Model Consistency ensured by Metamodel Integration
MoConseMI
6th GEMOC 2018, Copenhagen

Johannes Meier    Andreas Winter

Software Engineering Group
Department of Computing Science
Carl von Ossietzky University, Oldenburg, Germany

15. October 2018
Motivation

- various Artifacts in Software Development:
  - Diagrams, DSLs, Tools, …
  - Artifacts are technically separated
  - Artifacts are interrelated regarding content

- Ensure Consistency between Artifacts automatically
Motivation

- Various Artifacts in Software Development:
  - Diagrams, DSLs, Tools, ...
  - Artifacts are technically separated
  - Artifacts are interrelated regarding content

- Ensure Consistency between Artifacts automatically

<table>
<thead>
<tr>
<th>ID</th>
<th>Author</th>
<th>Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>Andreas Winter</td>
<td>The student must be able to register for an event.</td>
</tr>
<tr>
<td>R2</td>
<td>Johannes Meier</td>
<td>The student must be enrolled at the university.</td>
</tr>
</tbody>
</table>
Motivation

- various Artifacts in Software Development:
  - Diagrams, DSLs, Tools, …
  - Artifacts are technically separated
  - Artifacts are interrelated regarding content

- Ensure Consistency between Artifacts automatically

<table>
<thead>
<tr>
<th>ID</th>
<th>Author</th>
<th>Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>Andreas Winter</td>
<td>The student must be able to register for an event.</td>
</tr>
<tr>
<td>R2</td>
<td>Johannes Meier</td>
<td>The student must be enrolled at the university.</td>
</tr>
</tbody>
</table>

University[1] students[*] University

Student
  +Name : String
Motivation

- various Artifacts in Software Development:
  - Diagrams, DSLs, Tools, ...
  - Artifacts are technically separated
  - Artifacts are interrelated regarding content

- Ensure Consistency between Artifacts automatically
Motivation

- various Artifacts in Software Development:
  - Diagrams, DSLs, Tools, ...
  - Artifacts are technically separated
  - Artifacts are interrelated regarding content
- Ensure Consistency between Artifacts automatically

<table>
<thead>
<tr>
<th>ID</th>
<th>Author</th>
<th>Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>Andreas Winter</td>
<td>The student must be able to register for an event.</td>
</tr>
<tr>
<td>R2</td>
<td>Johannes Meier</td>
<td>The student must be enrolled at the university.</td>
</tr>
</tbody>
</table>

![Diagram and code snippet]

Johannes Meier, Andreas Winter
Motivation

- various Artifacts in Software Development:
  - Diagrams, DSLs, Tools, ...
  - Artifacts are technically separated
  - Artifacts are interrelated regarding content

- Ensure Consistency between Artifacts automatically
Motivation

- various Artifacts in Software Development:
  - Diagrams, DSLs, Tools, ...
  - Artifacts are technically separated
  - Artifacts are interrelated regarding content
- Ensure Consistency between Artifacts automatically
Motivation

- Various Artifacts in Software Development:
  - Diagrams, DSLs, Tools, ...
  - Artifacts are technically separated
  - Artifacts are interrelated regarding content
- Ensure Consistency between Artifacts automatically
Motivation

- various Artifacts in Software Development:
  - Diagrams, DSLs, Tools, …
  - Artifacts are technically separated
  - Artifacts are interrelated regarding content
- Ensure Consistency between Artifacts automatically
Motivation

- various Artifacts in Software Development:
  - Diagrams, DSLs, Tools, ...
  - Artifacts are technically separated
  - Artifacts are interrelated regarding content
- Ensure Consistency between Artifacts automatically
Motivation

- various Artifacts in Software Development:
  - Diagrams, DSLs, Tools, ...
  - Artifacts are technically separated
  - Artifacts are interrelated regarding content
- Ensure Consistency between Artifacts automatically
Problem

There are further Software Development Projects:
- e.g. with formal Specifications, C++, Test Cases, Documentation, Project Management, Build Tools
- Traceability
- → further Consistency issues

General Problem:
- Artifacts are technically separated, but interrelated contentwise
- specific Consistency Rules have to be fulfilled automatically

Goal

Ensure Consistency between Artifacts automatically!
- Artifact == Model + Metamodel (structural formalization [CNS12])
- → Model Integration
Challenges
Challenges

1. Formalize Consistency Rules

<table>
<thead>
<tr>
<th>ID</th>
<th>Author</th>
<th>Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>Andreas Winter</td>
<td>The student must be able to register for an event.</td>
</tr>
<tr>
<td>R2</td>
<td>Johannes Meier</td>
<td>The student must be enrolled at the university.</td>
</tr>
</tbody>
</table>

Motivation
Challenges

1. Formalize Consistency Rules

Sourcecode and Class Diagrams describe the same set of classes, identified by their class name.
Challenges

1. Formalize Consistency Rules

Each Requirement is linked with all Methods whose Name is contained in the Text of the Requirement.

