

---

# Software Engineering

## **UML definition**

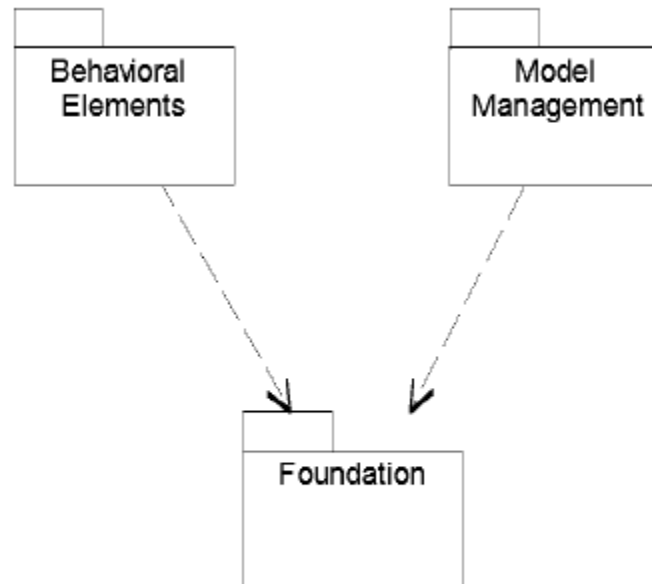
Dan CHIOREAN  
Babeş-Bolyai University  
`chiorean@cs.ubbcluj.ro`



