# 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

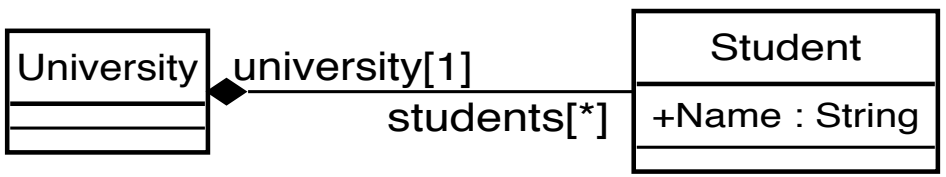
- various Artifacts in Software Development:
  - Diagrams, DSLs, Tools, ...
  - Artifacts are technically separated
  - Artifacts are interrelated regarding content
- Ensure Consistency between Artifacts automatically

| ID | Author | Text |
|---|---|---|
| R1 | Andreas Winter | The student must be able to register for an event. |
| R2 | Johannes Meier | The student must be enroled at the university. |

# Motivation

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

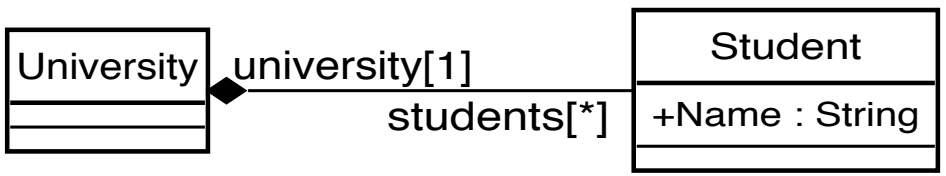
| ID | Author | Text |
|---|---|---|
| R1 | Andreas Winter | The student must be able to register for an event. |
| R2 | Johannes Meier | The student must be enroled at the university. |

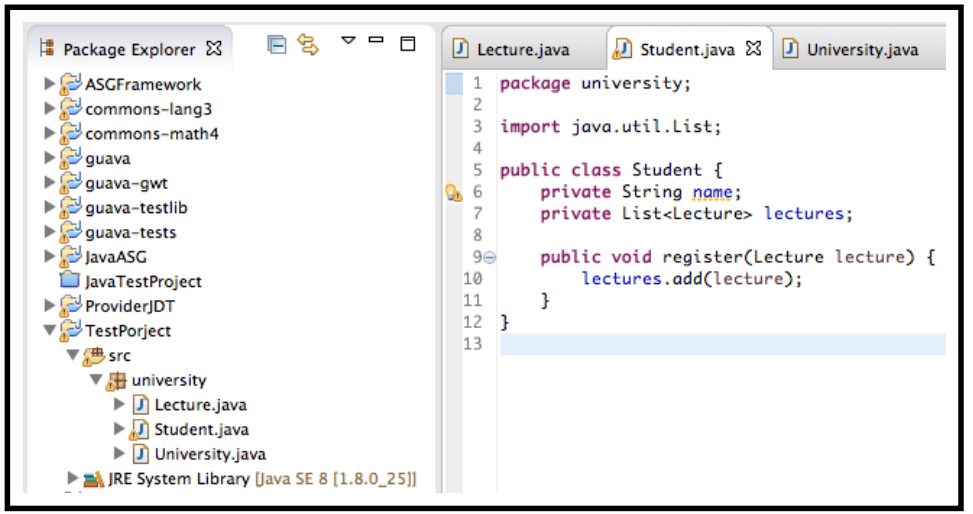


# Motivation

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

| ID | Author | Text |
|---|---|---|
| R1 | Andreas Winter | The student must be able to register for an event. |
| R2 | Johannes Meier | The student must be enroled at the university. |

