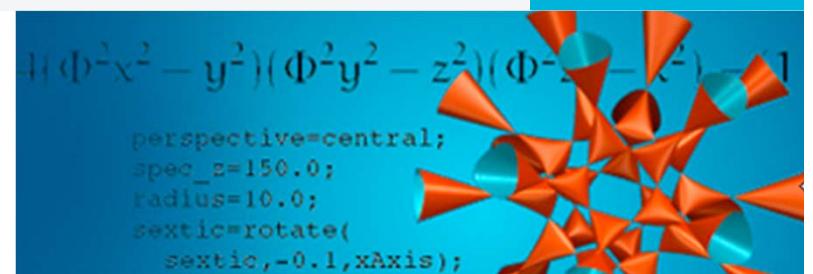


Improving the quality of EMF models using metrics, smells, and refactorings

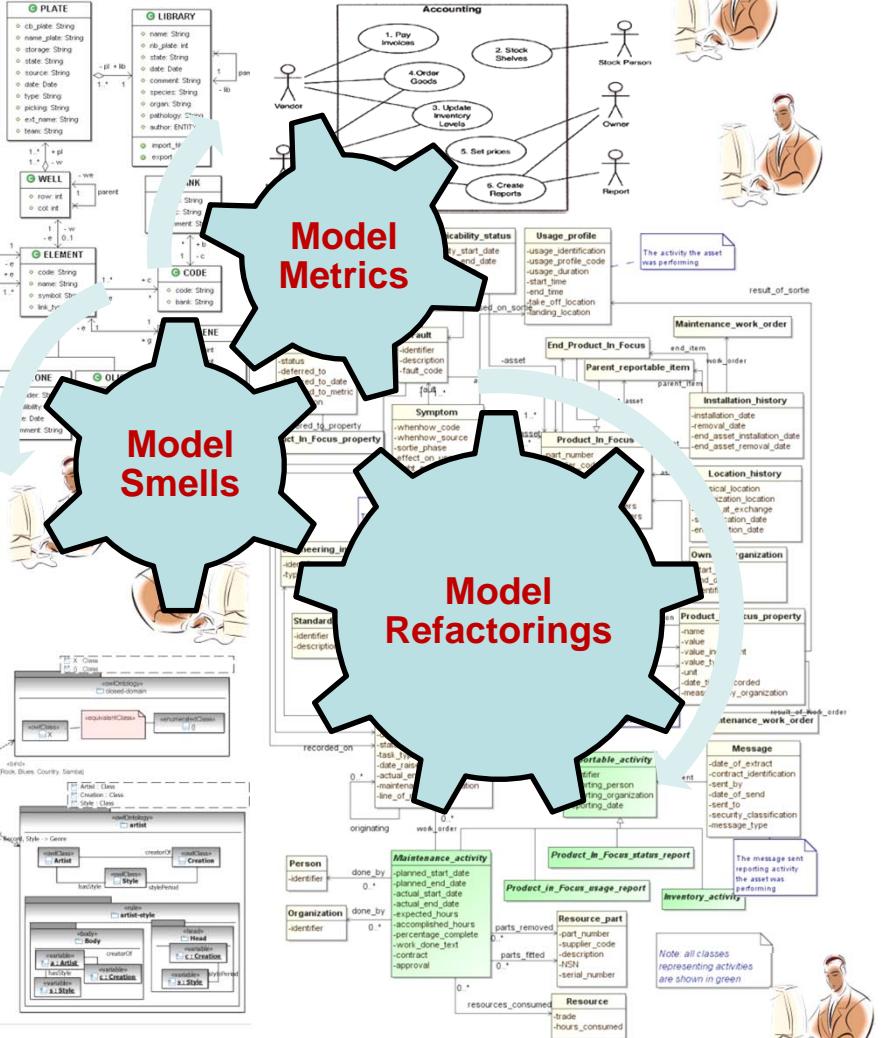
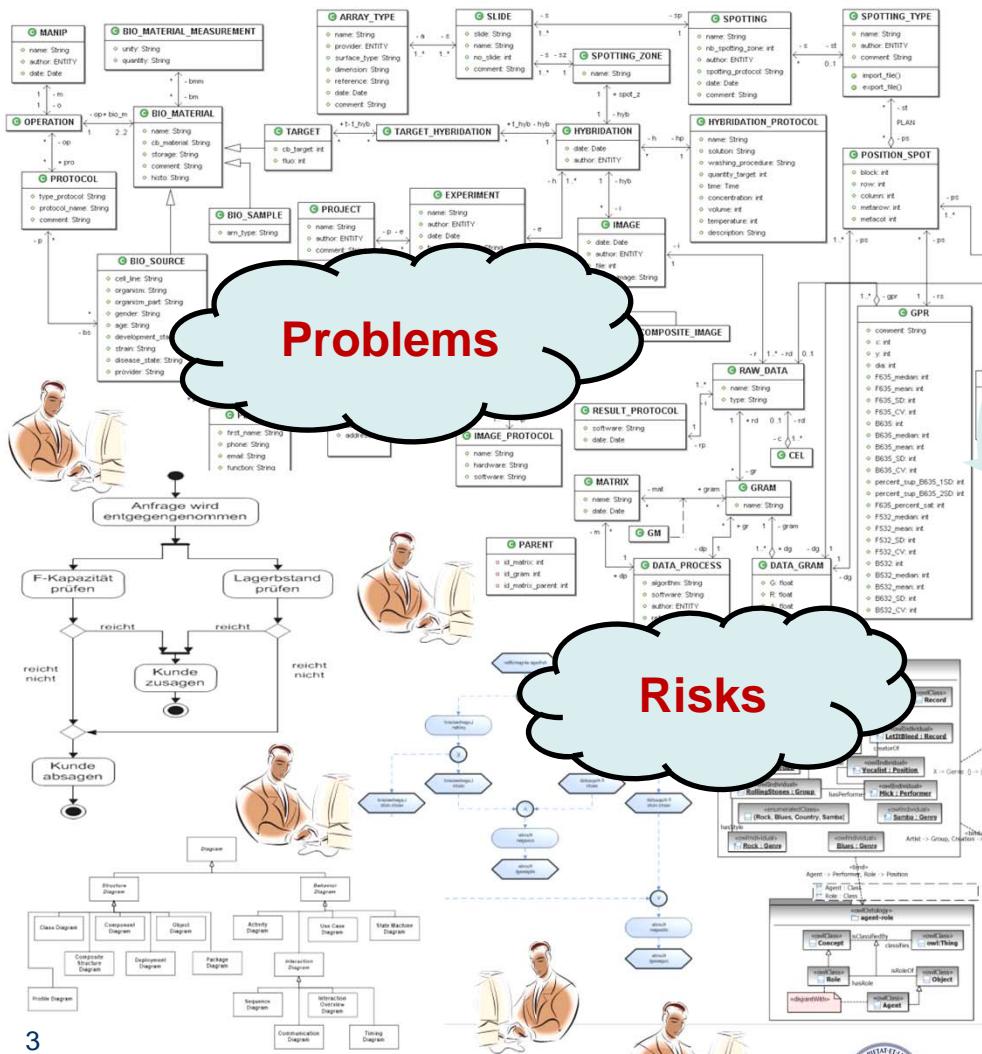
Gabriele Taentzer, Thorsten Arendt
Tutorial at ECMFA, July 2nd 2012, Lyngby, Denmark



Outline

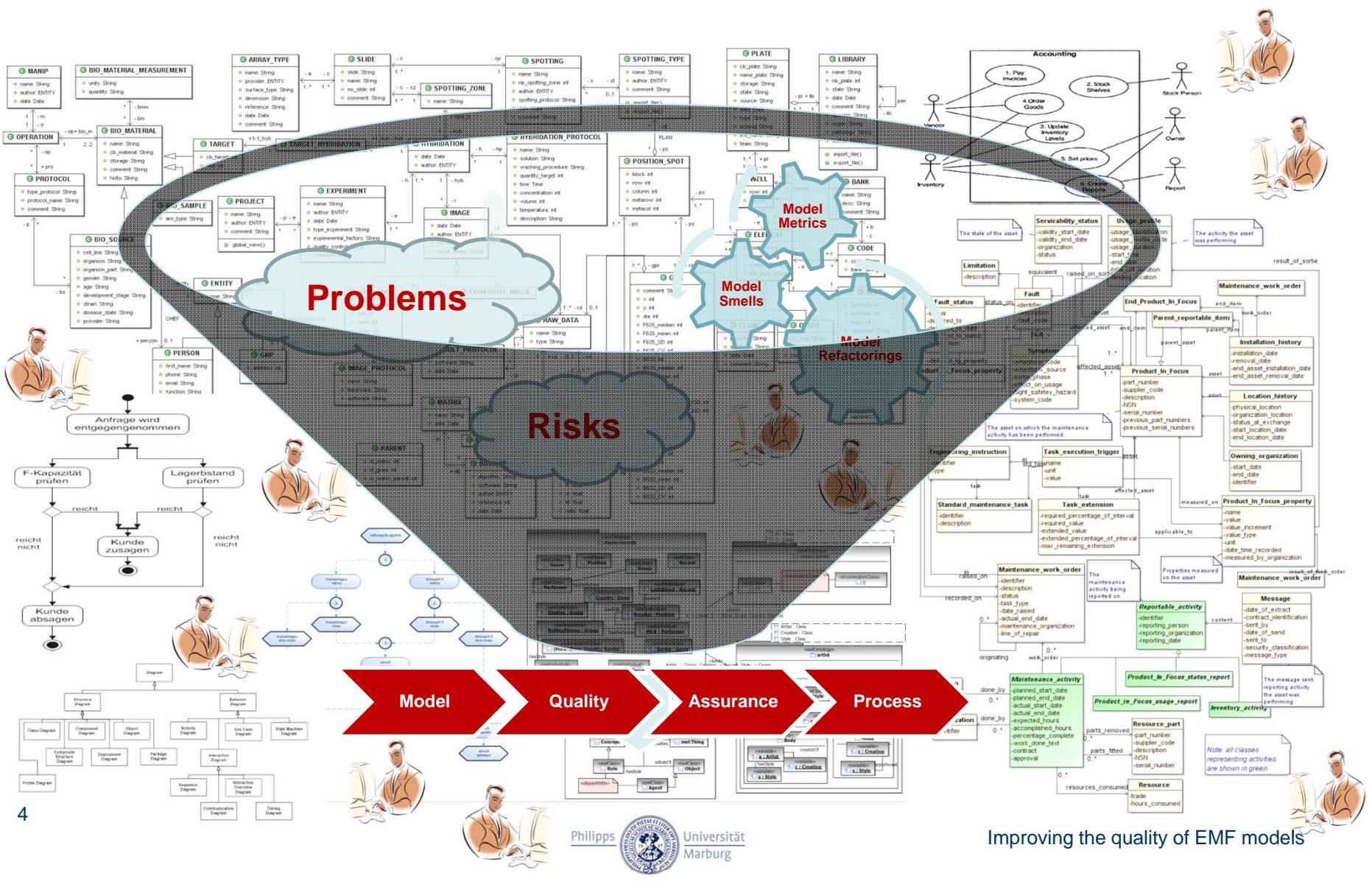
- Introduction
- Running Example
- Demo: Application of Model Quality Assurance Techniques
- Demo: Specification of Model Quality Assurance Techniques
- Conclusion

Motivation: Model quality assurance

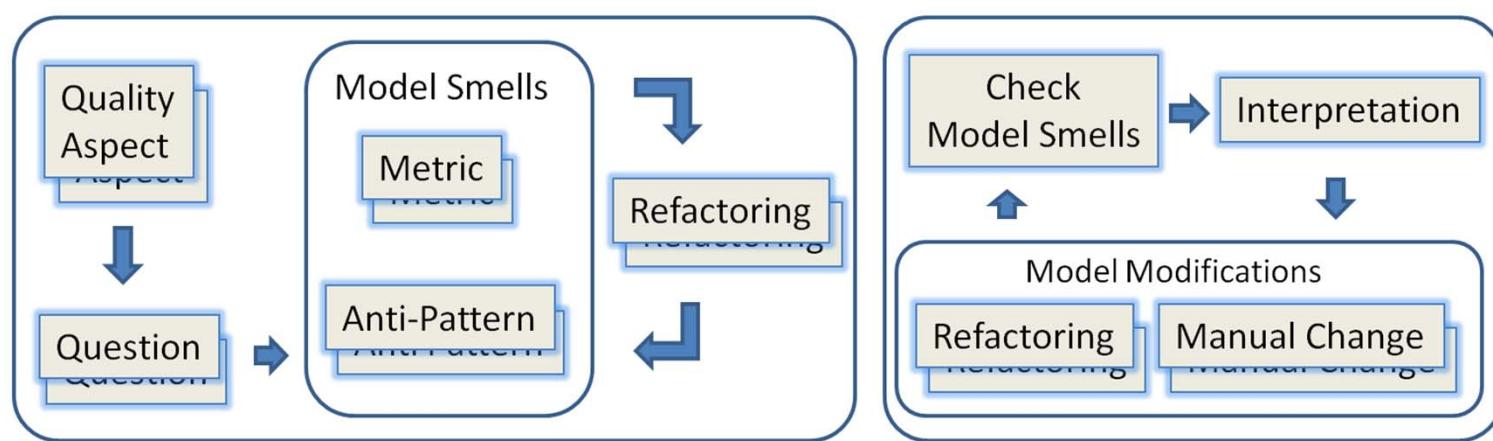


Improving the quality of EMF models

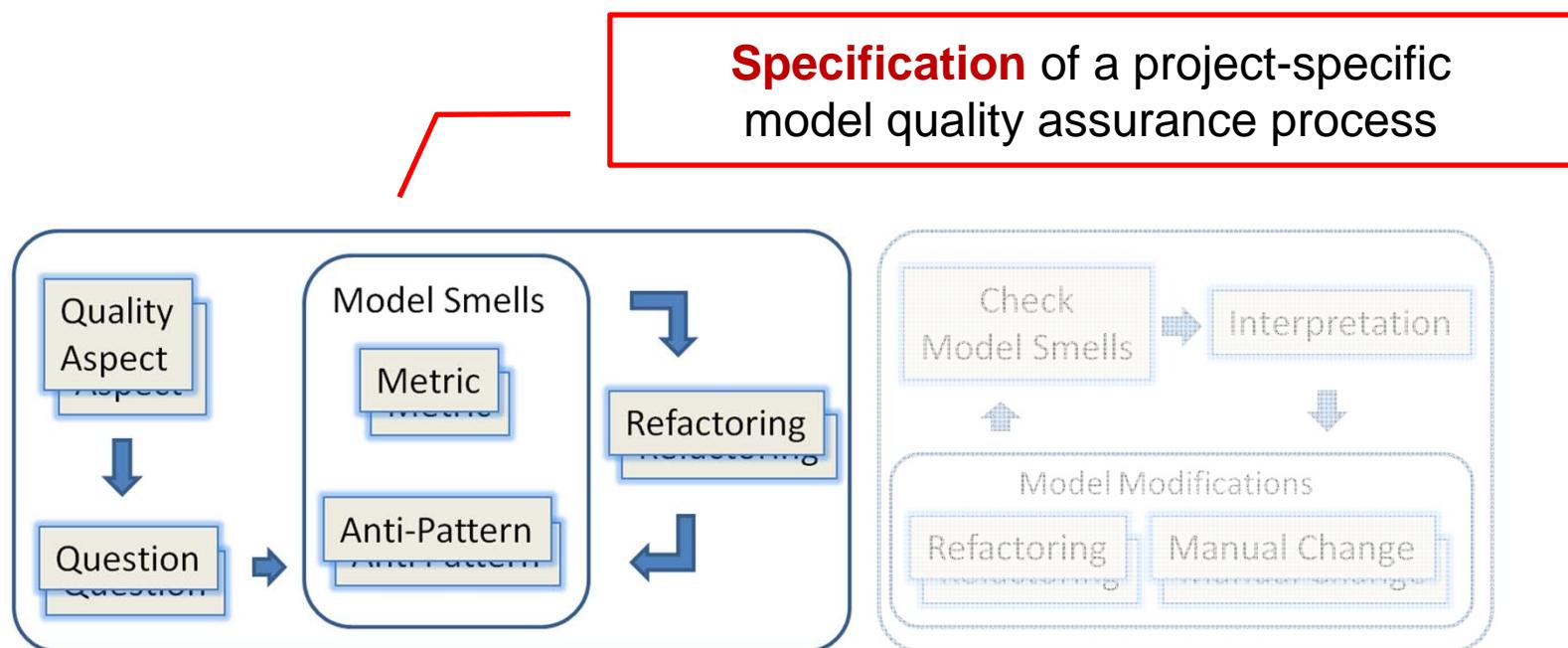
Motivation: Model quality assurance



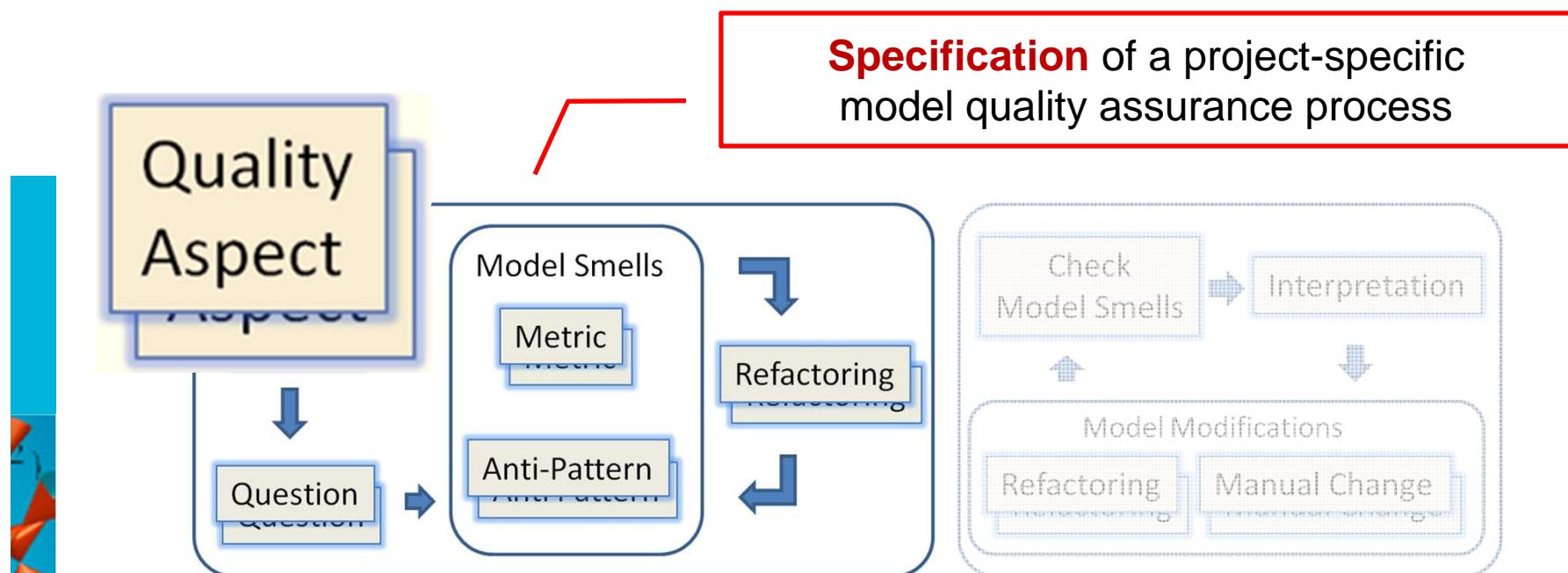
A model quality assurance process



A model quality assurance process



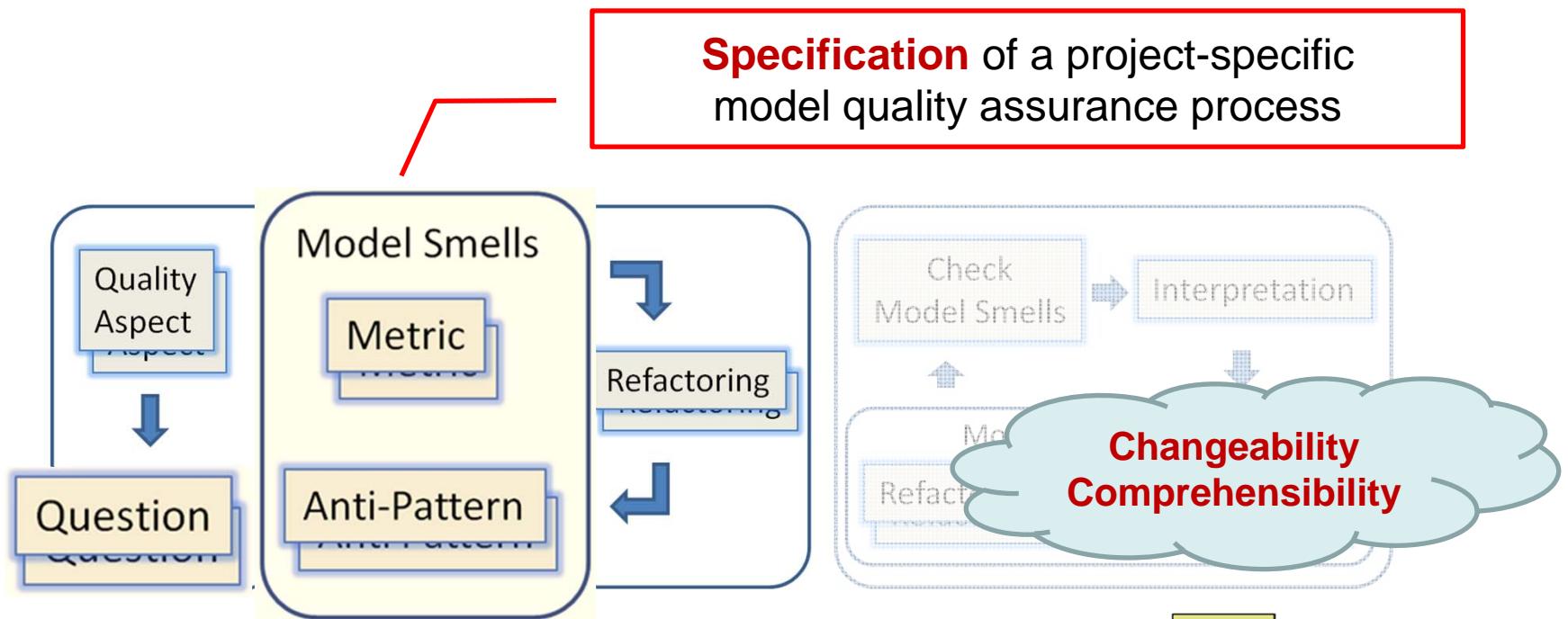
A model quality assurance process: 6C Goals¹



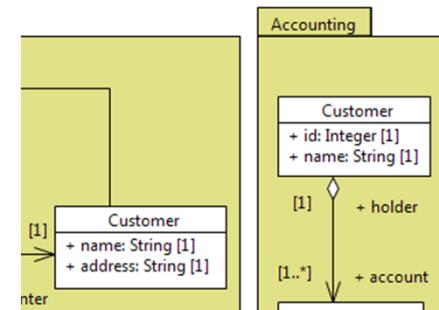
- **Correctness**
- **Completeness**
- **Changeability**
- **Comprehensibility**
- **Confinement**
- **Consistency**

¹ P. Mohagheghi, V. Dehlen, T. Neple:
Definitions and approaches to model quality in
model-based software development - A review of
literature. Information & Software Technology 51
(12): 1646-1669 (2009)