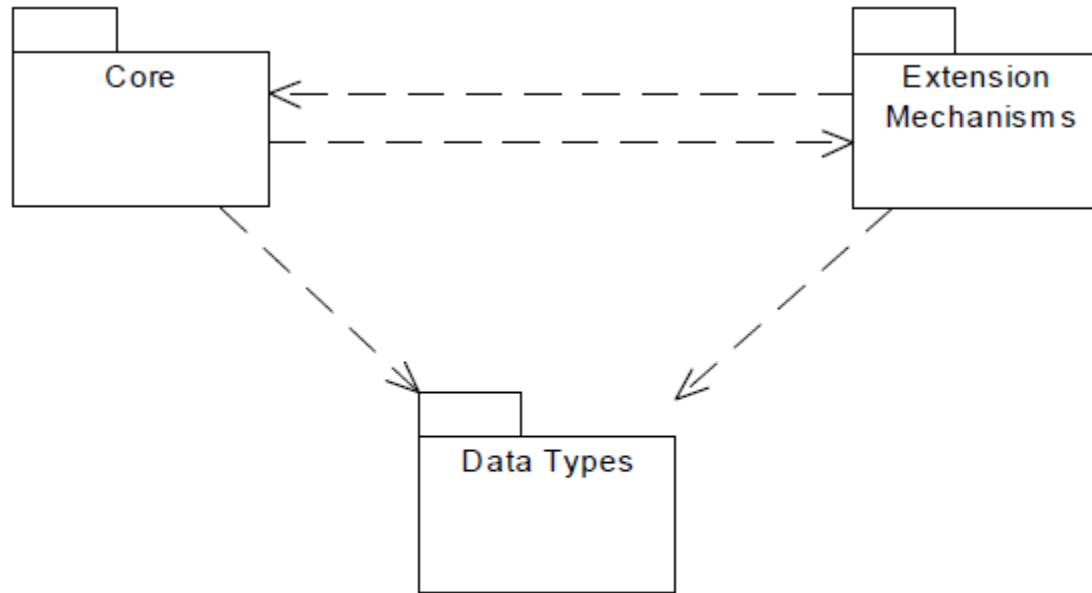
# UML\_definition – Top Level Packages

### Language definition (1.5.1)

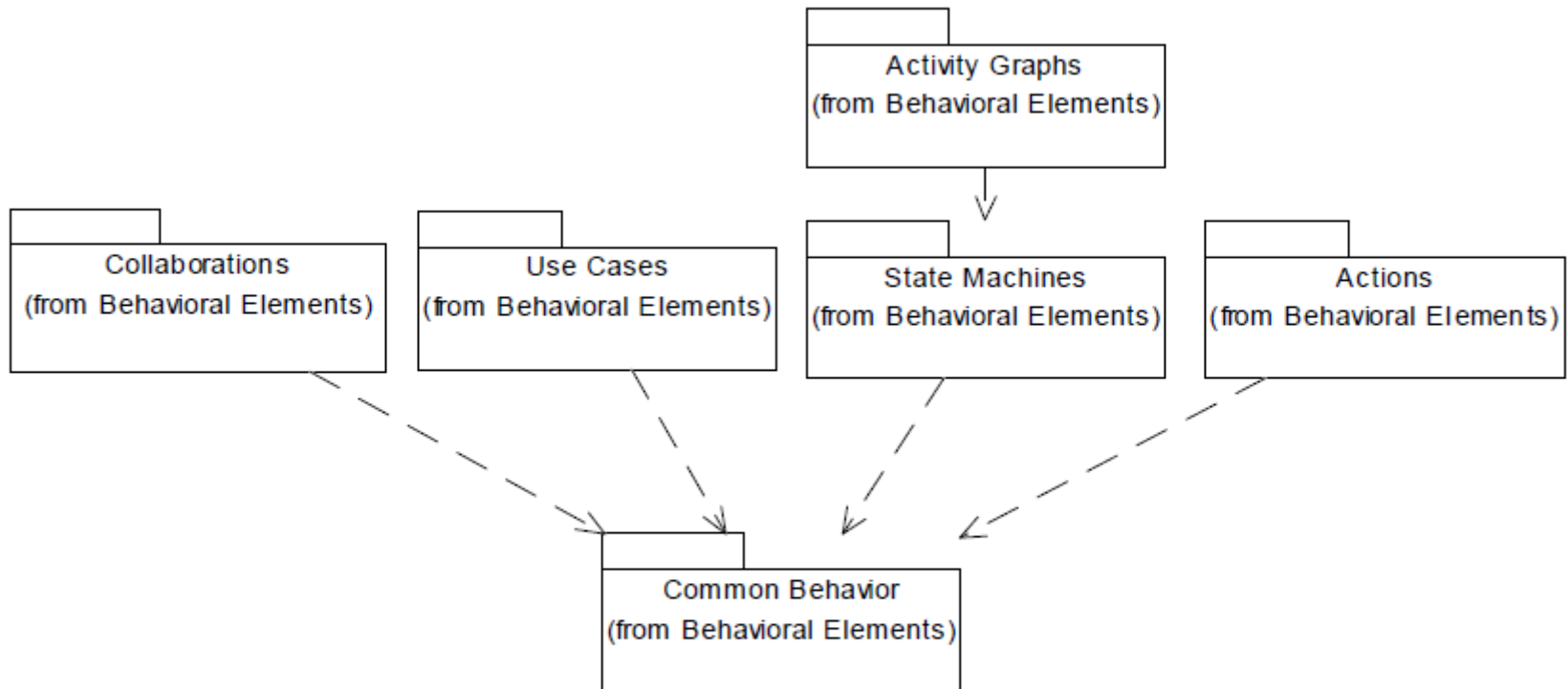
- concrete syntax
- abstract syntax (metamodel + constraints)
- semantics



