Graz University of Technology

Object-Oriented Analysis & Design (OAD)

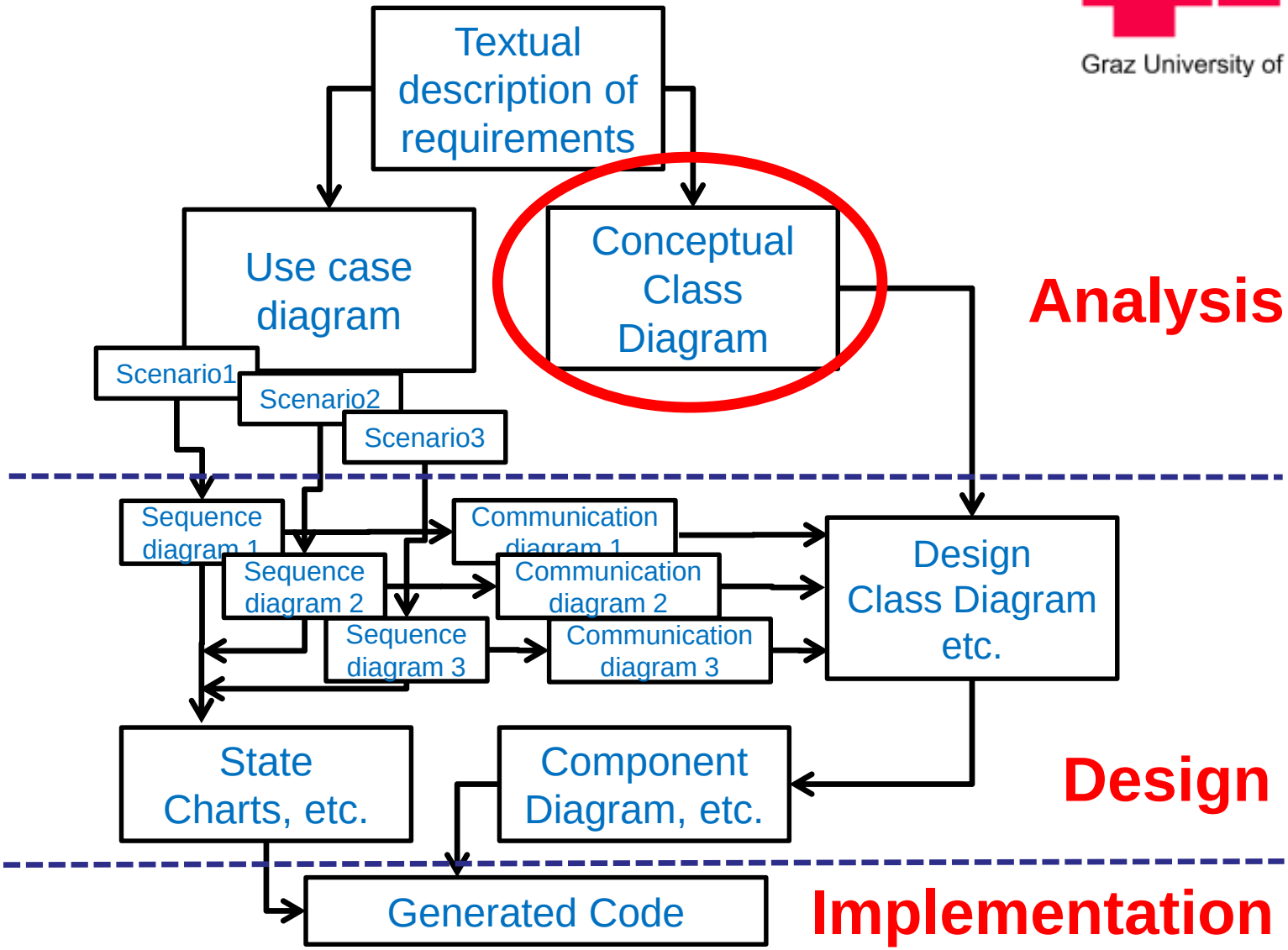
Class Diagrams

https://youtu.be/Gz29E1bMUGo

Alexander Felfernig
Institute for Software Technology
Inffeldgasse 16b/2

„Big Picture“

Graz University of Technology



Classes

Graz University of Technology

- **class**: a collection of well-distinguishable entities (objects) of our thinking
- **for example**: students, employees, hotels, invoices, products, countries, …
- **attributes**: classes have attributes that are used to describe each entity of the class, for example, name, address, #stars, …
- attributes have a **datatype**, for example, name:string, address: string, #stars:int[1..5], …

Objects

Graz University of Technology

- **object**: a uniquely identifiable entity of the collection described by a class
- **for example**:
  - class: *hotel* (name, #starts, address)
  - object 1: (“Hotel Post”, 4, “Dienten/Salzburg”)
  - object 2: (“Hotel Linden”, 3, “Innsbruck/Tirol”)
  - object 3: (“Plachutta”, 4, “Linz/OÖ”)

Associations

Graz University of Technology

- **association**: describes relationships between the objects of (different) classes
- representation in UML:

Multiplicities

Graz University of Technology

- **multiplicity**: further specification of a relationship in terms of the number of allowed/required connections
- **no multiplicity** means “exactly one”
- representation in **UML**:

Indicator	Meaning
0..1	Zero or one
1	One only
0..*	Zero or more
1..*	One or more
n	Only *n* (where *n* > 1)
0..n	Zero to *n* (where *n* > 1)
1..n	One to *n* (where *n* > 1)

Multiplicities

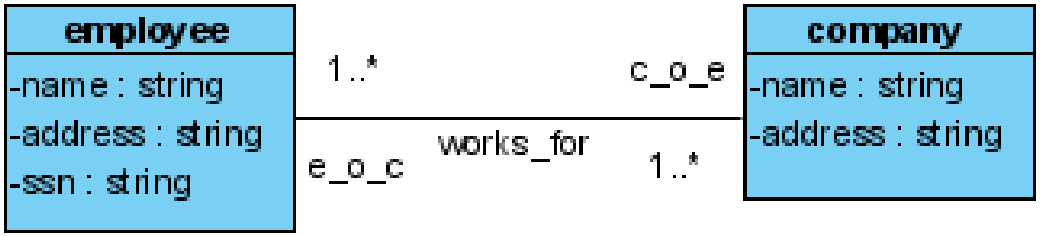
Graz University of Technology

Indicator	Meaning
0..1	Zero or one
1	One only
0..*	Zero or more
1..*	One or more
n	Only *n* (where *n* > 1)
0..n	Zero to *n* (where *n* > 1)
1..n	One to *n* (where *n* > 1)

