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

Week 05 – semester 01

Luc Doumen

Nathalie Fuchs

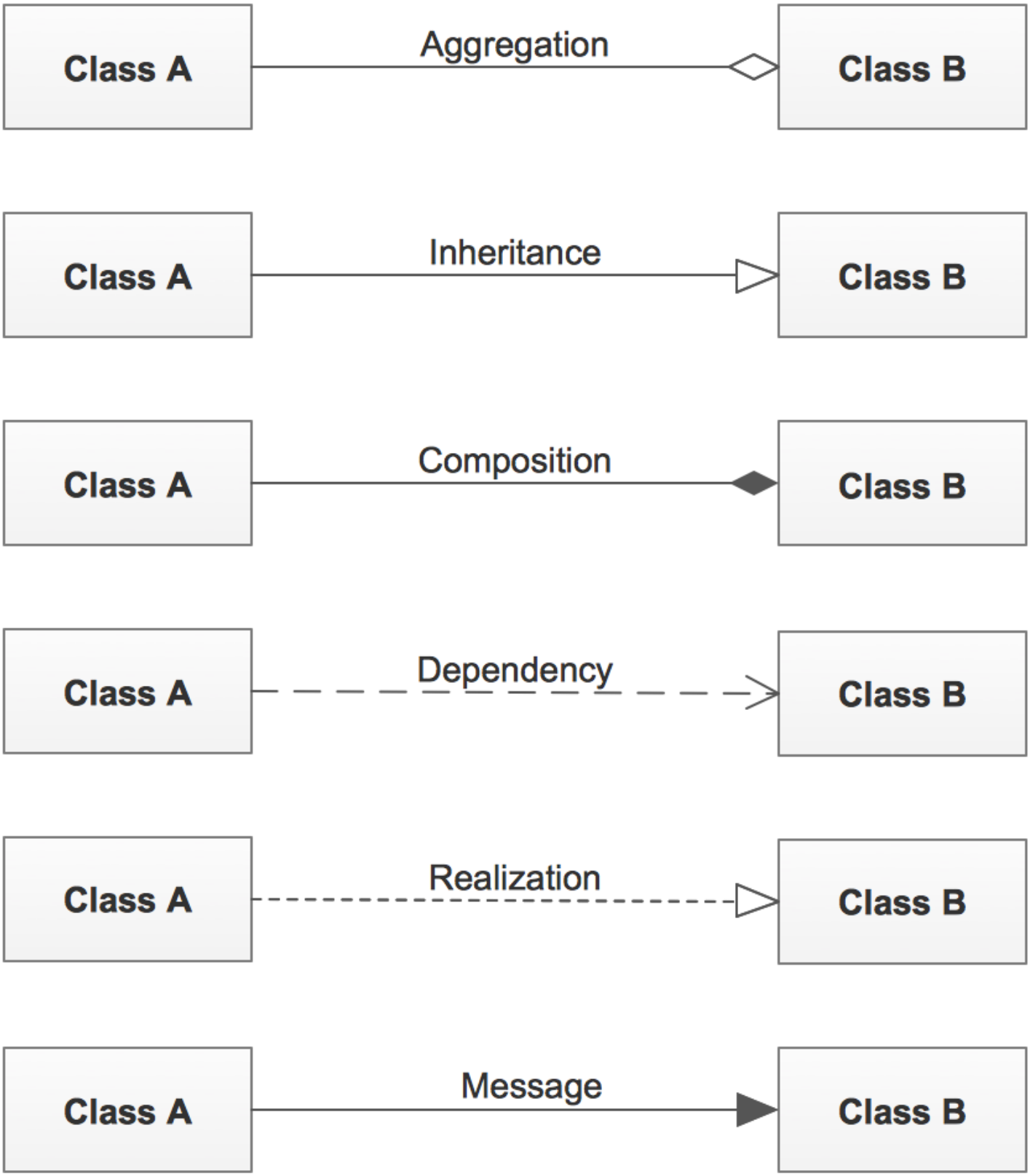
DE HOGESCHOOL MET HET NETWERK

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

UML Domain modeling (1)

Shape	Description
Package1	**Package** A collection of interaces and classes.
«interface»IFoo +Method1()	**Interface** Microsoft guidelines specify that interfaces should start with I. This graphic can also sometimes be used as an abstract class.
Class1 -Property1 +Method1()	**Class** Properties or attributes sit at the top, methods or operations at the bottom. + indicates public and # indicates protected.
	These are both typically drawn vertically:
B ——▷ A	**Inheritence** - B inherits from A. "is-a" relationship.
B - - - - ▷ A	**Generalization** - B implements A,
A —— B	**Association** - A and B call each other
A ——→ B	**One way Association.** A can call B's properties/methods, but not visa versa.
A ◇—— B	**Aggregation** A "has-a"instance of B. B can survive if A is disposed.
A ◆—— B	**Composition** A has an instance of B, B cannot exist without A.
	A note Some descriptive text attached to any item.

UML Domain modeling (2)



UML Domain modeling (3)

Multiplicity	UML Multiplicity Notation	Association with Multiplicity	Association Meaning
Exactly 1	1 or *leave blank*	Employee — Works for 1 — Department Employee — Works for — Department	An employee works for one and only one department.
Zero or 1	0..1	Employee — Has 0..1 — Spouse	An employee has either one or no spouse.
Zero or more	0..* or *	Customer — Makes 0..* — Payment Customer — Makes * — Payment	A customer can make no payment up to many payments.
1 or more	1..*	University — Offers 1..* — Course	A university offers at least 1 course up to many courses.
Specific range	7..9	Team — Has scheduled 7..9 — Game	A team has either 7, 8, or 9 games scheduled

UML Domain modeling – General steps

1. **Prepare problem statement for the system being developed**
   a) Most of the time a BUC description

2. **Identify concepts (these are the classes & objects)**
   a) Read descriptions, input documentation carefully
   b) Look for nouns ➔ and underline these
   c) Words with definite and indefinite articles (“the”, “it“, “a”)
   d) Look for verbs because nouns execute these
      - Verbs indicate an action
   e) Look for adjectives because these tell something about the nouns
   f) 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”

UML Domain modeling – General steps

1. **Prepare problem statement for the system being developed**
   a) Most of the time a BUC description

2. **Identify concepts (these are the classes & objects)**
   g) 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, …

3. Develop a common vocabulary, dictionary, glossary
   a) Make an alphabetic list
   - Count the occurrences
   - Make a glossary of terms ➔ domain classes
   - Create a first domain class diagram

4. Identify associations between concepts

UML Domain modeling – General steps

1. **Prepare problem statement for the system being developed**
   a) Most of the time a BUC description

2. **Identify concepts (these are the classes & objects)**

5. Assign attributes to the concepts
6. Check for multiplicities and indicate in domain model
7. Iterate and refine the model