PXL – IT
42TIN280 Software Analysis - System & System Context – Domain Model

Week 05 – semester 01

Luc Doumen

Nathalie Fuchs

DE HOGESCHOOL MET HET NETWERK

Hogeschool PXL – Elfde-Liniestraat 24 – B-3500 Hasselt
www.pxl.be - www.pxl.be/facebook

Content

- Problems & context of requirements gathering
- Some definitions
  - Object, class structure, attributes, operations
- Definition of a domain model
- Example of a domain & a domain model
- Characteristics and benefits of a domain model
- Definition of a class diagram
- The process of domain modeling
- General steps in domain modeling
- Developing and documenting Domain Models

Problems & context of requirements gathering

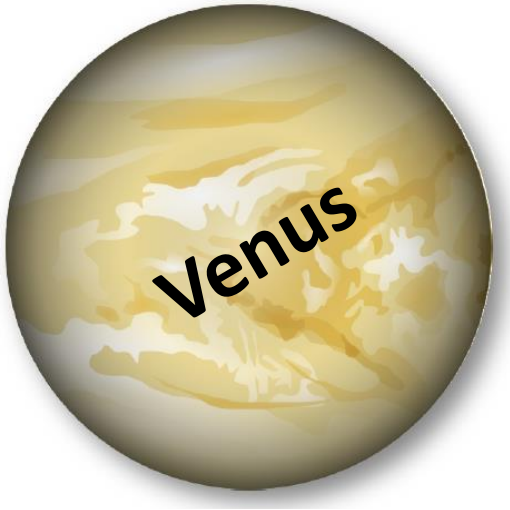
- Unknown knowledge field
- Unknown jargon
- Unknown problem
- Unknown ….

➔ Unknown problem domain

➔ Ambiguous communication

➔ A lot of communication

Problems & context of requirements gathering



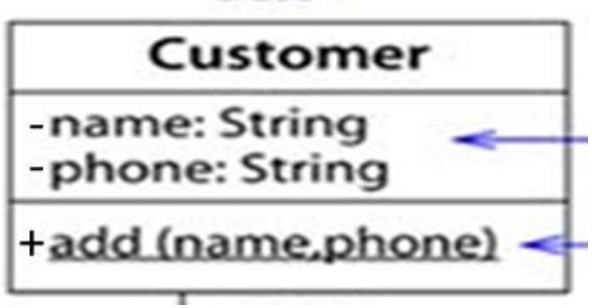
Client

Developers

Other Developers

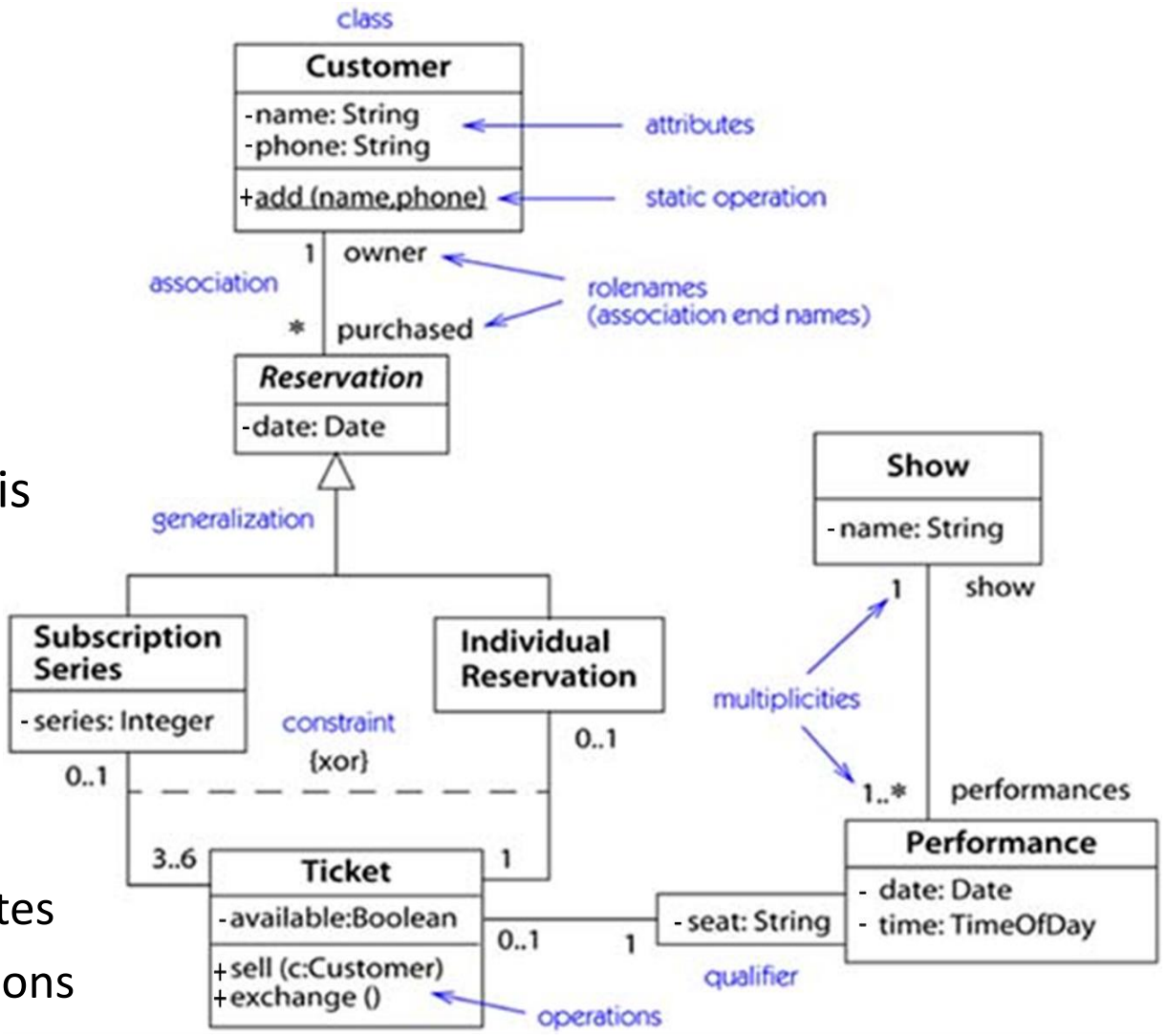
Some definitions (1)