Roles

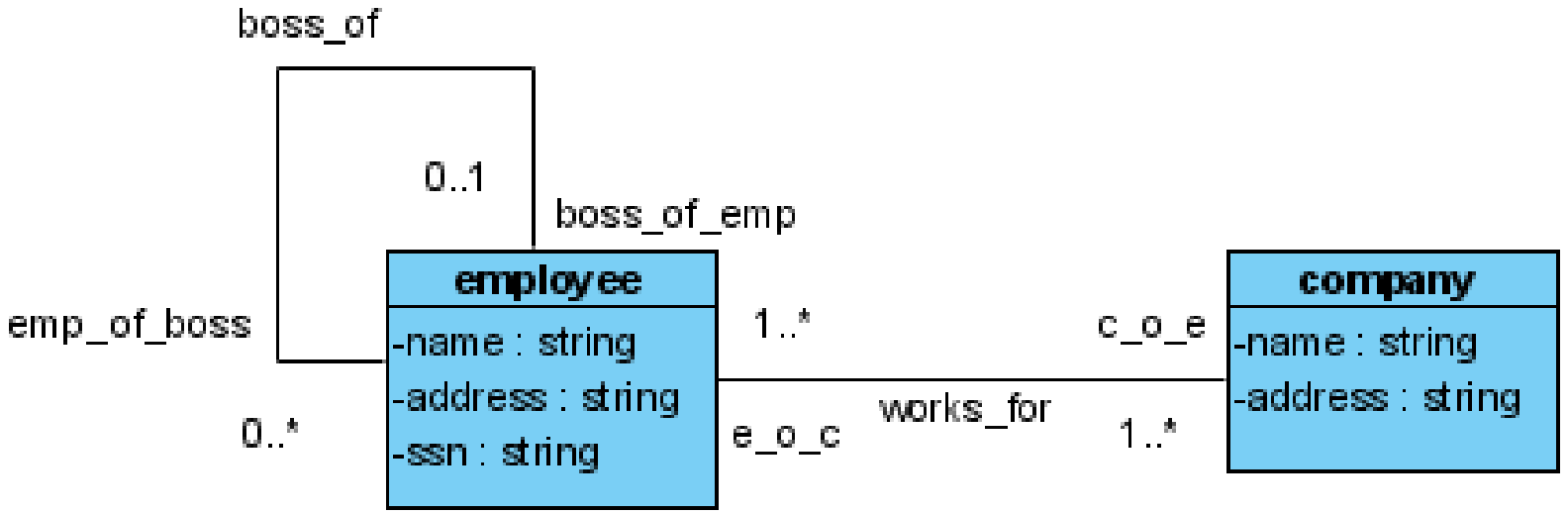
Graz University of Technology

- **role**: further specification of a relationship in terms of the “role” of an object
- representation in **UML**:
  - company of employee: “c_o_e”
  - employee of company: “e_o_c”



Roles

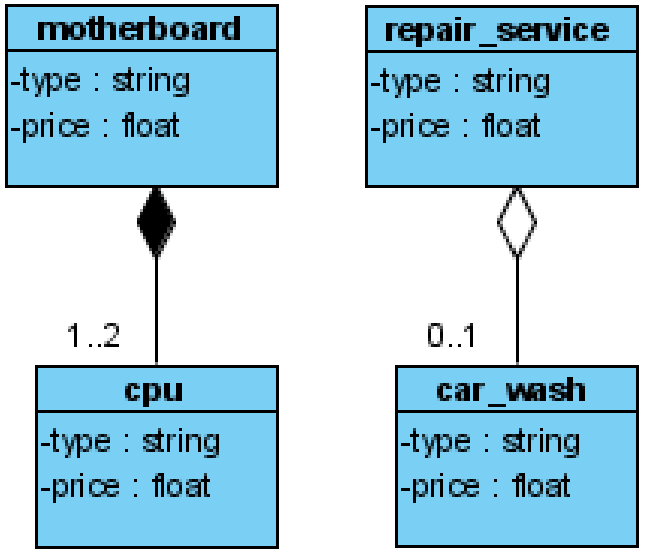
- each employee has 0..1 boss („emp_of_boss“)
- each employee is boss of 0..* employees („boss_of_emp”)



Aggregations

Graz University of Technology

- **aggregations**: part-whole relationships between objects
- **specific type of association**
- **for example**: each motherboard consists of 1..2 CPUs



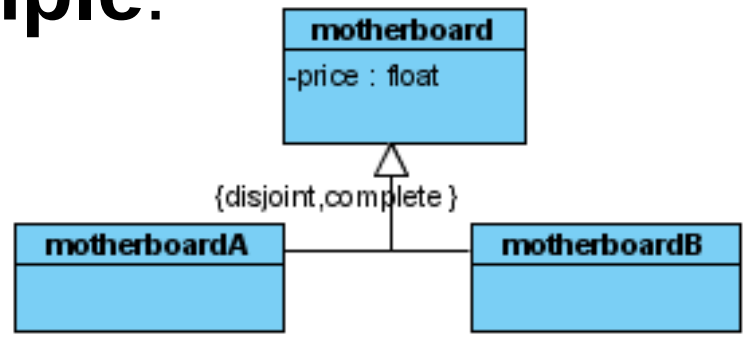
physical containment (“composite”)

parts can exist independently of whole (“shared”)

Generalizations

Graz University of Technology

- **generalization**: taxonomic relationship between a more general element and a more specific element. The more specific element is fully consistent with the more general element and may contain additional information.
- **for example**:



Building Class Diagrams: Noun Method

Graz University of Technology

- **underline all nouns** in requirements document
- identify **important terms as classes**
- **advantage**:
  - simple
  - developer thinks in terms of the customer
- **disadvantage**:
  - human language imprecise, depends on writing quality

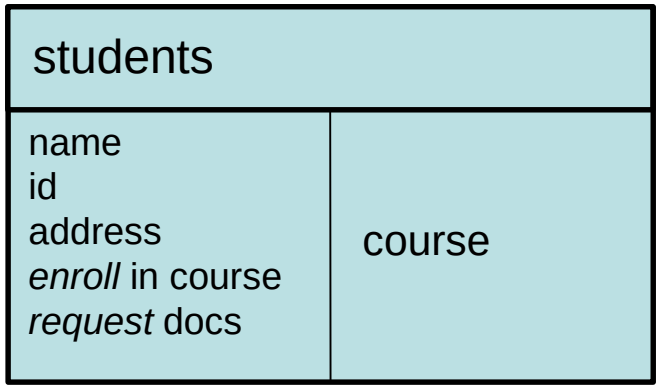
Noun Method

Graz University of Technology

"Computers are managed in an enterprise network. From each computer we know the serial number and the type. Each computer is located in exactly one office room and each room has one or more network connections."