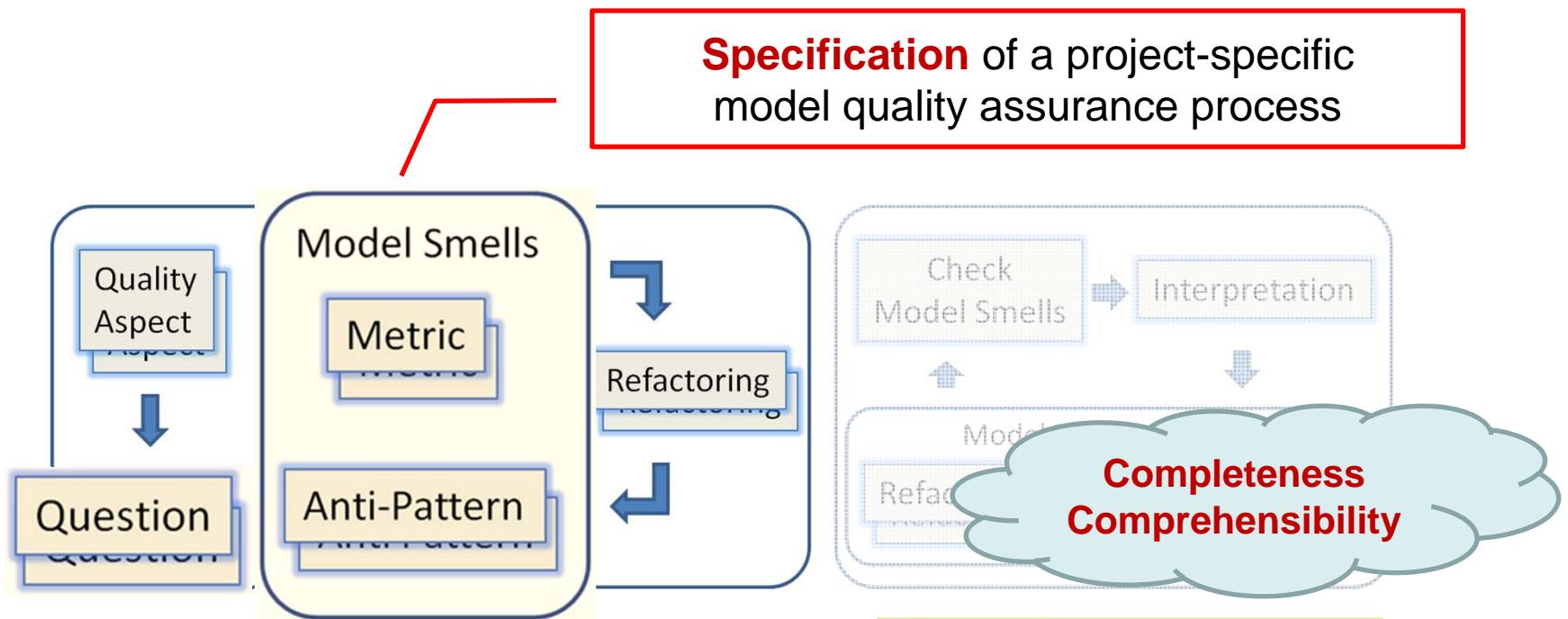
A model quality assurance process: Smells



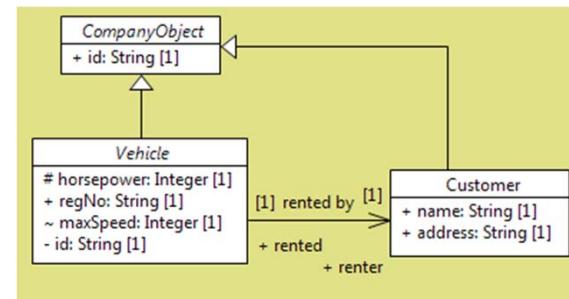
- Example: *Are there equally named classes in different packages?*
 - Smell: **Multiple definitions of classes with equal names**



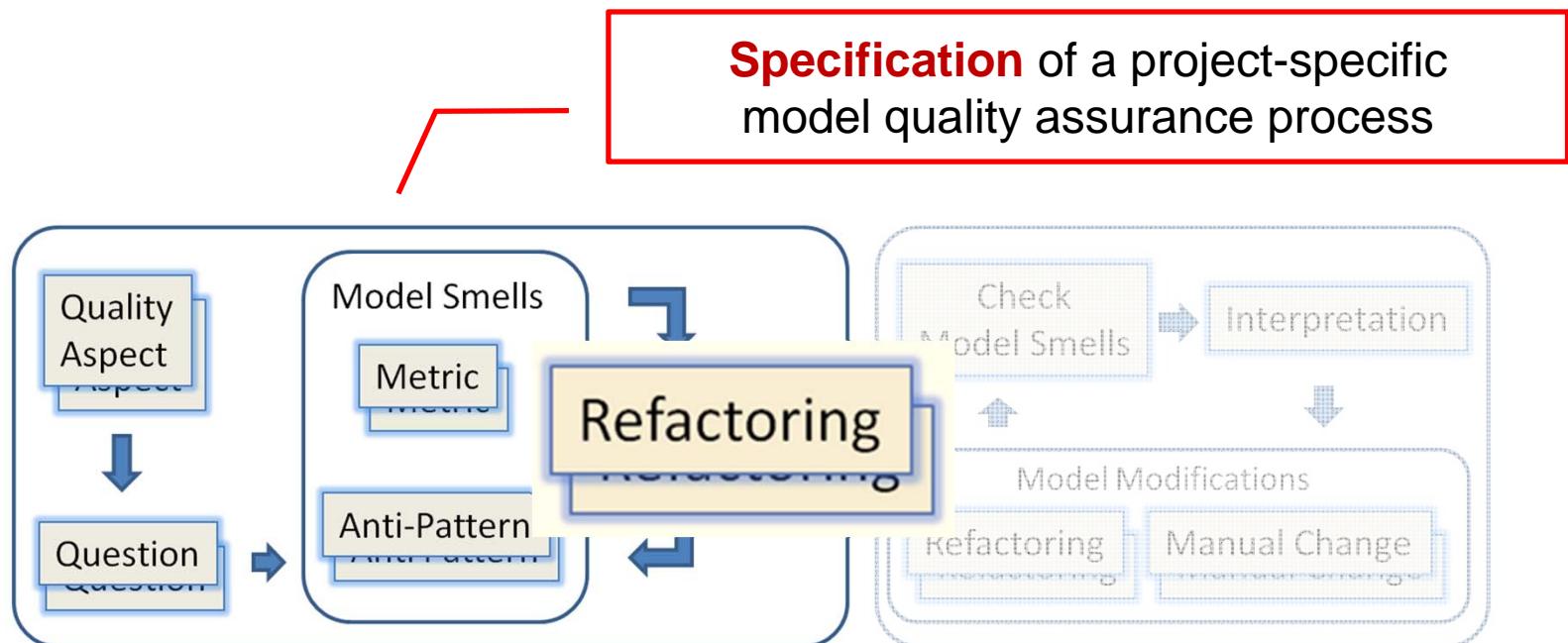
A model quality assurance process: Smells



- Example: *Are there abstract classes without any concrete subclasses?*
 - Smell: **No concrete subclass**

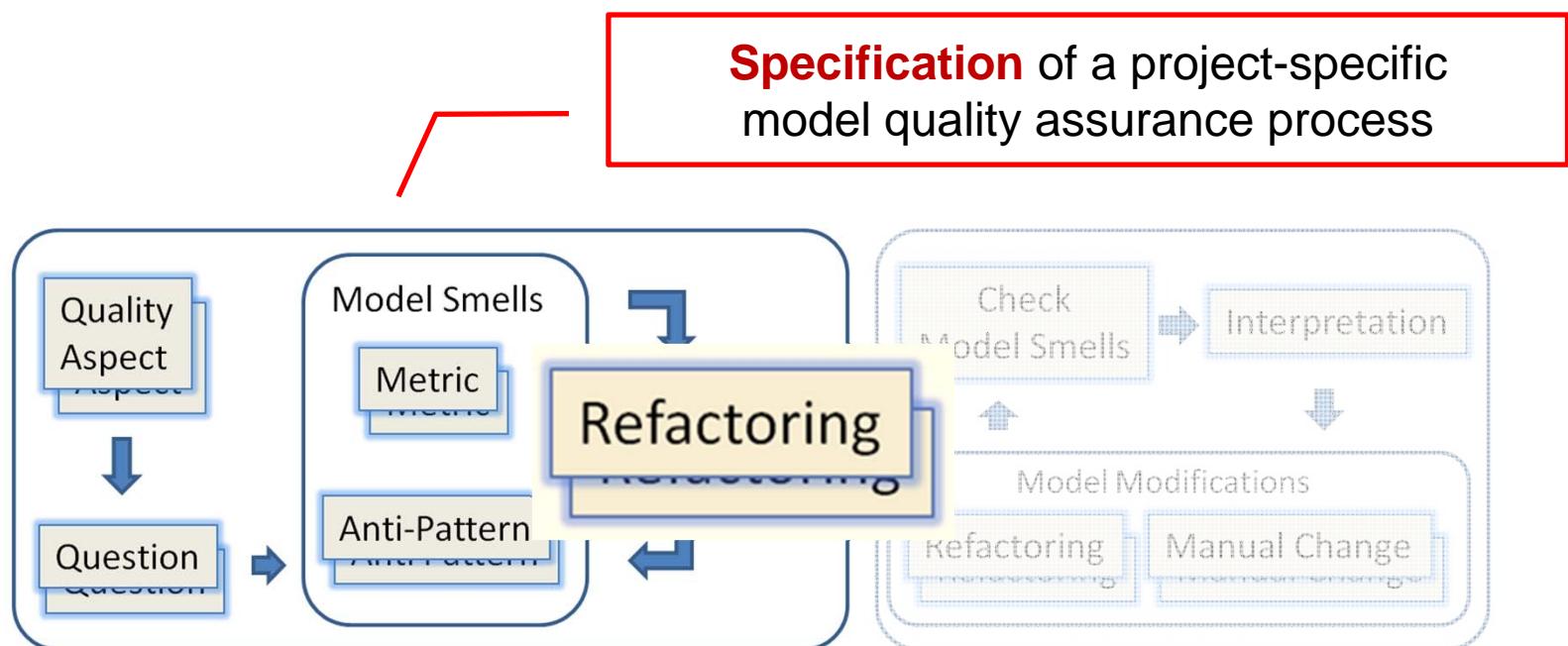


A model quality assurance process: Refactoring

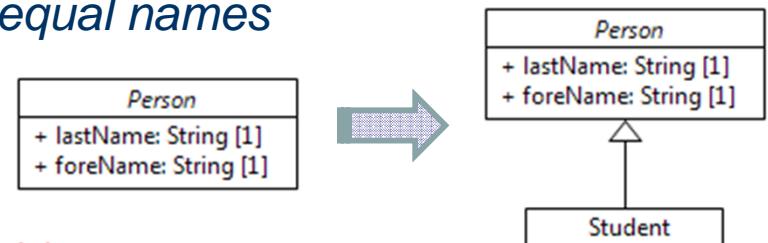


- Smell: *Multiple definitions of classes with equal names*
 - Refactoring: **Rename Class**

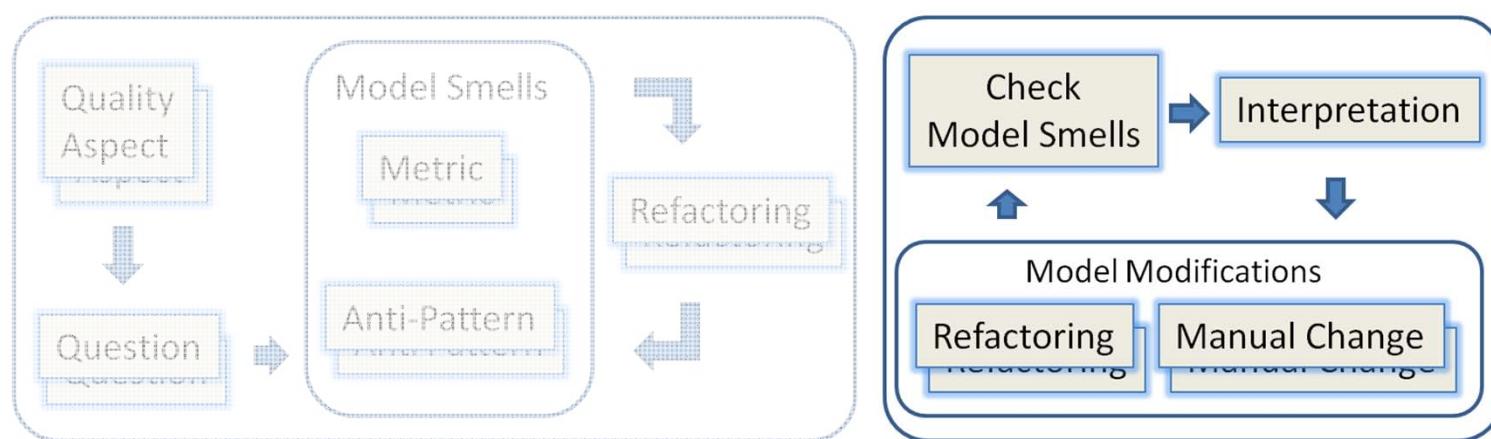
A model quality assurance process: Refactoring



- Smell: *Multiple definitions of classes with equal names*
 - Refactoring: **Rename Class**
- Smell: *No concrete subclass*
 - Refactoring: **Insert Concrete Subclass**

