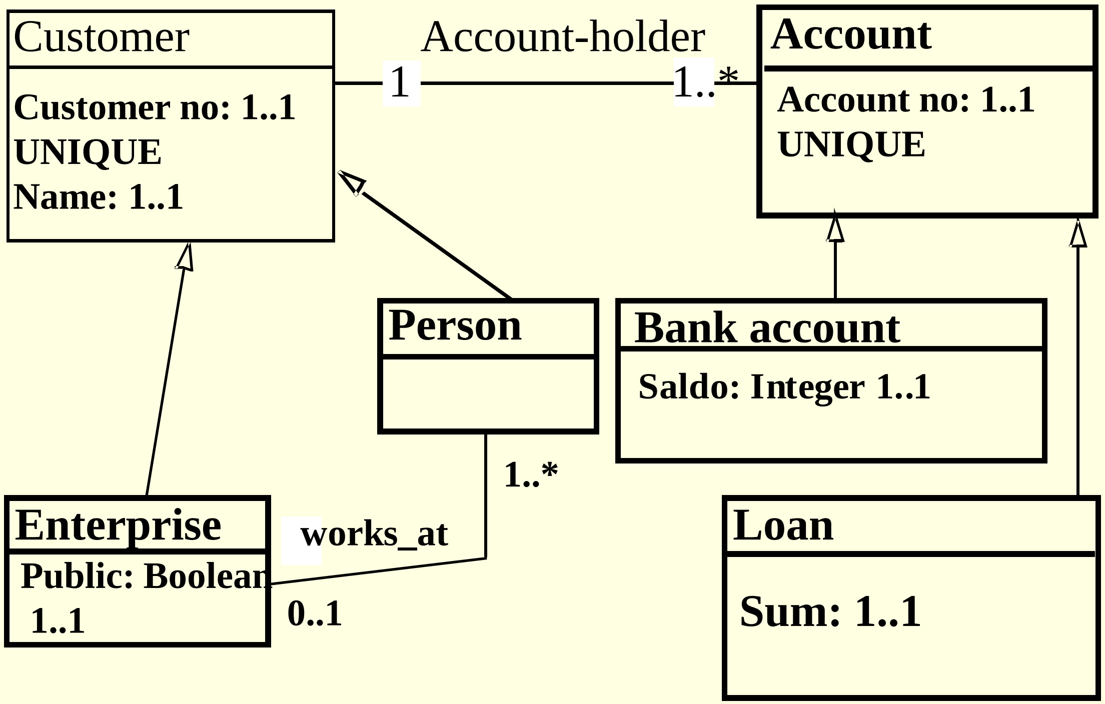
Partial attribute = an attribute with minimum value = 0
Total attribute = an attribute with minimum value > 0

Inheritance



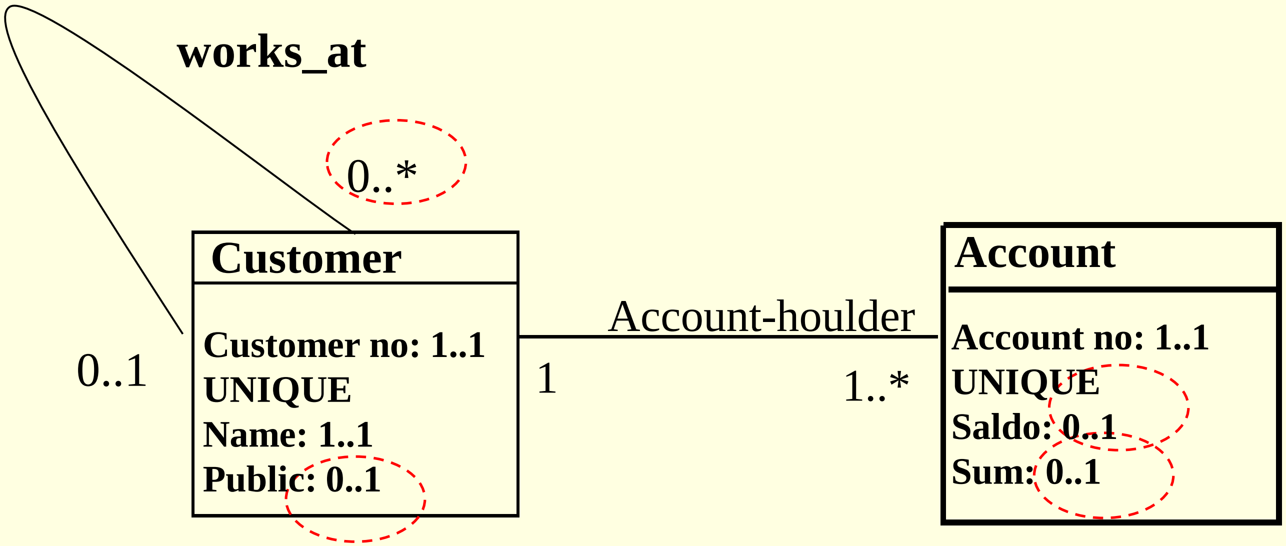
What are the multiplicities for an isa-relation?