Building Class Diagrams: CRC Cards

Graz University of Technology



- **CRC Card**
  - **c**lass (collection of similar objects)
  - **r**esponsibilities (something the class knows or does)
  - **c**ollaborators (other classes the class interacts with)
- **Process**
  - find classes
  - find responsibilities
  - define collaborators
  - move cards around

CRC Cards in Visual Paradigm

Graz University of Technology

Student

Super Classes:

Sub Classes:

Description: Students

Attributes:

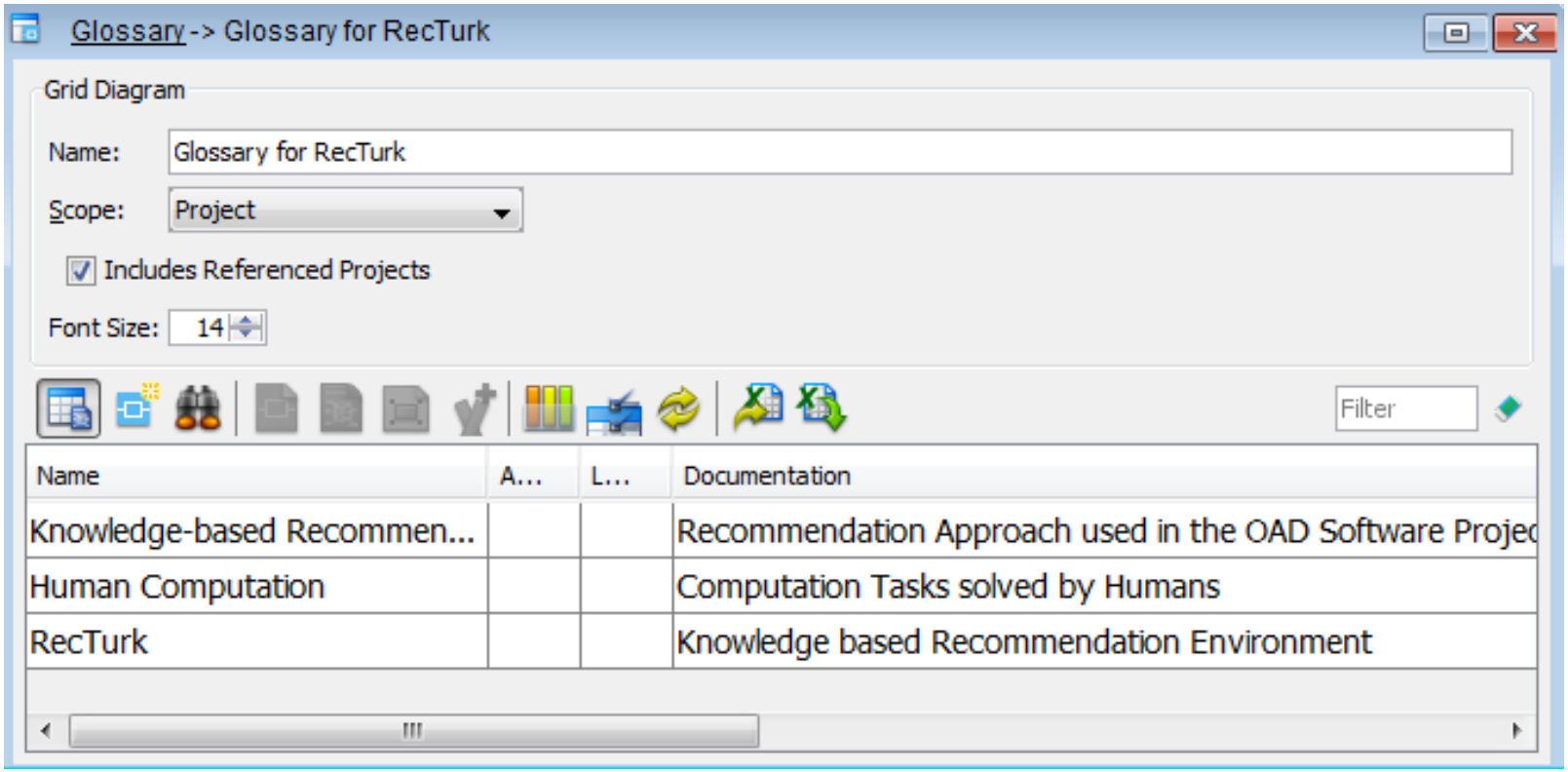
Name	Description
name	student name
id	student id
address	address of the student

Responsibilities:

Name	Collaborator
enroll in course	course
request docs	course

Glossary in Visual Paradigm

Graz University of Technology

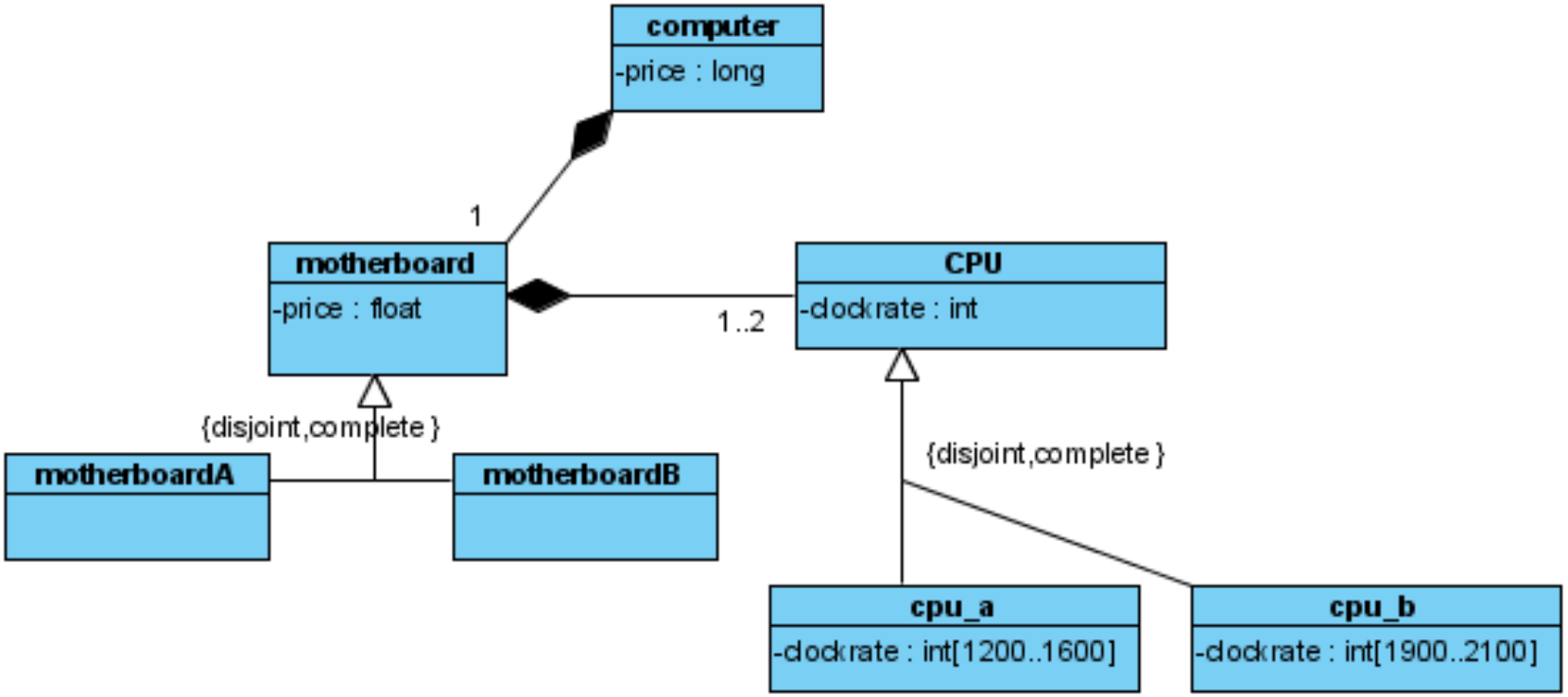


Additional Constraints

Graz University of Technology

OCL Constraints, e.g.,

context computer: self.motherboard.cpu->size > 1;

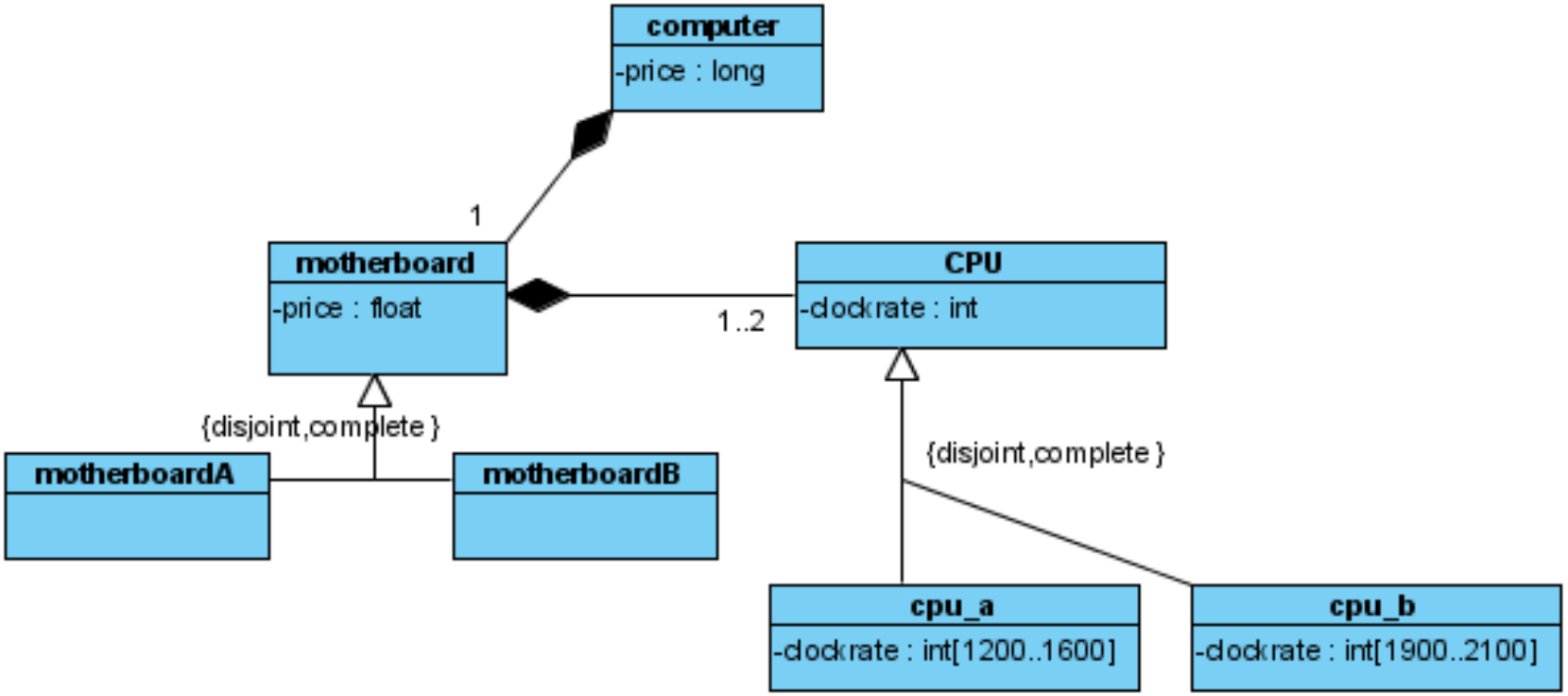


Additional Constraints

Graz University of Technology

context computer:

self.motherboard.cpu->select(isOclTypeOf(cpu_a))->size<2;