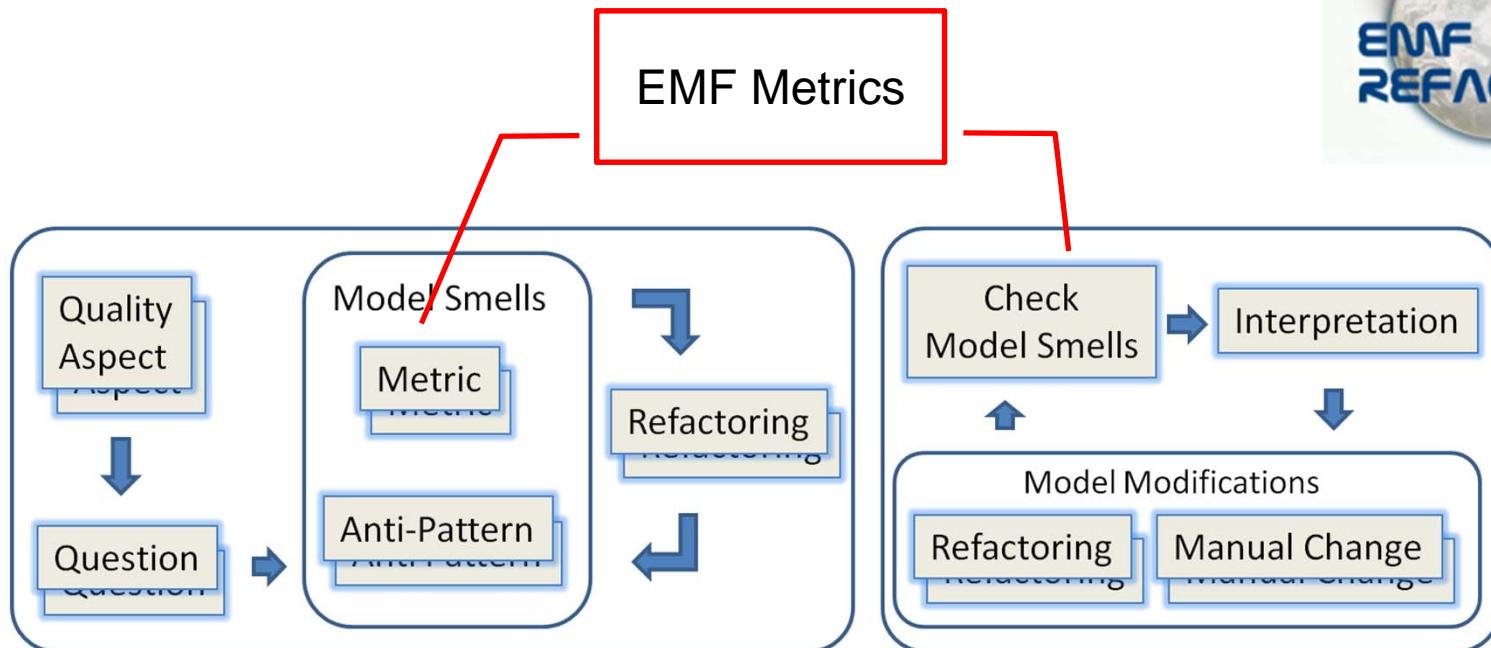


A model quality assurance process

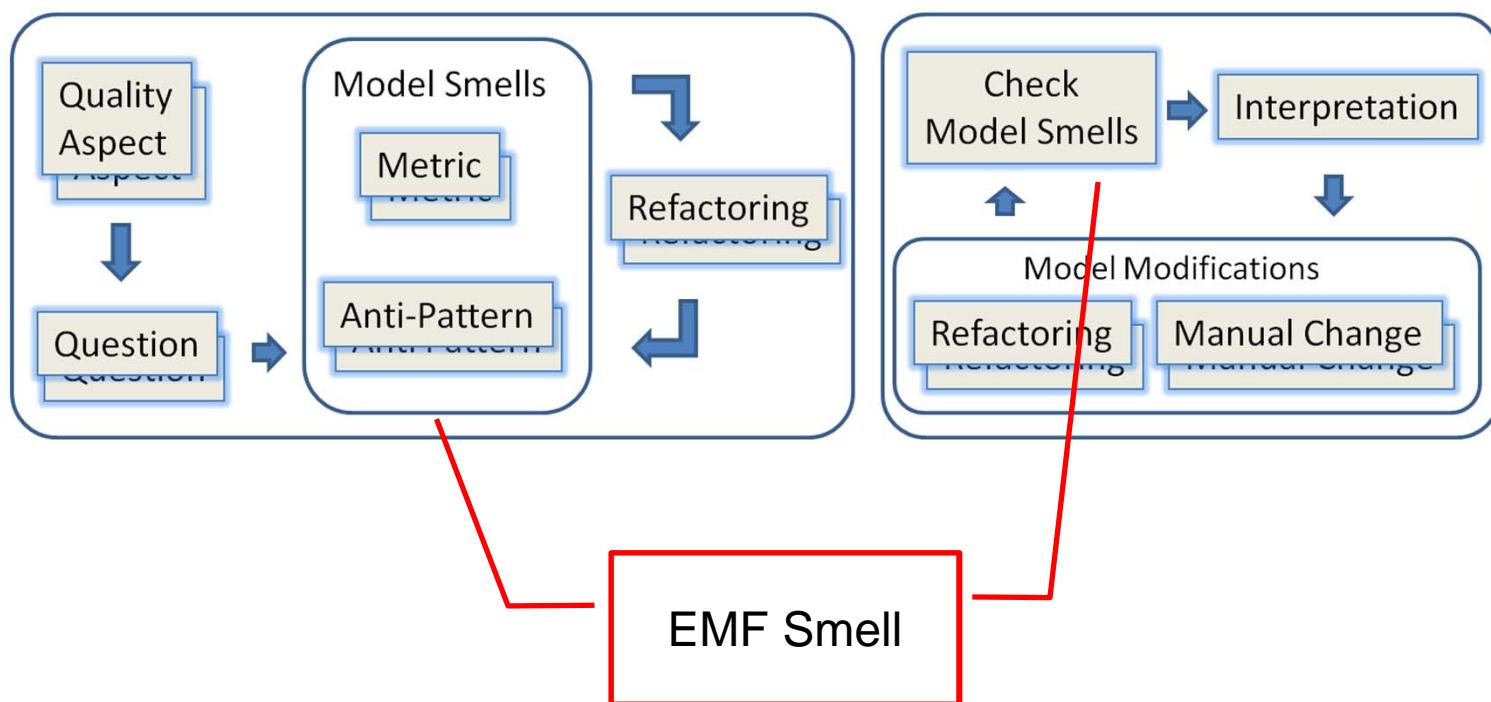


Application of the specified process to concrete software models

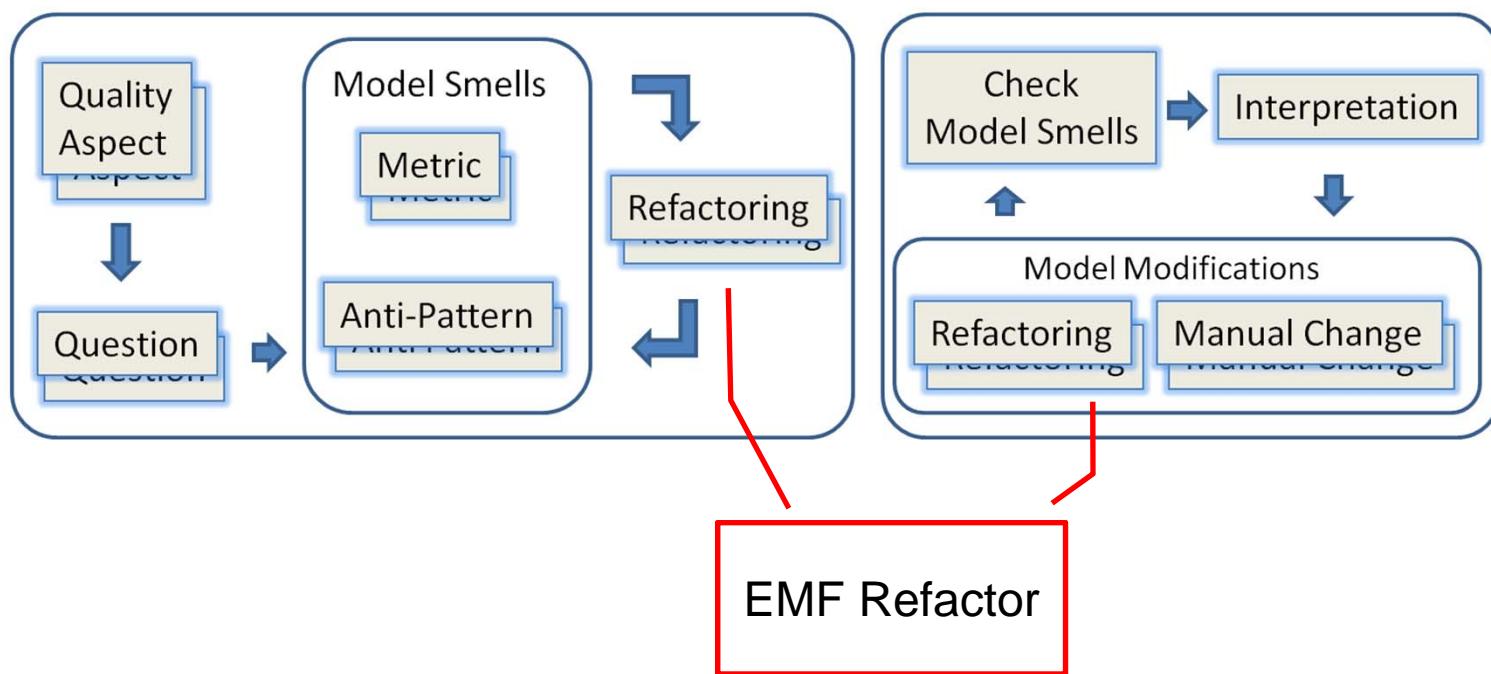
Tool support in Eclipse



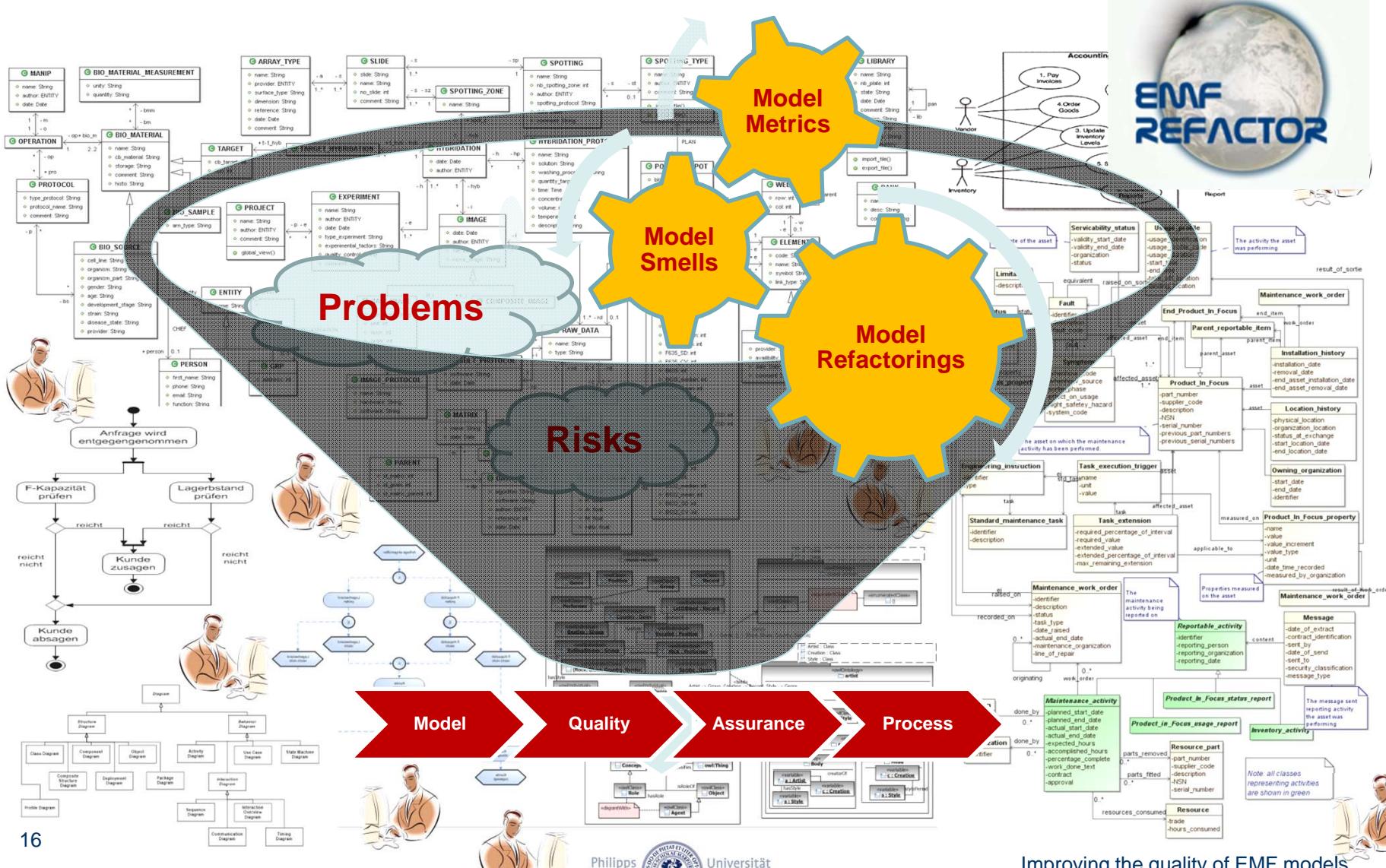
Tool support in Eclipse



Tool support in Eclipse



Topic of Tutorial

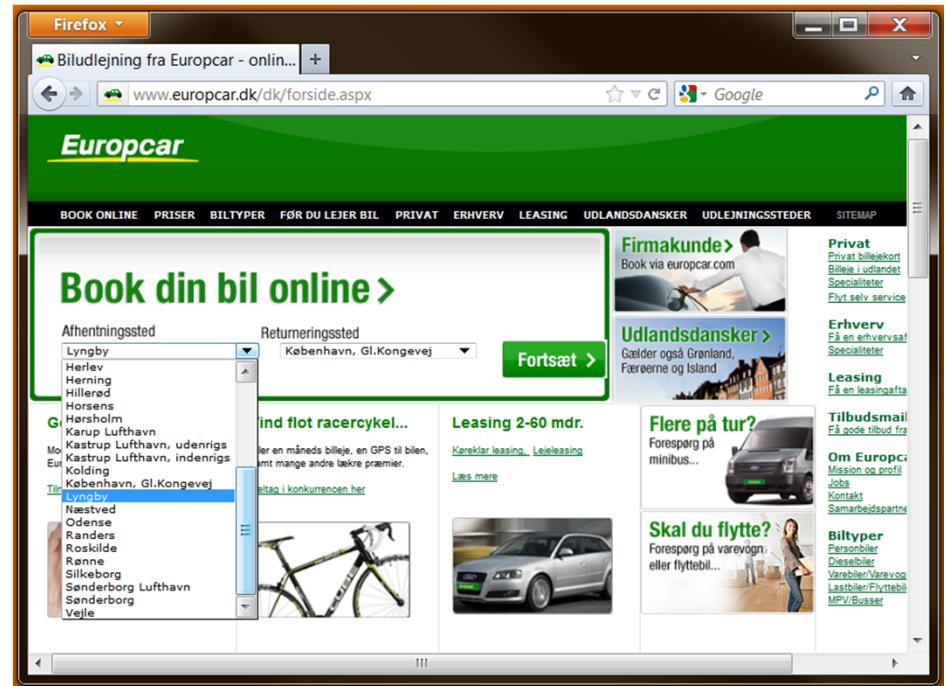


Outline

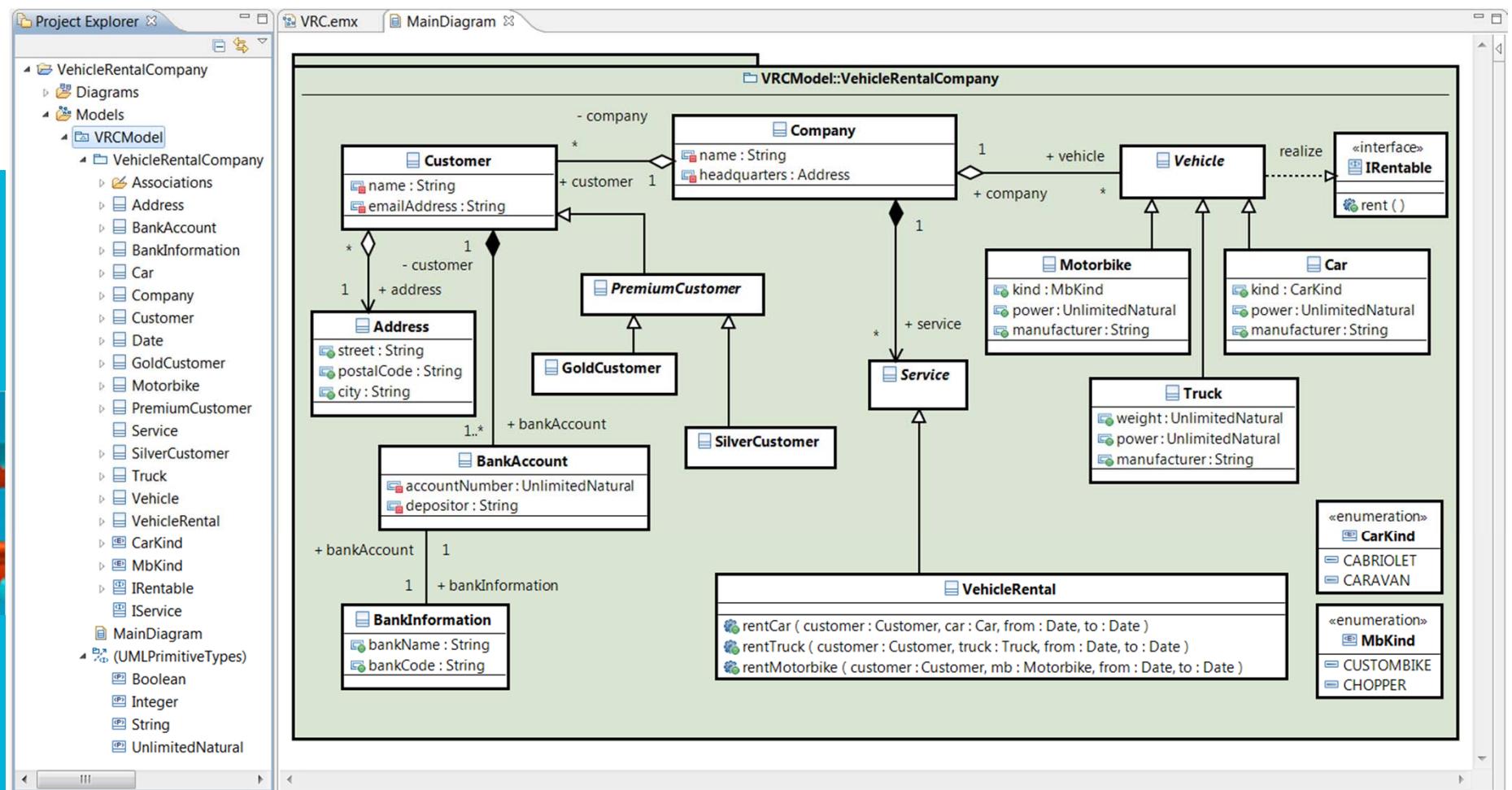
- Introduction
- **Running Example**
- Demo: Application of Model Quality Assurance Techniques
- Demo: Specification of Model Quality Assurance Techniques
- Conclusion

Example: Vehicle Rental Company

- Project
 - Development of a web application for renting vehicles
- Semantic domain
 - Company with
 - Vehicles
 - Customers
 - Services
- Technical domain
 - Web application
- Modeling purpose
 - Modeling of the semantic domain



Running example: UML class model



Outline

- Introduction
- Running Example
- Demo: **Application of Model Quality Assurance Techniques**
- Coffee Break
- Demo: Specification of Model Quality Assurance Techniques
- Conclusion



EMF Metrics: configuration dialog

Properties for VehicleRentalCompany

type filter text

Resource
Builders
EMF Metrics
EMF Refactor
EMF Relations
EMF Smells
Project References
Refactoring History
Run/Debug Settings

EMF Metrics

Please select EMF metrics from the list below to be supported by the project.

<http://www.eclipse.org/uml2/2.1.0/UML>

Selected	Name	Description
<input type="checkbox"/>	NPM	Number of packages in the model.
<input type="checkbox"/>	NCM	Number of classes in the model.
<input type="checkbox"/>	NASM	Number of associations in the model.
<input type="checkbox"/>	NAGM	Number of aggregations in the model.
<input type="checkbox"/>	NIM	Number of inheritance relations in the model.
<input checked="" type="checkbox"/>	NATM	Number of attributes in classes within the model.
<input checked="" type="checkbox"/>	NOM	Number of operations in classes within the model.
<input type="checkbox"/>	NACTM	Number of attributes with class type in the model.
<input type="checkbox"/>	NNOEM	Number of navigable owned association ends in the model.
<input type="checkbox"/>	NACM	Number of all ancestors of all classes in the model.
<input type="checkbox"/>	ASvsC	Ratio between number of associations and number of classes in the model.
<input type="checkbox"/>	AGvsC	Ratio between number of aggregations and number of classes in the model.
<input type="checkbox"/>	GEvsC	Ratio between number of inheritance relations and number of classes in the ...
<input type="checkbox"/>	ATvsCC	Ratio between number of attributes and number of classes in the models.
<input type="checkbox"/>	OvsC	Ratio between number of operations and number of classes in the model.
<input type="checkbox"/>	NTDM	Number of class type dependencies in the model.
<input type="checkbox"/>	NDEPM	Number of class dependencies in the model.
<input type="checkbox"/>	MaxHAgg	Maximum of aggregation trees.
<input checked="" type="checkbox"/>	MaxDIT	Maximum of all depths of inheritance trees.
<input checked="" type="checkbox"/>	ANA	Average number of ancestors of all classes in the model.
<input checked="" type="checkbox"/>	NIH	Total number of inheritance hierarchies in the model.
<input type="checkbox"/>	TNME	Total number of elements in the model.

org.eclipse.uml2.uml.Package

Selected	Name	Description
----------	------	-------------

② OK Cancel

EMF Metrics: result view



The screenshot shows the EMF Refactor interface. The top part displays a UML class diagram for the 'VehicleRentalCompany' package. The bottom part shows the 'Metrics' tab of the results table.

UML Class Diagram (MainDiagram):

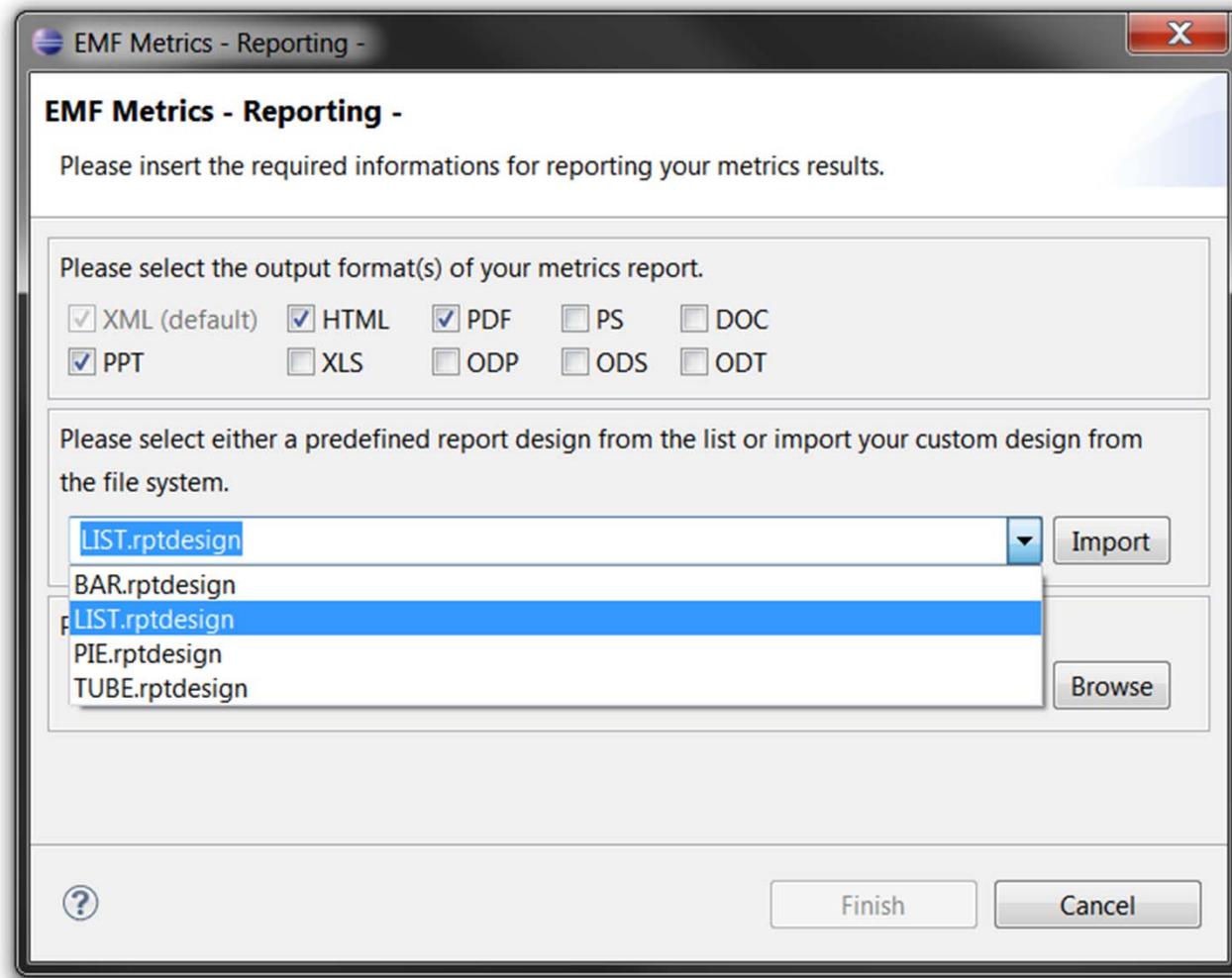
```

classDiagram
    package VRCModel {
        package VehicleRentalCompany {
            Customer
            Address
            BankAccount
            BankInformation
            Car
            Company
            Date
            GoldCustomer
            Motorbike
            PremiumCustomer
            Service
            SilverCustomer
            Truck
            Vehicle
            VehicleRental
        }
        Company
        Vehicle
        IRentable
        rent()
        Service
        Motorbike
        Car
        Truck
        PremiumCustomer
        GoldCustomer
        SilverCustomer
        Address
        BankAccount
    }
    Customer "*" -- "1" Company : - company
    Customer "*" -- "1" Address : + customer
    Address "*" -- "1" Customer : - address
    Address "*" -- "1.." BankAccount : + bankAccount
    Company "1" -- "1" Vehicle : + vehicle
    Company "1" -- "1" Service : + service
    Vehicle "*" -- "1" Company : + company
    Service "*" -- "1" Vehicle : + vehicle
    Service "*" -- "1" Motorbike : + vehicle
    Service "*" -- "1" Car : + vehicle
    Service "*" -- "1" Truck : + vehicle
    Motorbike "*" -- "1" Vehicle : + vehicle
    Car "*" -- "1" Vehicle : + vehicle
    Truck "*" -- "1" Vehicle : + vehicle
    PremiumCustomer --> GoldCustomer
    PremiumCustomer --> SilverCustomer
    GoldCustomer --> SilverCustomer
    Company "1" -- "1" IRentable : realize
    IRentable "1" -- "1" rent()
    
```

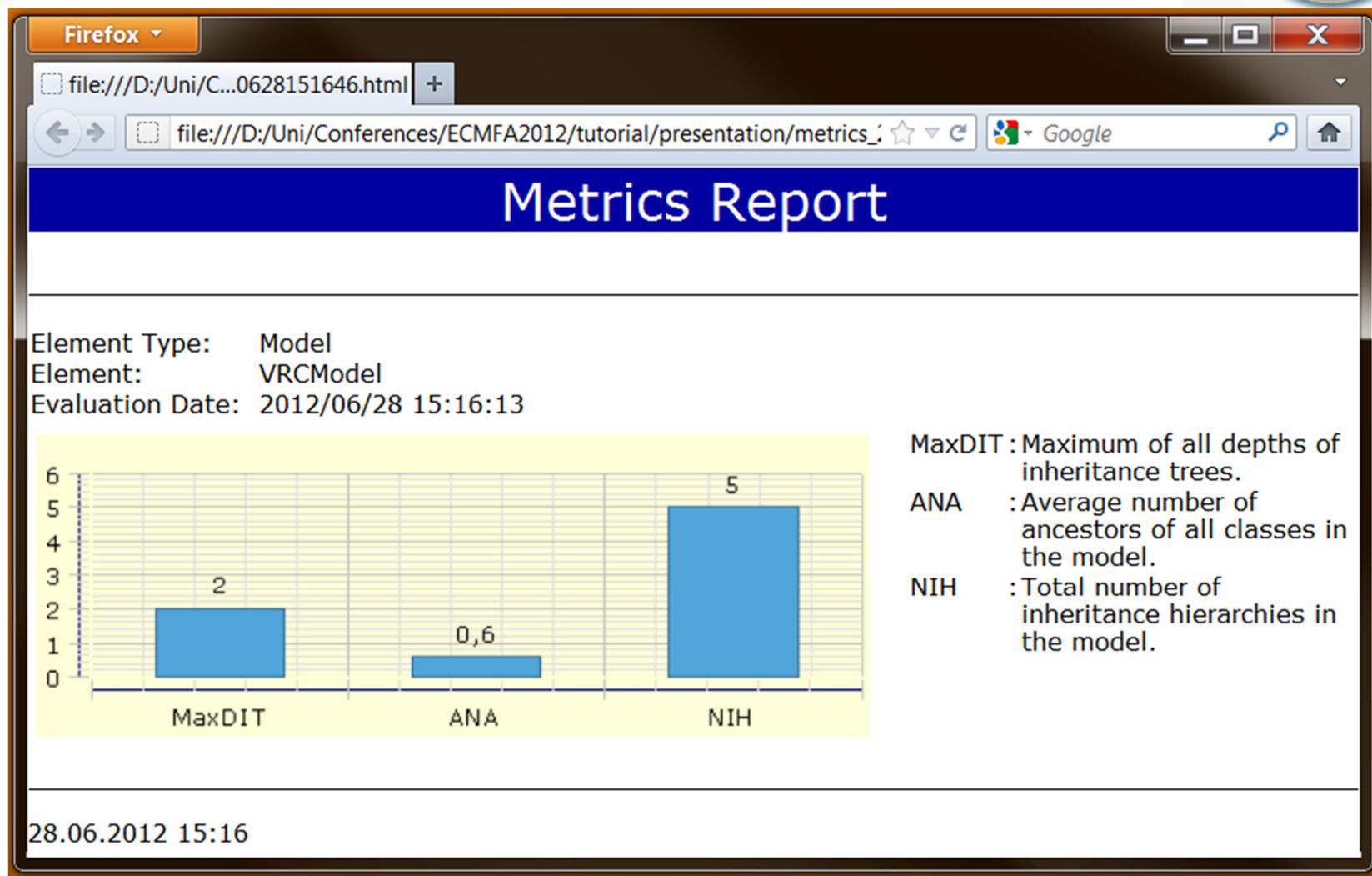
Metrics Results Table:

Time	Context	Metric	Description	Result
2012/06/28 15:13:50	Model VRCModel	NATM	Number of attributes in classes within the model.	33.00
2012/06/28 15:13:50	Model VRCModel	NOM	Number of operations in classes within the model.	4.00
2012/06/28 15:13:50	Model VRCModel	MaxDIT	Maximum of all depths of inheritance trees.	2.00
2012/06/28 15:13:50	Model VRCModel	ANA	Average number of ancestors of all classes in the model.	0.60
2012/06/28 15:13:50	Model VRCModel	NIH	Total number of inheritance hierarchies in the model.	5.00
2012/06/28 15:13:50	Package VehicleRentalCompany	NCP	Number of classes within the package.	15.00
2012/06/28 15:13:50	Package VehicleRentalCompany	NIP	Number of interfaces within the package.	2.00
2012/06/28 15:13:50	Package VehicleRentalCompany	NAP	Number of associations within the package.	6.00
2012/06/28 15:13:50	Package VehicleRentalCompany	NAGGR	Number of aggregation relationships within the package.	5.00
2012/06/28 15:13:50	Package VehicleRentalCompany	A	Ratio between number of abstract classes and number of classes within the package.	0.20
2012/06/28 15:13:50	Package VehicleRentalCompany	NAVCP	Ratio between number of associations and number of classes within the package.	0.40

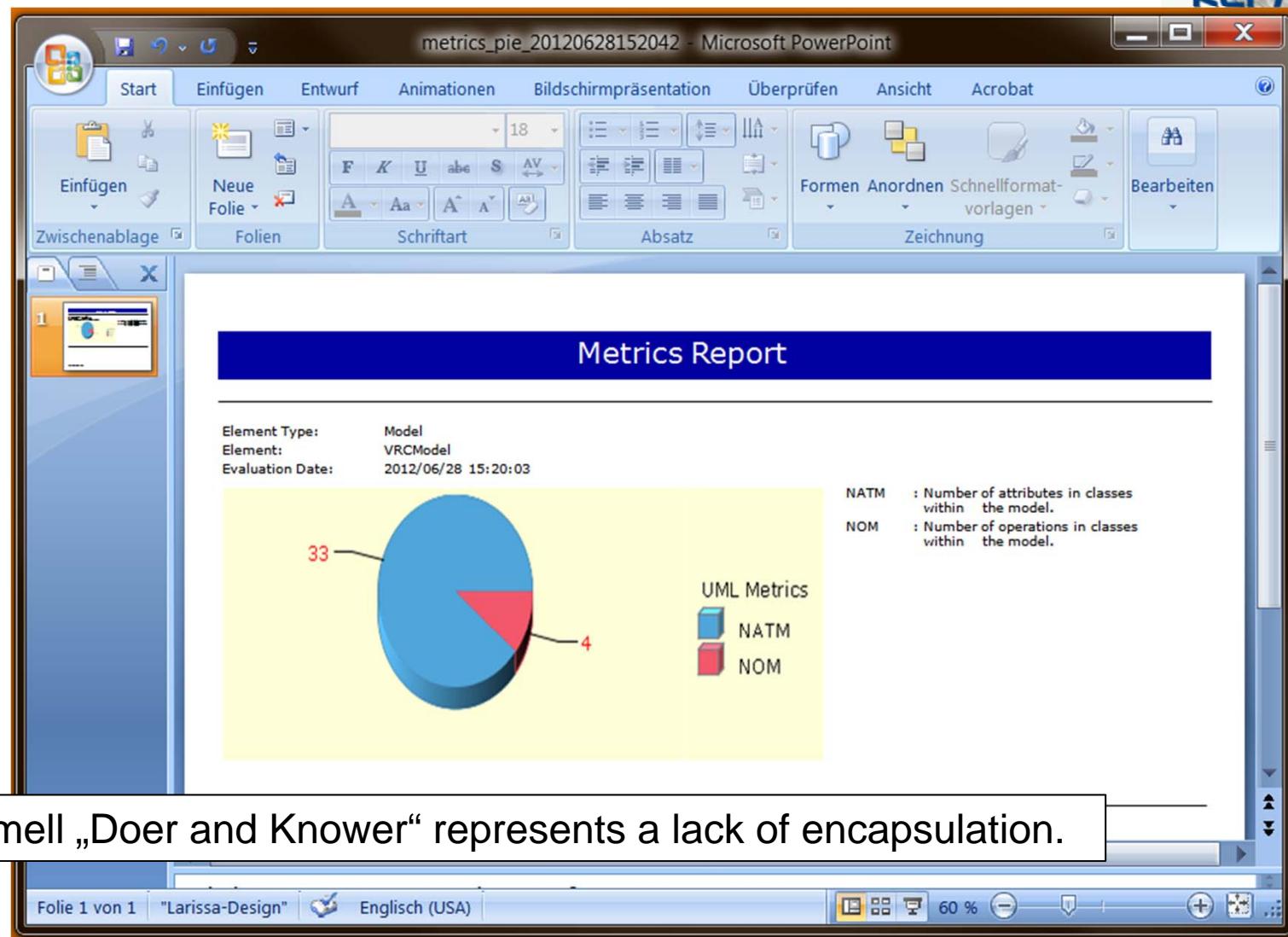
EMF Metrics: export dialog



EMF Metrics: HTML export (bar diagram)



EMF Metrics: PPT export (pie diagram)



EMF Smell: configuration dialog



Properties for VehicleRentalCompany

type filter text

Resource
Builders
EMF Metrics
EMF Refactor
EMF Relations
EMF Smells
Project References
Refactoring History
Run/Debug Settings

Please select EMF smells from the list below to be supported by the project.