*Figure 2-1 Top-Level Packages*



*Figure 2-2 Foundation Packages*



*Figure 2-3 Behavioral Elements Packages*

# UML definition – Core Package - Backbone

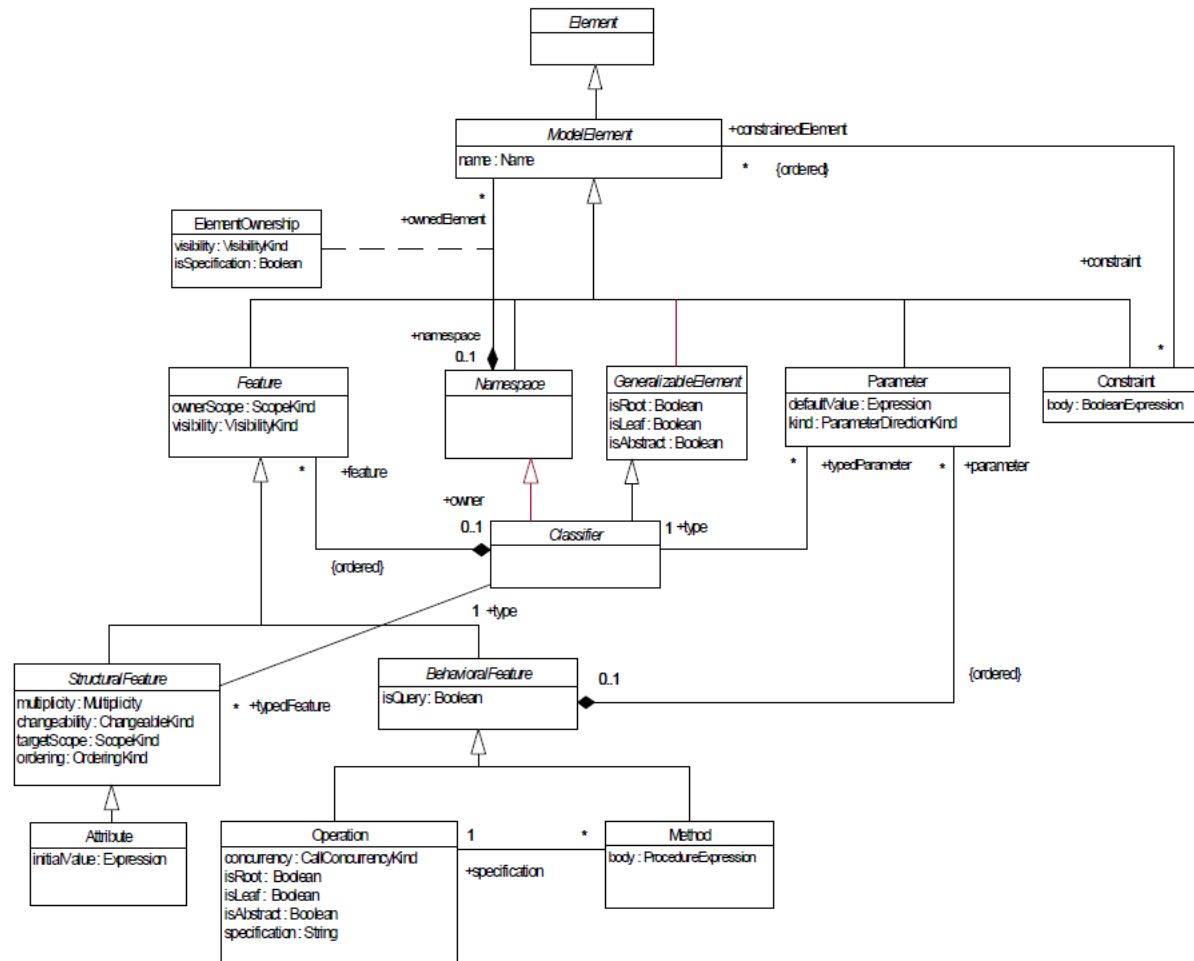
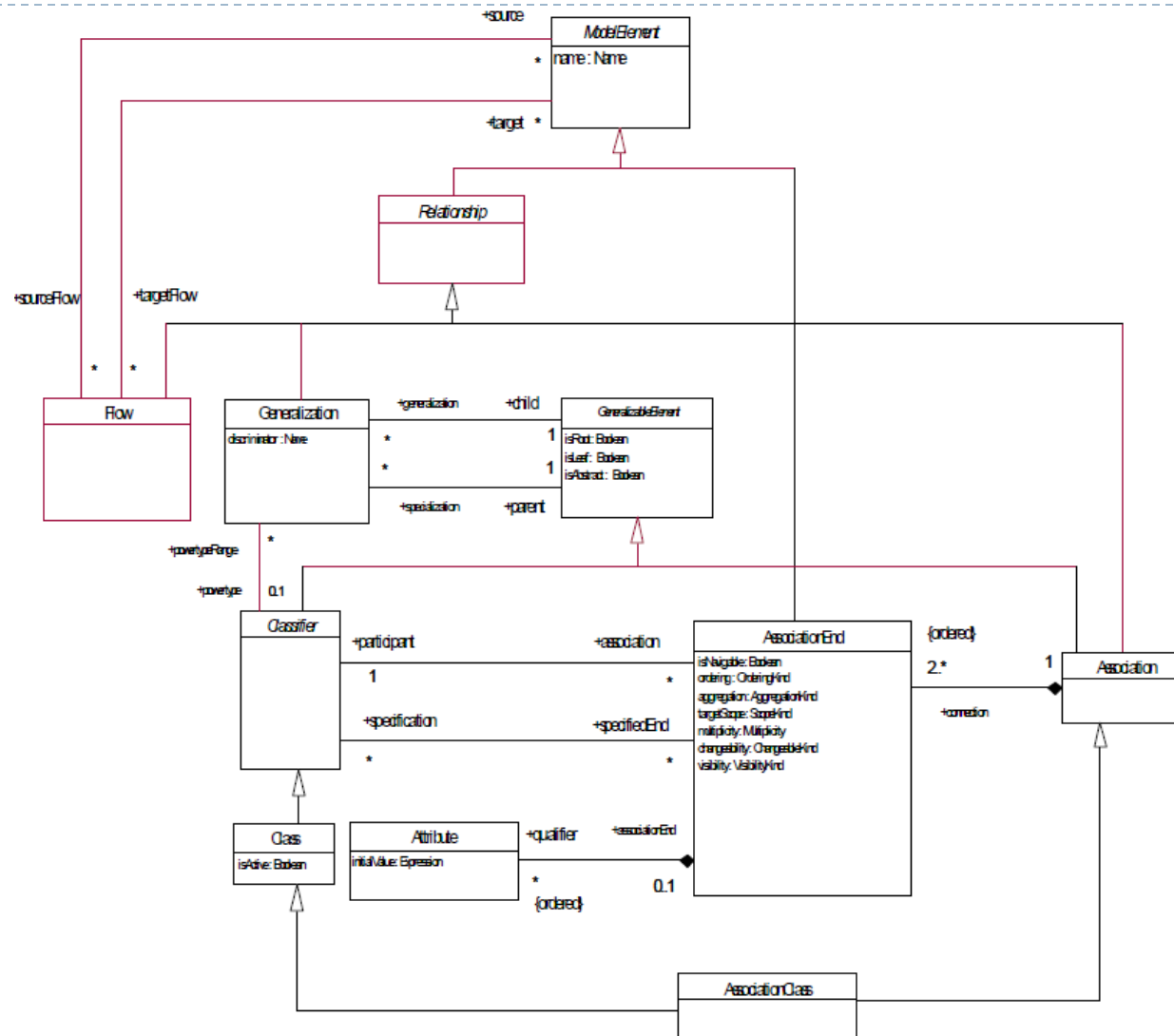