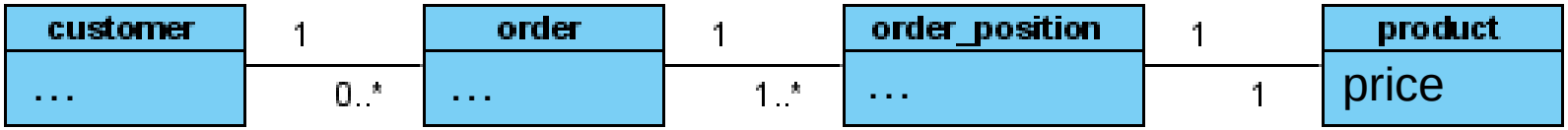


Additional Constraints

Graz University of Technology

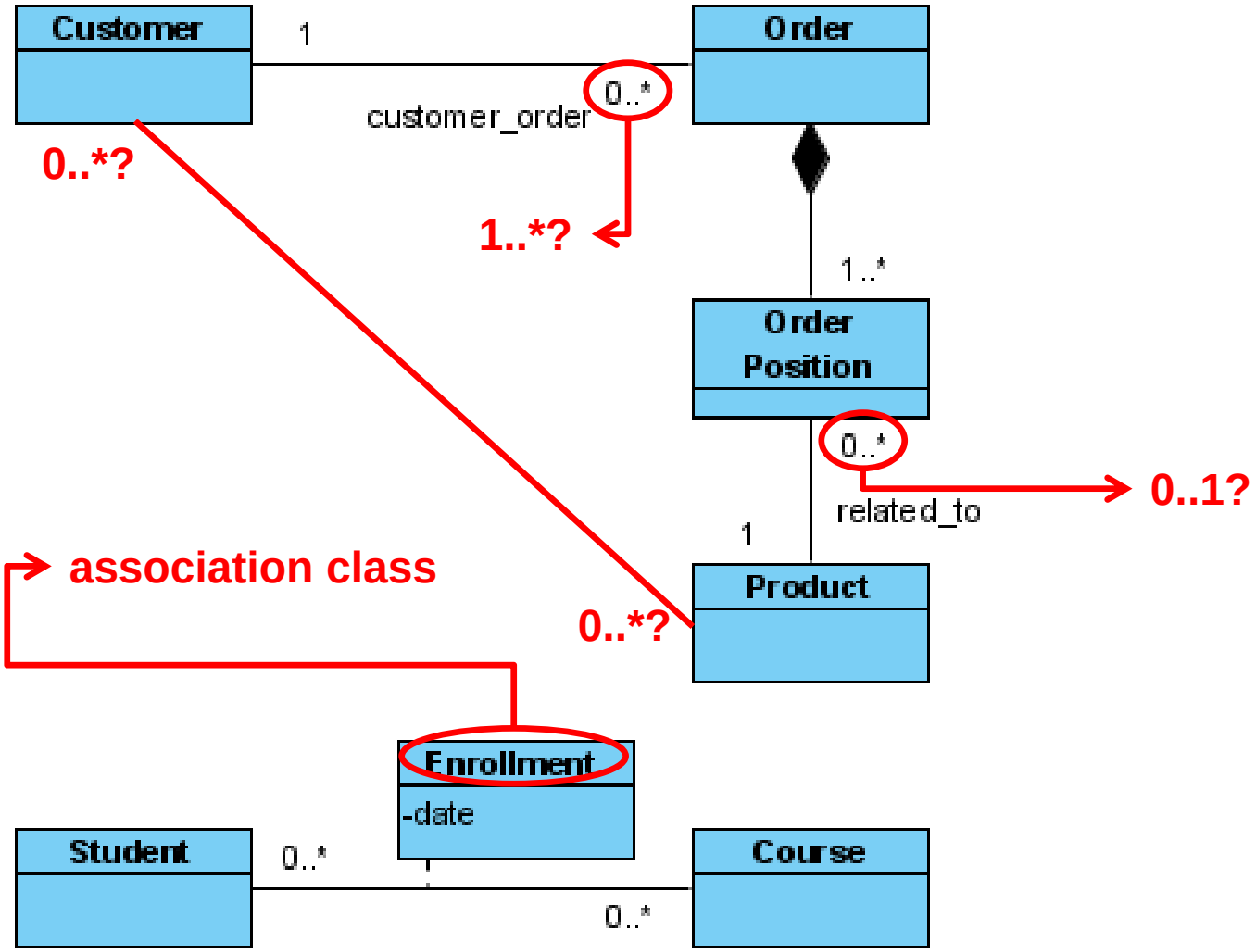
context order:

self.order_position.product.price->sum>500;