Sourcecode and Class Diagrams describe the same set of classes, identified by their class name.
Challenges

1. Formalize Consistency Rules
2. Create explicit SUM(M)
Challenges

1. Formalize Consistency Rules
2. Create explicit SUM(M)

- reuse Model Techniques which work only with one Model
- used as single Point of Truth
- Single Underlying Model [ASB09]
- SUMM and SUM are explicit
Challenges

1. Formalize Consistency Rules
2. Create explicit SUM(M)

- reuse Model Techniques which work only with one Model
- used as single Point of Truth
- Single Underlying Model [ASB09]
- SUMM and SUM are explicit

Motivation
Challenges

1. Formalize Consistency Rules

2. Create explicit SUM(M)
Challenges

1. Formalize Consistency Rules
2. Create explicit SUM(M)
3. Support initial (Meta)Models:
Challenges

1. Formalize Consistency Rules
2. Create explicit SUM(M)
3. Support initial (Meta)Models:
   a. Reuse initial Models

- existing Metamodels: DSLs, Environments, Tools, …
- existing Models: ongoing projects, legacy data, …
Challenges

1. Formalize Consistency Rules
2. Create explicit SUM(M)
3. Support initial (Meta)Models:
   a. Reuse initial Models

- existing Metamodels: DSLs, Environments, Tools, …
- existing Models: ongoing projects, legacy data, …
Challenges

1. Formalize Consistency Rules
2. Create explicit SUM(M)
3. Support initial (Meta)Models:
   a. Reuse initial Models
   b. Fix initial Inconsistencies
Challenges

1. Formalize Consistency Rules
2. Create explicit SUM(M)
3. Support initial (Meta)Models:
   a. Reuse initial Models
   b. Fix initial Inconsistencies
   c. Consistent initial Models
Challenges

1. Formalize Consistency Rules
2. Create explicit SUM(M)
3. Support initial (Meta)Models:
   a. Reuse initial Models
   b. Fix initial Inconsistencies
   c. Consistent initial Models
4. Ensure Model Consistency
Related Work

<table>
<thead>
<tr>
<th></th>
<th>synthetic</th>
<th>OSM</th>
<th>projectional</th>
<th>Vitruvius</th>
<th>GEMOC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Formalize Consistency Rules</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>2. Create explicit SUM(M)</td>
<td>✗</td>
<td>✔️</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>3a. Reuse initial Models</td>
<td>✔️</td>
<td>✗</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>3b. Fix initial Inconsistencies</td>
<td>✔️</td>
<td>–</td>
<td>✗</td>
<td>?</td>
<td>✔️</td>
</tr>
<tr>
<td>3c. Consistent initial Models</td>
<td>✔️</td>
<td>–</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>4. Ensure Model Consistency</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
</tbody>
</table>

- ISO Standard 42010:2011 [IEE11]: synthetic vs. projectional
- synthetic: TGGs [SK08], QVT-R [RJV09], explicit correspondences [EEC+14]
- OSM: Single Underlying (Meta)Model (SUM(M)) [ASB09]
- Vitruvius [KBL13, BHK+14]
- GEMOC Approach [LDC18]
Metamodel Integration

**Challenges**

1. Formalize Consistency Rules
2. Create explicit SUM(M)
3. Support initial (Meta)Models:
   a. Reuse initial Models
   b. Fix initial Inconsistencies
   c. Consistent initial Models
4. Ensure Model Consistency

**Activities**
Metamodel Integration

Challenges

1. Formalize Consistency Rules
2. Create explicit SUM(M)
3. Support initial (Meta)Models:
   a. Reuse initial Models
   b. Fix initial Inconsistencies
   c. Consistent initial Models
4. Ensure Model Consistency

Activities

1. Configuration of Operators

SUM(M)
Metamodel Integration

**Challenges**

1. Formalize Consistency Rules
2. Create explicit $\text{SUM}(M)$
3. Support initial (Meta)Models:
   a. Reuse initial Models
   b. Fix initial Inconsistencies
   c. Consistent initial Models
4. Ensure Model Consistency

**Activities**

1. Configuration of Operators

3. Support initial (Meta)Models:
   a. Reuse initial Models
   b. Fix initial Inconsistencies
   c. Consistent initial Models

Requirements → 1 Add Association → 2 Change Multiplicity → 3 Change Multiplicity → 4 Merge Classes → 5 Merge Attributes → 6 $\text{SUM}(M)$
### Challenges

1. Formalize Consistency Rules

2. Create explicit SUM(M)

3. Support initial (Meta)Models:
   - a. Reuse initial Models
   - b. Fix initial Inconsistencies
   - c. Consistent initial Models

4. Ensure Model Consistency

### Activities

1. Configuration of Operators

2. Initialization of SUM

3. Consistency Assurance

---

**Java**

1. Add Association

2. Change Multiplicity

3. Merge Classes

4. Merge Attributes

**ClassDiagram**

Challenges

1. Formalize Consistency Rules
2. Create explicit SUM(M)
3. Support initial (Meta)Models:
   a. Reuse initial Models
   b. Fix initial Inconsistencies
   c. Consistent initial Models
4. Ensure Model Consistency