Figure 2-5 Core Package - Backbone



# Software Engineering



# UML definition – Foundation WFRs & AOs

---

At most one AssociationEnd may be an aggregation or composition.

```
self.allConnections->select(aggregation <#none)->size <= 1
```

The names of the AssociationEnds and the StructuralFeatures do not overlap.

```
self.allConnections->forAll( ar |  
    self.allFeatures->forAll( f |  
        f.ocllsKindOf(StructuralFeature) implies ar.name <> f.name ))
```

The operation allConnections results in the set of all AssociationEnds of the AssociationClass, including all connections defined by its parent (transitive closure).

```
allConnections : Set(AssociationEnd);  
allConnections = self.connection->union(self.parent->select  
    (s | s.ocllsKindOf(Association))->collect (a : Association |  
        a.allConnections))->asSet
```

# UML definition – Core Package - Classifiers

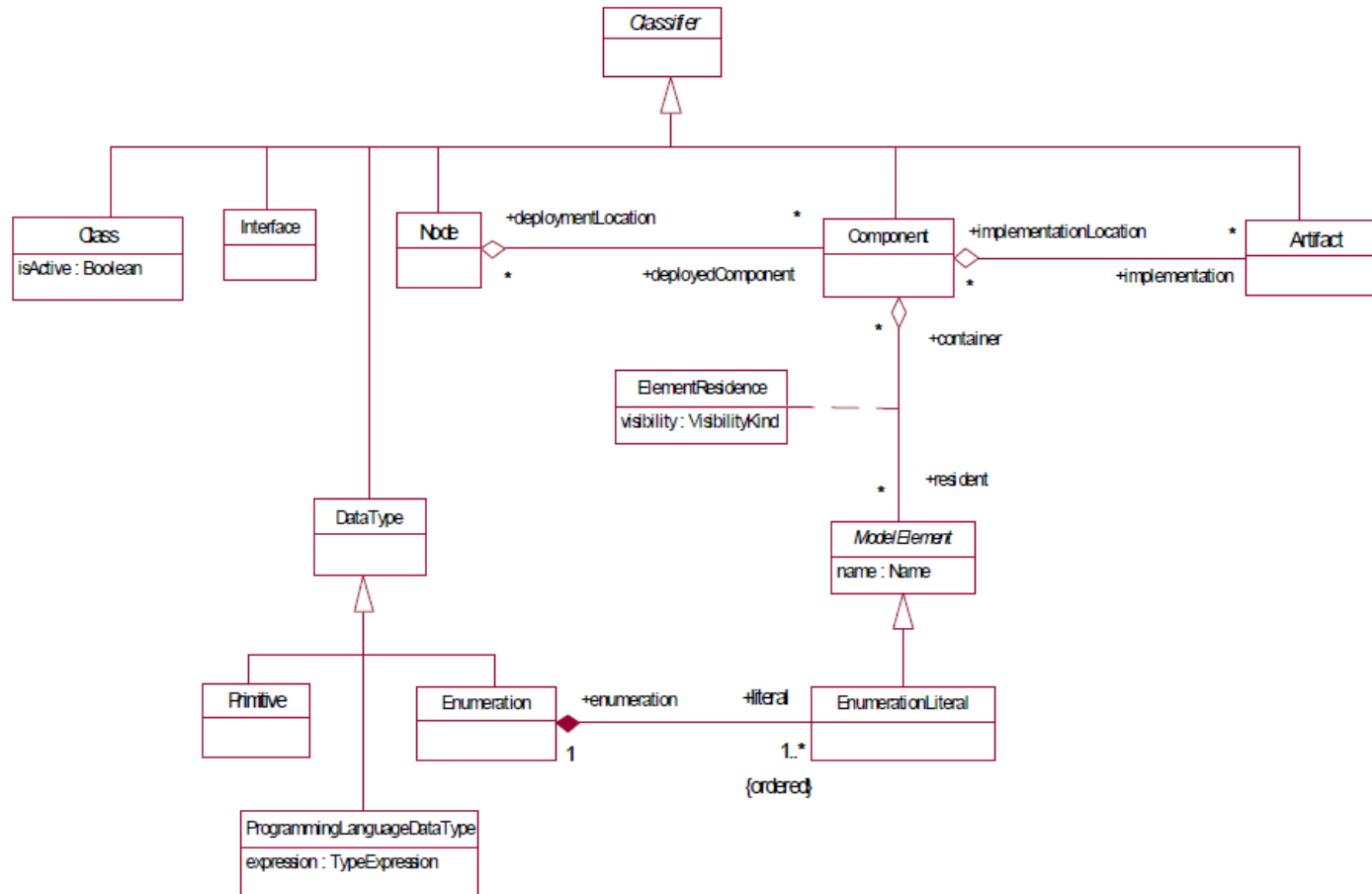
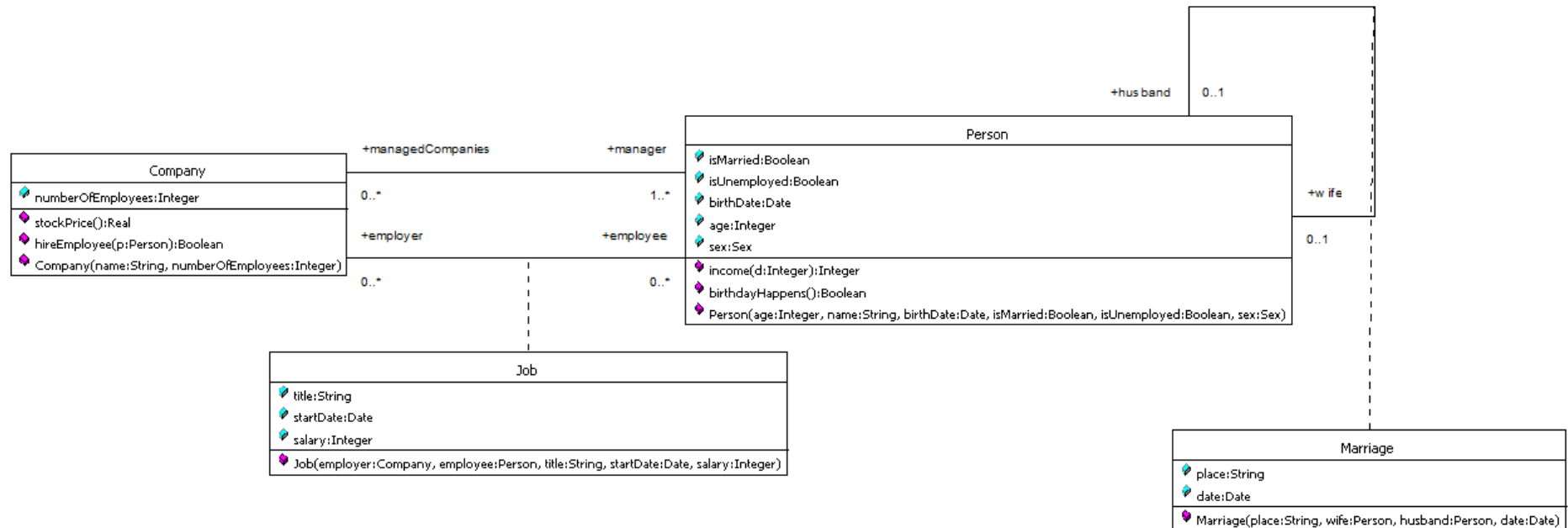