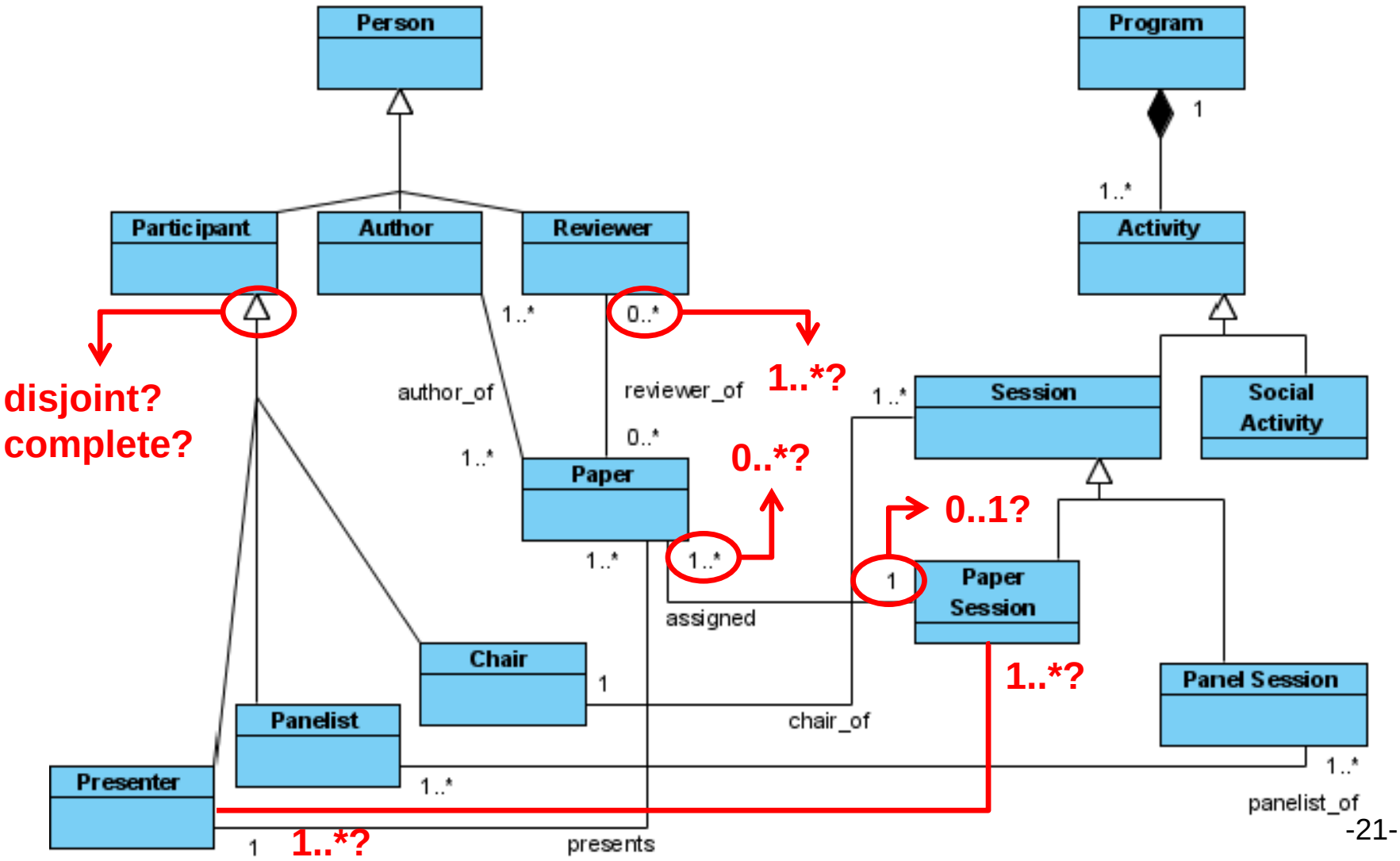
Example Class Diagrams

Graz University of Technology



Example Class Diagram: Conference Administration

Graz University of Technology



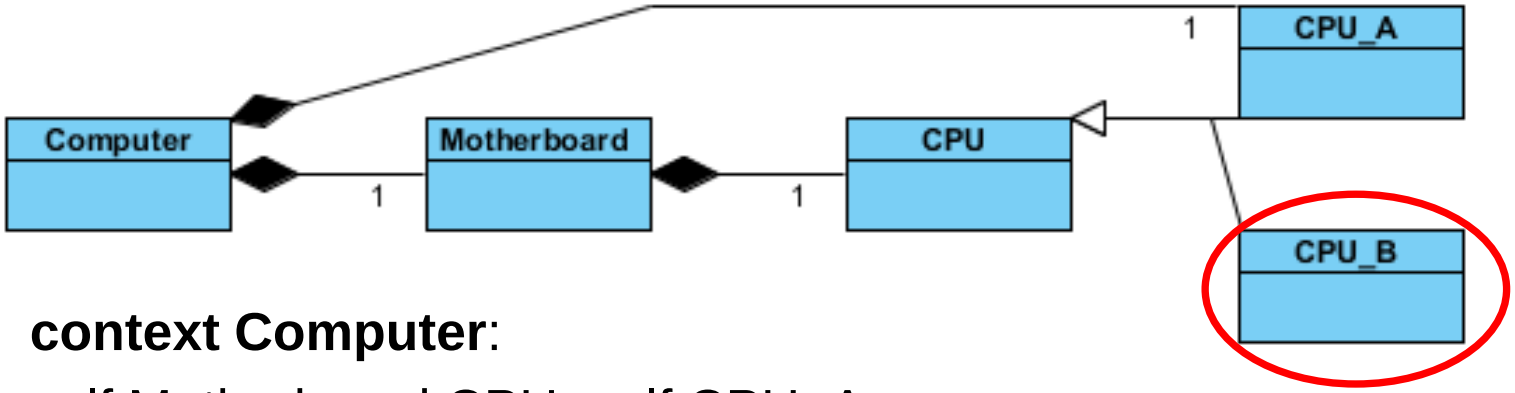
Conference Administration Additional Constraints

Graz University of Technology

- Chairs must not be authors
  - **context Person**: not(self.oclIsTypeOf(Author).and(self.oclIsTypeOf(Chair)))
- Each reviewer must review at least 3 papers
  - **context Reviewer**: self.Paper->size() > 2
- Only authors are allowed to present papers
  - **context Paper**: self.Author->includes(self.Presenter)
- On Nov. 3rd no panel sessions must take place
  - **context Panel Session**: self.date <> “3.11.2013”

Correctness?

Graz University of Technology



context Computer:

self.Motherboard.CPU=self.CPU_A;

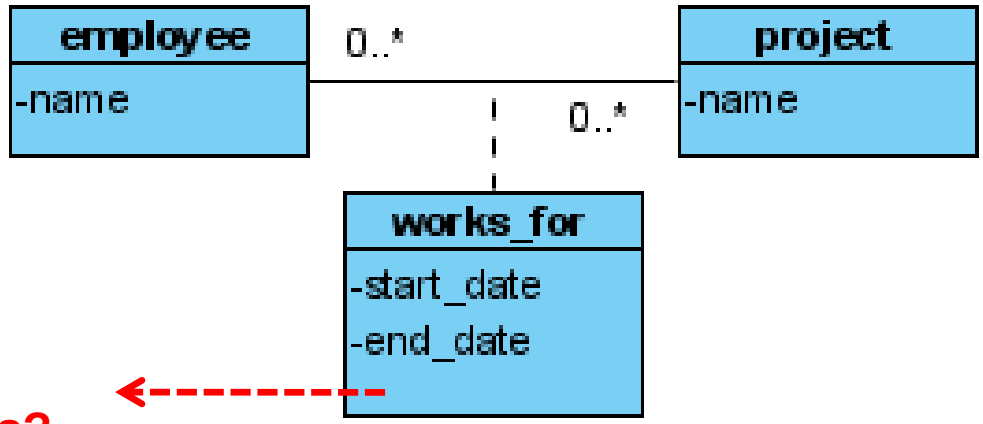
∃solution for each class?

„dead class“

Association Class

Graz University of Technology

- When an association has additional properties, **association classes** are used to model it.
- The association between **project** and **employee** is modeled using the association class **works_for**.