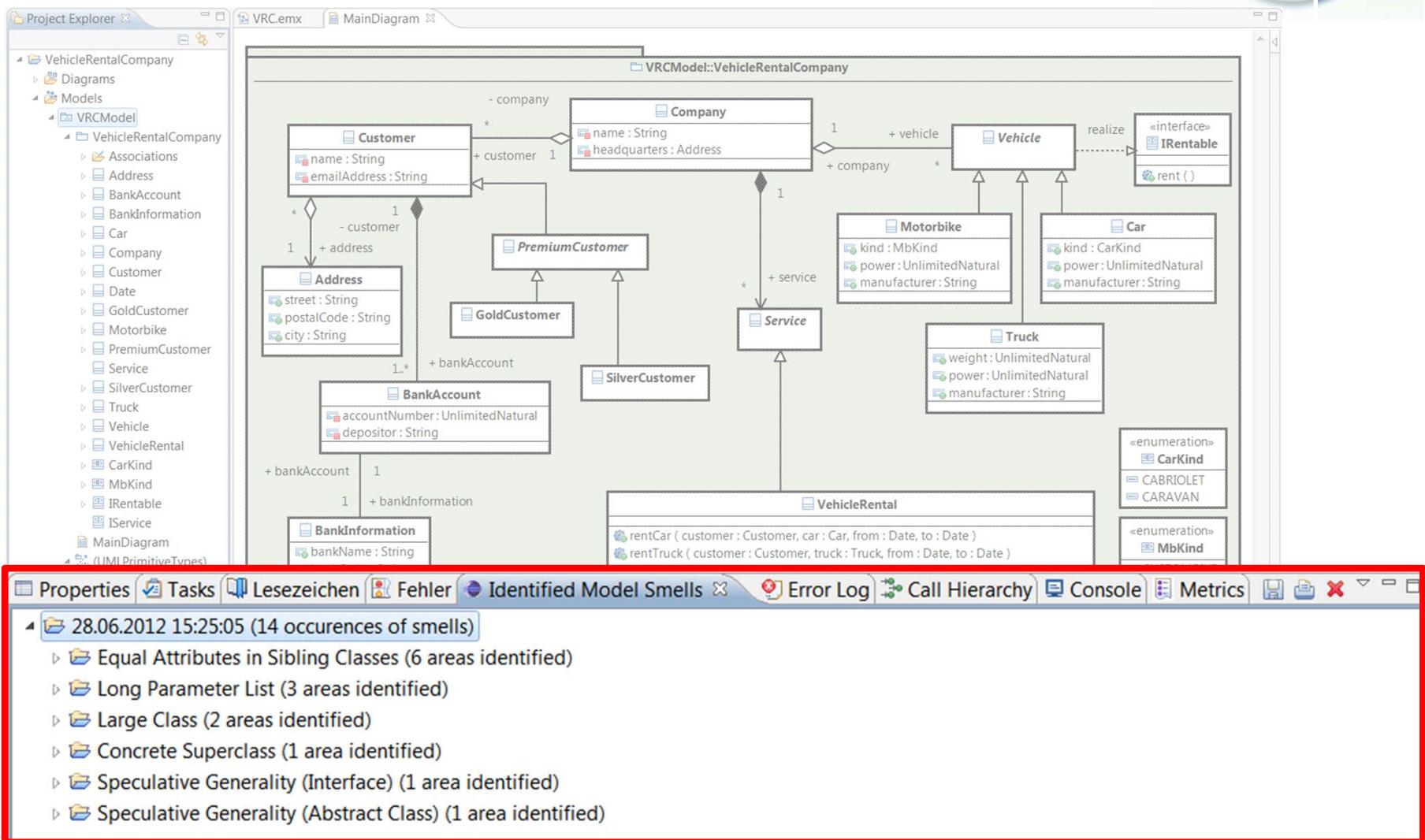
http://www.eclipse.org/uml2/2.1.0/UML

Selected	Name	Description	Limit
<input type="checkbox"/>	Unused Use Case	The model contains a use case that is not associated to any actors.	
<input type="checkbox"/>	Extends	The model contains a use case that extends another use case.	
<input type="checkbox"/>	No Incoming	The model contains a state without incoming transitions.	
<input type="checkbox"/>	Unnamed Class	The model contains a class without a name.	
<input type="checkbox"/>	Unnamed Interface	The model contains an interface without a name.	
<input type="checkbox"/>	Unused Enumeration	The model contains an enumeration whose literals are not used as ...	
<input checked="" type="checkbox"/>	Unused Class	The model contains a class that has no child or parent classes, that i...	
<input checked="" type="checkbox"/>	Multiple Definition of Classes with...	The model contains two classes (in different packages) having the s...	
<input type="checkbox"/>	Attribute name overridden	The model contains an attribute with the same name as an inherite...	
<input checked="" type="checkbox"/>	Concrete Superclass	The model contains an abstract class with a concrete superclass.	
<input checked="" type="checkbox"/>	Specialization Aggregation	The model contains a generalization hierarchy between associations.	
<input checked="" type="checkbox"/>	Speculative Generality (Interface)	The model contains an interface that is implemented by one single ...	
<input checked="" type="checkbox"/>	Speculative Generality (Abstract C...	The model contains an abstract class that is inherited by one single ...	
<input checked="" type="checkbox"/>	Large Class	The model contains a class owning more features (attributes and o... 4.0	
<input checked="" type="checkbox"/>	Long Parameter List	The model contains an operation with more input parameters than ... 3.0	
<input type="checkbox"/>	Primitive Obsession (Constants)	The model contains a class with more constant attributes than the s... 0.0	
<input checked="" type="checkbox"/>	Primitive Obsession (Primitive Typ...	The model contains a class with more attributes having primitive ty... 3.0	
<input type="checkbox"/>	Data Clumps (Attributes)	The model contains classes with more than a specific number of eq... 0.0	
<input checked="" type="checkbox"/>	Data Clumps (Parameters)	The model contains operations with more than a specific number o... 3.0	
<input checked="" type="checkbox"/>	Equal Attributes in Sibling Classes	Each sibling class of the owning class of an attribute contains an eq...	

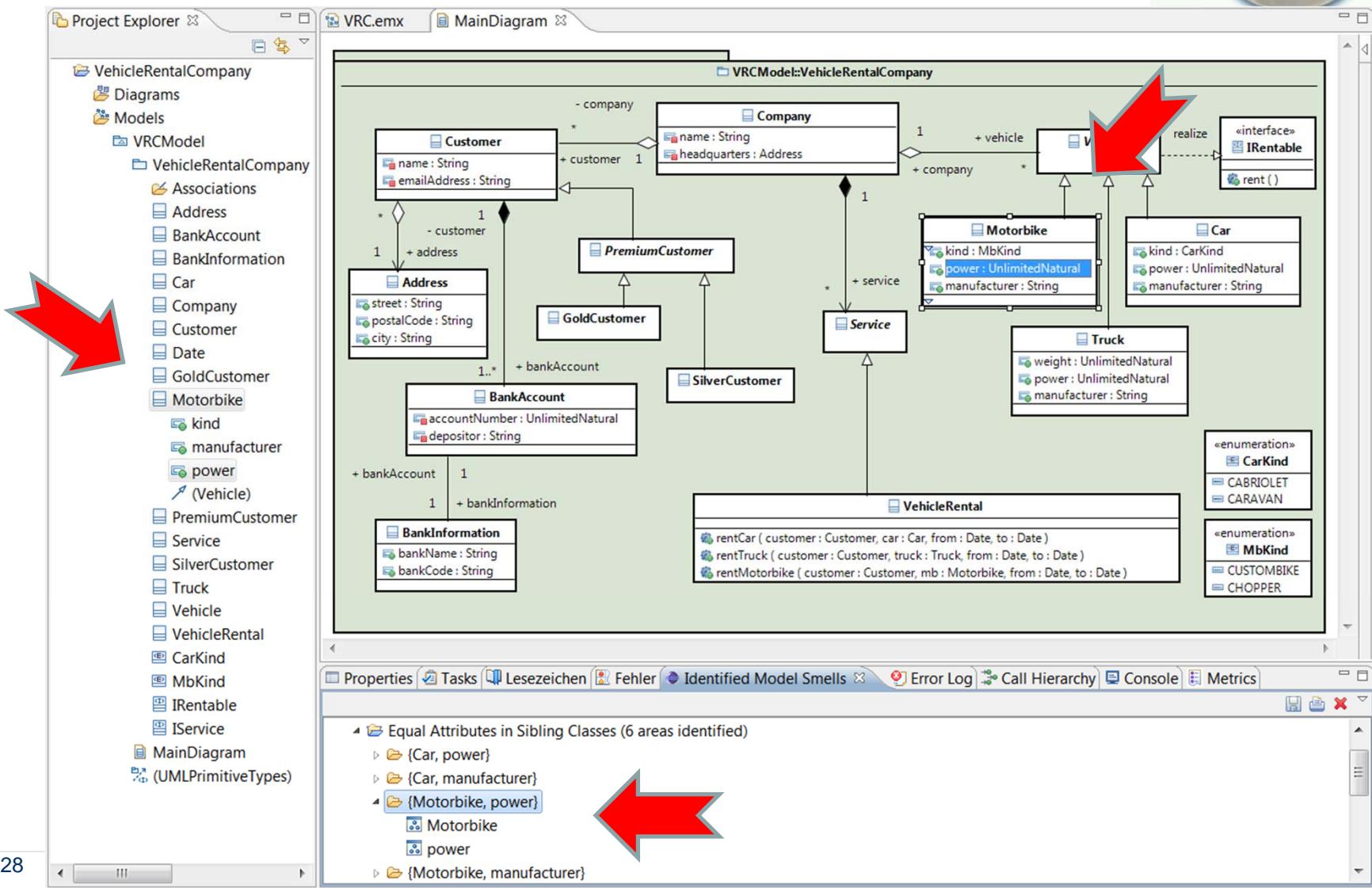
?

OK Cancel

EMF Metrics: result view



EMF Smell: highlighting



EMF Smell: PDF export (list)



smells_20120628153128.pdf - Adobe Acrobat Pro

Datei Bearbeiten Anzeige Fenster Hilfe

Erstellen Lesezeichen Werkzeuge Kommentar Freigaben

Smell Detection Analysis Report

Smell	Description
Concrete Superclass	The model contains an abstract class with a concrete superclass.
Equal Attributes in Sibling Classes	Each sibling class of the owning class of an attribute contains an equal attribute.
Large Class	
Long Parameter List	
Speculative Generality (Abstract Class)	
Speculative Generality (Interface)	

Analysis No. 1
Analysed File VRC.emx
Date of Analysis Thu Jun 28 15:28:44 CEST 2012

Smell Concrete Superclass

Description The model contains an abstract class with a concrete superclass.

Occurrences 1

Occurrence No. 1
Involved Element VRCModel::VehicleRentalCompany::PremiumCustomer

Smell Equal Attributes in Sibling Classes

Description Each sibling class of the owning class of an attribute contains an equal attribute.

Occurrences 6

Occurrence No. 1
Involved Element VRCModel::VehicleRentalCompany::Car

models

Relation: smell erasing refactorings



Properties for de.unimarburg.swt.emf.fix.relations

type filter text

- Resource
- Builders
- EMF Metrics
- EMF Refactor
- EMF Relations**
- EMF Smells
- > Plug-in Development
- Project References
- Run/Debug Settings

EMF Relations

Please set the relations between model smells and model refactorings as defined in this project.

Meta model: <http://www.eclipse.org/uml2/2.1.0/UML>

Refactorings useable to eliminate a given smell Smells possibly introduced by a given refactoring

Smell: Equal Attributes in Sibling Classes

Selected	Name
<input checked="" type="checkbox"/>	Extract Superclass
<input checked="" type="checkbox"/>	Extract Superclass (only Attributes)
<input type="checkbox"/>	Hide Attribute
<input type="checkbox"/>	Inline Class
<input type="checkbox"/>	Introduce Parameter Object
<input type="checkbox"/>	Move Attribute
<input type="checkbox"/>	Move Operation
<input checked="" type="checkbox"/>	Pull Up Attribute
<input type="checkbox"/>	Pull Up Operation
Push Down Attribute	

Restore Defaults Apply

OK Cancel

Relation: smell causing refactorings



Properties for de.unimarburg.swt.emf.fix.relations

type filter text

- Resource
- Builders
- EMF Metrics
- EMF Refactor
- EMF Relations**
- EMF Smells
- Plug-in Development
- Project References
- Run/Debug Settings

EMF Relations

Please set the relations between model smells and model refactorings as defined in this project.

Meta model: <http://www.eclipse.org/uml2/2.1.0/UML>

Refactorings useable to eliminate a given smell Smells possibly introduced by a given refactoring

Refactoring: Pull Up Attribute

Selected	Name	Description
<input type="checkbox"/>	Data Clumps (Parameters)	The model contains operations with more than a specific number of ...
<input checked="" type="checkbox"/>	Equal Attributes in Sibling Classes	Each sibling class of the owning class of an attribute contains an equ...
<input type="checkbox"/>	Extends	The model contains a use case that extends another use case.
<input checked="" type="checkbox"/>	Large Class	The model contains a class owning more features (attributes and op...
<input type="checkbox"/>	Long Parameter List	The model contains an operation with more input parameters than t...
<input type="checkbox"/>	Multiple Definition of Classes with E...	The model contains two classes (in different packages) having the sa...
<input type="checkbox"/>	No Incoming	The model contains a state without incoming transitions.
<input type="checkbox"/>	Primitive Obsession (Constants)	The model contains a class with more constant attributes than the sp...
<input checked="" type="checkbox"/>	Primitive Obsession (Primitive Types)	The model contains a class with more attributes having primitive typ...
<input type="checkbox"/>	Specialization Aggregation	The model contains a generalization hierarchy between associations

Restore Defaults Apply

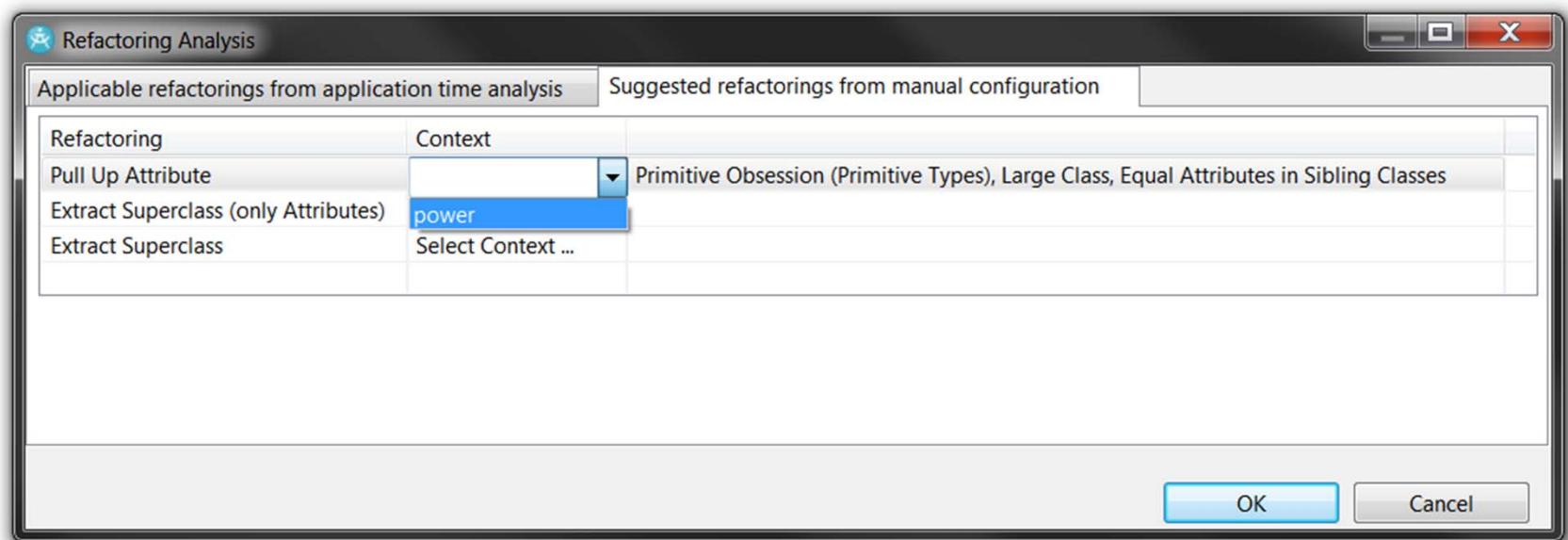
?

OK Cancel

Dialog: suggested refactorings



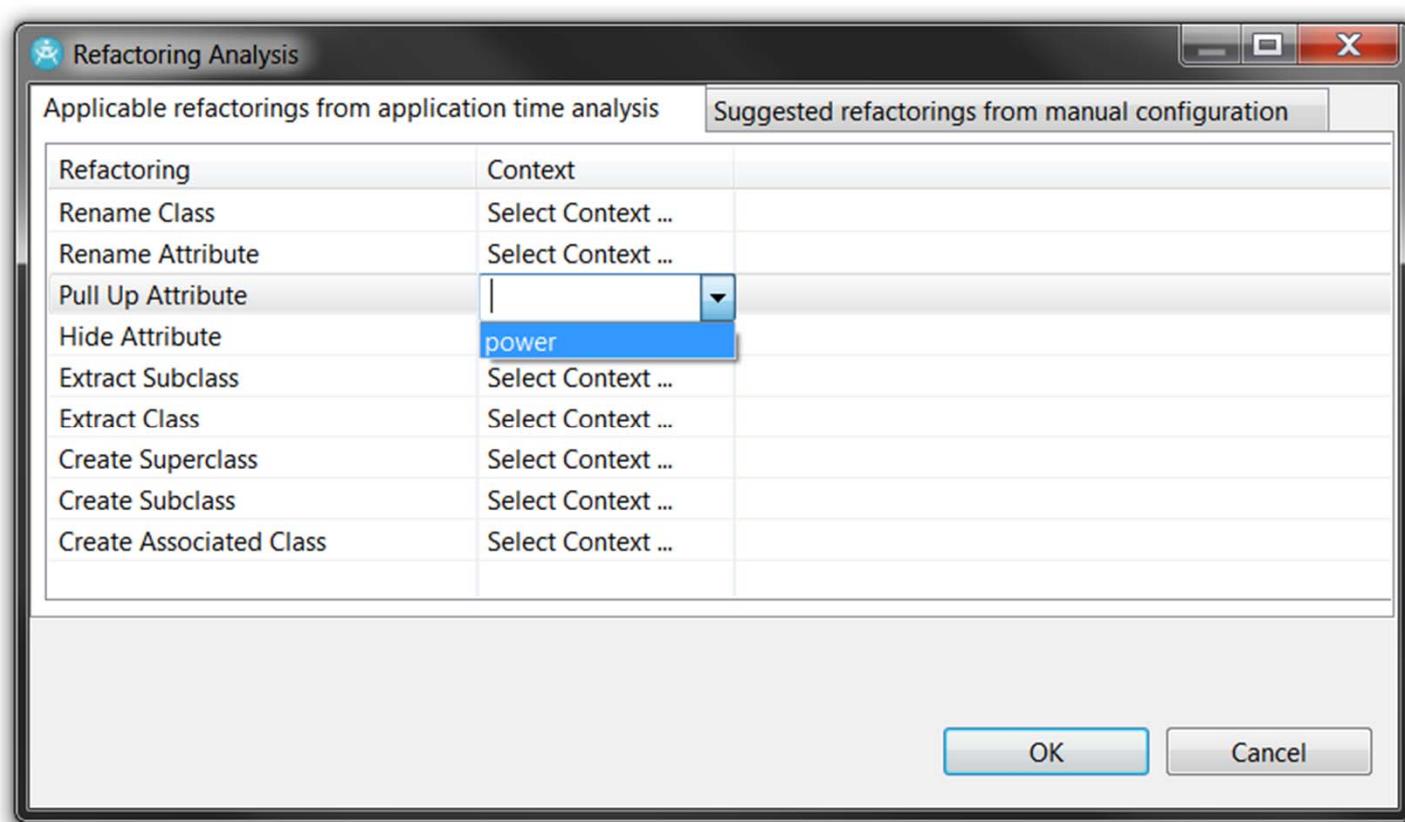
- Invocation on Smell **Equal Attributes in Sibling Classes**
 - Occurrence **{ Motorbike, power }**