Activities

1. Configuration of Operators
2. Initialization of SUM
3. Consistency Assurance

Requirements

1. Add Association
2. Change Multiplicity
3. Change Multiplicity
4. Merge Classes
5. Merge Attributes
6. SUM(M)
Metamodel Integration

**Challenges**

1. Formalize Consistency Rules
2. Create explicit $SUM(M)$
3. Support initial (Meta)Models:
   a. Reuse initial Models
   b. Fix initial Inconsistencies
   c. Consistent initial Models
4. Ensure Model Consistency

**Activities**

1. Configuration of Operators
2. Initialization of $SUM$
Metamodel Integration

**Challenges**

1. Formalize Consistency Rules
2. Create explicit SUM(M)
3. Support initial (Meta)Models:
   a. Reuse initial Models
   b. Fix initial Inconsistencies
   c. Consistent initial Models
4. Ensure Model Consistency

**Activities**

1. Configuration of Operators
2. Initialization of SUM
3. Consistency Assurance

---

**Requirements**

1. Add Association
2. Change Multiplicity
3. Change Multiplicity
4. Merge Classes
5. Merge Attributes
6. SUM(M)

---

Java

ClassDiagram
Metamodel Integration

**Challenges**

1. Formalize Consistency Rules
2. Create explicit $\text{SUM}(M)$
3. Support initial (Meta)Models:
   a. Reuse initial Models
   b. Fix initial Inconsistencies
   c. Consistent initial Models
4. Ensure Model Consistency

**Activities**

1. Configuration of Operators
2. Initialization of $\text{SUM}$
3. Consistency Assurance

---

Java

ClassDiagram

1. Add Association
2. Change Multiplicity
3. Change Multiplicity
4. Merge Classes
5. Merge Attributes
6. $\text{SUM}(M)$

Requirements

---

Johannes Meier, Andreas Winter
Metamodel Integration

Challenges

1. Formalize Consistency Rules
2. Create explicit SUM(M)
3. Support initial (Meta)Models:
   a. Reuse initial Models
   b. Fix initial Inconsistencies
   c. Consistent initial Models
4. Ensure Model Consistency

Activities

1. Configuration of Operators
2. Initialization of SUM
3. Consistency Assurance

SUM(M)

Requirements 1 Add Association 2 Change Multiplicity 3 Change Multiplicity 4 Merge Classes 5 Merge Attributes 6 SUM(M)

Java ClassDiagram

15. October 2018
1. Configuration of Operators
1. Configuration of Operators

- **Operator**:
  - **MM**: Metamodel Decisions
  - **M**: Model Decisions
  - **MM'**: Metamodel Decisions
  - **M'**: Model Decisions

- **Requirements**:
  - Java
  - ClassDiagram

- **Steps**:
  1. Add Association
  2. Change Multiplicity
  3. Change Multiplicity
  4. Merge Classes
  5. Merge Attributes
  6. SUM(M)
1. Configuration of Operators

![Diagram of Metamodel Integration]

- **MM** → **Δ_{MM}** → **MM’**
- **M** → **Δ_{M}** → **M’**

Operator connects Metamodel Decisions and Model Decisions.

- **Requirements**
  - 1. Add Association
  - 2. Change Multiplicity
  - 3. Change Multiplicity
  - 4. Merge Classes
  - 5. Merge Attributes
  - 6. SUM(M)

Chechik, Nejati, Sabetzadeh: A Relationship-Based Approach to Model Integration (2012)

1. Configuration of Operators

- **Chechik, Nejati, Sabetzadeh**: *A Relationship-Based Approach to Model Integration* (2012)
  - merge, composition, weaving

- **Herrmannsdoerfer et al.**: *An Extensive Catalog of Operators for the Coupled Evolution of Metamodels and Models* (2011)
  - extended Coupled Operators

---

**Diagram:**

```
    MM  \[\Delta_{MM}\]  MM'
    \uparrow                      \downarrow
     \quad Operator\quad
    \downarrow                      \uparrow
    M  \[\Delta_{M}\]  M'
```

**Requirements:**

1. Add Association
2. ClassDiagram
3. Change Multiplicity
4. Change Multiplicity
5. Merge Classes
6. Merge Attributes

**Java**

**SUM(M)**
1. Configuration of Operators

- **MM**: Metamodel
- **M**: Model
- **MM'**: Metamodel
- **M'**: Model
- **MM'**: Metamodel
- **M': Model
- **M'**: Model
- **M''**: Model
- **Δ_{MM}**: Metamodel Decision
- **Δ_{M}**: Model Decision
- **Δ'_{M}**: Inverse Model Decision
- **Δ'_{MM}**: Inverse Metamodel Decision

**Operators**

- **Operator**
- **Inverse Operator**

**Decision Flows**

- **MM**: Metamodel
- **M**: Model
- **MM'**: Metamodel
- **M'**: Model
- **M'**: Model
- **MM'**: Metamodel