Figure 2-8 Core Package - Classifiers





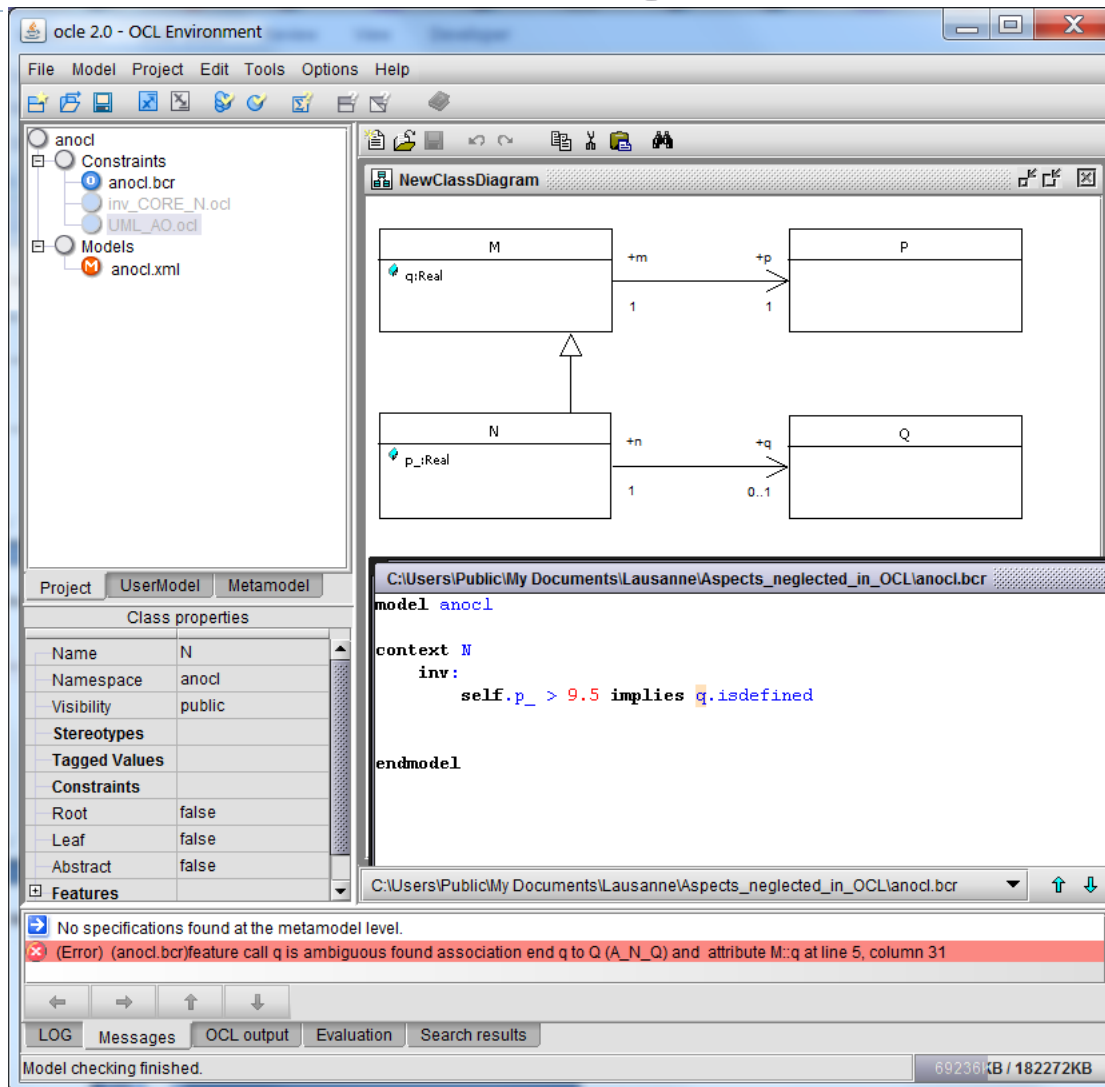
# UML model – Instances of Foundation (meta)classes