Dialog: applicable refactorings



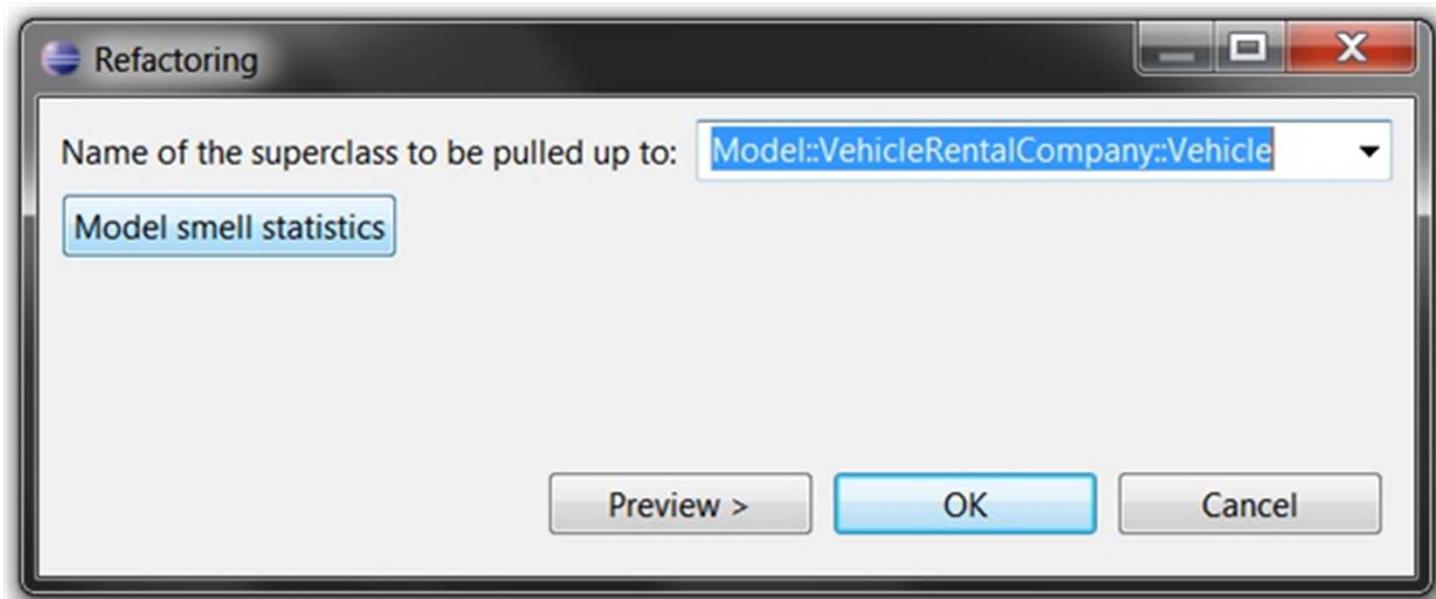
- Invocation on Smell **Equal Attributes in Sibling Classes**
 - Occurrence *{ Motorbike, power }*



EMF Refactor: parameter dialog



- Refactoring **Pull Up Attribute**
 - Context ***VehicleRentalCompany::Motorbike::power***



EMF Refactor: smell analysis



Changes in model smell occurrences

The following model smell occurrences are found before applying refactoring 'Pull Up Attribute':

- Equal Attributes in Sibling Classes: 6
- Long Parameter List: 3
- Large Class: 2
- Speculative Generality (Abstract Class): 1
- Speculative Generality (Interface): 1
- Concrete Superclass: 1

The following model smell occurrences are found after applying refactoring 'Pull Up Attribute':

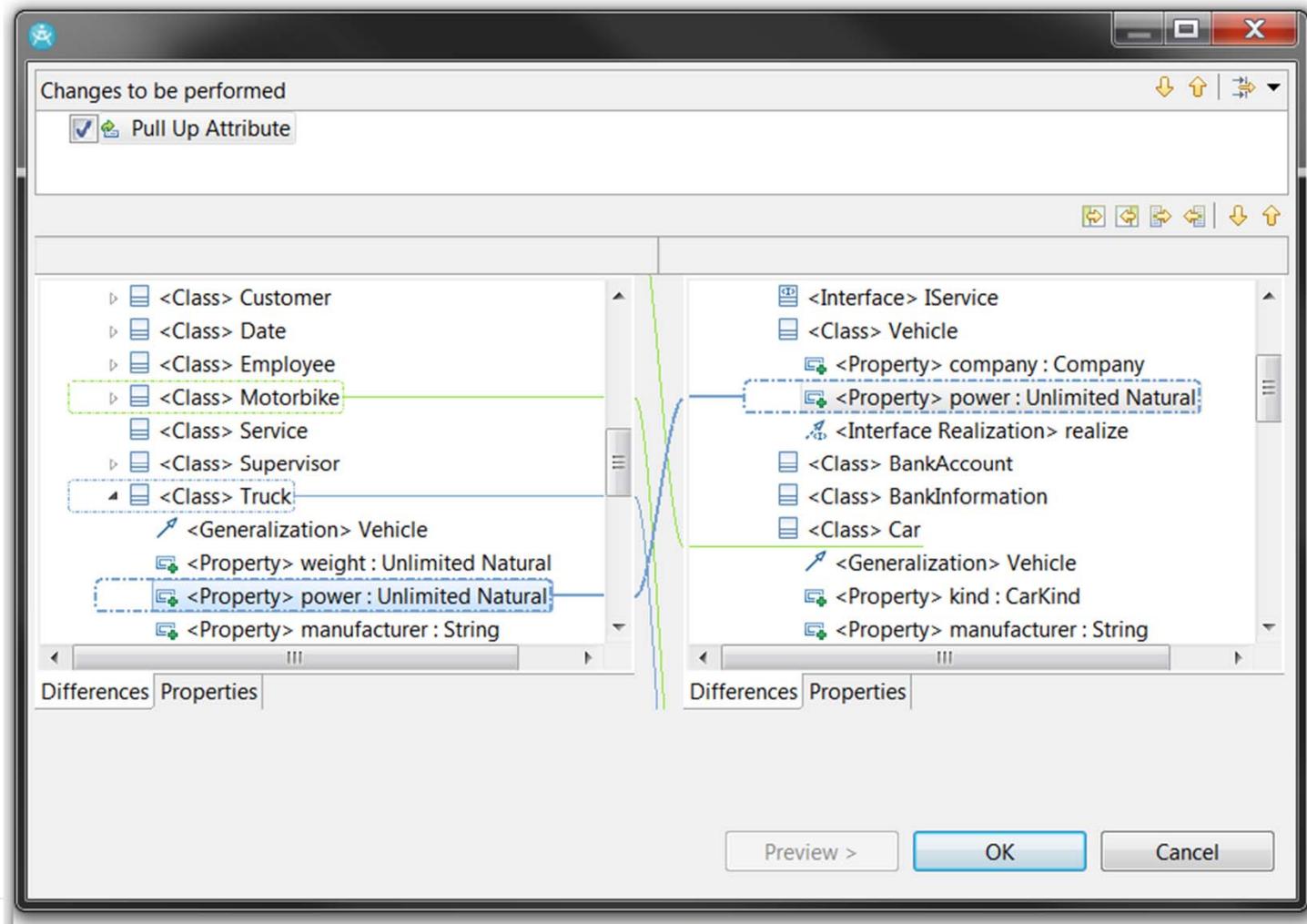
- Long Parameter List: 3
- Equal Attributes in Sibling Classes: 3
- Large Class: 2
- Speculative Generality (Abstract Class): 1
- Speculative Generality (Interface): 1
- Concrete Superclass: 1

The following model smell occurrences will change when applying refactoring 'Pull Up Attribute':

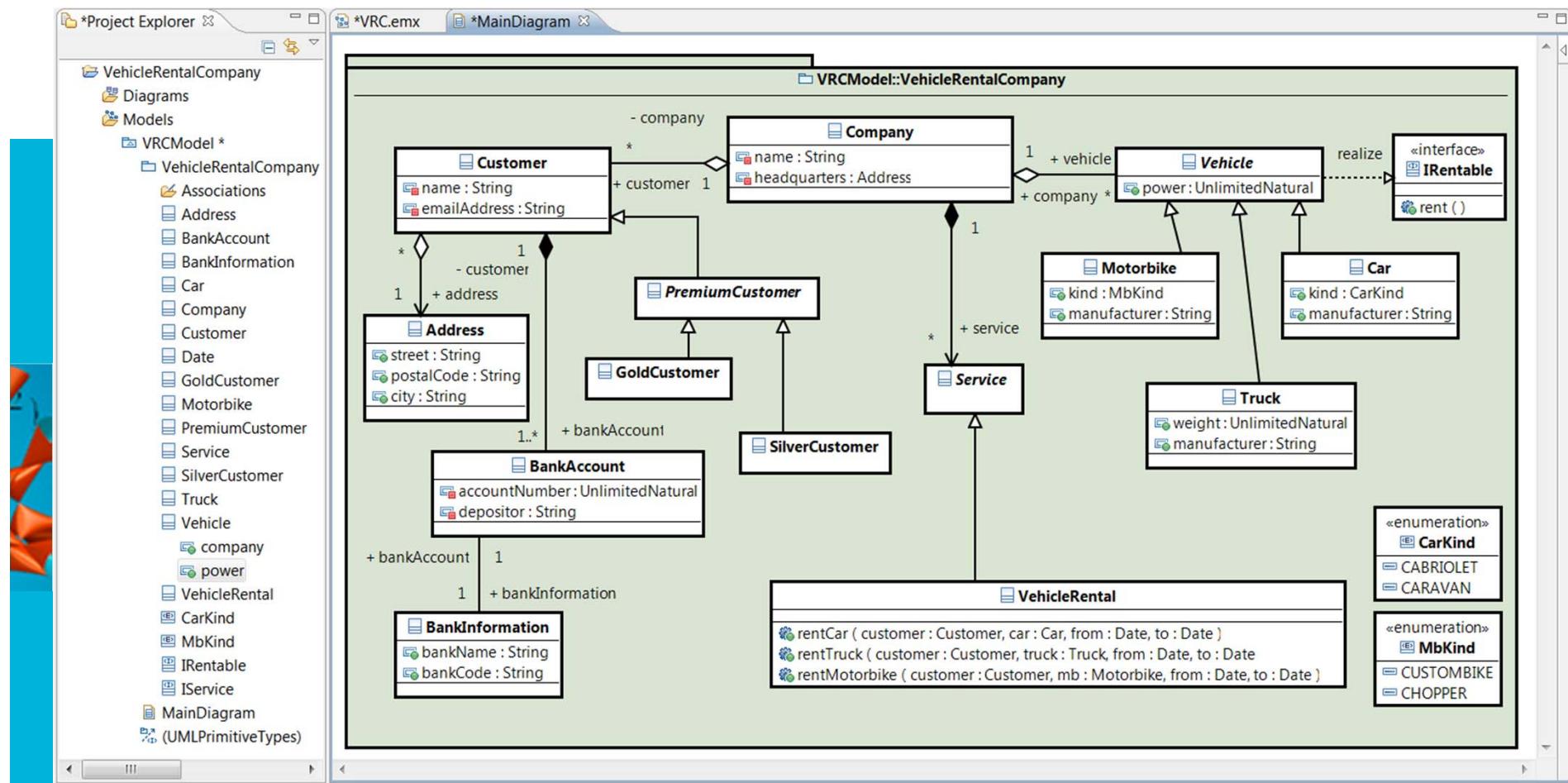
- Equal Attributes in Sibling Classes: -3

OK

EMF Refactor: result preview



EMF Refactor: refactored model



Outline

- Introduction
- Running Example
- Demo: Application of Model Quality Assurance Techniques
- Coffee Break
- Demo: **Specification of Model Quality Assurance Techniques**
- Conclusion

Used UML specification diagrams

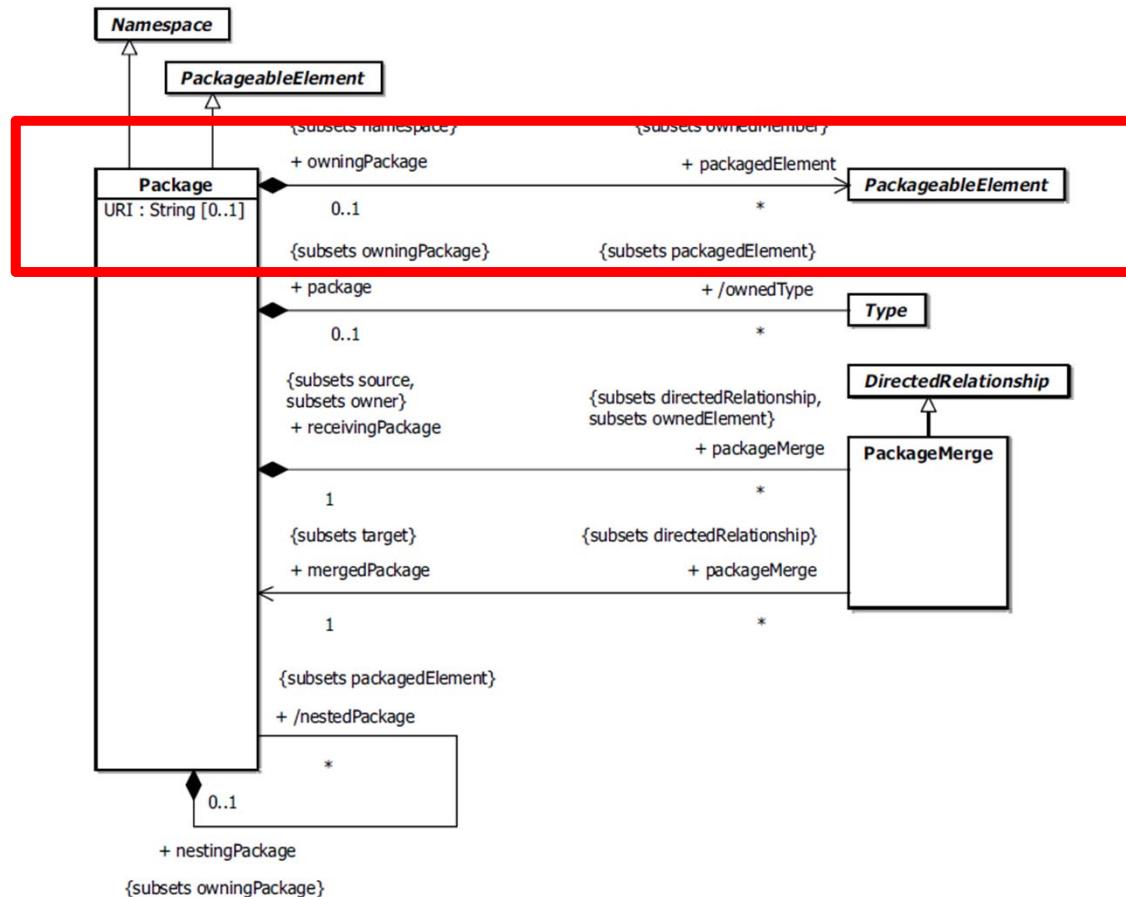
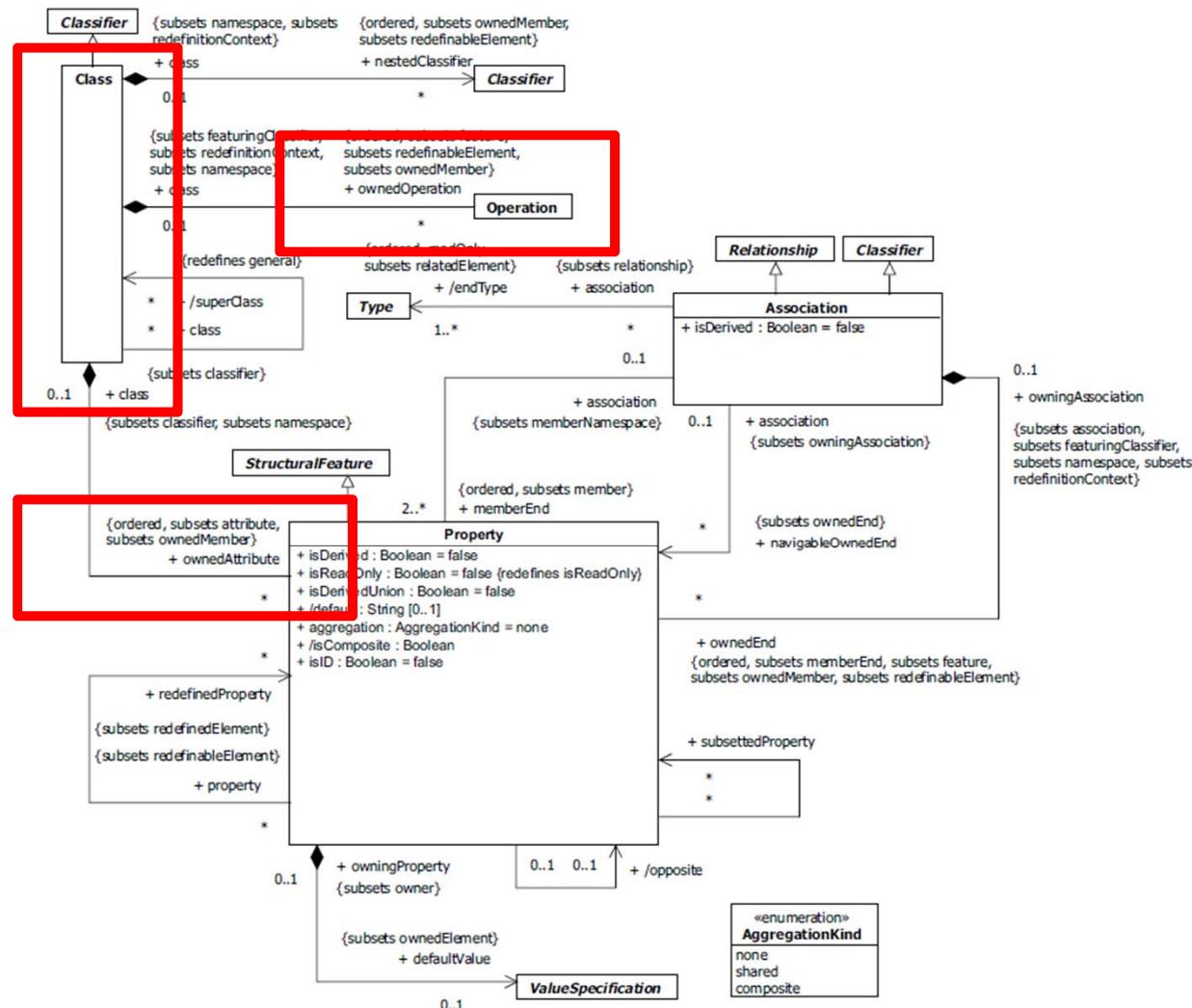
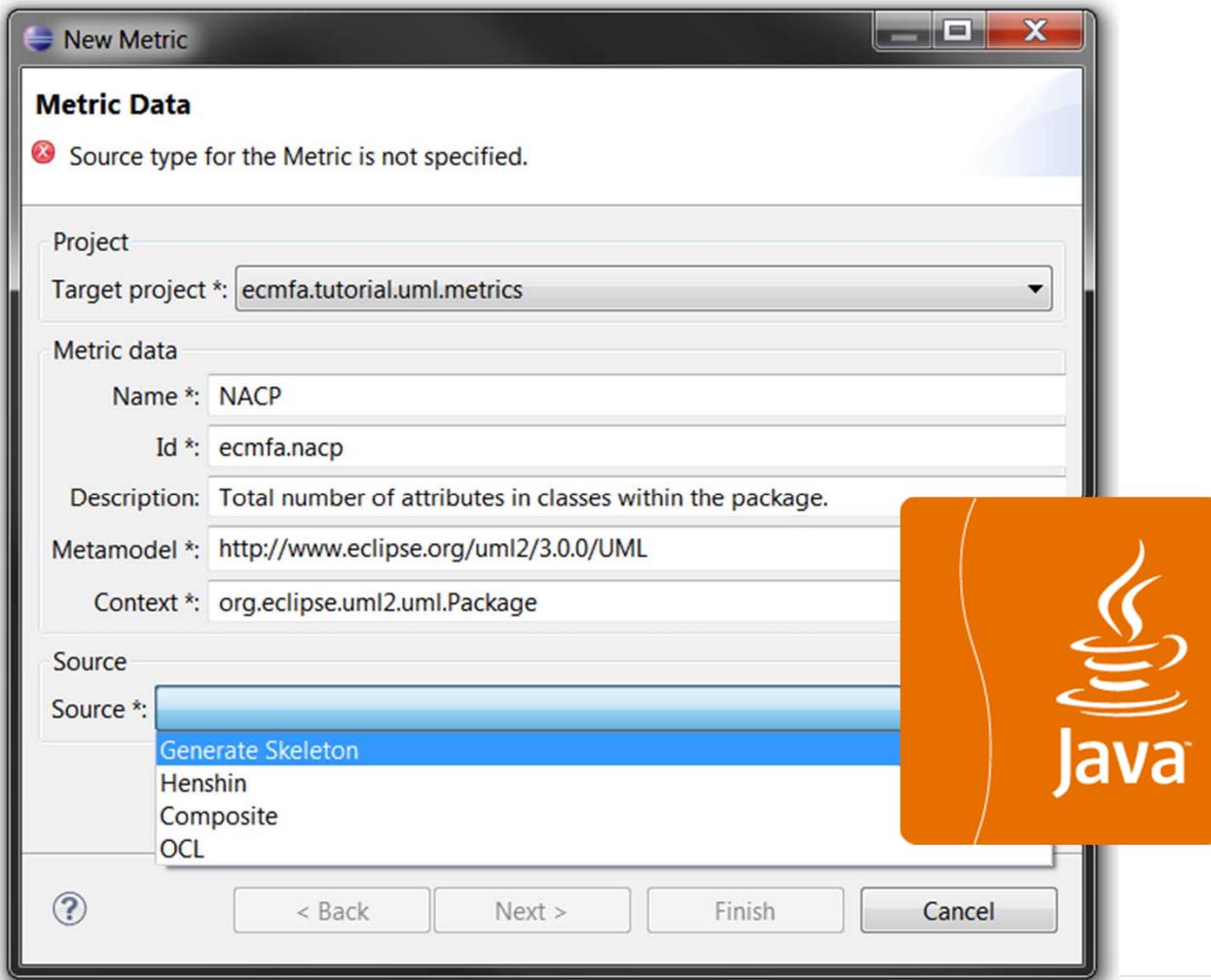


Figure 7.14 - The Packages diagram of the Kernel package

Used UML specification diagrams



Specification of new metric NACP (Total number of attributes in classes within the package): general data



Specification of new metric NACP (Total number of attributes in classes within the package): Java code

