Quality Assurance of Software Models within Eclipse using Java and OCL

Dr. Thorsten Arendt

Modellgetriebene Softwareentwicklung mobiler Anwendungen
Wintersemester 2014/15
17. Dezember 2014
Outline

• Why model quality assurance?
• A structured model quality assurance process
• Tailoring the process for a given modeling project
  – Specification process
  – Application process
• Tool support: EMF Refactor
Why model quality assurance?

Problems

Risks

Model Metrics

Model Smells

Model Refactorings
Why model quality assurance?
Software quality ≠ software model quality

- Installability
  - Conformance
  - Replaceability
  - Adaptability
- Portability
- Maintainability
  - Stability
  - Analyzability
  - Changeability
  - Testability
- Efficiency
- Usability
- Functionality
- Reliability
- Security
- Suitability
  - Accuracy
  - Interoperability
  - Compliance
- Maturity
  - Recoverability
  - Fault tolerance
- Learnability
  - Understandability
  - Operability

ISO/IEC 9126-1
Software quality ≠ software model quality

Software quality

- Installability
- Conformance
- Replaceability
- Adaptable

- Portability

- Maintainability
- Stability
- Analyzerability
- Changeability
- Testability

- Efficiency
- Time
- Behaviour
- Resource
- Behaviour

- Functionality

ISO/IEC 9126-1

- Reliability
- Maturity
- Recoverability
- Fault tolerance

- Usability
- Learnability
- Understandability
- Operability

6C model quality goals

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

[ISO/IEC 9126]

[Mohagheghi]
Model smells

- Adopted from the concept of **code smell**

- Smells are **parts** of the **model** that
  - ... contain symptoms of poor model quality
  - ... are worthy of an inspection
  - ... are potential candidates for improvements by refactoring

- Smells are **not synonyms** for **problems**
  - ‘... smells don't *always* indicate a problem. ... You have to look deeper to see if there is an underlying problem there - smells aren't inherently bad on their own - they are often an indicator of a problem rather than the problem themselves.’

[Fowler]
A structured model quality assurance process (1)
Discussing important quality aspects

- Correctness
- Completeness
- Changeability
- Comprehensibility
- Confinement
- Consistency
Specifying relevant model smells (1)

- Question: *Are there equally named classes in different packages?*
- Smell: *Multiple definitions of classes with equal names*
Specifying relevant model refactorings (1)

• Smell: *Multiple definitions of classes with equal names*
  – Refactorings: **Merge Classes**, **Rename Class**
• Question: *Are there abstract classes without any concrete subclasses?*
  
  - Smell: **No concrete Subclass**
Specifying relevant model refactorings (2)

- Smell: No concrete subclass
  - Refactoring: Insert Concrete Subclass
UML specific refactorings

Refactoring:
Change Attribute to Association End

Person
+ forename: String [1]
+ surname: String [1]
+ birthday: Date [1]
+ email: String [1]

Address
+ street: String [1]
+ number: String [1]
+ postalCode: String [1]
+ city: String [1]

Date
+ day: UnlimitedNatural [1]
+ month: UnlimitedNatural [1]
+ year: UnlimitedNatural [1]

Refactoring:
Change Association End to Attribute

Person
+ forename: String [1]
+ surname: String [1]
+ email: String [1]

Address
+ street: String [1]
+ number: String [1]
+ postalCode: String [1]
+ city: String [1]

Date
+ day: UnlimitedNatural [1]
+ month: UnlimitedNatural [1]
+ year: UnlimitedNatural [1]
UML class attribute specification

Refactoring: Change Attribute to Association End

[OMG: UML]
A structured model quality assurance process (2)

Specification process

- Quality Aspect
- Metric
- Anti-Pattern
- Refactoring

Model Analysis
- Calculate Model Metrics
- Check Model Smells
- Interpretation

Model Modifications
- Refactoring
- Manual Change

Application process

Manager  Designer  Modeler  Reviewer
Tool environment for model quality assurance

Added functionality!

Model

Metrics

Risks

Model

Refactorings

Problems

Model

Smells

Model

Quality

Assurance

Process

Initial functionality
Quality Assurance and Textual Modeling in Eclipse
Scenario: Vehicle Rental Company

• Project
  – Development of a web application for renting vehicles using MDD

• Semantic domain
  – Company with vehicles, customers, services, ..