**References**

- Chechik, Nejati, Sabetzadeh: A Relationship-Based Approach to Model Integration (2012)
1. Configuration of Operators

- **Operator**
  - $\Delta_{MM}$
  - $\Delta_{M}$
  - $\Delta'_{M}$
  - $\Delta'_{MM}$

- **Inverse Operator**

- **Metamodel Decisions**
- **Model Decisions**

- **Metamodel** $MM$ 
  - $\Delta_{MM}$ 
  - $\Delta'_{MM}$ 
  - $= ?$ 

- **Model** $M$ 
  - $\Delta_{M}$ 
  - $\Delta'_{M}$ 

- **Model $M'$**

- **Model $M''$**

- **Model $M'''$**

- **Requirements**
  - Java
  - Add Association
  - Class Diagram
  - Change Multiplicity
  - Change Multiplicity
  - Merge Classes
  - Merge Attributes
1. Configuration of Operators

Metamodel Integration

- **Operator**
  - MM $\rightarrow$ MM'
  - M $\rightarrow$ M'
  - M'' $\rightarrow$ M'
  - MM $\rightarrow$ MM'

- **Inverse Operator**
  - M' $\rightarrow$ M
  - M' $\rightarrow$ M''
  - MM' $\rightarrow$ MM

- **Metamodel Decisions**
  - $\Delta_{MM}$
  - $\Delta_{M}$
  - $\Delta_{M'}$

- **Model Decisions**
  - $\Delta^{-1}_{MM}$
  - $\Delta^{-1}_{M}$
  - $\Delta^{-1}_{M'}$
2. Initialization of SUM: Overview

- **1. Add Association**
  - Java
  - ClassDiagram

- **2. Change Multiplicity**
  - Add Association

- **3. Change Multiplicity**
  - Add Association

- **4. Merge Classes**
  - Add Association

- **5. Merge Attributes**
  - Add Association

- **6. SUM(M)**

Requirements: J. Meier, A. Winter
2. Initialization of SUM: Overview

Requirements
1. Java
2. Add Association
3. Change Multiplicity
4. Change Multiplicity
5. Merge Classes
6. Merge Attributes

SUM(M)
2. Initialization of SUM: Overview

Requirements

1. Add Association
2. Change Multiplicity
3. Change Multiplicity
4. Merge Classes
5. Merge Attributes
6. SUM(M)
2. Initialization of SUM: Overview

**Requirements**
1. Add Association
2. Change Multiplicity
3. Change Multiplicity
4. Merge Classes
5. Merge Attributes
6. SUM(M)

**Java**

**ClassDiagram**

**ProjectData**
- integrator [0..1]
  - containsRequirementsSpecification [+]

**JavaASG**
- asg [1]
  - classes [+]

**ClassDiagram**
- Class
  - className : EString [1]

**ClassType**
- name : EString [1] = "Student"

**Method**
- name : EString [1] = "register"

**Method**
- name : EString [1] = "start"

**Requirement**
- rowNumber : EInt [0..1] = 1
- id : EString [0..1] = "R1"
- author : EString [0..1] = "Andreas Winter"
- text : EString [0..1] = "The student must be able to register for an event."

**Requirement**
- rowNumber : EInt [0..1] = 2
- id : EString [0..1] = "R2"
- author : EString [0..1] = "Johannes Meier"
- text : EString [0..1] = "The student must be enrolled at the university."
2. Initialization of SUM: Overview

- **Requirements**
  - Java
  - Add Relation
  - Change Multiplicity
  - Change Multiplicity
  - Merge Classes
  - Merge Attributes

- **ClassDiagram**
  - **ProjectData**
    - integrator [0..1]
    - containsRequirementsSpecification [+]
  - **JavaASG**
    - asg [1]
    - classes [+]
  - **RequirementsSpecification**
    - container [1]
    - content [+]
  - **Requirement**
    - rowNumber : EInt
    - id : EString
    - author : EString
    - text : EString

- **ClassType**
  - name : EString
  - className : EString
  - className = "Student"
  - className = "University"

- **Method**
  - name : EString
  - M1 : Method
    - name = "register"
  - M2 : Method
    - name = "start"

- **ProjectData**
  - integrator [0..1]
  - containsJavaASG [+]

- **JavaASG**
  - asg [1]
  - classes [+]

- **RequirementsSpecification**
  - container [1]
  - content [+]

- **Requirement**
  - rowNumber : EInt
  - id : EString
  - author : EString
  - text : EString

- **ClassDiagram**
  - **ProjectData**
    - integrator [0..1]
    - containsRequirementsSpecification [+]
  - **JavaASG**
    - asg [1]
    - classes [+]
  - **RequirementsSpecification**
    - container [1]
    - content [+]
  - **Requirement**
    - rowNumber : EInt
    - id : EString
    - author : EString
    - text : EString