Inheritance



What would the schema look like if isa-relations were not used?

Inheritance


works_at

0..*

0..1

Customer

Customer no: 1..1
UNIQUE
Name: 1..1
Public: 0..1

Account-houlder

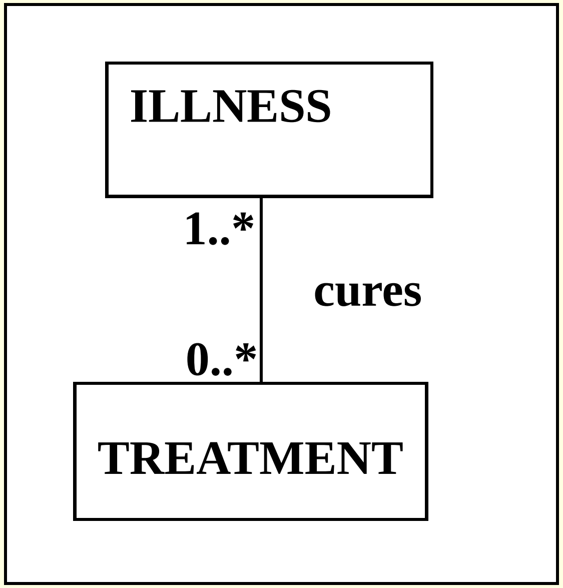
1

1..*

Account

Account no: 1..1
UNIQUE
Saldo: 0..1
Sum: 0..1

Reification



ILLNESS

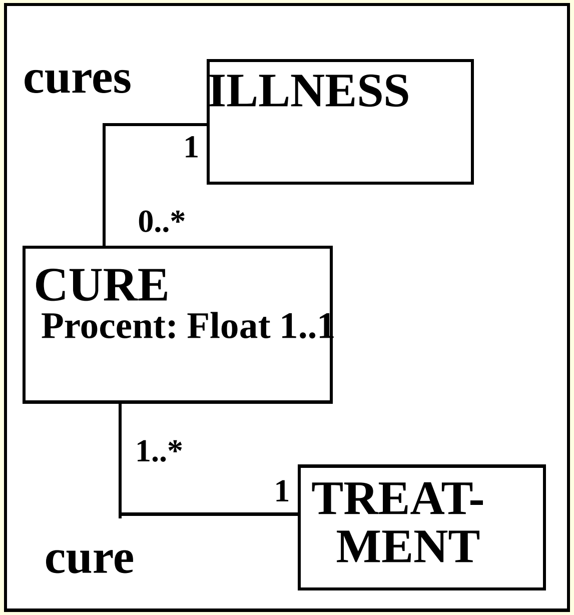
1..*

cures

0..*

TREATMENT

blir:



cures

ILLNESS

1

0..*

CURE

Procent: Float 1..1

1..*

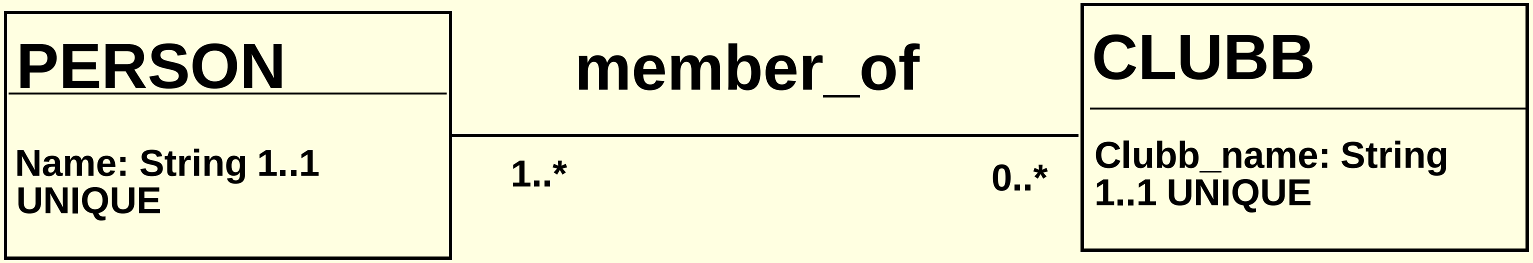
1

TREAT-MENT

cure

The association “cures” is M:M. If we want to store information about the relation “cures” it must be reified, i.e. made into a class. Other M:M-associations may be left as they are at the modelling level but will be turned into tables in a DBMS!

Excercise reification



PERSON

Name: String 1..1 UNIQUE

member_of

1..*

0..*

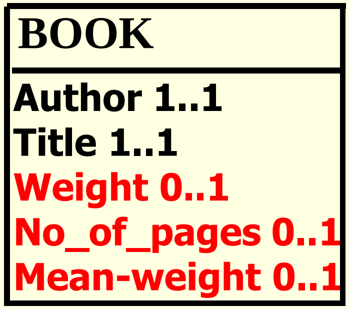
CLUBB

Clubb_name: String 1..1 UNIQUE

Extend the conceptual schema above so that it can represent that a certain person entered a certain clubb a certain date!

Template-copy

- How is the word ”bok” used in the sentences below?
  - Jules Verne skrev många böcker, en heter ’Kapten Grants barn’
  - Biblioteket i Vällingby har många böcker
  - Om ’Kapten Grants barn’ trycks i A4-format blir antalet sidor i en tryckt bok 100 och den väger 150 gram.
  - Bokhandlaren i Jönköping sålde många böcker



BOOK

Author 1..1
Title 1..1
Weight 0..1
No_of_pages 0..1
Mean-weight 0..1

Template-Copy structures (power types, ”category classes”)

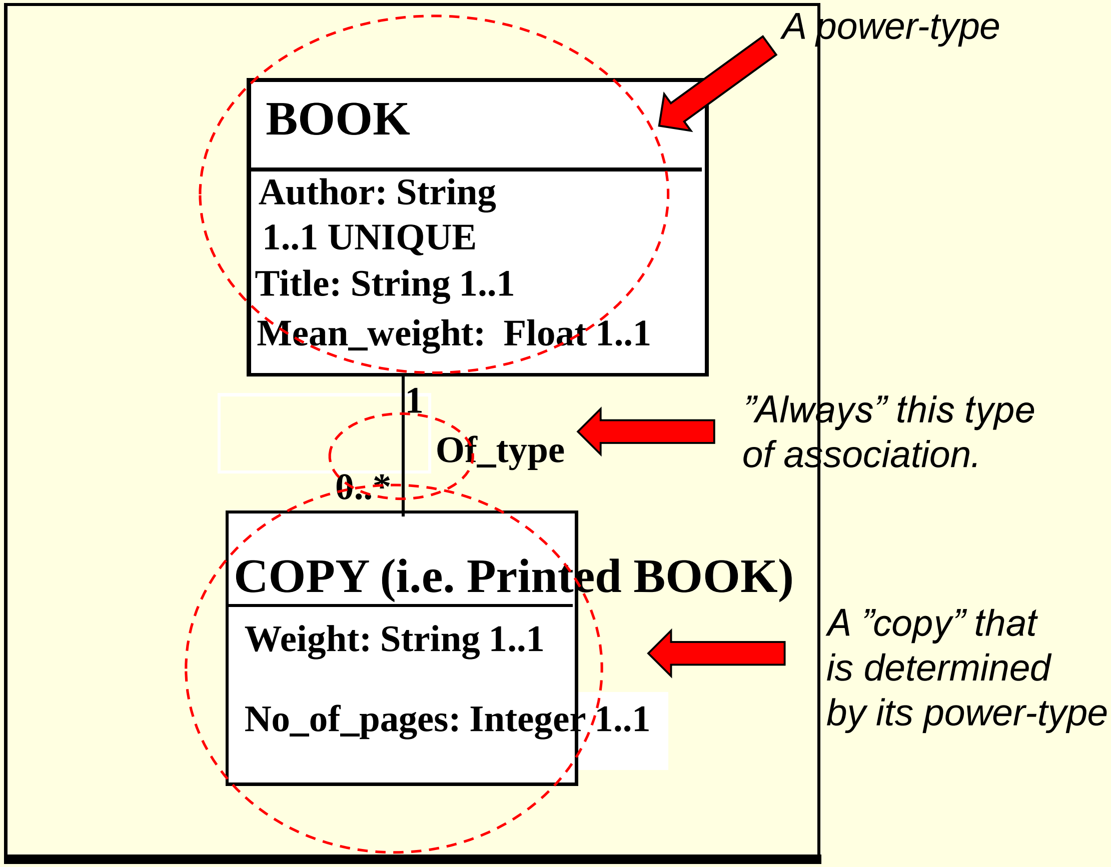
Certain objects may be seen as templates for other objects - copies.