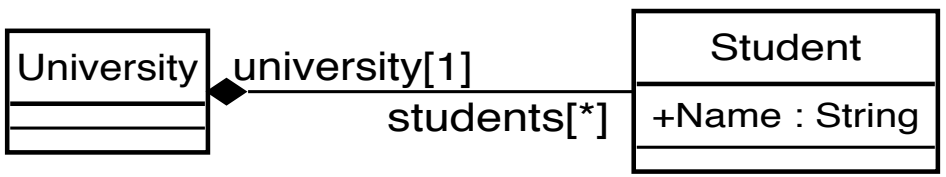


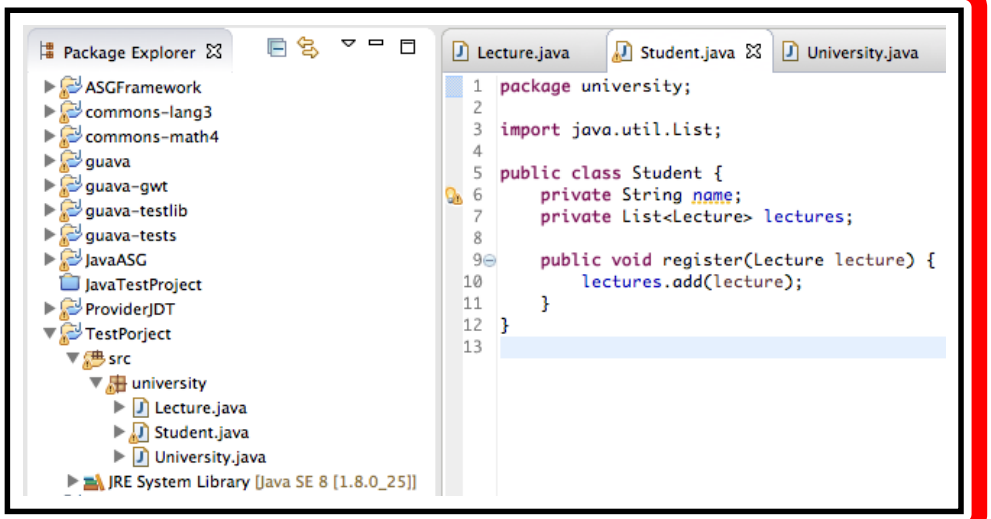



# Motivation

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

| ID | Author | Text |
|---|---|---|
| R1 | Andreas Winter | The student must be able to register for an event. |
| R2 | Johannes Meier | The student must be enroled at the university. |

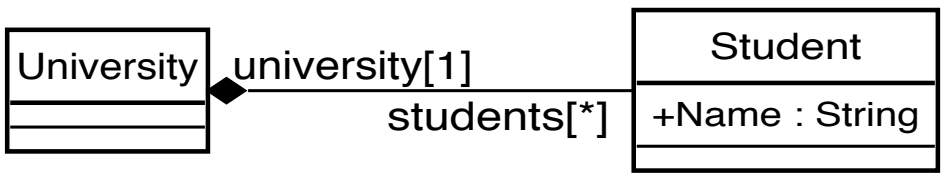


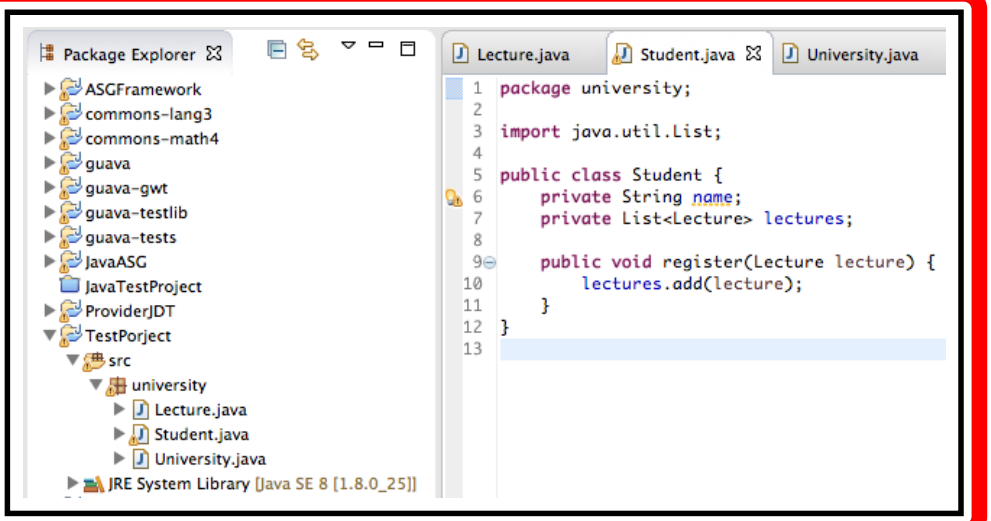



# Motivation

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

| ID | Author | Text |
|---|---|---|
| R1 | Andreas Winter | The student must be able to register for an event. |
| R2 | Johannes Meier | The student must be enroled at the university. |

