



**DE HOGESCHOOL
MET HET NETWERK**

Elfde-Liniestraat 24, 3500 Hasselt, www.pxl.be

PXL – Digital

42TIN280 Software Analysis -

System & System Context – Domain Model

Cheat sheet

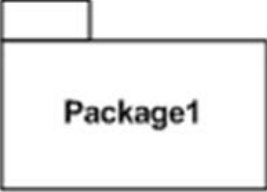
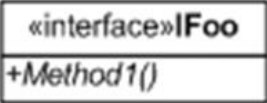
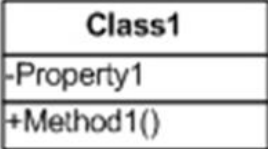







Week 05 – semester 01

Luc Doumen

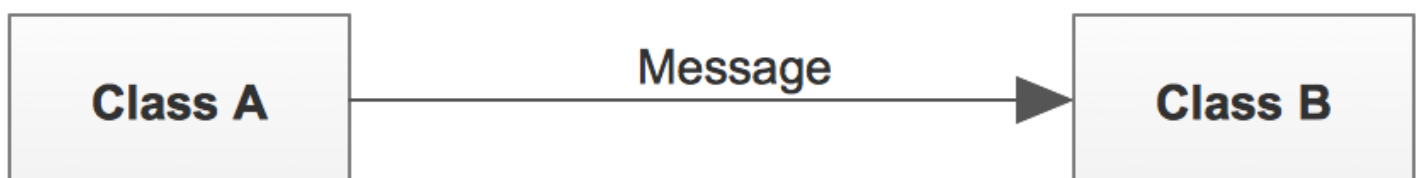
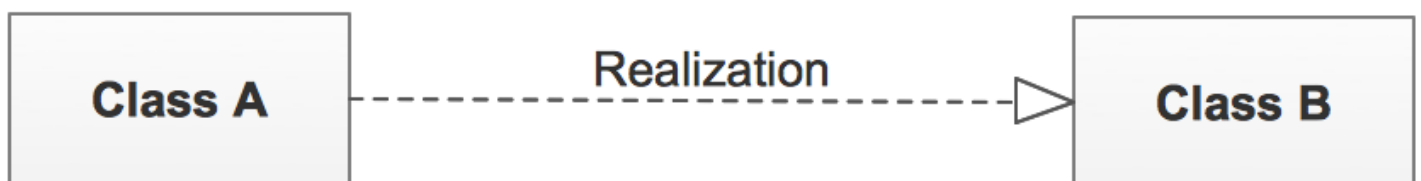
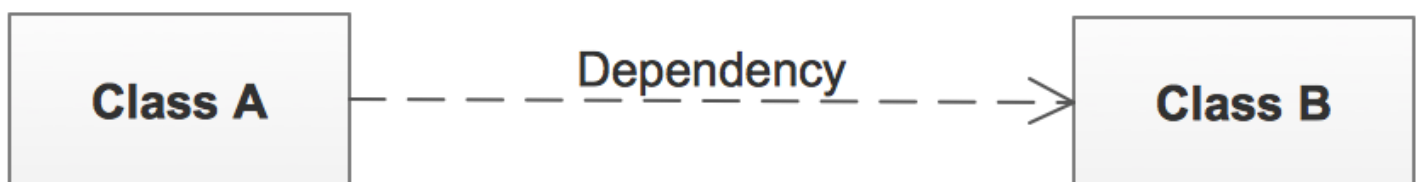
Nathalie Fuchs



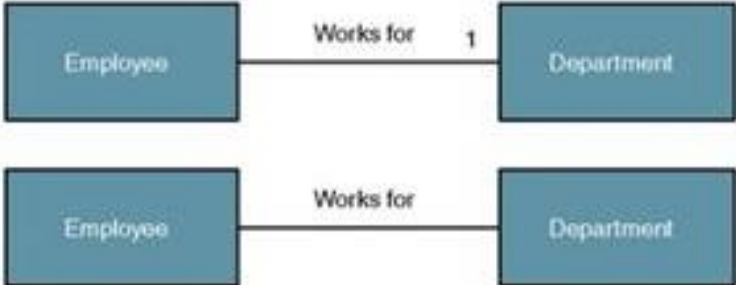


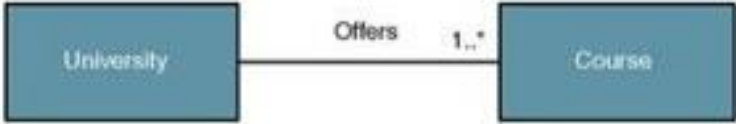
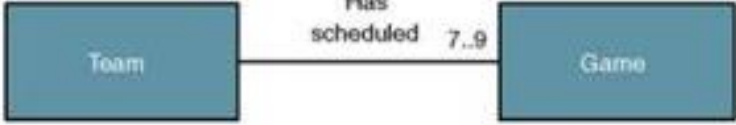
UML Domain modeling (1)

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

UML Domain modeling (2)



UML Domain modeling (3)

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

UML Domain modeling – General steps

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