- Definition of Object
  - An object is a *self-contained entity* with well-defined characteristics (properties or attributes) and behaviors (operations)
  - **Example of real-life objects include: School, Teacher, Client, Course, Account, etc.**



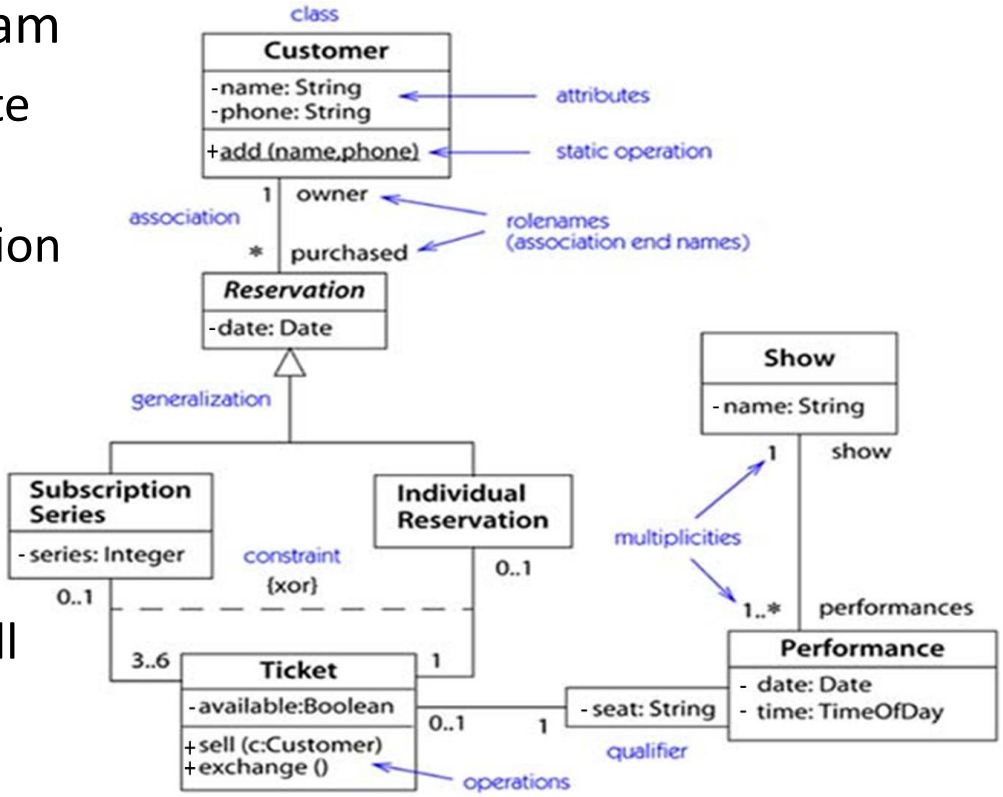
Some definitions (2)

- Class Structure
  - A class is a specification of *a set of objects*, not the actual object
  - In UML, a class is represented by a rectangle divided into 3 parts
    - class name
    - list of attributes
    - list of operations



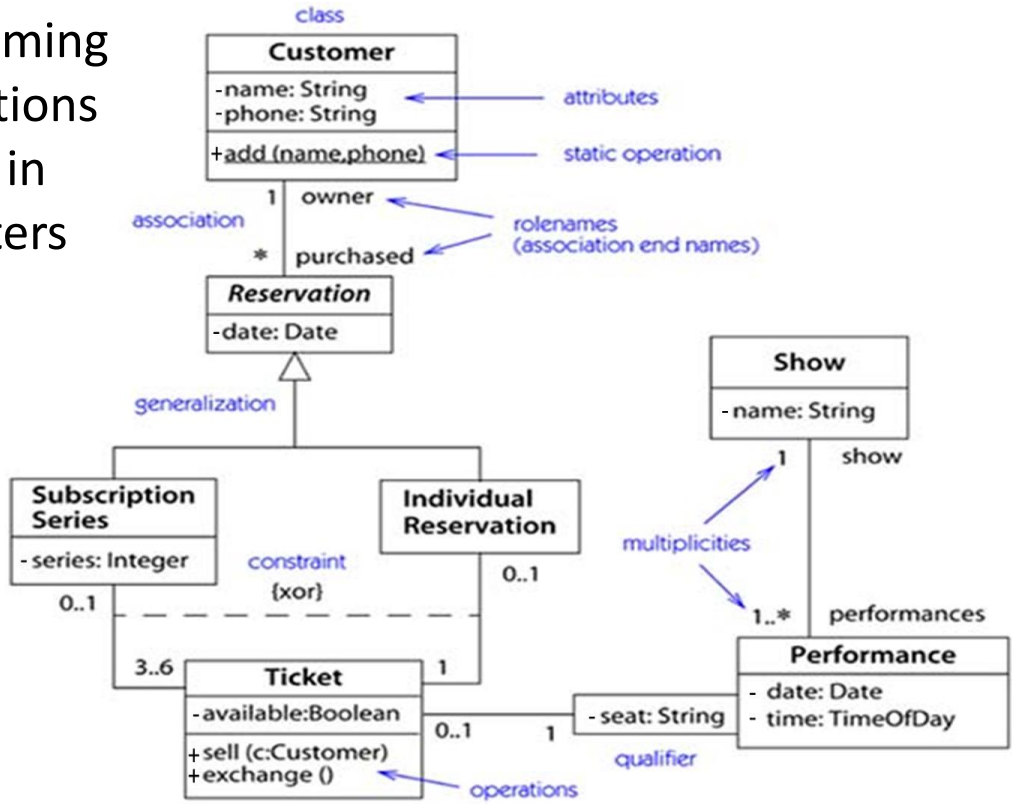
Some definitions (3)

- Class Structure (continued)
  - Only attributes and operations relevant to the current context will be shown in a diagram
    - "-" denotes an attribute within that class
    - "+" denotes an operation within that class
  - Attribute
    - Refer to properties that define the class
    - E.g. class **Customer** will have attributes name and phone
    - E.g. class **Performance** will have attributes date and time