- **ClassType**
  - name : EString
  - className : EString
  - className = "Student"
  - className = "University"

- **Method**
  - name : EString
  - M1 : Method
    - name = "register"
  - M2 : Method
    - name = "start"

- **ProjectData**
  - integrator [0..1]
  - containsJavaASG [+]

- **JavaASG**
  - asg [1]
  - classes [+]

- **RequirementsSpecification**
  - container [1]
  - content [+]

- **Requirement**
  - rowNumber : EInt
  - id : EString
  - author : EString
  - text : EString
2. Initialization of SUM: Overview

Requirements

1. Add Association
2. Change Multiplicity
3. Change Multiplicity
4. Merge Classes
5. Merge Attributes
6. SUM(M)
2. Initialization of SUM: Overview

- **Add Association**
- **Change Multiplicity**
- **Change Multiplicity**
- **Merge Classes**
- **Merge Attributes**
- **SUM(M)**
2. Initialization of SUM: Overview

**Requirements**
- **Add Association**
- **Change Multiplicity**
- **Merge Classes**
- **Merge Attributes**

**Java**

**ClassDiagram**

- **ProjectData**
- **JavaASG**
- **RequirementsSpecification**

**Class**
- **ClassType**
- **Method**

**Association**

- **data**
- **containsClassDiagram**

**RequirementsSpecification**
- **rowNumber : EInt [0..1]**
- **id : EString [0..1]**
- **author : EString [0..1]**
- **text : EString [0..1]**

**JavaASG**
- **name : EString [1]**
- **className : EString [1]**

**ClassType**
- **name : EString [1]**
- **className : EString [1]**

**Method**
- **name : EString [1]**
- **calledBy : [0..1]**
- **calling : [0..1]**
- **fulfilled : [0..1]**

**Class**
- **className : EString [1]**

**Association**
- **name : EString [1]**
- **lowerBound : EInt [0..1]**
- **upperBound : EInt [0..1]**

**ClassDiagram**
- **diagram : [1]**
- **classes : [1]**
- **name : EString [1]**
- **className : EString [1]**

**Requirements**

- **R1 : Requirement**
  - **rowNumber : EInt [0..1] = 1**
  - **id : EString [0..1] = "R1"**
  - **author : EString [0..1] = "Andreas Winter"**
  - **text : EString [0..1] = "The student must be able to register for an event."**

- **R2 : Requirement**
  - **rowNumber : EInt [0..1] = 2**
  - **id : EString [0..1] = "R2"**
  - **author : EString [0..1] = "Johannes Meier"**
  - **text : EString [0..1] = "The student must be enrolled at the university."**

**ClassType**
- **name : EString [1] = "Student"**
- **className : EString [1] = "University"**

**Method**
- **name : EString [1] = "register"**
- **name : EString [1] = "start"**

**ClassDiagram**
- **diagram : [1]**
- **classes : [0..1]**
- **name : EString [1]**
- **className : EString [1]**

**RequirementsSpecification**
- **contain : [1]**
- **content : [0..1]**
- **rowNumber : EInt [0..1]**
- **id : EString [0..1]**
- **author : EString [0..1]**
- **text : EString [0..1]**

**ProjectData**
- **integrator : [0..1]**
- **containsClassDiagram : [1]**
- **containsJavaASG : [1]**

**SUM(M)**
2. Initialization of SUM: Overview

- Initialization of SUM: Overview

**Requirements**
1. **Java**
2. **Add**
3. **ClassDiagram**
4. **Change Multiplicity**
5. **Merge Classes**
6. **Merge Attributes**

**SUM(M)**

**Java**

**ClassDiagram**

1. **Add**
2. **Association**
3. **Change Multiplicity**
4. **Change Multiplicity**
5. **Merge**
6. **Merge**

**Java**

**ClassDiagram**

1. **Add**
2. **Association**
3. **Change Multiplicity**
4. **Change Multiplicity**
5. **Merge**
6. **Merge**

**SUM(M)**
2. Initialization of SUM: Overview

- **Initialization of SUM**: Overview
  - **Requirements**
    - **1:** Initialize SUM
      - **Add Association**
      - **2:** Change Multiplicity
      - **3:** Change Multiplicity
      - **4:** Change Multiplicity
      - **5:** Merge Classes
      - **6:** Merge Attributes
      - **SUM(M)**

- **Summarized Overview**:
  - **Requirements**
  - **ClassDiagram**
  - **ProjectData**
  - **JavaASG**
  - **RequirementsSpecification**

- **Details**:
  - **ClassDiagram**
  - **JavaASG**
  - **RequirementsSpecification**

- **Diagram**:
  - **ClassDiagram**
  - **JavaASG**
  - **RequirementsSpecification**

- **Example**:
  - **ClassDiagram**
    - **ClassType**
      - **Student**
        - **University**
      - **Class**
        - **Methods**
          - **register**
    - **Class**
      - **Methods**
        - **start**

- **Java**
  - **ClassDiagram**
  - **ProjectData**
  - **JavaASG**
  - **RequirementsSpecification**

- **Project Data**
  - **ClassDiagram**
  - **JavaASG**
  - **RequirementsSpecification**

- **Requirements**
  - **Requirement**
    - **R1**
      - **Row Number**: 1
      - **Id**: R1
      - **Author**: Andreas Winter
      - **Text**: The student must be able to register for an event.
    - **R2**
      - **Row Number**: 2
      - **Id**: R2
      - **Author**: Johannes Meier
      - **Text**: The student must be enrolled at the university.