A template describes the general features of the copy - the copy may contain other features.

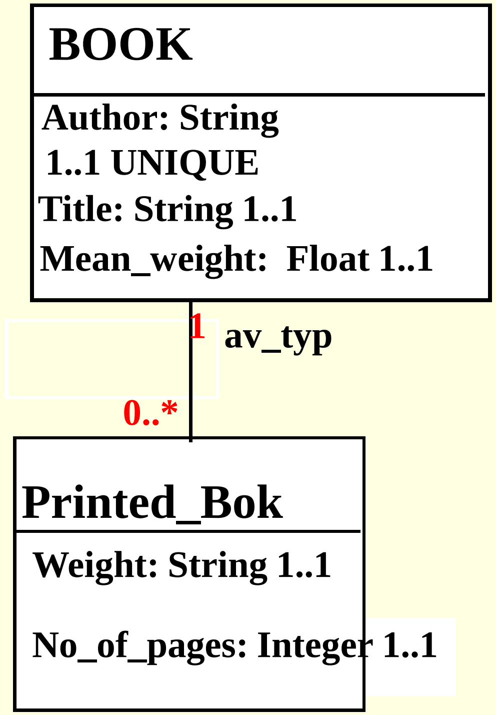
Templates (power types) often model abstract phenomena while copies more often model concrete dittos.

BOOK is a typical example of a template , the book as a piece of art. The BOOK has a title, an author etc. The individual copies are the physical printed copies of the book. A copy has attributes of its own such as weight, number of pages etc.

OBS, a COPY is NOT 'isa' a BOOK! Template-copy relationships are not 'isa'-relationships.



Template-Copy structures (power types, ”category-klasses”) compared to inheritance:



BOOK

Author: String 1..1 UNIQUE

Title: String 1..1

Mean_weight: Float 1..1

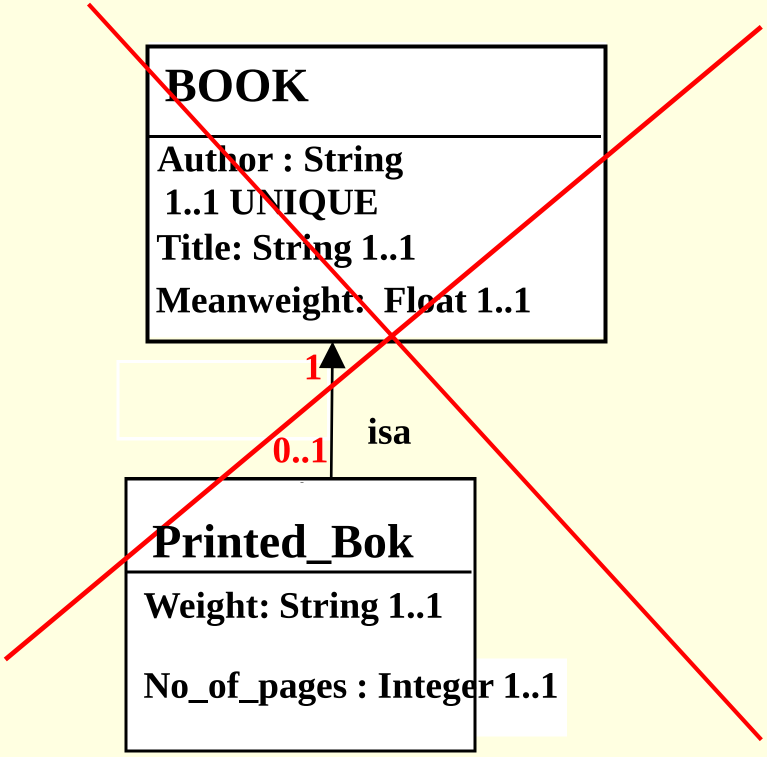
1 av_typ 0..*

Printed_Bok

Weight: String 1..1

No_of_pages: Integer 1..1

VS



BOOK

Author : String 1..1 UNIQUE

Title: String 1..1

Meanweight: Float 1..1

1 isa 0..1

Printed_Bok

Weight: String 1..1

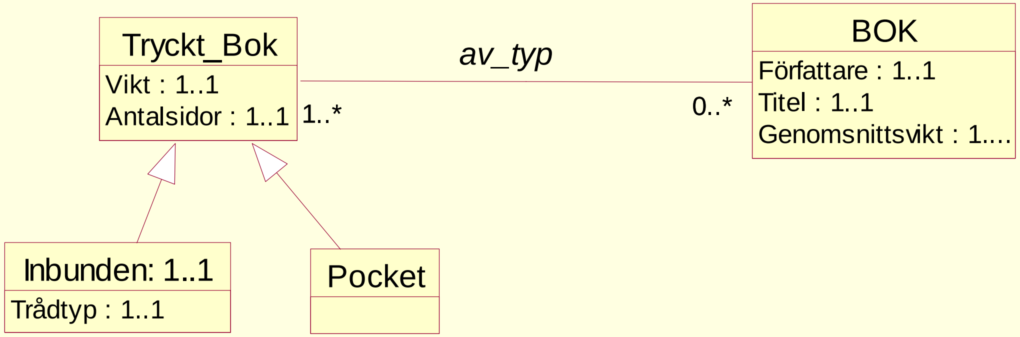
No_of_pages : Integer 1..1

Multiplicity as measure of model quality?

Rule of thumb: All attributes in a class should be total (= min value 1)

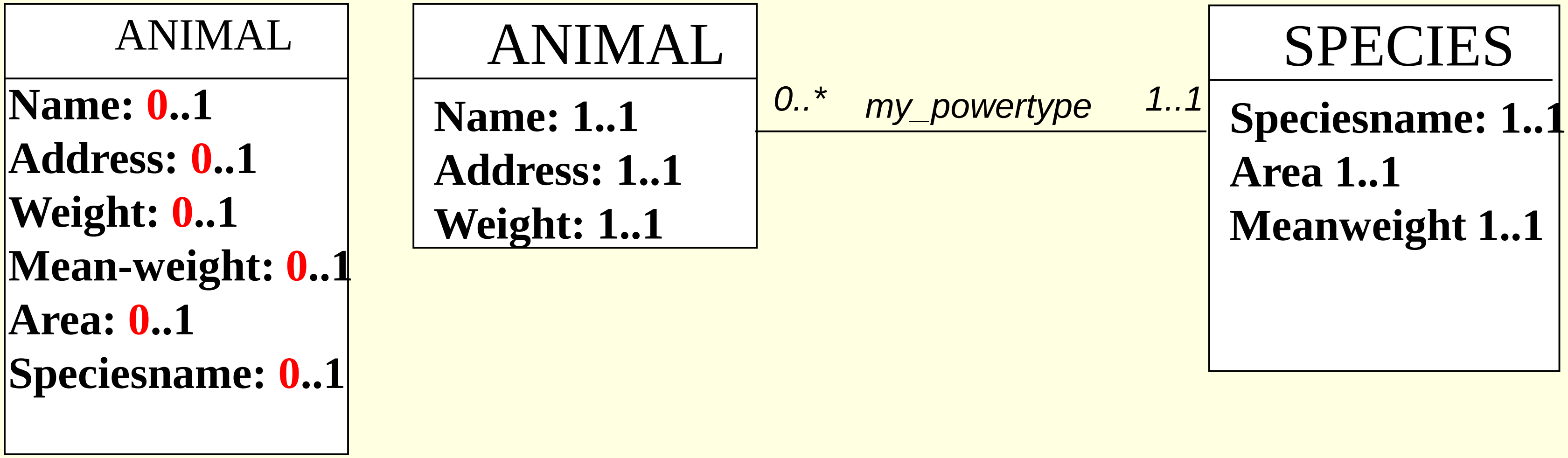
A large number of partial attributes (i.e. attributes with minumum value = 0) means that not all instances of the class has a value on said attributes, which **may** mean that **several** different phenomena are modelled in **one and the same class**. This may be remedied by using inheritance (when there is several subsets of one and the same phenomenon) or via power types (when the same phenomenon exists on different levels of abstraction).