# Managing UML models persistency – XMI files

```
C:\Users\Dan\OneDrive\Temporary\Company_Example.xml
</UML:Classifier.feature>
<UML:Namespace.ownedElement>
  <UML:Generalization xmi.id="S.387" parent="S.349" child="S.237" name="Bank_is_a_NamedEntities" namespace="S.237" visibility="public" isSpecification="false"/>
</UML:Namespace.ownedElement>
</UML:Class>
<UML:Class xmi.id="S.12" isActive="false" instance="S.94 S.136" typedParameter="S.397" isRoot="false" isLeaf="true" isAbstract="false" generalization="S.398" name="Company" namespace="S.1" visibility="public" isSpecification="false">
  <UML:Classifier.feature>
    <UML:Attribute xmi.id="S.142" attributeLink="S.253 S.140" type="S.327" ordering="unordered" targetScope="instance" changeability="changeable" owner="S.12" visibility="public" ownerScope="instance">
      <UML:Attribute.initialValue>
        <UML:Expression xmi.id="S.388" body="" language=""/>
      </UML:Attribute.initialValue>
      <UML:StructuralFeature.multiplicity>
        <UML:Multiplicity xmi.id="S.389">
          <UML:Multiplicity.range>
            <UML:MultiplicityRange xmi.id="S.390" multiplicity="S.389" lower="1" upper="1"/>
          </UML:Multiplicity.range>
        </UML:Multiplicity>
      </UML:StructuralFeature.multiplicity>
    </UML:Attribute>
    <UML:Operation xmi.id="S.391" isRoot="false" isLeaf="false" isAbstract="false" concurrency="sequential" isQuery="false" owner="S.12" visibility="public" ownerScope="instance" name="stockPrice">
      <UML:BehavioralFeature.parameter>
        <UML:Parameter xmi.id="S.358" type="S.357" kind="return" behavioralFeature="S.391" name="stockPrice.Return" visibility="public" isSpecification="false"/>
      </UML:BehavioralFeature.parameter>
    </UML:Operation>
    <UML:Operation xmi.id="S.392" isRoot="false" isLeaf="false" isAbstract="false" concurrency="sequential" isQuery="false" owner="S.12" visibility="public" ownerScope="instance" name="hireEmployee">
      <UML:BehavioralFeature.parameter>
        <UML:Parameter xmi.id="S.347" type="S.19" kind="inout" behavioralFeature="S.392" name="p" visibility="public" isSpecification="false">
          <UML:Parameter.defaultValue>
            <UML:Expression xmi.id="S.393" body="" language=""/>
          </UML:Parameter.defaultValue>
        </UML:Parameter>
      </UML:BehavioralFeature.parameter>
    </UML:Operation>
  </UML:Classifier.feature>
</UML:Class>
```