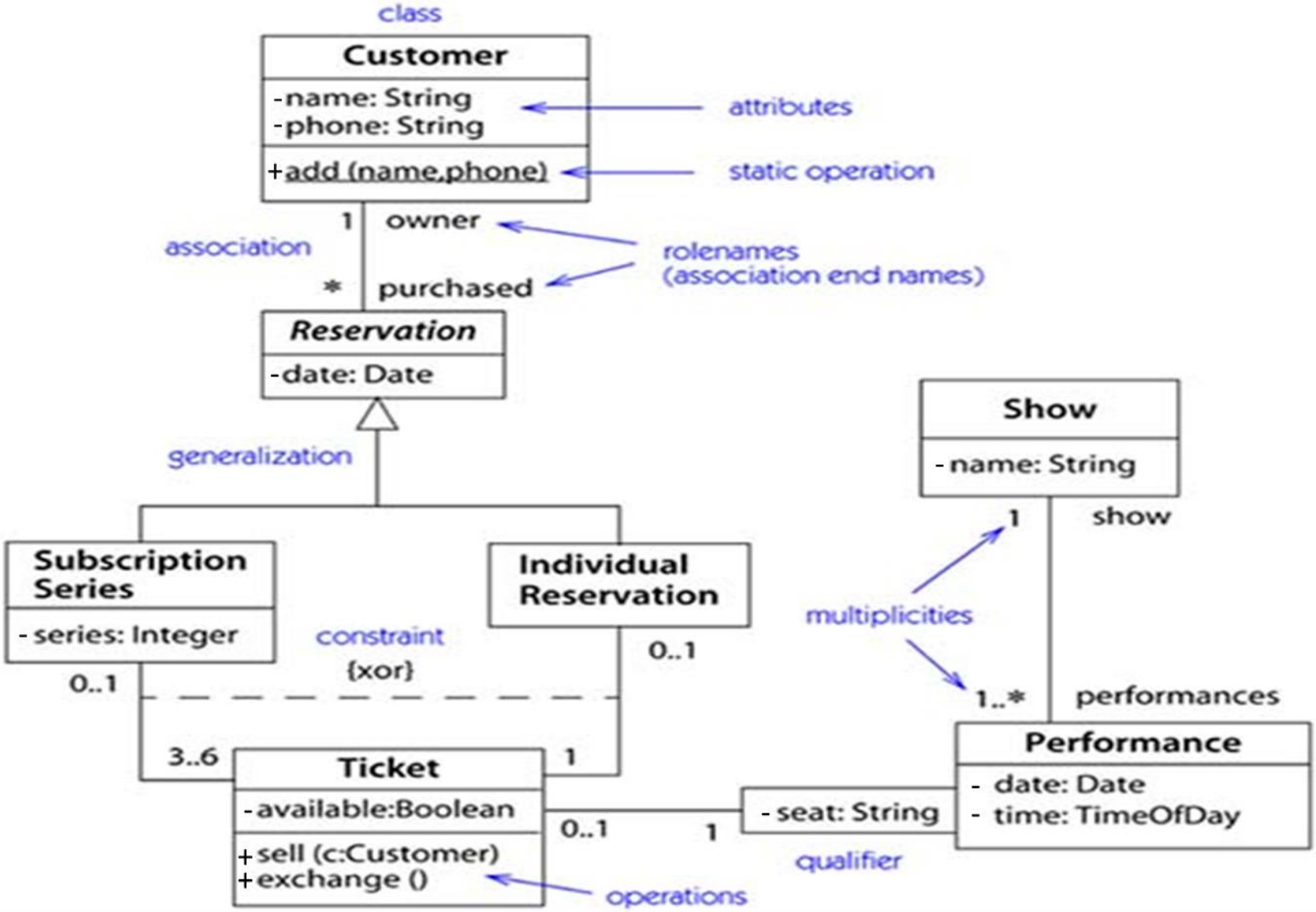


Some definitions (4)

- Class Structure (continued)
  - Operations
    - Functions which can be performed and related to a specific class
    - Also commonly known as methods
    - Expandable to programming language, where operations are similar to functions in that they have parameters and return values.
    - E.g.: class **Customer** may have operation: add(name, phone)
    - E.g.: class **Ticket** has operations: sell(c:Customer) and exchange()



Some definitions (5) - overview



Definition of a domain model

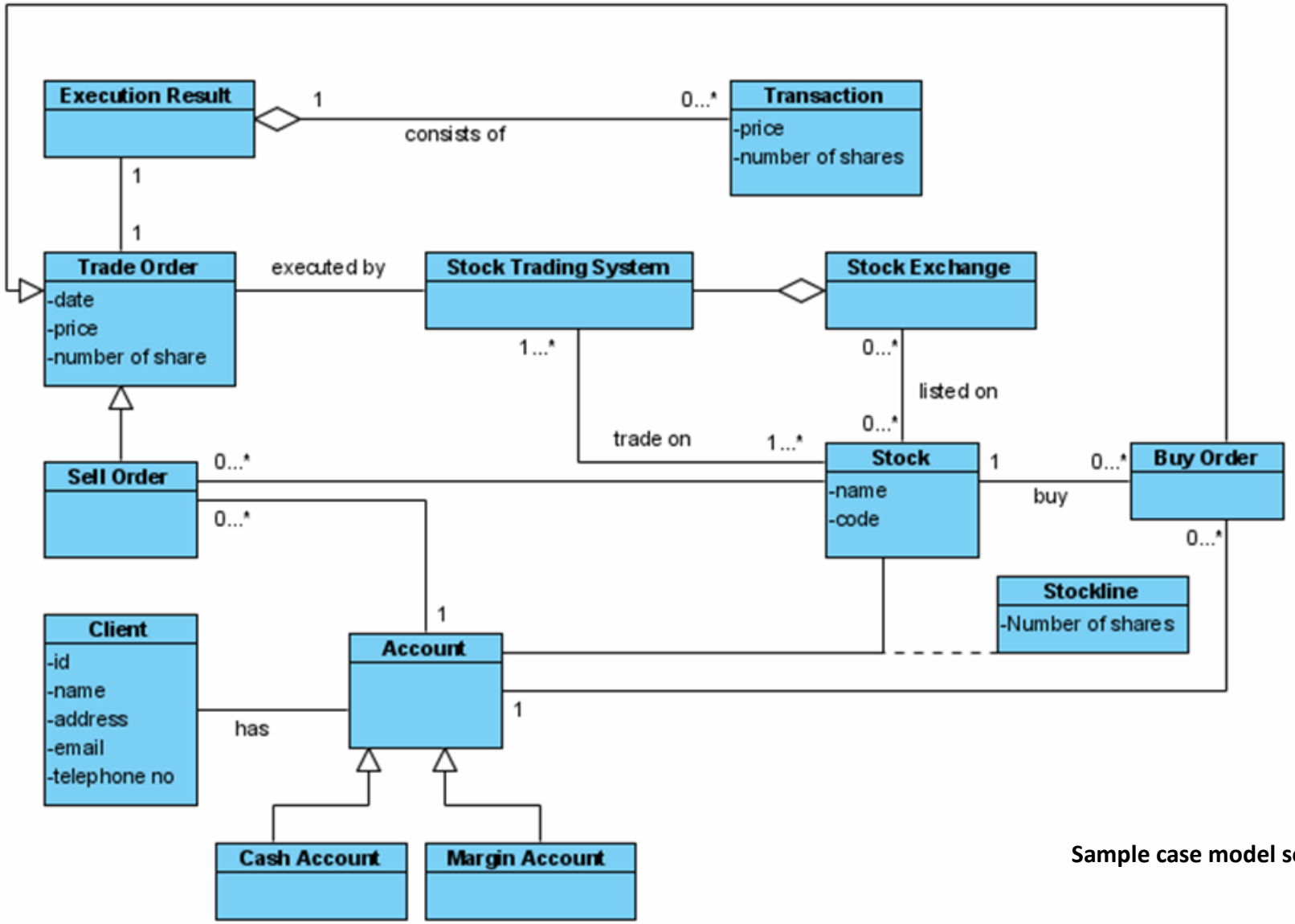
- Captures the most important types of objects in a system
- Describing "things" in a system and how these things are related to each other
- A "thing" can be an object, a class, an interface, a package or a subsystem, which is part of the system being developed
- Very important process because it is employed throughout the entire system development life cycle
- **Thus → Model of the problem domain with**
  - **Glossary of terms**
  - **Domain Objects – domain classes**
  - **Relations between terms**