- **Association**
  - **Name**: EString
  - **Lower Bound**: EInt
  - **Upper Bound**: EInt

- **Class Type**
  - **Name**: EString
  - **Class Name**: EString

- **Method**
  - **Name**: EString
  - **Called By**: Method
    - **Calling**: Method
  - **Fulfilled By**: Method
    - **Fulfilled**: Method

- **Diagram**
  - **Classes**: Class
    - **Methods**: Method
  - **Assemblies**: Class
    - **Methods**: Method
  - **Requirements**: Requirement
    - **Row Number**: EInt
    - **Id**: EString
    - **Author**: EString
    - **Text**: EString

- **Metamodel Integration**
  - **15. October 2018 8 / 12**

- **Author**: Johannes Meier, Andreas Winter
2. Initialization of SUM: Overview

Requirements

1. Java
2. Add Association
3. Change Multiplicity
4. Change Multiplicity
5. Merge Classes
6. Merge Attributes

SUM(M)

Johannes Meier, Andreas Winter

Metamodel Integration

15. October 2018 8 / 12
2. Initialization of SUM: Details

Java

ClassDiagram

1. Add Association
2. Change Multiplicity
3. Change Multiplicity
4. Merge Classes
5. Merge Attributes
6. SUM(M)

Requirements

1. 2. 3. 4. 5. 6.结局
2. Initialization of SUM: Details

Requirements → 1 Add Association → 2 ClassDiagram → 3 Change Multiplicity → 4 Change Multiplicity → 5 Merge Classes → 6 Merge Attributes → SUM(M)

Java

ClassDiagram

SUM(M)

Metamodel Integration

Johannes Meier, Andreas Winter

Metamodel Integration

15. October 2018 9 / 12
2. Initialization of SUM: Details

3a. Forward Execution to create the initial SUM

3b. Backward Execution to fix the initial Models inverse to each other

Java

ClassDiagram

ClassType
name : String

Class
className : String

: ClassType
name = “University”
className = “University”

: ClassType
name = “Student”
className = “Student”

Requirements

1 Add Association
2 Change Multiplicity
3 Change Multiplicity
4 Merge Classes
5 Merge Attributes
6 SUM(M)
2. Initialization of SUM: Details

3a. Forward Execution to create the initial SUM

3b. Backward Execution to fix the initial Models inverse to each other

Requirements

1. Add Association
2. Change Multiplicity
3. Change Multiplicity
4. Merge Classes
5. Merge Attributes
6. SUM(M)

Java

ClassDiagram

ClassType
name : String

Class
className : String

: ClassType
name = “University”

className = “University”

: ClassType
name = “Student”

className = “Student”

SUM(M)

Metamodel Integration

15. October 2018
2. Initialization of SUM: Details

3a. Forward Execution

to create the initial SUM

3b. Backward Execution

to fix the initial Models

inverse to each other

Java

ClassDiagram

SUM(M)

Metamodel

Changes

Model

Changes

1. Add Association

2. Change Multiplicity

3. Change Multiplicity

4. Merge Classes

5. Merge Attributes

SUM(M)

Requirements

Java

ClassDiagram
2. Initialization of SUM: Details

3a. Forward Execution
   to create the initial SUM

3b. Backward Execution
   to fix the initial Models

Merge
Classes

SUM(M)

ClassDiagram

Java

ClassType
name : String

Class
className : String

: ClassType
name = “University”
className = “University”

: ClassType
name = “Student”

SUM(M)

ClassType
name : String
className : String

: ClassType
name = “University”
className = “University”

: ClassType
name = “Student”
className = “Student”

Metamodel Integration
Johannes Meier, Andreas Winter
15. October 2018
2. Initialization of SUM: Details

3a. Forward Execution to create the initial SUM

3b. Backward Execution to fix the initial Models

SUM(M)

Metamodel Changes

Model Changes

Model Changes

Metamodel Changes

Split Class

Merge Classes

Class

ClassType

name : String
className : String

name = “University”
className = “University”

name = “Student”
className = “Student”

Class

ClassType

name : String
className : String

name = “University”
className = “University”

name = “Student”
className = “Student”

Java

ClassDiagram

ClassType

name : String
className : String

name = “University”
className = “University”

name = “Student”
className = “Student”

Add Association

Change Multiplicity

Split Class

SUM(M)