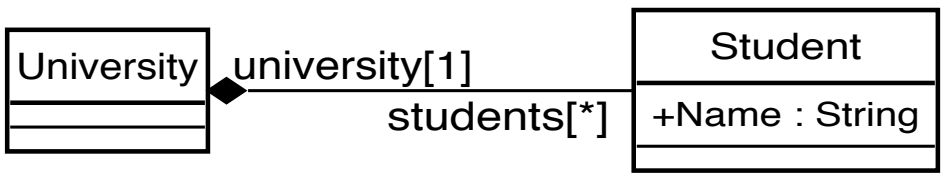


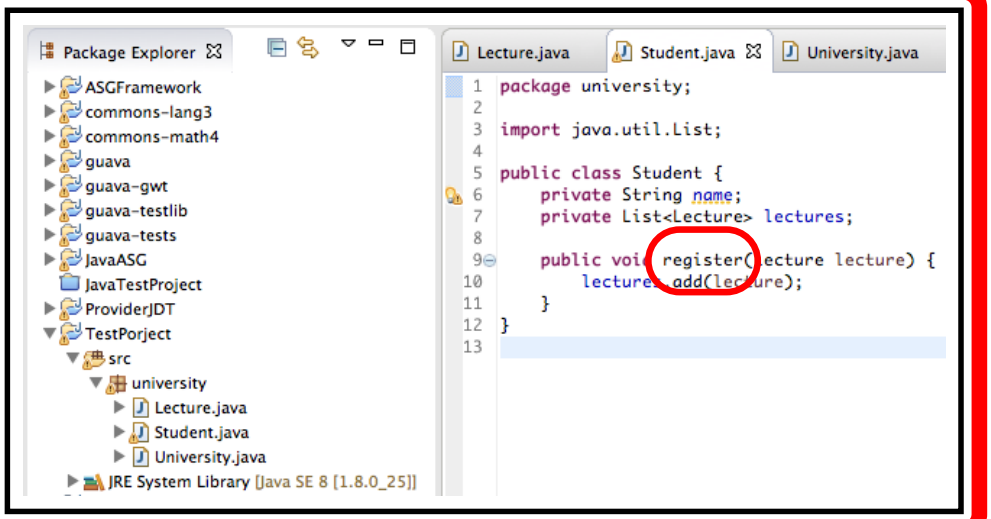



# Motivation

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

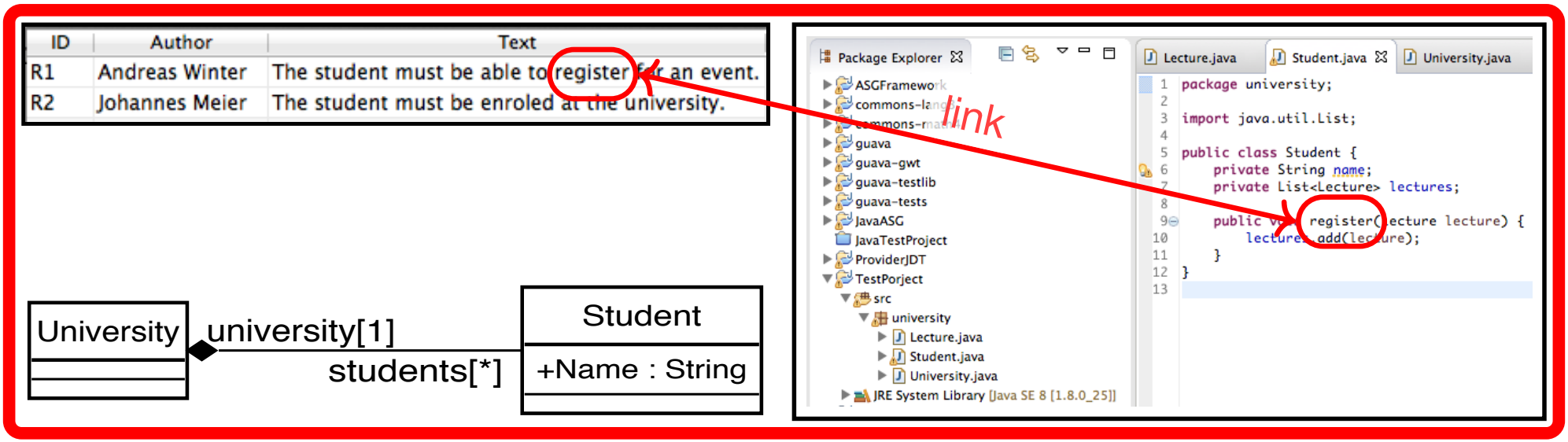
| ID | Author | Text |
|---|---|---|
| R1 | Andreas Winter | The student must be able to register for an event. |
| R2 | Johannes Meier | The student must be enroled at the university. |