Example of a domain

- Application for the lending out of books
  - The domain consists of the whole range of things, data and rules that have to do with the lending of books

- Application for an internet shop with computer items
  - The domain consists of everything that has to do with the sale of those items

Example of a domain model



Sample case model solution

Characteristics & benefits of a domain model (1)

- System of **abstractions** describing selected aspects of sphere of knowledge, influence, or activity
- Is **visual representation** of **meaningful real-world concepts**, **conceptual classes** pertinent to the domain needing to be modeled in software
- Thus: identify & relate **key concepts** in a domain
  - Also called "conceptual modeling"
- Shows **associations and relationships** between concepts
  - E.g. Payment PAYS-FOR Sales
- Shows **attributes** for information content
  - E.g. Sale records DATE and TIME

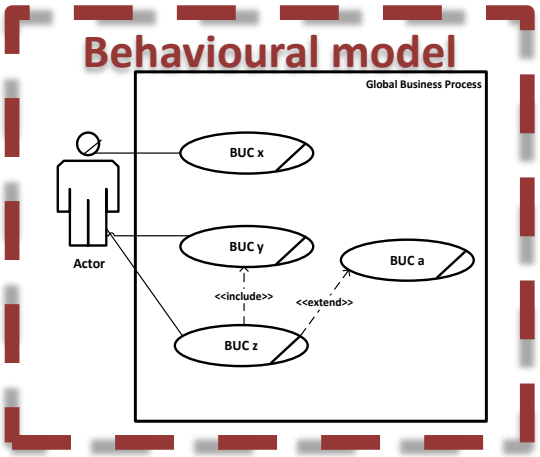
Characteristics & benefits of a domain model (2)

- Concepts of domain model: **data** involved in the business and business rules used in relation to that data
- Used **to solve problems** related to that domain, business
- Generally uses **vocabulary of the domain** so that representation of the model can be used to **communicate with non-technical stakeholders**
- **Does not include operations / functions**
- **Does not describe software classes**
- **Does not describe software responsibilities**
- **Part of Object-Oriented Analysis**
  - **i.e. analysis of the problem space**

Definition of a class diagram

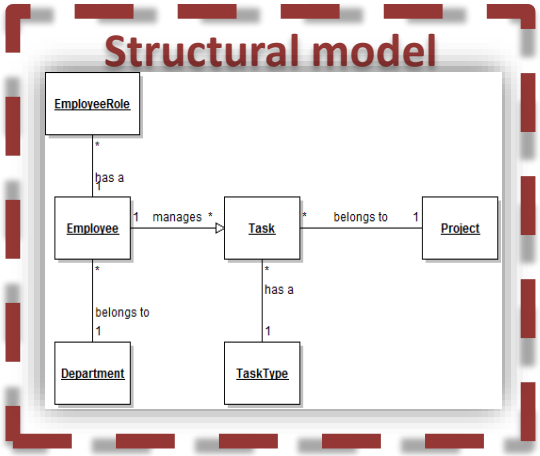
- A diagram that describe structure of system by showing the system's:
  - classes
  - its attributes and operations
  - relationships between classes

The process of domain modeling (1)



Test plans

Domain model



Source code

The process of domain modeling (2)

- **Why domain modeling?**
  - You may **not know** the **domain** well
    - Details matter!
    - E.g.: does every student have exactly one major?
  - You don't want to forget **key concepts**
    - E.g.: a student's home college affects registration
  - You want to **agree** on **common set of terms**
    - E.g.: a freshman/sophomore vs. first-year/second-year
  - **Prepare to design**
    - Domain concepts are good candidates for OO classes

General steps in domain modeling (1)

- A **recommended flow** for domain modeling is shown below in sequential order:
  1. Prepare problem statement for the system being developed
  2. Identify concepts (these are the classes & objects)
  3. Develop a common vocabulary, dictionary, glossary
     a) Make an alphabetic list
     b) Count the occurrences
     c) Make a glossary of terms ➔ domain classes
     d) Create a first domain class diagram
  4. Identify associations between concepts
  5. Assign attributes to the concepts
  6. Check for multiplicities and indicate in domain model
  7. Iterate and refine the model

Step 01 - Prepare problem statement

- Use interview notes of requirements
- Check for business use case descriptions
- Reuse existing models
- Gather other input information, first own notes of requirements
- Analyze typical use scenarios, analyze behavior
- Execute a brainstorming
- Collect first; organize, filter, and revise later

Step 02 – Identify concepts, classes & objects (1)