Metamodel Changes

Model Changes

Model Changes

Metamodel Changes

Requirements

1. Java

2. Add Association

3. Change Multiplicity

4. Change Multiplicity

5. Split Class

Johannes Meier, Andreas Winter

Metamodel Integration

15. October 2018 9 / 12
2. Initialization of SUM: Details

3a. Forward Execution to create the initial SUM

3b. Backward Execution to fix the initial Models

SUM(M)

Model Changes

Model Changes

Model Changes

Metamodel Changes

Java

ClassDiagram

Merge Classes

Split Class

ClassType

name : String

className : String

Class

name = "University"

className = "University"

: ClassType

name = "Student"

className = "Student"

ClassType

name : String

className : String

: Class

name = "University"

className = "University"

ClassType

name : String

className : String

: Class

name = "Student"

className = "Student"

ClassType

name : String

className : String

Class

name = "University"

className = "University"

ClassType

name : String

className : String

: Class

name = "Student"

className = "Student"

ClassType

name : String

className : String

Class

name = "University"

className = "University"

Java

Add Association

Change Multiplicity

Change Multiplicity

Split Class

Johannes Meier, Andreas Winter

Metamodel Integration

15. October 2018 9 / 12
3. Consistency Assurance: Details

1. Initialization (forward)
2. Run (backward)
3. Run (forward)

Requirements

Java

ClassDiagram

Add Association

Change Multiplicity

Change Multiplicity

Merge Classes

Merge Attributes

SUM(M)
3. Consistency Assurance: Details

1. Initialization (forward)
2. Run (backward)
3. Run (forward)

Requirements

Java

ClassDiagram

Add Association

Change Multiplicity

Change Multiplicity

Merge Classes

Merge Attributes

SUM(M)
3. Consistency Assurance: Details

1. Initialization (forward)

1. Create Link between R2 "enrole" and M2 "start"
2. Rename Method M1 from "register" to "enrole"

2. Run (backward)

3. Run (forward)

Add Association

Requirements

<table>
<thead>
<tr>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>text : String</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>name : String</td>
</tr>
</tbody>
</table>

Java

<table>
<thead>
<tr>
<th>R1 : Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>text = &quot;... register ...&quot;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>M1 : Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>name = &quot;register&quot;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>R2 : Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>text = &quot;... enrole ...&quot;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>M2 : Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>name = &quot;start&quot;</td>
</tr>
</tbody>
</table>

SUM(M)

<table>
<thead>
<tr>
<th>Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>text : String</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>name : String</td>
</tr>
</tbody>
</table>

Add Association

<table>
<thead>
<tr>
<th>R1 : Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>text = &quot;... register ...&quot;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>M1 : Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>name = &quot;register&quot;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>R2 : Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>text = &quot;... enrole ...&quot;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>M2 : Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>name = &quot;start&quot;</td>
</tr>
</tbody>
</table>

Model Changes

Johannes Meier, Andreas Winter
Metamodel Integration
15. October 2018
3. Consistency Assurance: Details

1. Initialization (forward)

Add Association

R1 : Requirement
   text = "... register ...

M1 : Method
   name = "register"

R2 : Requirement
   text = "... enrol ...

M2 : Method
   name = "start"

2. Rename Method M1 from "register" to "enrole"

3. Run (forward)

4. Run (backward)

Add Association

SUM(M)

R1 : Requirement
   text = "... register ...

M1 : Method
   name = "register"

R2 : Requirement
   text = "... enrol ...

M2 : Method
   name = "start"

User

1. Create Link between R2 "enrole" and M2 "start"

Model Changes

Requirements

ClassDiagram

Java

SUM(M)

Model Changes

Java

SUM(M)

Metamodel Changes

Johannes Meier, Andreas Winter
Metamodel Integration
15. October 2018
3. Consistency Assurance: Details

1. Create Link between R2 “enrole” and M2 “start”
2. Rename Method M1 from “register” to “enrole”

Model Changes

Requirements

Java

SUM(M)

1. Initialization (forward)
2. Run (backward)

1. Delete Association
2. Change Multiplicity
3. Change Multiplicity
4. Split Class Into Two
5. Split Attribute
6. SUM(M)
3. Consistency Assurance: Details

1. Initialization (forward)
   - Create Link between R2 “enrole” and M2 “start”
   - Rename Method M1 from “register” to “enrole”

2. Run (backward)
   - Delete Association

Requirements

- R1: Requirement
  - text: "... register ...

- R2: Requirement
  - text: "... enrole ...

Java

- M1: Method
  - name: "register"

- M2: Method
  - name: "start"

SUM(M)

- R1: Requirement
  - text: "... register ...

- R2: Requirement
  - text: "... enrole ...