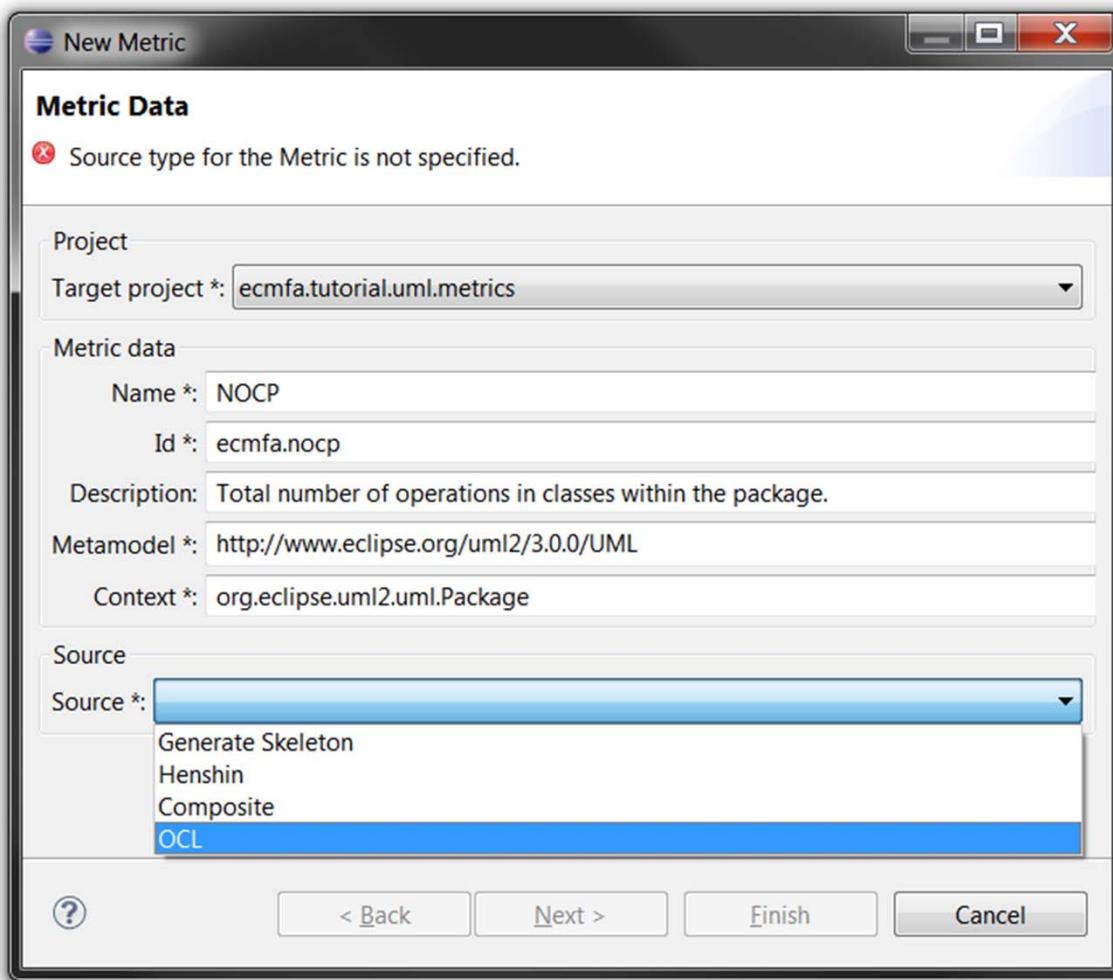
```
VehicleRentalCompany.di NACP.java
1 package de.unimarburg.swt.emf.metric;
2
3 import java.util.List;
4
5
6 public final class NACP implements ICalculateClass {
7
8     private List<EObject> context;
9
10    @Override
11    public void setContext(List<EObject> context) {
12        this.context=context;
13    }
14
15    @Override
16    public double calculate() {
17        org.eclipse.uml2.uml.Package in =
18            (org.eclipse.uml2.uml.Package) context.get(0);
19        double ret = 0.0;
20
21        // TODO fill 'ret'
22
23        return ret;
24    }
25
26
27 }
```

generated code

```
VehicleRentalCompany.di NACP.java
1 package de.unimarburg.swt.emf.metric;
2
3 import java.util.List;
4
5
6 public final class NACP implements ICalculateClass {
7
8     private List<EObject> context;
9
10    @Override
11    public void setContext(List<EObject> context) {
12        this.context=context;
13    }
14
15    @Override
16    public double calculate() {
17        org.eclipse.uml2.uml.Package in =
18            (org.eclipse.uml2.uml.Package) context.get(0);
19        double ret = 0.0;
20
21        // begin custom code
22        for (PackageableElement pe : in.getPackagedElements()) {
23            if (pe instanceof Class) {
24                Class cl = (Class) pe;
25                for (Property attr : cl.getAllAttributes()) {
26                    if (attr.getAssociation() == null) ret++;
27                }
28            }
29        }
2
30        // end custom code
31        return ret;
32    }
33
34
35
36
37 }
```

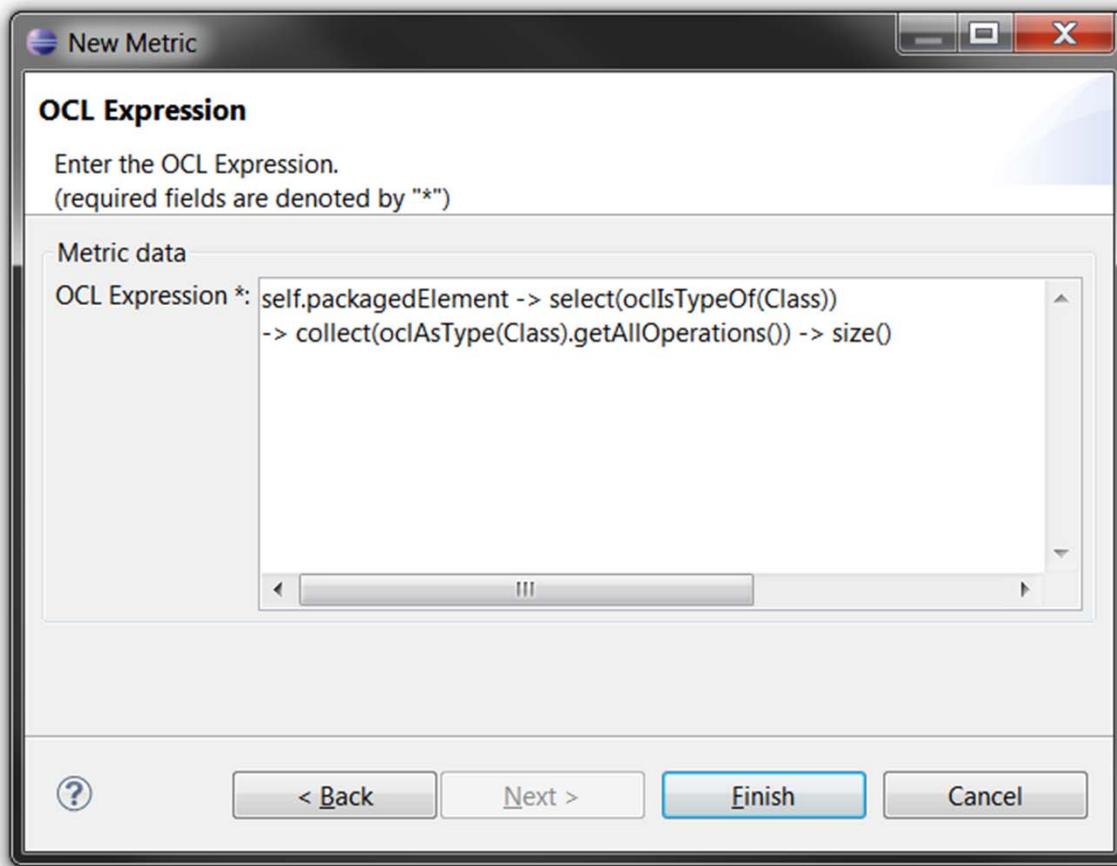
completed code

Specification of new metric NOCP (Total number of operations in classes within the package): general data

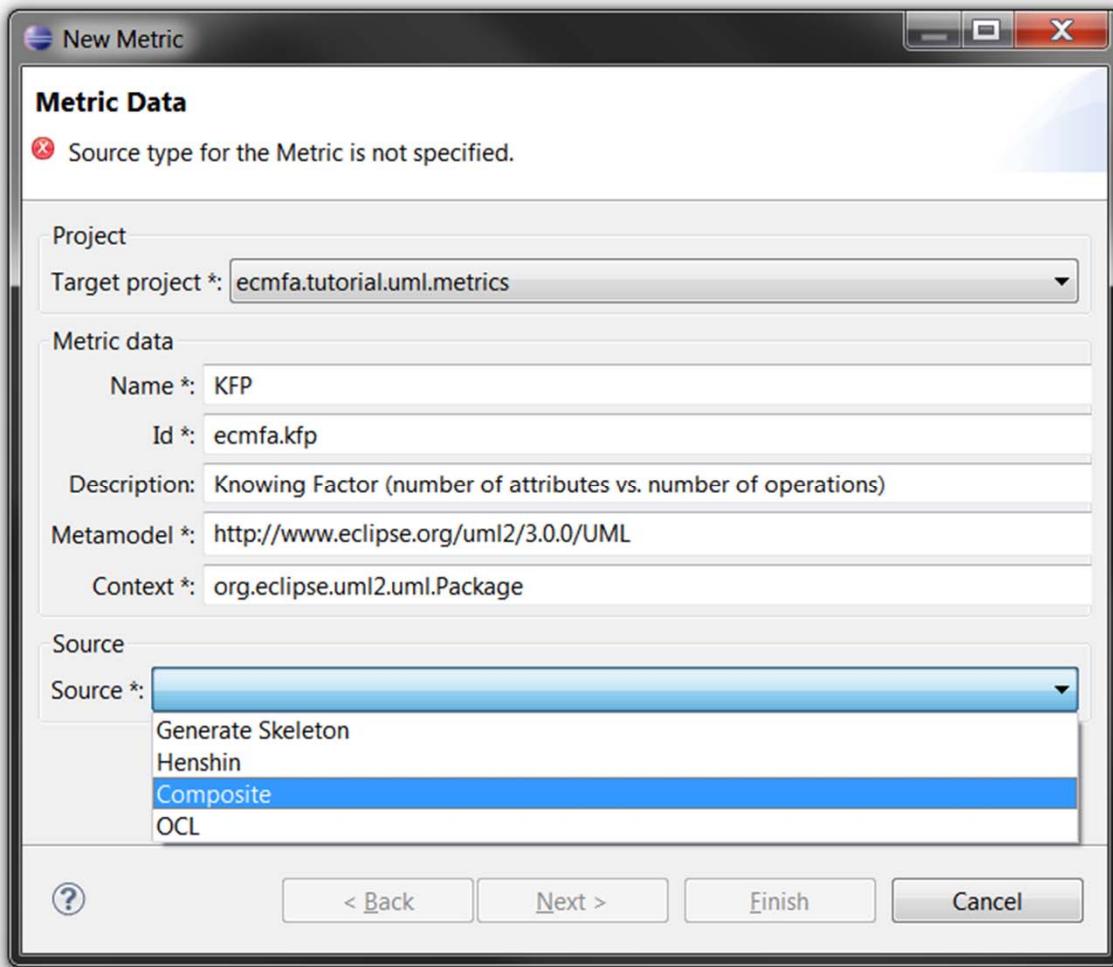


OCL

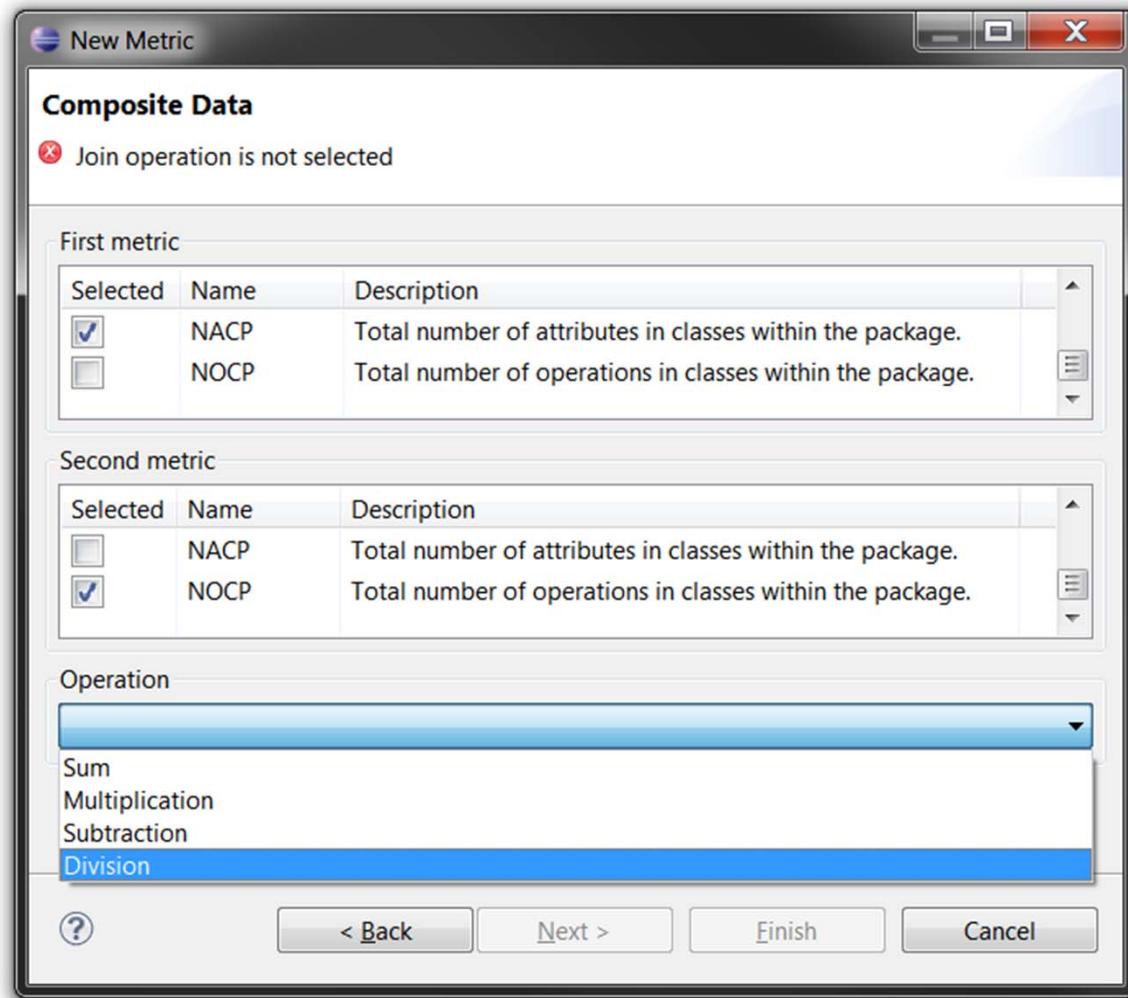
Specification of new metric NOCP (Total number of operations in classes within the package): OCL expr.



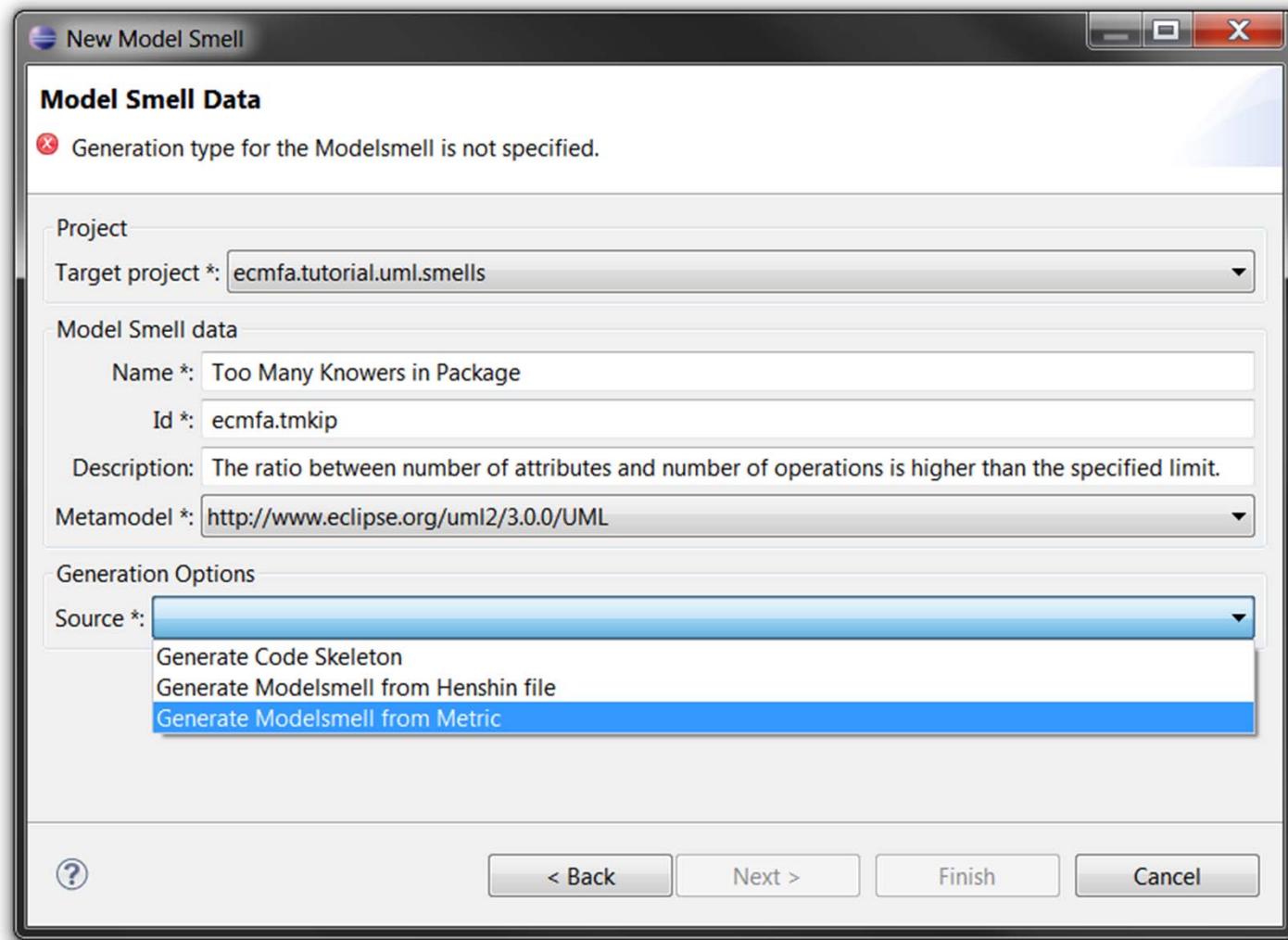
Specification of new metric KFP (Knowing Factor): general data



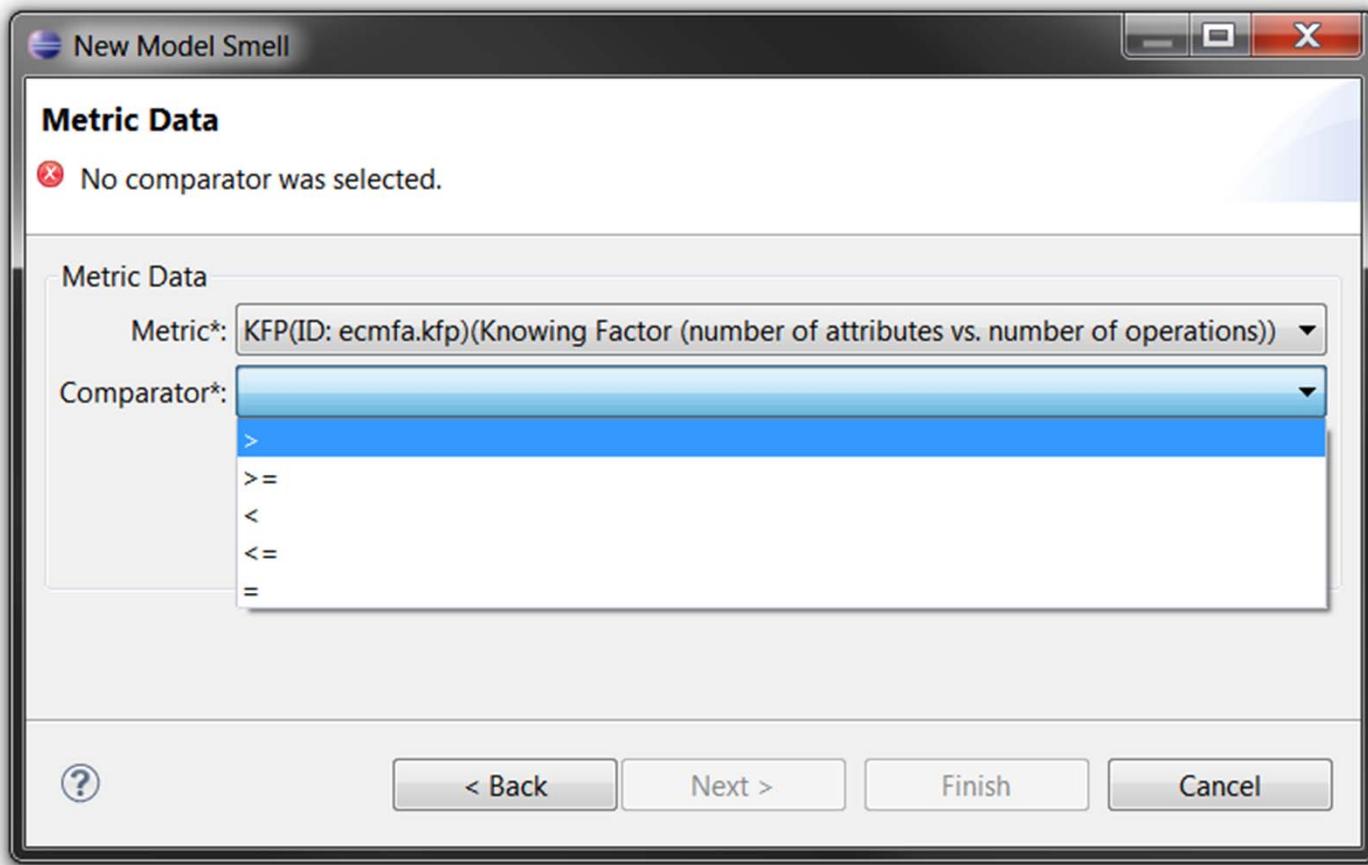
Specification of new metric KFP (Knowing Factor): metric data



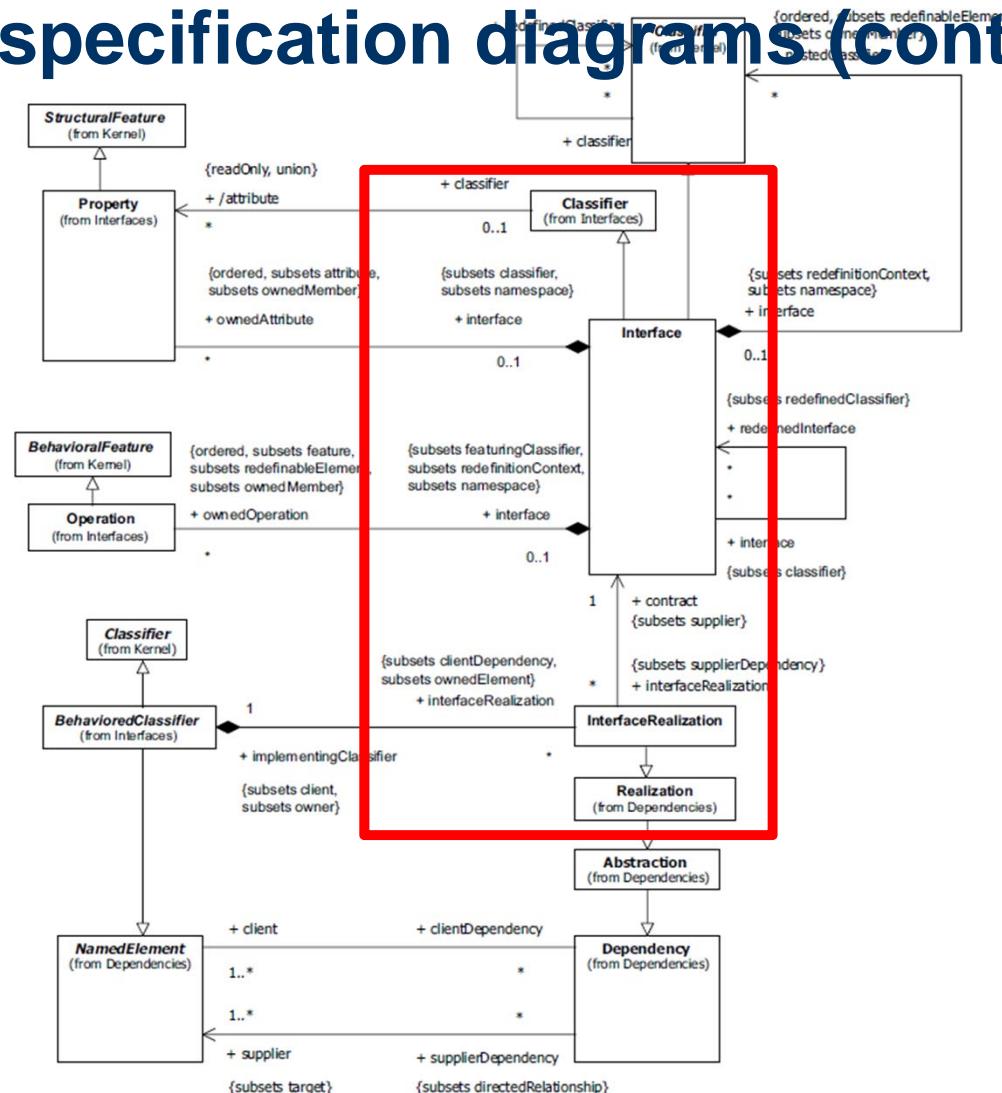
Specification of new smell ‘Too Many Knowers in Package’: general data



Specification of new smell ‘Too Many Knowers in Package’: metric data



Used UML specification diagrams (cont.)