- Read descriptions, input documentation carefully
- Look for **nouns** ➔ **and underline these**
  - Words with definite and indefinite articles (“the”, “it“, “a”)
- Look for **verbs** because nouns execute these
  - Verbs indicate an action
- Look for **adjectives** because these tell something about the nouns
  - Characteristics, properties (=attributes)
  - Attributes (usually) derived from sentence structures as:
  - “X” has a “Y” and a “Z”
  - “X” is made up of a “Y” and a “Z”
  - “X” consists of a “Y” and a “Z”

Step 02 – Identify concepts, classes & objects (2)

- Other hint for identifying: use a category list
  - **Tangible things**: cars, telemetry data, terminals, classroom, playground, …
  - **Conceptual**: course, module, …
  - **Events**: landing, purchase, request, test, examination, seminar, …
  - **External organizations**: publisher, supplier, …
  - **Roles played**: mother, teacher, researcher, student, …
  - **Other system(s)**: admission system, grade reporting system, …
  - **Interactions**: loan, meeting, intersection, …
  - **Attributes**: cash balance, color, …
  - **Structure**, **devices**, **organizational units**, …

Step 02 – Identify concepts, classes & objects (3)

- Underlined words = domain concepts
- User does not count? Why?
  - Is an actor!
  - Underline but use another color

- Conclusion
  - Method is no guarantee for finding of all domain classes
  - In a lot of situations this is an ideal starting point for further investigation

Step 01 - Prepare problem statement (1)

- Let's start with **an easy example**!
- The business use case description of a microwave oven is given
- Title of the BUC: "Microwave oven: how to heat food?"
- See next slide

Step 01 - Prepare problem statement (2)

“Microwave oven: how to heat food?”

ID	BUC_08, version v3.0	
Title	The heating of food	
Actor(s)	Cook	
Precondition(s)	The microwave oven is inactive, but it is plugged in The door is closed The food to heat is in close reach	
Normal flow	**Step**	**Description**
	01	The user opens the door
	02	The light goes on
	03	The user puts the food into the oven
	04	The user closes the door
	05	The light turns out
	06	The user presses the button once to set the working time to 60 seconds

Step 01 - Prepare problem statement (2)

“Microwave oven: how to heat food?”

Normal flow	Step	Description
(continued)	07	The light goes on
	08	Microwave tube starts
	09	The working time becomes visible on the display
	10	The working time decreases and it is visible in the window. When the time is up:
	11	The light turns out
	12	A beep sounds
	13	The user opens the door
	14	The light goes on
	15	The user takes out the food
	16	The user closes the door
	17	The light turns out
Post condition	The food is heated	

Step 02 – Identify concepts, classes & objects (1)

"Microwave oven: how to heat food?"

Underline

ID	BUC_08, version v3.0	
Title	The heating of food	
Actor(s)	Cook, the user	
Precondition(s)	The microwave oven is inactive, but it is plugged in The door is closed The food to heat is in close reach	
Normal flow	**Step**	**Description**
	01	The user opens the door
	02	The light goes on
	03	The user puts the food into the oven
	04	The user closes the door
	05	The light turns out
	06	The user presses the button once to set the working time to 60 seconds

Step 02 – Identify concepts, classes & objects (2)

"Microwave oven: how to heat food?"

Normal flow	Step	Description
(continued)	07	The light goes on
	08	Microwave tube starts
	09	The working time becomes visible on the display
	10	The working time decreases and it is visible in the window. When the time is up:
	11	The light turns out
	12	A beep sounds
	13	The user opens the door
	14	The light goes on
	15	The user takes out the food
	16	The user closes the door
	17	The light turns out
Post condition	The food is heated	

Step 03 - Develop a common vocabulary (1)

- Develop a common vocabulary, dictionary, glossary
  - Count the occurrences

Nouns (1)	#
Door	4
Light	6
Food	2
Oven	1

Nouns (2)	#
Button	1
Working time	3
Seconds	1
Microwave tube	1

Nouns (3)	#
Display	1
Window	1
Time	1
Beep	1

  - Make an alphabetic list

Nouns (1)	#
Beep	1
Button	1
Display	1
Door	4

Nouns (2)	#
Food	2
Light	6
Microwave tube	1
Oven	1

Nouns (3)	#
Seconds	1
Time	1
Window	1
Working time	3

Step 03 - Develop a common vocabulary (2)

- Which domain concepts play an essential role?
- Consider whether concepts belong together
- Consider different words meaning same thing (synonyms)
- Consider same words possibly meaning something else (homonyms)
- Then
  - Annotate alphabetical list by indicating “Tangible”, “Concept”, “Event”, “External organization”, “Role”, “Other system”, “Interaction”, “Attribute”, “Structure”, “Device”, “Organizational unit”
  - Complete dictionary by adding definitions and comments

Step 03 - Develop a common vocabulary (3)

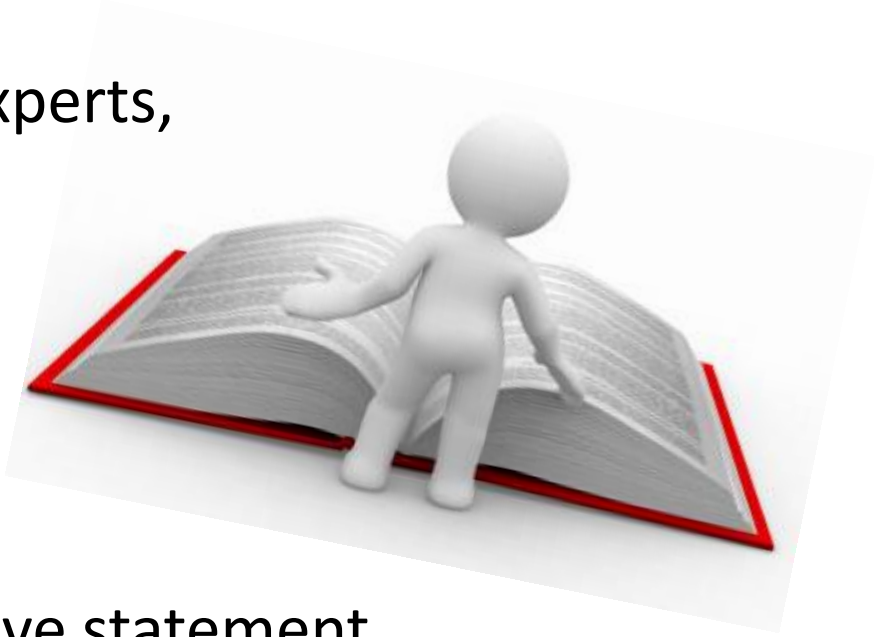
Nouns	#	Definitions and comments
Button	1	Tangible thing. Button is a critical concept.
Display	1	Tangible thing. Display is a critical concept.
Door	4	Tangible thing. Door is a critical concept.
Food	2	Interaction. Food is not part of the system, the oven …
Light	6	Tangible thing. Light is a critical concept.
Microwave tube	1	Tangible thing. Microwave tube is a critical concept.
Oven	1	Tangible thing. Oven tube is a critical concept. It is the “case” that keeps the whole together.
Seconds	1	Attribute. Unit of measure for working time.
Time	1	Concept. Time means the same as working time.
Window	1	Tangible thing. Window means the same a s display.
Working time	3	Concept. Working time is a critical concept. We need a clock, a timepiece