Reification of Associations

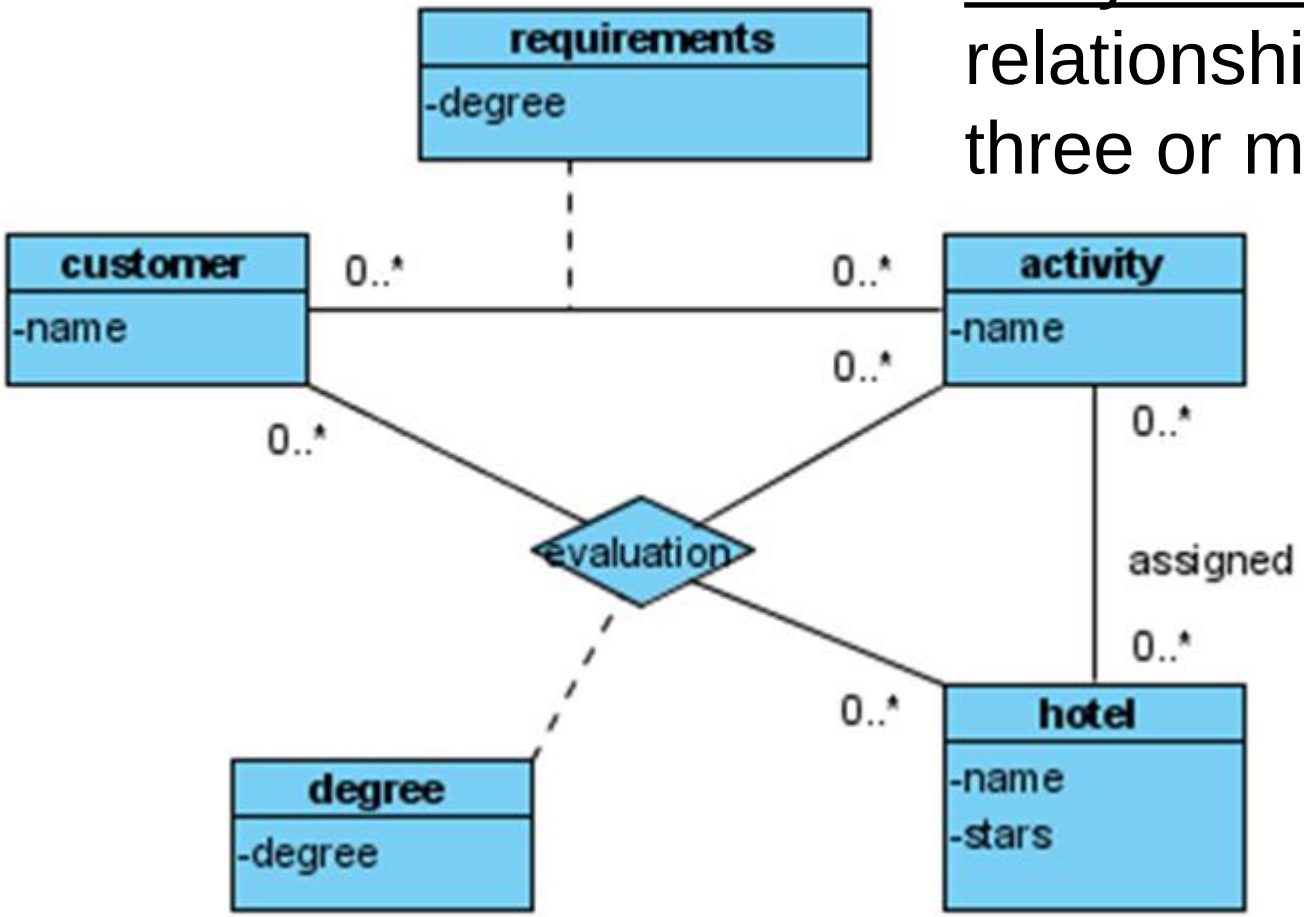
Graz University of Technology

- Reification = „Vergegenständlichung“
- An association can be reified to a class
- The **association class is transformed to a class**, the multiplicities have to be extended accordingly.