• Technical domain
  – Web application

• Modeling purpose
  – Modeling of the semantic domain as well as the information mandatory for code generation using a (textual) DSML

• Credo
  – The better the model, the better the code
Xtext Grammar for SWM

grammar org.eclipse.emf.refactor.examples.SimpleWebModel
    with org.eclipse.xtext.common.Terminals

generate simpleWebModel
    "http://www.eclipse.org/SimpleWebModel/1.0"

WebModel:
    'webmodel' name=ID '{'
        dataLayer=DataLayer
        hypertextLayer=HypertextLayer
    '}' ;

DataLayer:
    'data' '{' [DataLayer]
        entities+=Entity
    '}' ;

Entity:
    'entity' name=ID '{'
        attributes+=Attribute
        references+=Reference
    '}' ;

Attribute:
    'att' name=ID ':' type=SimpleType

enum SimpleType:
    Boolean | Email | Float | Integer | String

Reference:
    'ref' name=ID ':' type=[Entity]

HypertextLayer:
    'hypertext' '{'
        pages+=Page+
        startPage is' startPage=[StaticPage]
    '}' ;

Page:
    StaticPage | DynamicPage

StaticPage:
    'static page' name=ID '{'
        links+=Link
    '}' ;

Link:
    'link to page' target=[Page]

DynamicPage:
    IndexPage | DataPage

IndexPage:
    'index page' name=ID ('shows entity' entity=[Entity])? '{'
        links+=Link
    '}' ;

DataPage:
    'data page' name=ID ('shows entity' entity=[Entity])? '{'
        links+=Link
    '}' ;
Scenario: Xtext Model Vehicle Rental Company

Is the model complete?

Is the model comprehensible, correct, ...?
Metrics calculation

Is the model complete?

<table>
<thead>
<tr>
<th>Context</th>
<th>Metric</th>
<th>Description</th>
<th>Result</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebModel VehicleRentalCompany</td>
<td>ANAWM</td>
<td>Average Number of Attributes in Entities within the Web Model</td>
<td>2.14</td>
<td>2013/09/13 14:07:00</td>
</tr>
<tr>
<td>WebModel VehicleRentalCompany</td>
<td>ANRWM</td>
<td>Average Number of References in Entities within the Web Model</td>
<td>0.43</td>
<td>2013/09/13 14:07:00</td>
</tr>
</tbody>
</table>
Reporting and interpreting

variously interpretable

Is the model complete?

Metrics Report

<table>
<thead>
<tr>
<th>Context</th>
<th>Metric</th>
<th>Description</th>
<th>Result</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebModel VehicleRentalCompany</td>
<td>NEM</td>
<td>Number of Entities in the Model.</td>
<td>7.00</td>
<td>2013/09/13 13:34:41</td>
</tr>
<tr>
<td>WebModel VehicleRentalCompany</td>
<td>NDPM</td>
<td>Number of Dynamic Pages in the Model.</td>
<td>4.00</td>
<td>2013/09/13 13:34:41</td>
</tr>
</tbody>
</table>
Smell detection

Is the model complete?
Locating and interpreting refactoring as quick-fix

Is the model complete?

MDDMA WS 2014/2015

locating and interpreting refactoring as quick-fix

Is the model complete?

MDDMA WS 2014/2015
Refactoring application

Is the model complete?

Model partially completed

Refactoring Quick Fix Dialog

Create Index Page And Data Page

Motorbike

Problems

No Dynamic Page (3 areas identified)
- (Address)
- (BankAccount)
- (Service)

Empty Entity (1 area identified)
- (Employee)

Unused Entity (1 area identified)
- (Service)
EMF Refactor: Architecture

- Specification Language
- Generation module
- Application module
- metrics, smells, refactorings
- extension points
- Eclipse Modeling Framework (EMF)
Configuration of relevant quality assurance techniques

Metrics configuration

<table>
<thead>
<tr>
<th>Selected</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓</td>
<td>A</td>
<td>Quota of abstract classes within the package</td>
</tr>
<tr>
<td>✓</td>
<td>AvASCP</td>
<td>Average number of Associations per Class</td>
</tr>
<tr>
<td>✓</td>
<td>Ca</td>
<td>Number of incoming class dependencies</td>
</tr>
<tr>
<td>✓</td>
<td>Ce</td>
<td>Number of outgoing class dependencies</td>
</tr>
<tr>
<td>✓</td>
<td>I</td>
<td>Ratio between afferent coupling and total</td>
</tr>
<tr>
<td>✓</td>
<td>NACP</td>
<td>Number of abstract Classes within the Package</td>
</tr>
<tr>
<td>✓</td>
<td>NASP</td>
<td>Number of Associations within the Package</td>
</tr>
<tr>
<td>✓</td>
<td>NATP</td>
<td>Number of Attributes in Classes within the Model</td>
</tr>
<tr>
<td>✓</td>
<td>NCCP</td>
<td>Number of concrete Classes within the Package</td>
</tr>
<tr>
<td>✓</td>
<td>NINP</td>
<td>Number of Interfaces within the Package</td>
</tr>
<tr>
<td>✓</td>
<td>NNIAACP</td>
<td>Number of non-inherited abstract classes</td>
</tr>
<tr>
<td>✓</td>
<td>NOPP</td>
<td>Number of Operations in Classes and Interfaces</td>
</tr>
<tr>
<td>✓</td>
<td>TC</td>
<td>Total Coupling: afferent coupling + efferent coupling</td>
</tr>
<tr>
<td>✓</td>
<td>TNCP</td>
<td>Total number of Classes within the Package</td>
</tr>
</tbody>
</table>