Used UML specification diagrams (cont.)

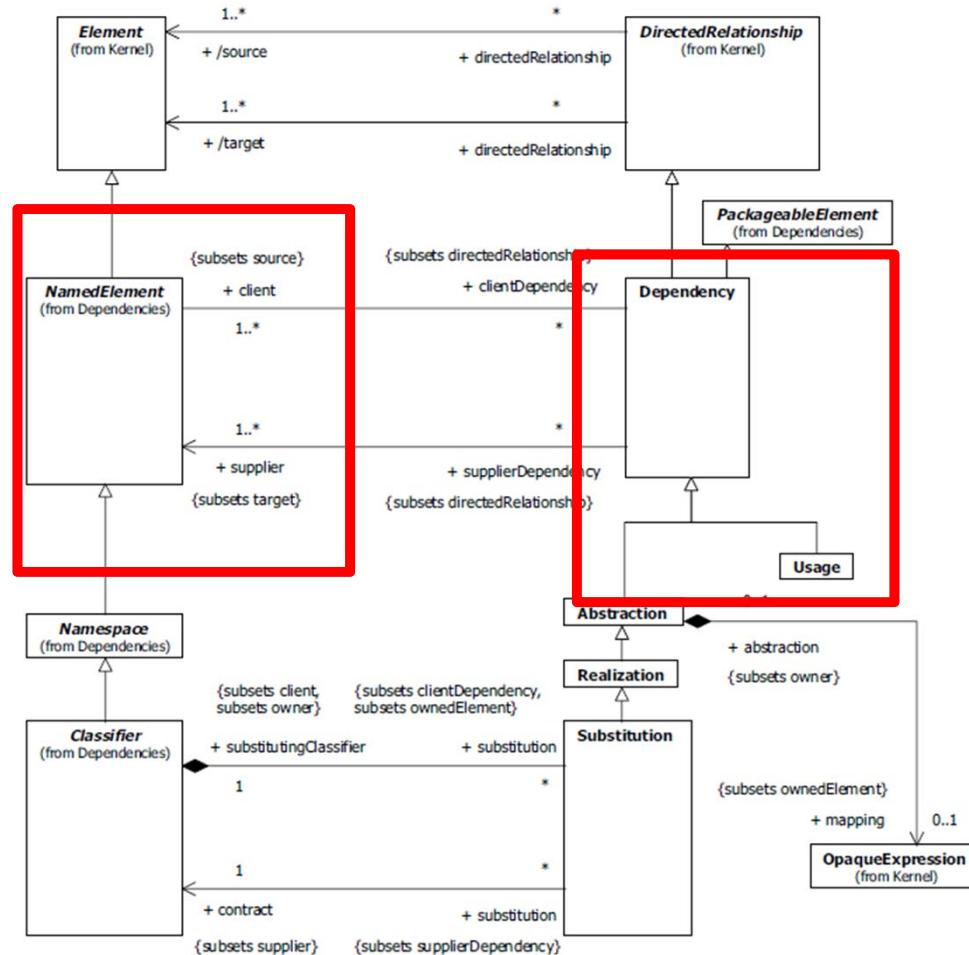


Figure 7.15 - Contents of Dependencies package

Used UML specification diagrams (cont.)

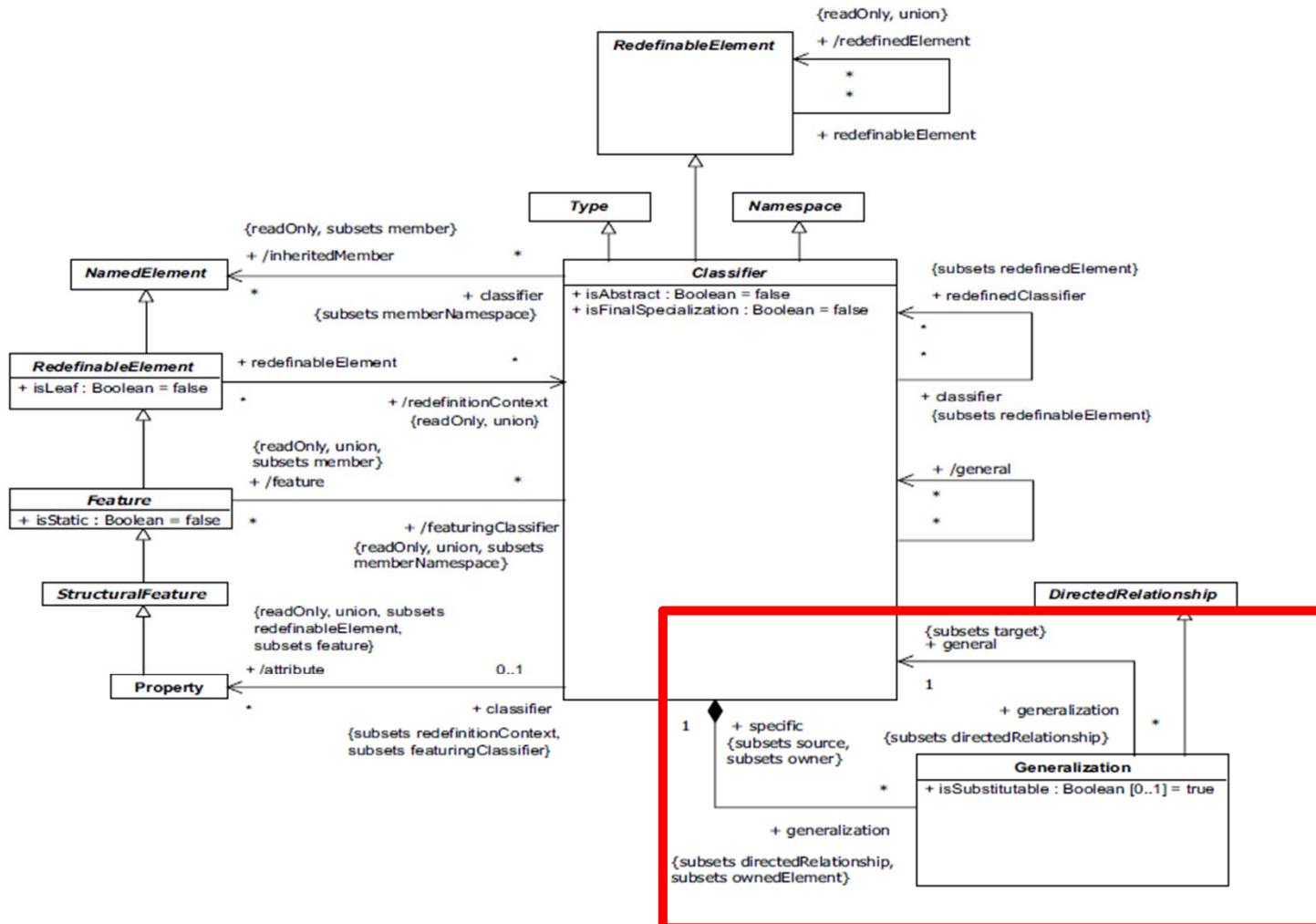
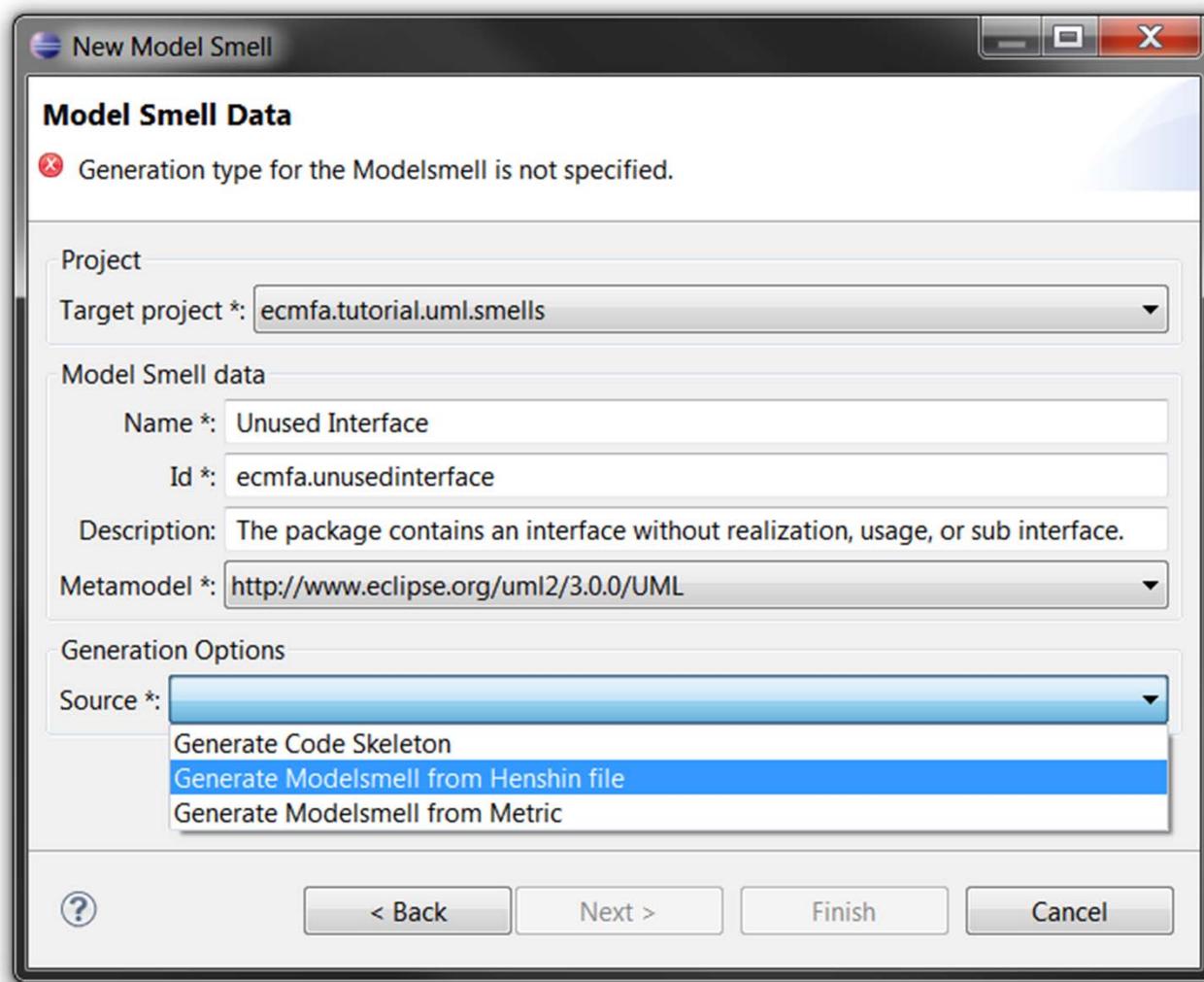
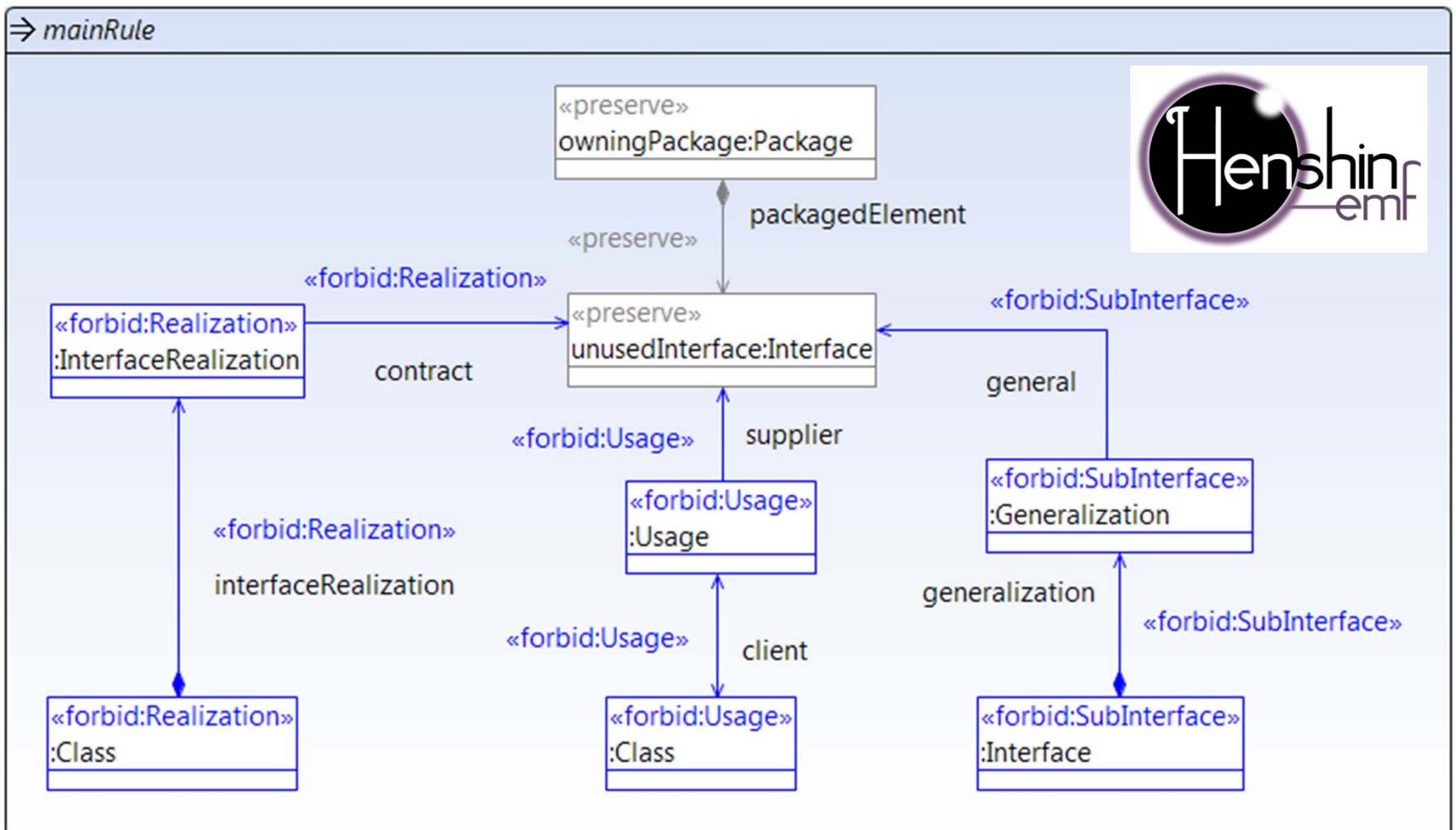


Figure 7.9 - Classifiers diagram of the Kernel package

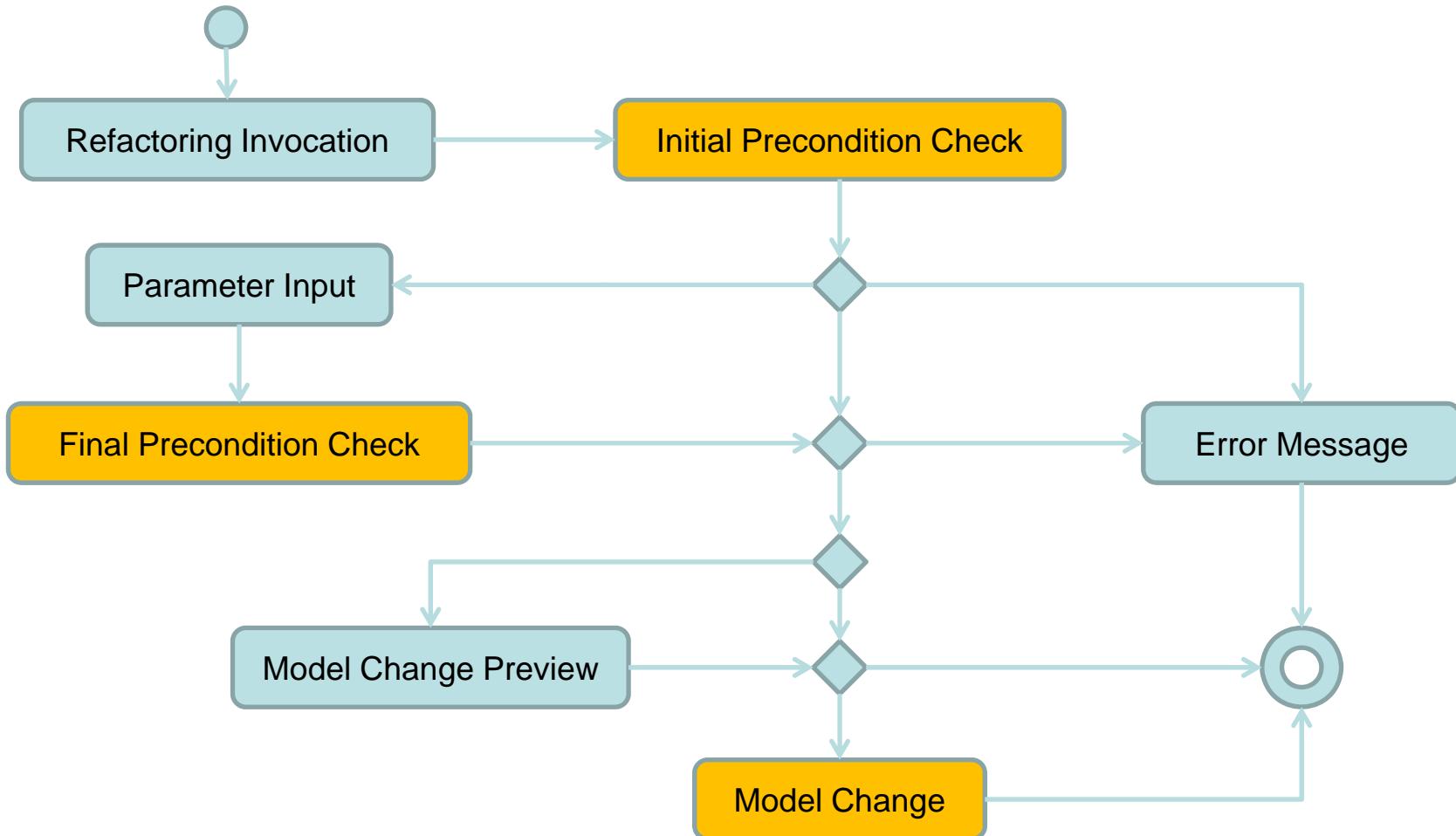
Specification of new smell ‘Unused Interface’: general data



Specification of new smell ‘Unused Interface’: Henshin pattern rule specification



Refactoring application workflow



Specification of new refactoring ‘Remove Unused Interface’: general data



EMF Refactor - Specify EMF Model Refactoring using Henshin

Please insert a text to be shown in menu (menu label). Otherwise, EMF Refactor will show the refactoring name.

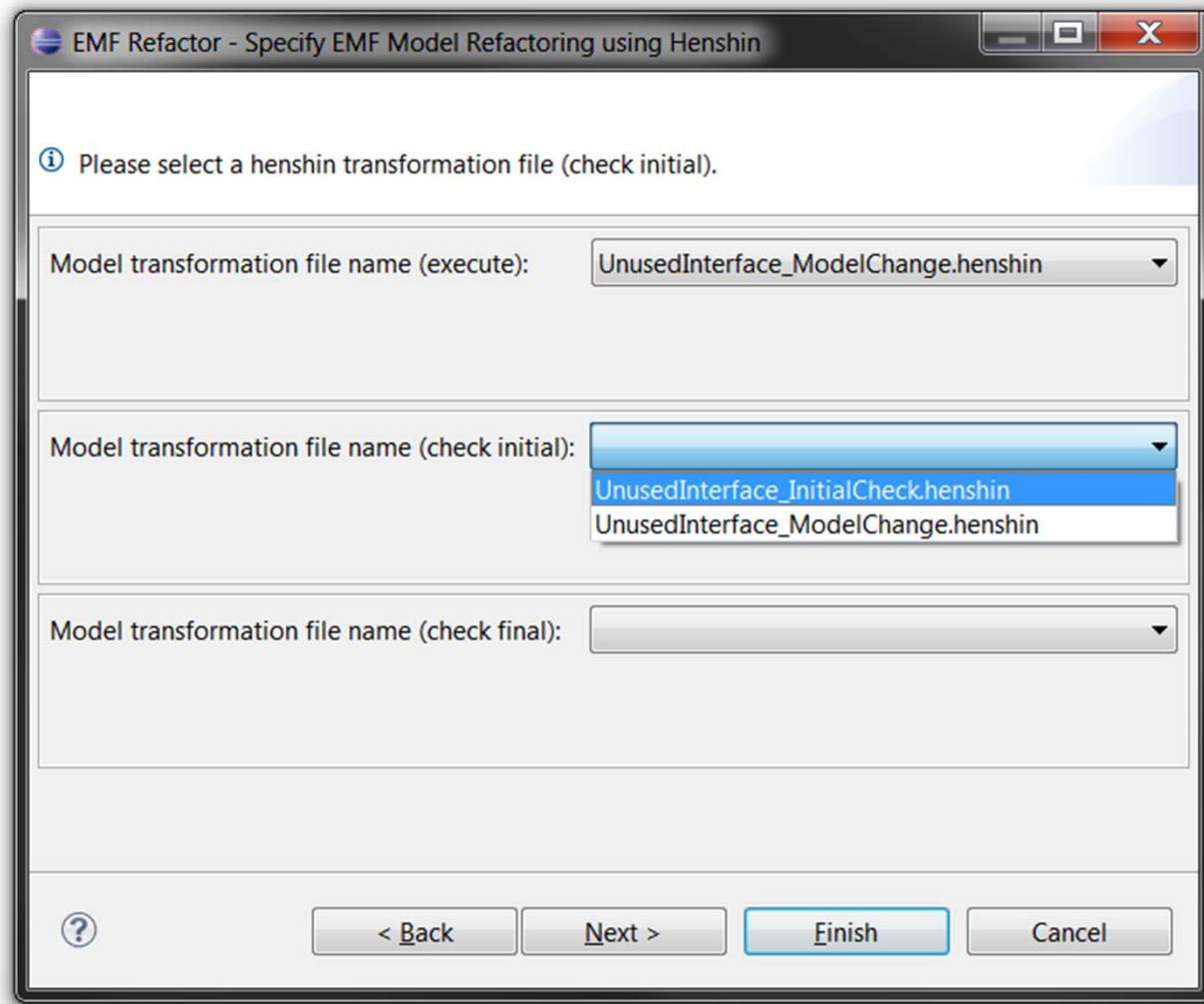
Name of the new EMF Model Refactoring (Id): ecmfa.removeunusedinterface
Text to be shown in menu (menu label): Remove unused Interface

Namespace-Uri: http://www.eclipse.org/uml2/3.0.0/UML
Class of selected EObject (context): org.eclipse.uml2.uml.Interface
jar-File to be included (required bundle): org.eclipse.uml2.uml