Step 03 - Develop a common vocabulary (4)

Nouns	#	Definitions and comments
Beep	1	Concept. Beep is a critical concept.
Button	1	Tangible thing. Button is a critical concept.
Display	1	Tangible thing. Display is a critical concept.
Door	4	Tangible thing. Door is a critical concept.
Food	2	Interaction. Food is not part of the system, the oven …
Light	6	Tangible thing. Light is a critical concept.
Microwave tube	1	Tangible thing. Microwave tube is a critical concept.
Oven	1	Tangible thing. Oven tube is a critical concept. It is the "case" that keeps the whole together.
Seconds	1	Attribute. Unit of measure for working time.
Time	1	Concept. Time means the same as working time.
Window	1	Tangible thing. Window means the same a s display.
Working time (= clock)	3	Concept. Working time is a critical concept. We need a clock, a timepiece

Step 03 - Develop a common vocabulary (5)

- Create a glossary of terms, in collaboration with domain experts, developers, etc.
- In our example
  - We have 8 domain concepts (see previous slide)



- Each concept gets a descriptive statement
- Useful for developers who know little of the sector for which they must make the application

Step 03 - Develop a common vocabulary (6)

- Domain classes

- The domain concepts thus represent several objects …
- A class however **is NOT the same as an** object
  - A class is a *description* for similar objects
  - An object is a thing that is made according to the description
  - One class can have many objects

Step 03 - Develop a common vocabulary (7)

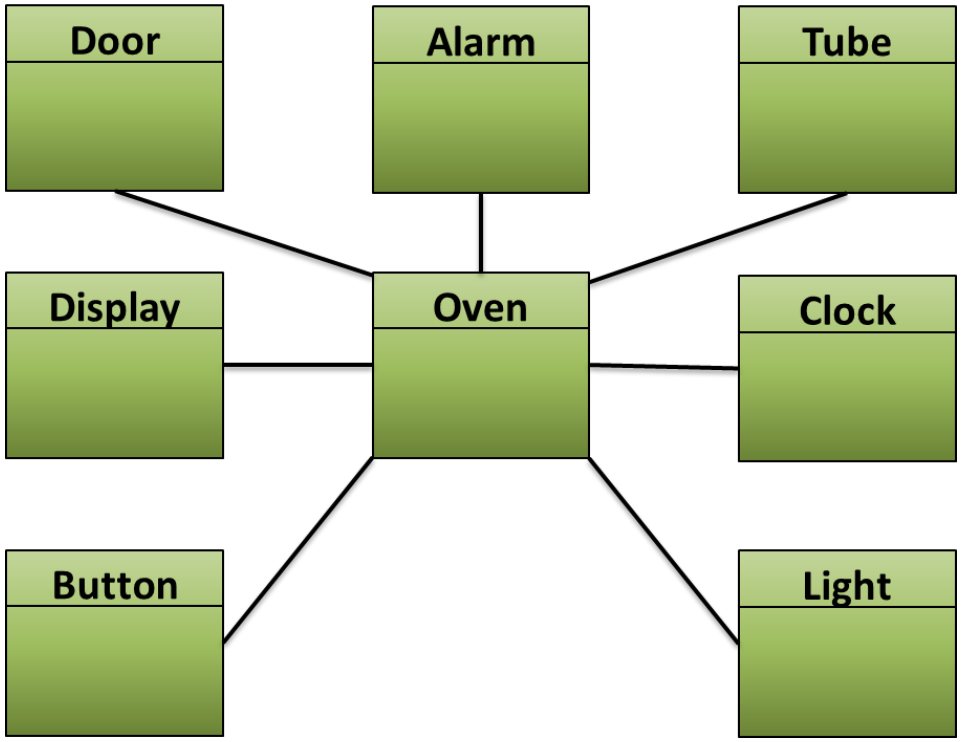
- So let us name the 8 domain classes of our example

Domain Class	Nouns	#	Definitions and comments
Alarm	Beep	1	Concept. Beep is a critical concept.
Button	Button	1	Tangible thing. Button is a critical concept.
Clock	Working time (= clock)	3	Concept. Working time is a critical concept. We need a clock, a timepiece
Display	Display	1	Tangible thing. Display is a critical concept.
Door	Door	4	Tangible thing. Door is a critical concept.
Light	Light	6	Tangible thing. Light is a critical concept.
Oven	Oven	1	Tangible thing. Oven tube is a critical concept. It is the “case” that keeps the whole together.
Tube	Microwave tube	1	Tangible thing. Microwave tube is a critical concept.