# Example of Ambiguous UML Model



# Checking Ambiguities in UML Models

ocle 2.0 - OCL Environment

File Model Project Edit Tools Options Help

anod  
NewClassDiagram  
M  
N  
P  
Q  
A\_M\_P  
A\_N\_Q

Project UserModel Metamodel

Class properties

Name	N
Namespace	anod
Visibility	public
Stereotypes	
Tagged Values	
Constraints	
Root	false
Leaf	false
Abstract	false
Features	
Active	true

Extension Mechanisms

- Core
  - Classifier
    - let innClassVisName: Bag(String) = (self.allAttributes.name->union(self.allContents.name))>reject(e|e.isUndefined)->reject(e|e="") in innClassVisName->excludesAll(self.oppositeAssociationEnds.name->reject(e|e=""))

Rule failed for context "N":Class

LOG Messages OCL output Evaluation Search results

Model checking finished.

116985KB 188416KB

NewClassDiagram

M

q:Real

N

p:Real

P

Q

C:\Users\Public\My Documents\Lausanne\Checking\_UML\_Models\inv\_CORE\_N.ocd

```
union(self.allContents.name)->includes(a.name)}
```

inv WFR\_5\_Classifier:

-- [5] The name of an opposite AssociationEnd may not be the same as the name of an Attribute or a ModelElement contained in the Classifier.

let innClassVisName: Bag(String) = (self.allAttributes.name->union(self.allContents.name))>reject(e|e.isUndefined)->reject(e|e="") in innClassVisName->excludesAll(self.oppositeAssociationEnds.name->reject(e|e=""))

inv WFR\_6\_Classifier:

-- [6] For each Operation in an specification realized by the Classifier, the Classifier must have a matching Operation.

C:\Users\Public\My Documents\Lausanne\Checking\_UML\_Models\inv\_CORE\_N.ocd

# Checking Ambiguities in UML Models

ocle 2.0 - OCL Environment

File Model Project Edit Tools Options Help

anocl  
Constraints  
anocl.bcr  
inv\_CORE\_N.ocd  
UML\_AO.ocd  
Models  
anocl.xml

Project UserModel Metamodel

Class properties

Name	N
Namespace	anocl
Visibility	public
Stereotypes	
Tagged Values	
Constraints	
Root	false
Leaf	false
Abstract	false
Features	
Active	true

NewClassDiagram

```
classDiagram
    class M {
        q:Real
    }
    class P
    class N {
        p:Real
    }
    class Q
    M "1" -- "+m" P
    N "1" -- "+n" Q
    N --|> M
```

C:\Users\Public\My Documents\Lausanne\Checking\_UML\_Models\inv\_CORE\_N.ocd

```
union(self.allContents.name)->includes(a.name)}

inv WFR_5_Classifier:
-- [5] The name of an opposite AssociationEnd may not be the same as the name of an Attribute or a ModelElement contained in the Classifier.

let innClassVisName: Bag(String) = (self.allAttributes.name->union(self.allContents.name))->reject(e| e.isUndefined)->reject(e| e=="") in
innClassVisName->excludesAll(self.oppositeAssociationEnds.name->reject(e| e==""))

inv WFR_6_Classifier:
-- [6] For each Operation in an specification realized by the Classifier, the Classifier must have a matching Operation
```

C:\Users\Public\My Documents\Lausanne\Checking\_UML\_Models\inv\_CORE\_N.ocd

Selection: Bag(String)=Bag{'q'}

Selection: Bag(String)=Bag{'q'}

Selection: Bag(String)=Bag{'p', 'q', 'Boolean', 'Integer', 'UnlimitedInteger', 'Real', 'String', 'undefined', 'M', 'N', 'P', 'Q'}

Selection: Bag(String)=Bag{'q'}

LOG Messages OCL output Evaluation Search results

Model checking finished.

117330KB 187392KB

---

Thanks for your patience!