- M1: Method
  - name: "enrole"

- M2: Method
  - name: "start"
3. Consistency Assurance: Details

1. Initialization (forward)
   - Create Link between R2 “enrole” and M2 “start”
   - Rename Method M1 from “register” to “enrole”

2. Run (backward)

- Requirements
- Java
- SUM(M)
- Add Association
- Delete Association
- ClassDiagram
- SUM(M)

- Requirement
text : String
- Method
name : String

- R1 : Requirement
text = "... register ..."
- M1 : Method
name = "register"

- R2 : Requirement
text = "... enrole ..."
- M2 : Method
name = "start"

- R1 : Requirement
text = "... register ...
- M1 : Method
name = "enrole"

- R2 : Requirement
text = "... enrole ...
- M2 : Method
name = "start"

- R1 : Requirement
text = "... register ...
- M1 : Method
name = "enrole"

- R2 : Requirement
text = "... enrole ...
- M2 : Method
name = "start"

- R1 : Requirement
text = "... register ...
- M1 : Method
name = "enrole"

- R2 : Requirement
text = "... enrole ...
- M2 : Method
name = "start"

Metamodel Integration

Johannes Meier, Andreas Winter
15. October 2018
3. Consistency Assurance: Details

```
<table>
<thead>
<tr>
<th>Requirement</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>text : String</td>
<td>name : String</td>
</tr>
</tbody>
</table>

```

- **R1**: Requirement
  - text = "... register ...

- **M1**: Method
  - name = "register"

- **R2**: Requirement
  - text = "... enrol ...

- **M2**: Method
  - name = "start"

**SUM(M)**
```
<table>
<thead>
<tr>
<th>Requirement</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>text : String</td>
<td>name : String</td>
</tr>
</tbody>
</table>
```

**Add Association**

1. Create Link between R2 "enrol" and M2 "start"
2. Rename Method M1 from "register" to "enrole"

**Delete Association**

- **R1**: Requirement
  - text = "... register ...

- **M1**: Method
  - name = "enrole"

- **R2**: Requirement
  - text = "... enrol ...

- **M2**: Method
  - name = "start"

**Model Changes**

1. Initialization (forward)
2. Run (backward)
3. Run (forward)

---

Johannes Meier, Andreas Winter

Metamodel Integration

15. October 2018 10 / 12
3. Consistency Assurance: Details

1. Initialization (forward)

2. Run (backward)

3. Run (forward)

4. Add Association

5. Delete Association

6. Add Association

1. Create Link between R2 "enrole" and M2 "start"

2. Rename Method M1 from "register" to "enrole"
3. Consistency Assurance: Details

1. Initialization (forward)
2. Run (backward)
3. Run (forward)

- User
- Requirements
- Java
- Metamodel Changes
- SUM(M)

Add Association
- R1 : Requirement
  text = "... register ...
  M1 : Method
  name = "register"
- R2 : Requirement
  text = "... enrol ...
  M2 : Method
  name = "start"

Delete Association
- R1 : Requirement
  text = "... register ...
  M1 : Method
  name = "enrole"
- R2 : Requirement
  text = "... enrol ...
  M2 : Method
  name = "start"

Add Association
- R1 : Requirement
  text = "... register ...
  M1 : Method
  name = "enrole"
- R2 : Requirement
  text = "... enrol ...
  M2 : Method
  name = "start"

1. Create Link between R2 "enrole" and M2 "start"
2. Rename Method M1 from "register" to "enrole"
3. Consistency Assurance: Details

1. Initialization (forward)
   - Create Link between R2 "enrole" and M2 "start"
   - Rename Method M1 from "register" to "enrole"

2. Run (backward)
   - Rename Method M1 from "enrole" to "register"
   - Create Link between R2 "enrole" and M2 "start"

3. Run (forward)
   - Rename Method M1 from "enrole" to "register"
   - Create Link between R2 "enrole" and M2 "start"
3. Consistency Assurance: Details

1. Initialization (forward)

2. Run (backward)

3. Run (forward)

1. Create Link between R2 "enrole" and M2 "start"
2. Rename Method M1 from "register" to "enrole"

**SUM(M)**

**Metamodel Changes**

**Model Changes**

**Requirements**

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>text : String</td>
<td>name : String</td>
</tr>
</tbody>
</table>

**Java**

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>text : String</td>
<td>name : String</td>
</tr>
</tbody>
</table>

**Add Association**

**Delete Association**

**Add Association**

**Inverse to each other**

Δ₁^M

Δ₂^M

Δ₃^M

ΔUser
Operators: Summary

- **Metamodel Change** $\Delta_{MM}$:
  small Change in the Metamodel

- **Model Change** $\Delta_{M}$:
  handle Model-Co-Evolution (Coupled Operators [H VW11])

- Configurations by the Methodologist:
  - **Metamodelling Decisions**: set Properties for wanted Metamodel Changes
  - **Model Decisions**: describe Model Changes for Consistency Rules

- **Bi-Directionality only for MM**: combine with inverse Operator

- currently 20 Operators implemented
Summary

Operator-based bottom-up SUM-Approach for Model Consistency:

- Methodologist configures arbitrary, but stable Chain of configured Operators (once)
- User applies Changes and Model Consistency is ensured automatically by executing the Operator Chain
- → separated Models are migrated to projectional Views on the SUM
Conclusion

Literature I


Literature II