- **Further indebt analysis needed to clarify if there are more domain classes**

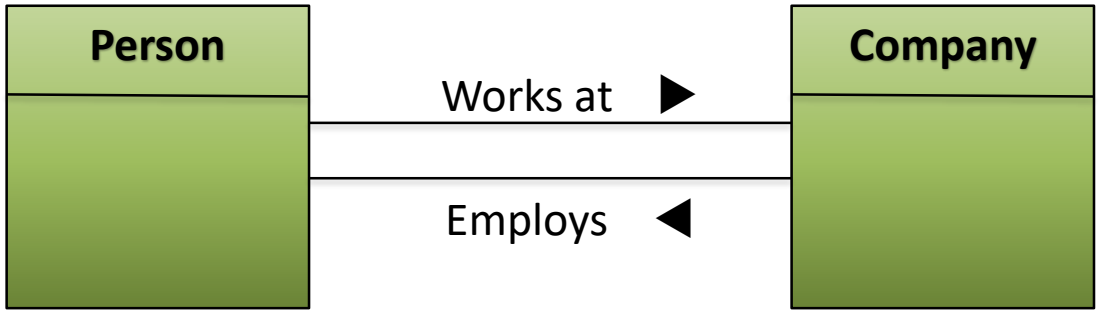
Step 03 - Develop a common vocabulary (8)

- Create a first **domain class diagram**
  - The oven is composed of the objects of the 7 other domain classes
  - "The oven is built of …."
  - The lines between the objects indicate that there is some kind of relationship between the domain classes



Step 04 - Associations between concepts (1)

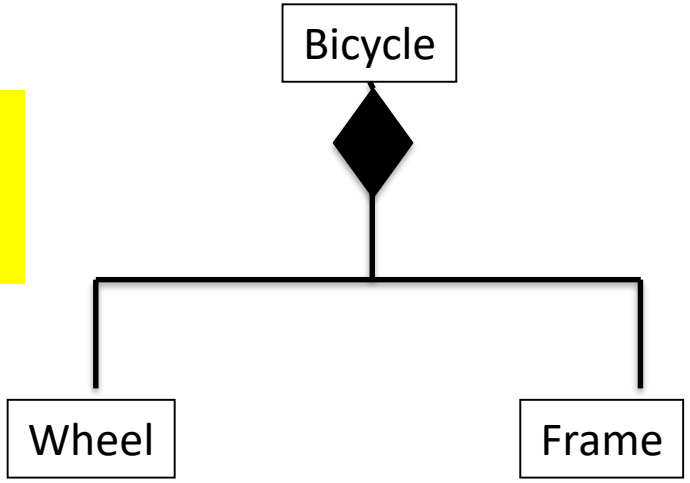
- Associations represent relationships between instances of classes
- For domain modeling associations represent **conceptual relationships**
  - Association = structural relation between 2 domain classes
    - Indicated by drawing a line between the 2 domain classes
    - Convention
      - Read names from left to right … but
      - Names can also be read from right to left (cf. arrow)



Step 04 - Associations between concepts (2)

- Possible **conceptual relationships**
  - Composition
    - A part can always only belong to a single entity
    - The part of the object can not independently continue to exist if the object itself, where the part is part of, disappears (not shared association)
    - Remark: use of a composite relationship is preferable to the aggregation relationship

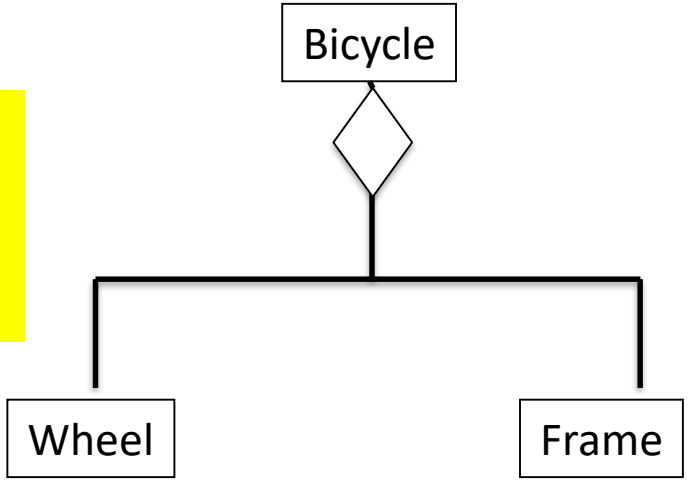
The bicycle is one whole
The bicycle has a composition relationship with its parts



Step 04 - Associations between concepts (3)

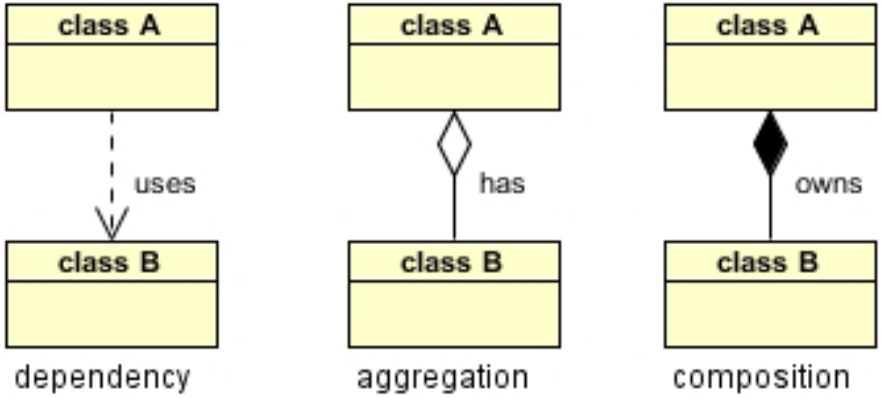
- Possible **conceptual relationships**
  - Aggregation
    - Indicates that one or more domain classes are part of another, different class
    - A shared association (also part of others)

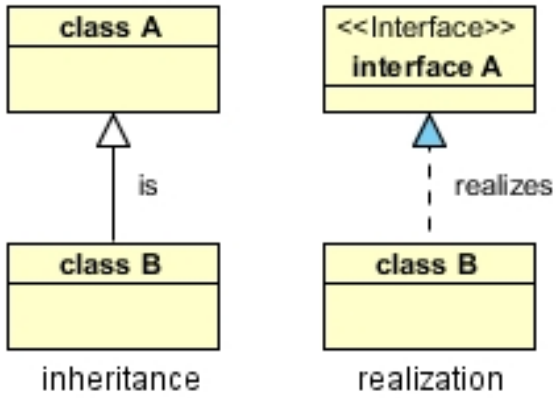
For a cyclist the bicycle is NOT one whole. The weather for example will determine the choice of the wheels which are put under the frame.