Model

<table>
<thead>
<tr>
<th>Selected</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ANA</td>
<td>Average number of ancestors of all classes</td>
</tr>
<tr>
<td></td>
<td>MaxDIT</td>
<td>Maximum of all depths of inheritance trees</td>
</tr>
<tr>
<td></td>
<td>NACM</td>
<td>Number of all ancestors of all classes in the</td>
</tr>
</tbody>
</table>

Smells configuration

<table>
<thead>
<tr>
<th>Selected</th>
<th>Name</th>
<th>Description</th>
<th>Limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓</td>
<td>Abstract Package</td>
<td>The model contains a Package</td>
<td>0.7</td>
</tr>
<tr>
<td>✓</td>
<td>Attribute Name Overridden</td>
<td>The model contains an attribute</td>
<td></td>
</tr>
<tr>
<td>✓</td>
<td>Concrete Superclass</td>
<td>There is an abstract Class having a concrete</td>
<td></td>
</tr>
<tr>
<td>✓</td>
<td>Data Clumps (Attributes)</td>
<td>The model contains classes with data clumps</td>
<td></td>
</tr>
<tr>
<td>✓</td>
<td>Data Clumps (Parameters)</td>
<td>The model contains operations with data clumps</td>
<td></td>
</tr>
<tr>
<td>✓</td>
<td>Diamond Inheritance</td>
<td>A Class inherits from another Class</td>
<td></td>
</tr>
<tr>
<td>✓</td>
<td>Empty Package</td>
<td>The model contains a package</td>
<td></td>
</tr>
<tr>
<td>✓</td>
<td>Equal Attributes in Sibling Classes</td>
<td>Each sibling class of the owning class has an equal attribute</td>
<td></td>
</tr>
<tr>
<td>✓</td>
<td>Equally Named Classes</td>
<td>The model contains two classes with an equal name</td>
<td></td>
</tr>
<tr>
<td>✓</td>
<td>Large Class</td>
<td>The model contains a class</td>
<td></td>
</tr>
<tr>
<td>✓</td>
<td>Long Parameter List</td>
<td>The model contains an operation with long</td>
<td></td>
</tr>
<tr>
<td>✓</td>
<td>No Specification</td>
<td>The model contains an abstract</td>
<td></td>
</tr>
<tr>
<td>✓</td>
<td>Primitive Obsession (Constants)</td>
<td>The model contains a primitive definition of</td>
<td></td>
</tr>
<tr>
<td>✓</td>
<td>Primitive Obsession (Primitive Type)</td>
<td>The model contains a primitive definition of</td>
<td></td>
</tr>
<tr>
<td>✓</td>
<td>Specialization Aggregation</td>
<td>The model contains a generalization</td>
<td></td>
</tr>
<tr>
<td>✓</td>
<td>Speculative Generality Class</td>
<td>The model contains an abstract</td>
<td></td>
</tr>
<tr>
<td>✓</td>
<td>Speculative Generality Interface</td>
<td>The model contains an interface</td>
<td></td>
</tr>
</tbody>
</table>


OK Cancel
Smell-Refactoring relations

Quick Fix Relations
- Metamodel: http://www.eclipse.org/simplewebmodel/1.0
- Smells possibly caused by Refactoring:
  - No Dynamic Page
- Usable Refactorings to erase Smell:
  - Insert Dynamic Pages

Quick Fix Relations
- Metamodel: http://www.eclipse.org/simplewebmodel/1.0
- Smells possibly caused by Refactoring:
  - No Dynamic Page
- Usable Refactorings to erase Smell:
  - Insert Dynamic Pages
- Refactoring:
  - Insert Dynamic Pages

Selected
- Empty Entity
- Equally Named Pages
- Missing Data Page
- Missing Index Page
- Missing Link
- Multiple Link Definitions
- No Dynamic Page
- Unused Entity

Description
- The entity does not have any attributes.
- There are pages with the same name.
- The entity is referenced by an index page.
- The entity is referenced by a data page.
- The index page is not linked by the... 
- The page has multiple links to the... 
- The entity is not referenced by a dynam...
EMF Refactor: Architecture

Eclipse Modeling Framework (EMF)

- Specification Language
- Generation module
- Application module

metrics, smells, refactorings
extension points
Specification languages

Eclipse Modeling Framework (EMF)

- OCL
- Metrics
- Refactorings

Java

Metrics, smells, refactorings
Integration of EMF Refactor tools

- 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
  - Textual Xtext editors
Flexible specification mechanisms in EMF Refactor

• **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
Conclusion

• Quality assurance for software models is an important project management task

• Software quality ≠ software model quality

• A structured model quality assurance process can be adapted to project and domain specific purposes
  – The application sub-process is performed during the modeling task, e.g., in a model review
  – The specification sub-process should be performed before (as far as possible)

• Tool support in Eclipse is given by EMF Refactor
References