Reification of n-ary Associations

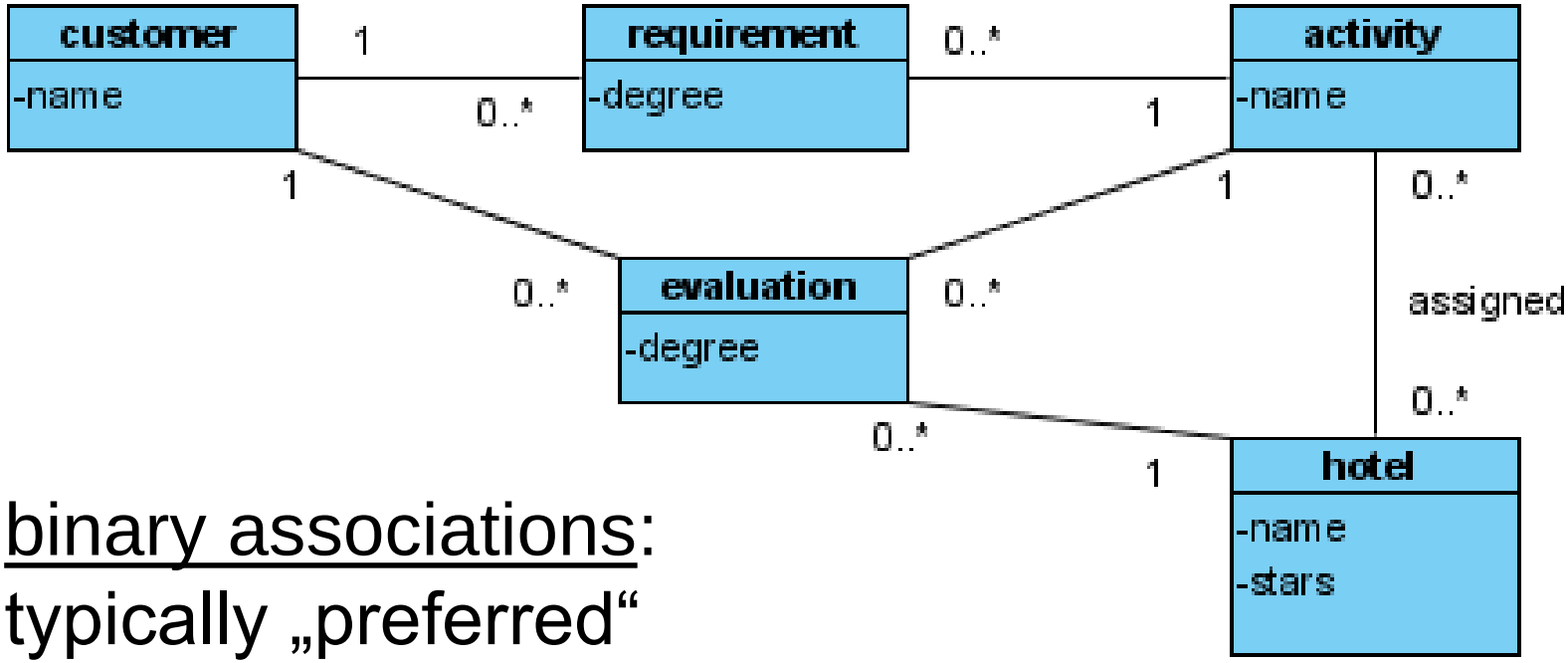
Graz University of Technology

n-ary association: relationship between three or more classes



Reification of n-ary Associations

Graz University of Technology



<u>binary associations</u>:
typically „preferred“
over n-ary associations

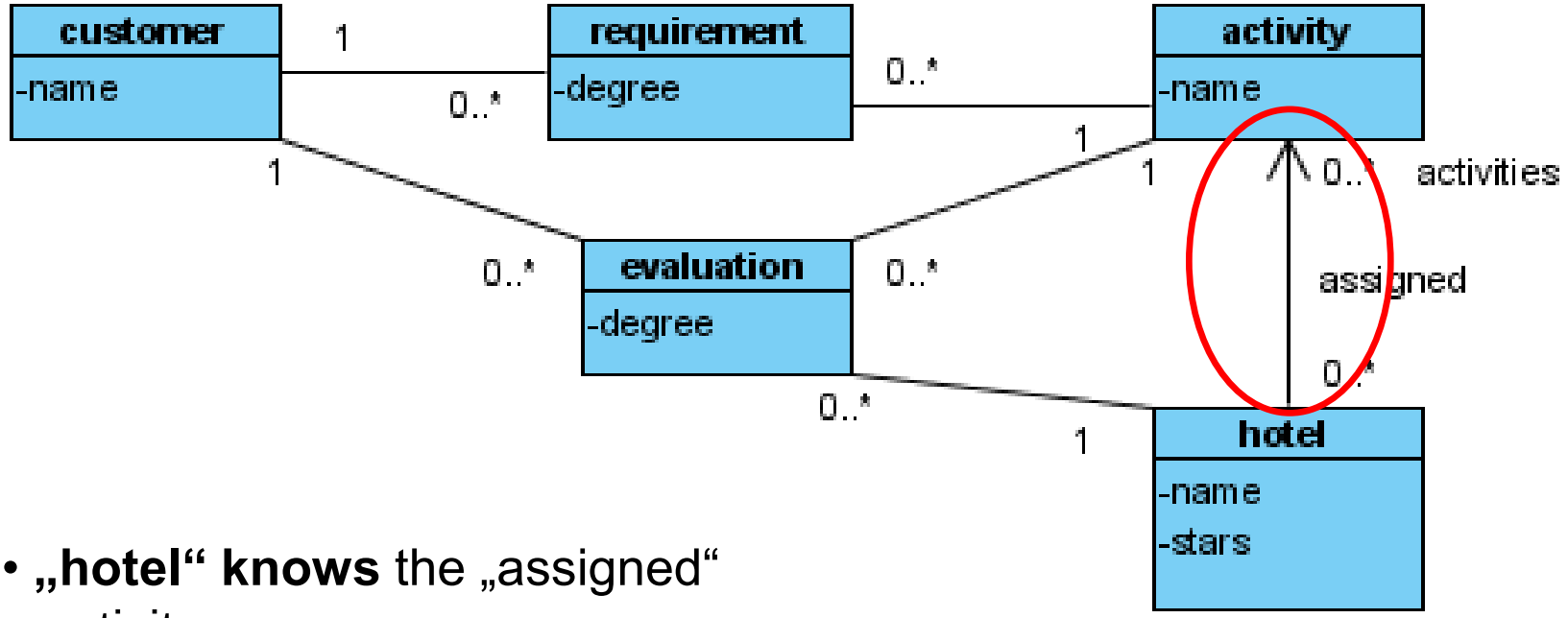
Navigability of Associations

Graz University of Technology

- Navigability: **direction in which an association can be traversed** (navigated)
- An **arrow** indicates the direction of navigability
- Binary associations are **navigable in both directions** (default)

Navigability of Associations

Graz University of Technology

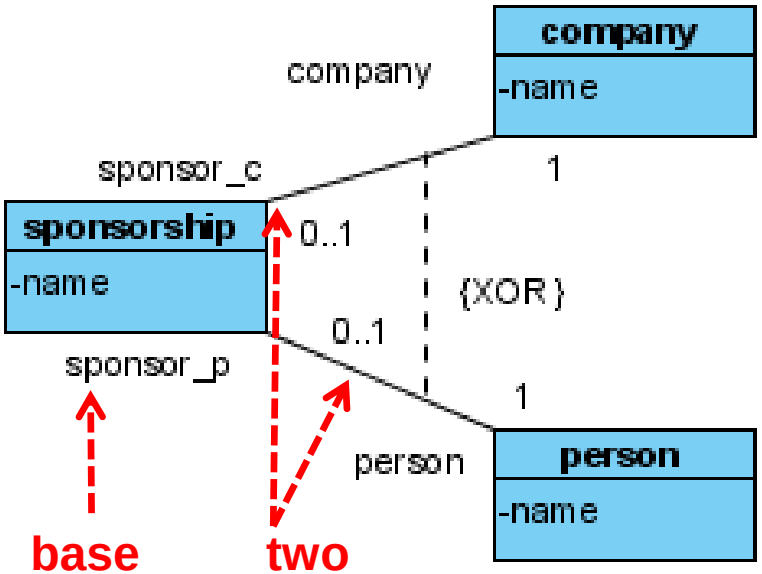


- **„hotel“ knows** the „assigned“ activity
- **„activity“ does not store** information regarding „hotel“

XOR Constraints

Graz University of Technology

- Constraints between **two or more associations connected to a single base class**
- Instances of the base class are allowed to participate **in only one of the associations**



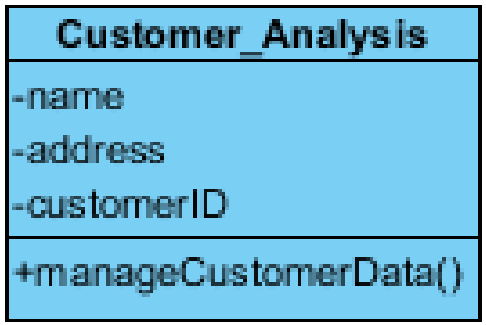
sponsor_c	company
s1	c1
s2	c2
s3	c1

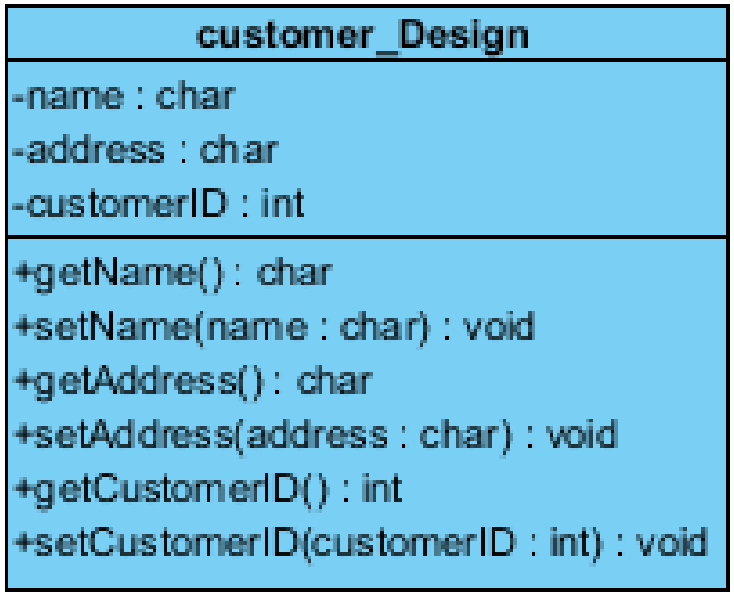
?

sponsor_p	person
s1	p1
s4	p2
s4	p3

Analysis & Design Classes

Graz University of Technology





Class diagrams serve **different purposes**, e.g.:

- understanding requirements (**analysis**)
- describing the **design**

- conceptual class diagram
- class responsibilities

- design class diagram

Class Specification

Graz University of Technology

- **Attribute specification**
  - visibility attributeName : type [multiplicity] = default value
- **Operation specification**
  - visibility operationName (parameterName: Type): returnType
- **Visibility**:
  - **+public** (any element that sees class), **#protected** (in the class + subclasses), **-private**(in the class)
- **Examples**
  - **-name : char = ““**
  - **+getCustomerID():int**

Visual Paradigm Intro

Graz University of Technology

- Class Diagram - Part 1: https://youtu.be/fz3sd-jZgAg
- Class Diagram - Part 2: https://youtu.be/LYJFwxmYVOM
- State Charts: https://youtu.be/x3RMjMMWzoc

Graz University of Technology

Thanks!

ase.ist.tugraz.at
www.felfernig.eu