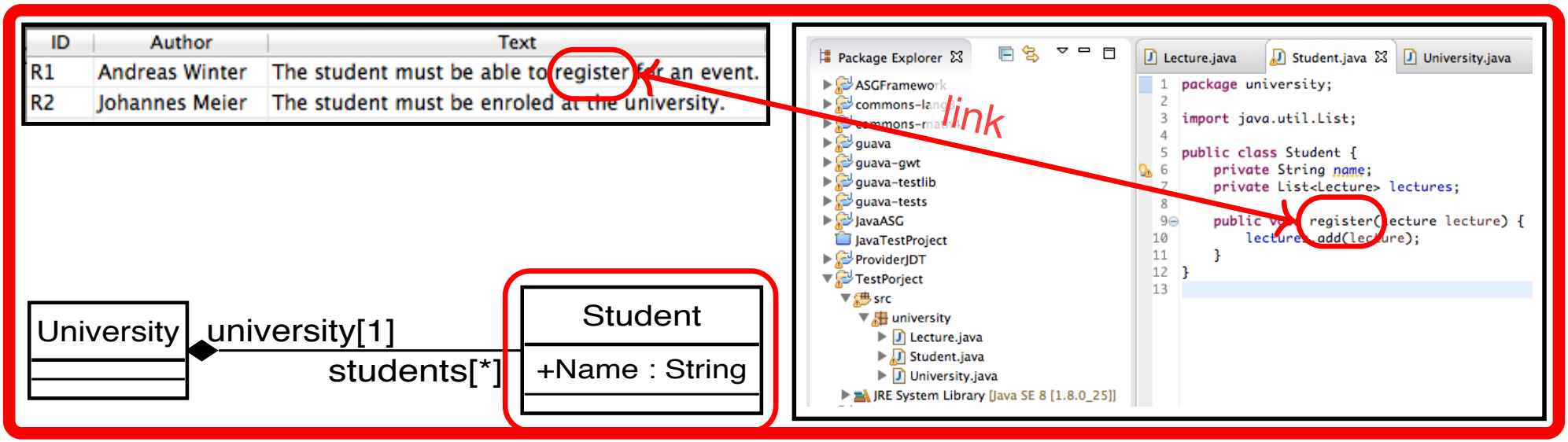
# Motivation

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



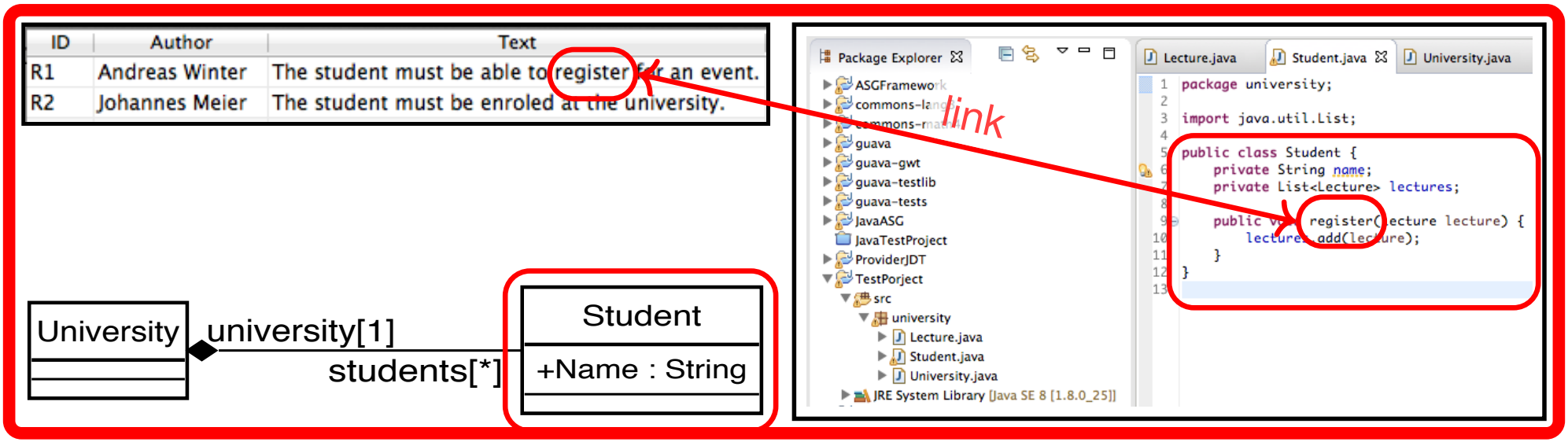
# Motivation

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



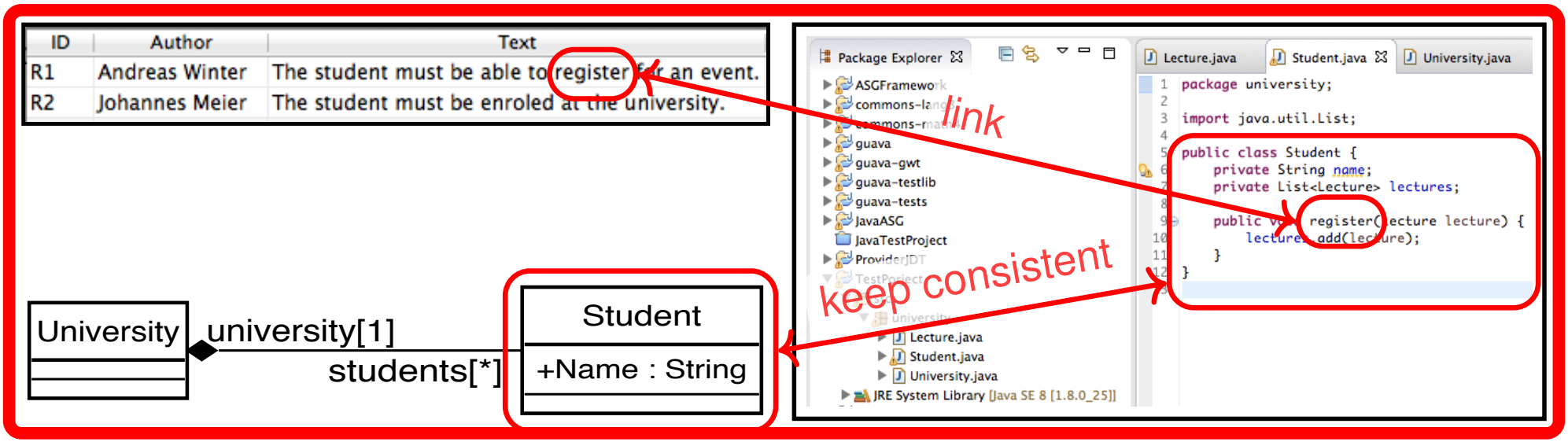
# Motivation

- 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
- $\rightarrow$ 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!