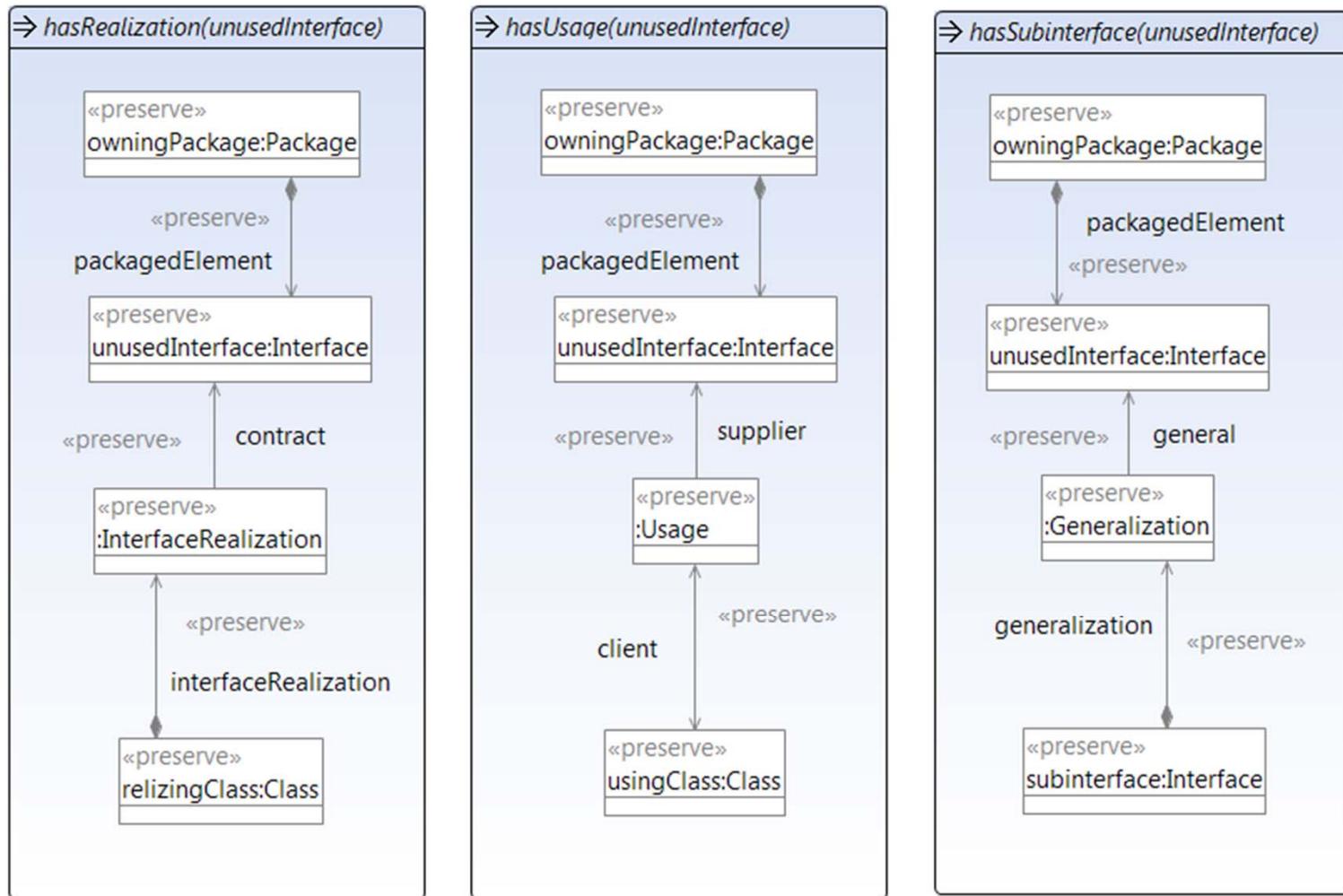
Plugin-Project that will contain the code: ecmfa.tutorial.uml.refactorings

< Back Next > Finish Cancel

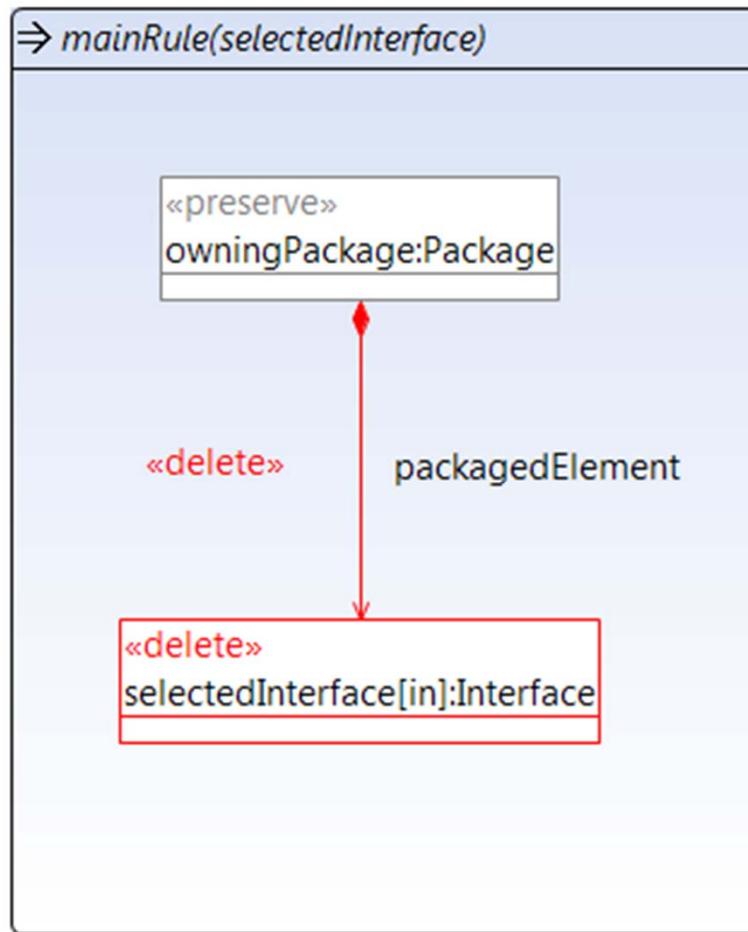
Specification of new refactoring ‘Remove Unused Interface’: Henshin data



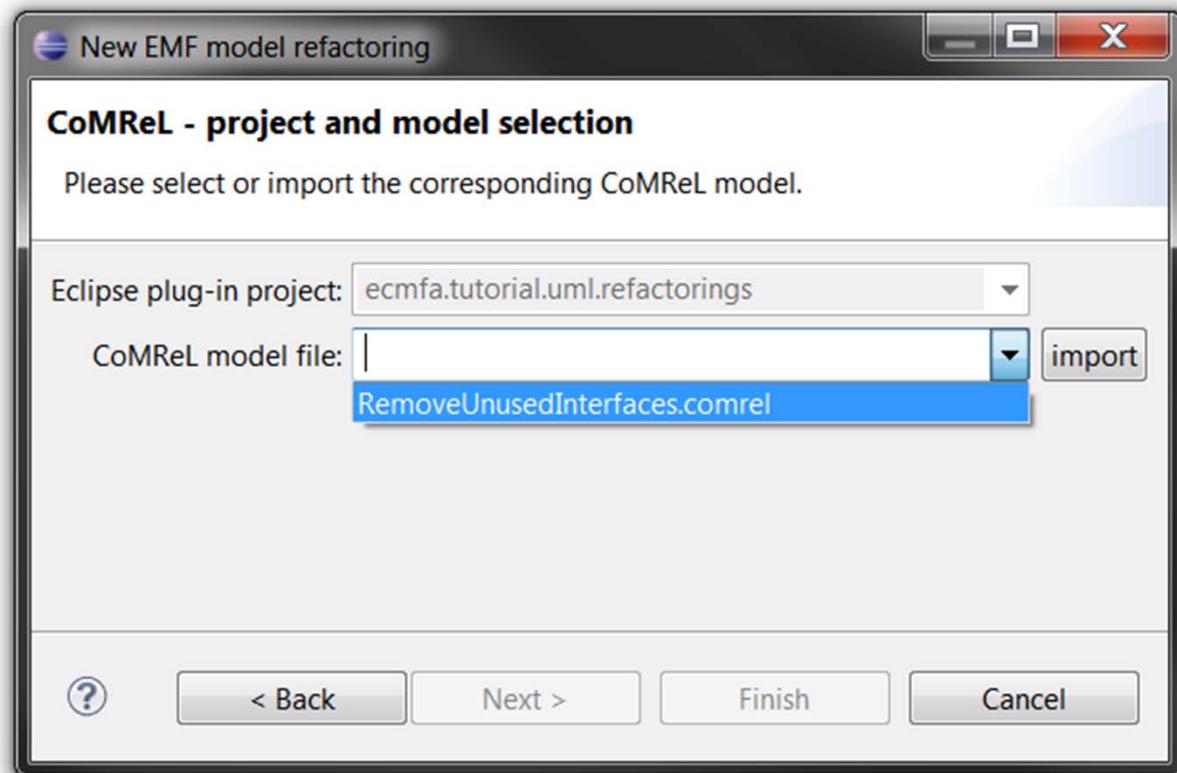
Specification of new refactoring ‘Remove Unused Interface’: Henshin’s initial check

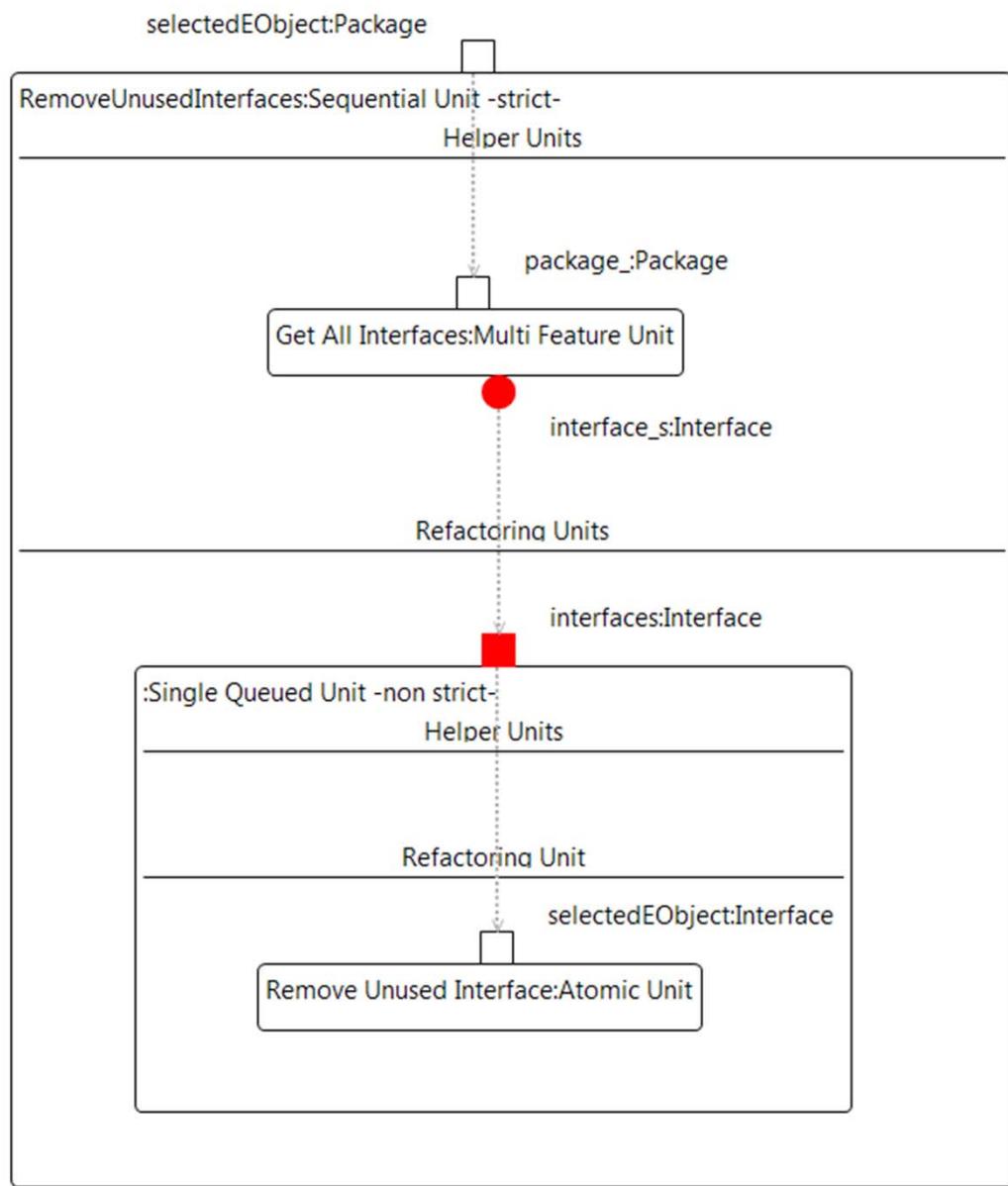


Specification of new refactoring ‘Remove Unused Interface’: Henshin’s model change



Specification of new refactoring ‘Remove Unused Interfaces from Package’: CoMReL data



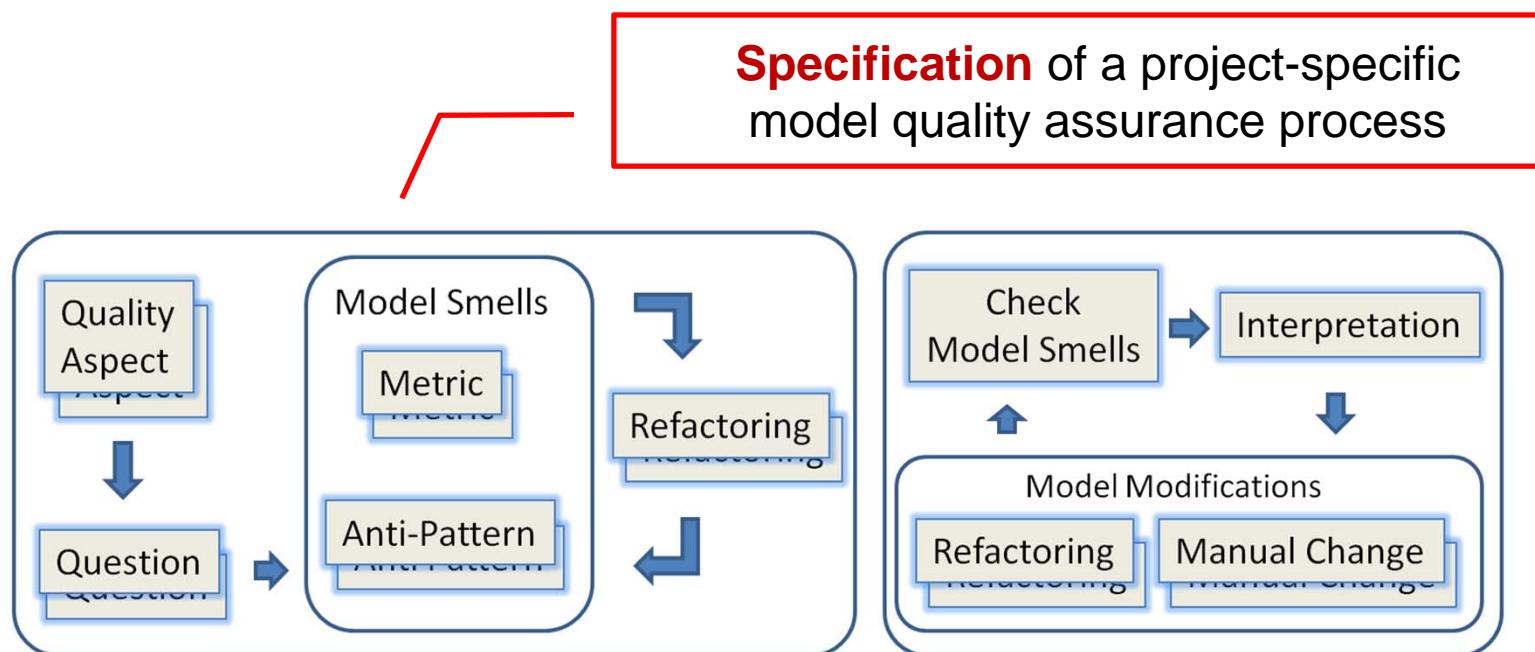


Specification of new refactoring ‘Remove Unused Interfaces from Package’: CoMReL specification

Outline

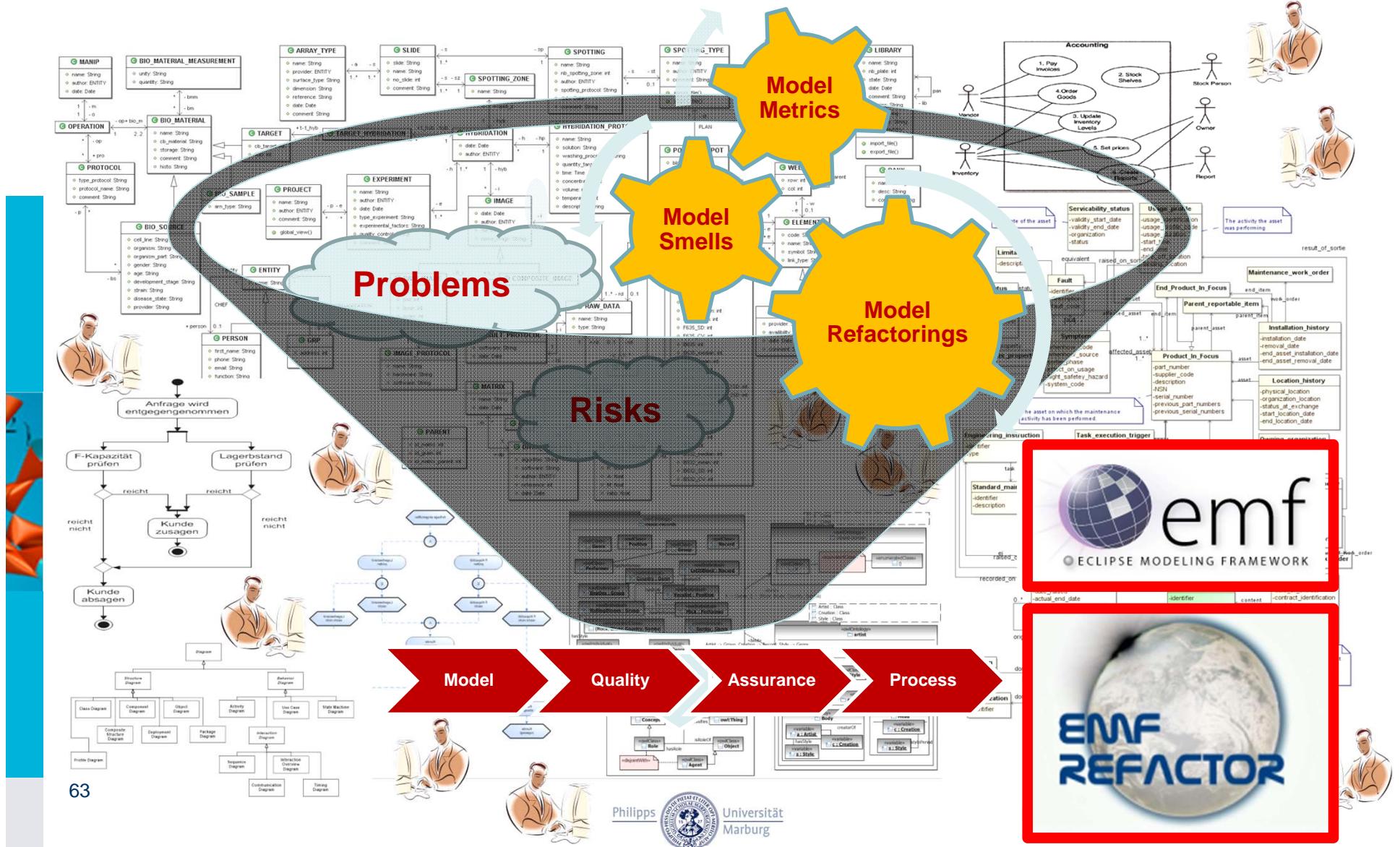
- Introduction
- Running Example
- Demo: Application of Model Quality Assurance Techniques
- Coffee Break
- Demo: Specification of Model Quality Assurance Techniques
- Conclusion

A model quality assurance process



Application of the specified process to concrete software models

Tool environment for model quality assurance



Tool environment for model quality assurance

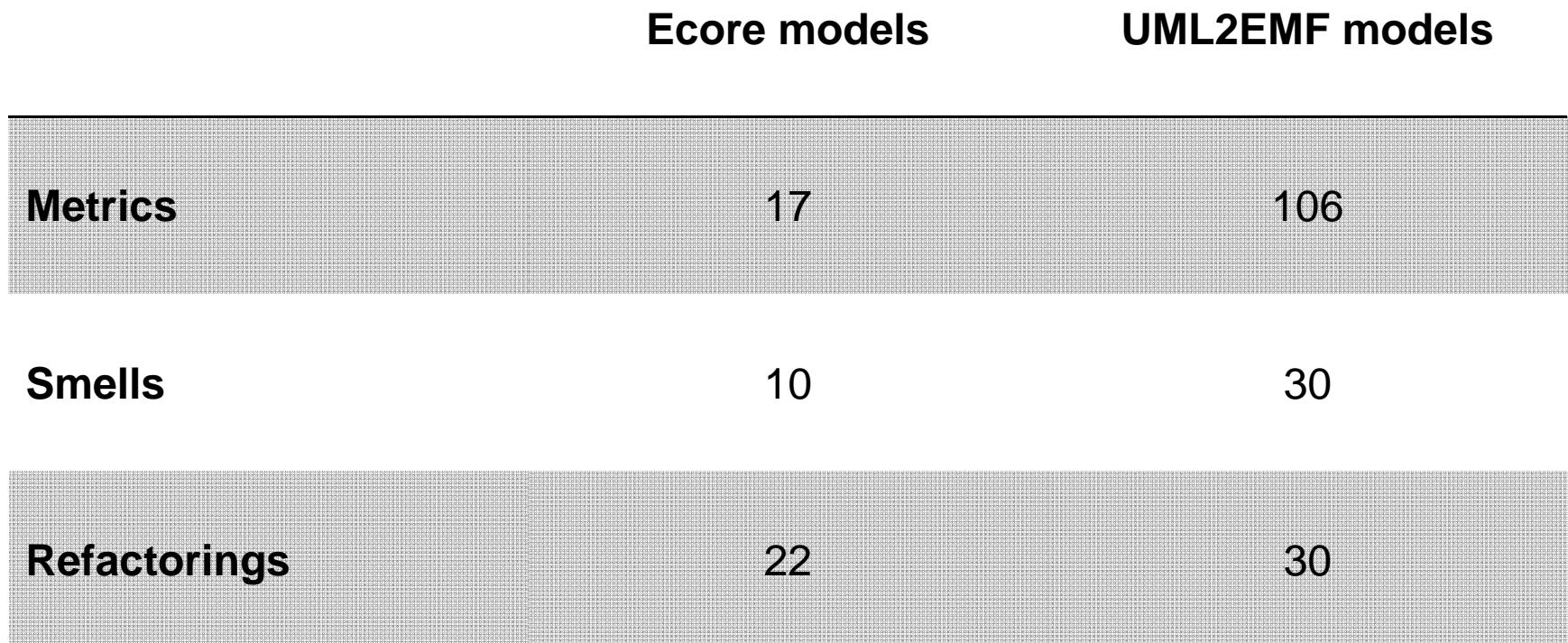
- Highly integrated

- Specific tools use each other
 - Metric-based model smells
 - Refactoring invocation from within model smell results view
 - Model smell analysis during refactoring application
- Each tool can be used directly on the model within a certain EMF-based modeling environment
 - EMF instance editor
 - Unified Modeling Language (UML)
 - Eclipse Plugin Papyrus
 - IBM Rational Software Architect

Tool environment for model quality assurance

- Flexible specification mechanisms
 - Usage of existing quality assurance techniques
 - Combination of existing model metrics
 - Metric-based model smells
 - Composite model refactorings
 - Usage of concrete specification languages
 - Java
 - OCL
 - EMF model transformation language Henshin
 - ... further languages can be integrated by new adapters

Implemented model quality assurance techniques



Tool environment for model quality assurance

- Download

- <http://www.eclipse.org/modeling/emft/refactor/>
 - Downloads
 - Specials