Author 1..1
Title 1..1
Weight 0..1
No_of_pages 0..1
Mean-weight 0..1
Binding: 0..1



Power-types – yet another example

Rule of thumb: All attributes in a class should be total (= min value 1)



ANIMAL

Name: 0..1
Address: 0..1
Weight: 0..1
Mean-weight: 0..1
Area: 0..1
Speciesname: 0..1

ANIMAL

Name: 1..1
Address: 1..1
Weight: 1..1

*0..** *my_powertype* *1..1*

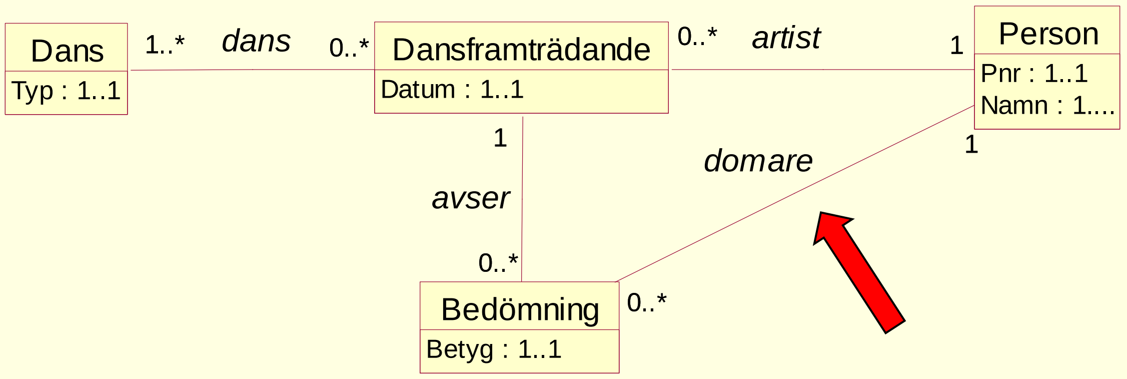
SPECIES

Speciesname: 1..1
Area 1..1
Meanweight 1..1

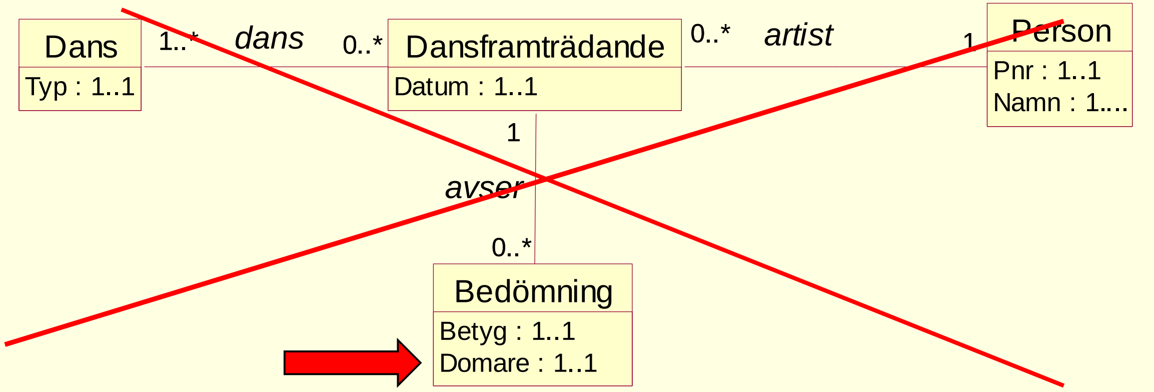
Classes are related via associations– NOT via attributes!

Compare the following two UML-diagram:

1.



2.



Summary: modelling mechanisms

- Classificaiton – from domain phenomenon to classes including attributes and associations
- Generalization – inheritance
- Reification
- Power-type – copy categorizations
- More? Yes!