Step 04 - Associations between concepts (4)

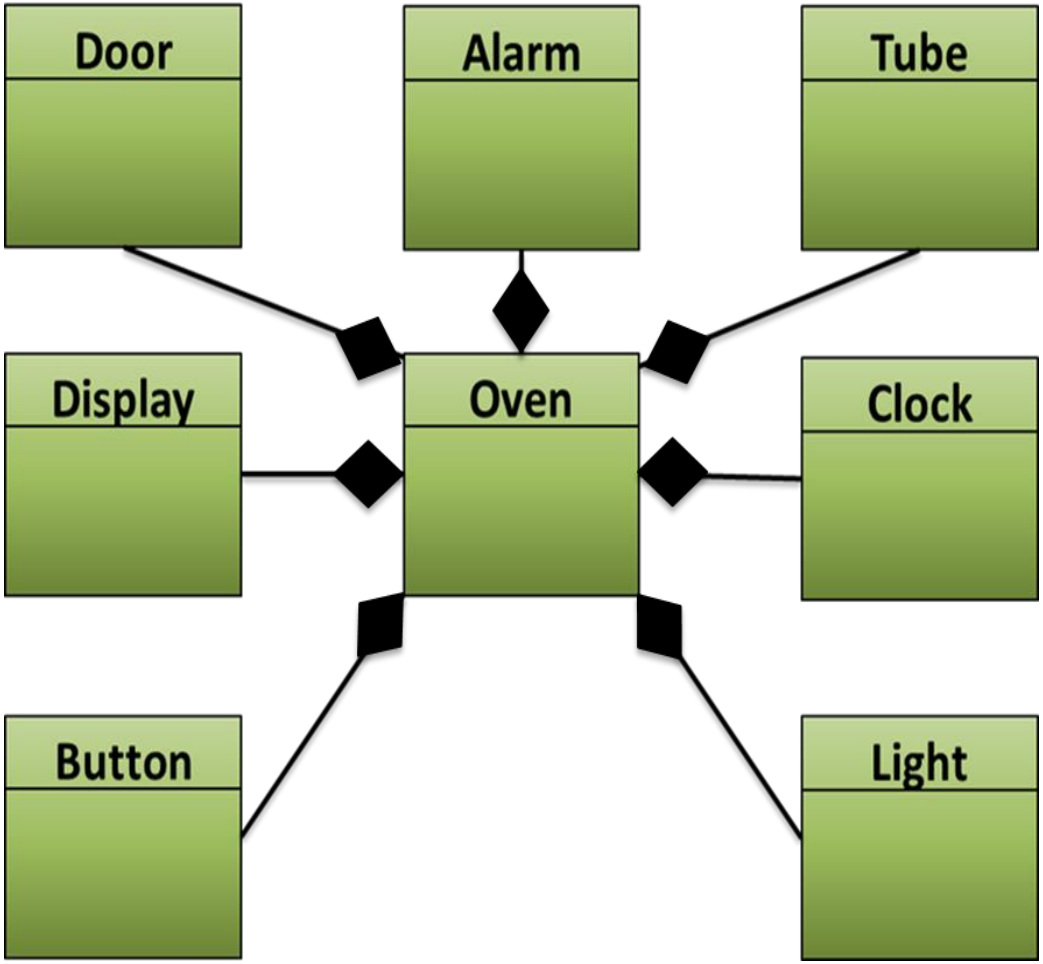
- The relationships are read as follows:
  - Dependency:
    - Class A uses class B
  - **Aggregation:**
    - Class A has a class B
  - **Composition:**
    - Class A owns a class B
  - Inheritance:
    - class B is a class A (or class A is extended by class B)
  - Realization:
    - Class B realizes class A (or class A is realized by class B)





Step 04 - Associations between concepts (5)

- Going back to our example of the microwave oven



Step 05 - Assign attributes to the concepts

- A class consists out of attributes & methods

- Attributes
  - Information or properties available for each object
  - These can be added to the domain model when it gives more clarity
- Methods
  - Operations, actions, executed by the objects

Remark: not covered in the example of the microwave oven

Step 06 - Check for multiplicities (1)

- For a better understanding of the association (s) between the domain classes
- Multiplicity is the number of instances of a particular class that is involved in an association in relation to exactly one instance of another class

Step 06 - Check for multiplicities (2)

- For example

- From left to right
  - Each student will participate in an indeterminate (* right) number of exams
- From right to left
  - Each exam is attended by an indefinite (* left) number of students

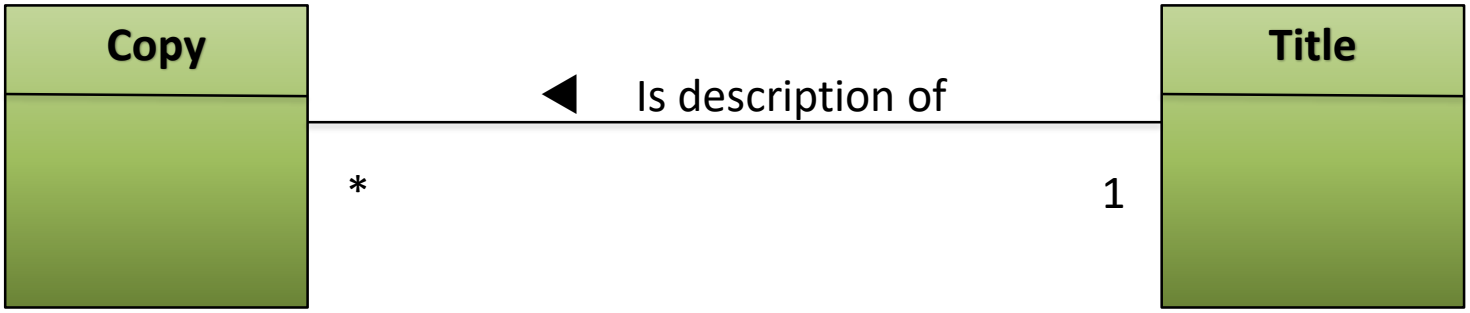
Step 06 - Check for multiplicities (3)

- Multiplicities in UML

Multiplicity	
1	Exactly 1
2	Exactly 2
1 .. 5	From 1 to 5 (included)
6, 7	6 or 7
1 .. *	Undefined number but at least 1
0 .. *	Undefined number, eventually 0 Is the same as *
*	Undefined number, eventually 0 Is the same as *
0 .. 1	0 or 1 Is the same as 0, 1

Step 06 - Check for multiplicities (4)

- Example - Library



- To every copy there only belongs one title
- For every title there exists an indefinite number (possibly 0) copies

Step 06 - Check for multiplicities (5)

- Example - Chess

- Every chessboard consists out of 64 fields
- Every field belongs to exactly one chessboard

Exercises

- MS Word doc
  - Tank station
  - Placing an order
  - Online Stock

Questions & answers