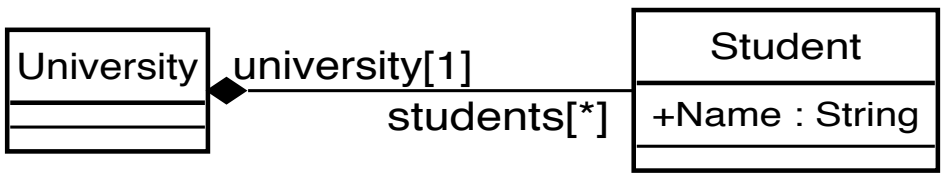
- Artifact == Model + Metamodel (structural formalization [CNS12])
- $\rightarrow$ Model Integration

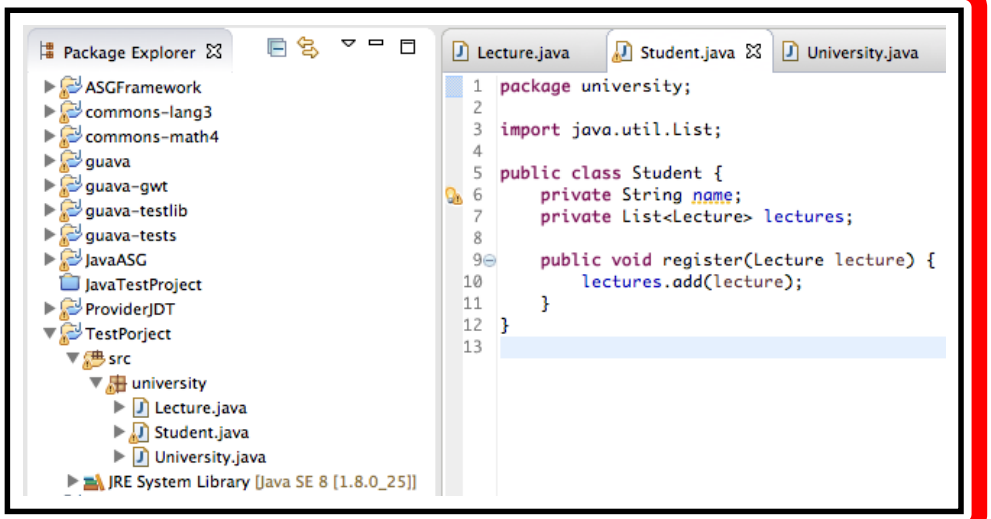
# Challenges

# Challenges

1. Formalize Consistency Rules

| ID | Author | Text |
|---|---|---|
| R1 | Andreas Winter | The student must be able to register for an event. |
| R2 | Johannes Meier | The student must be enroled at the university. |

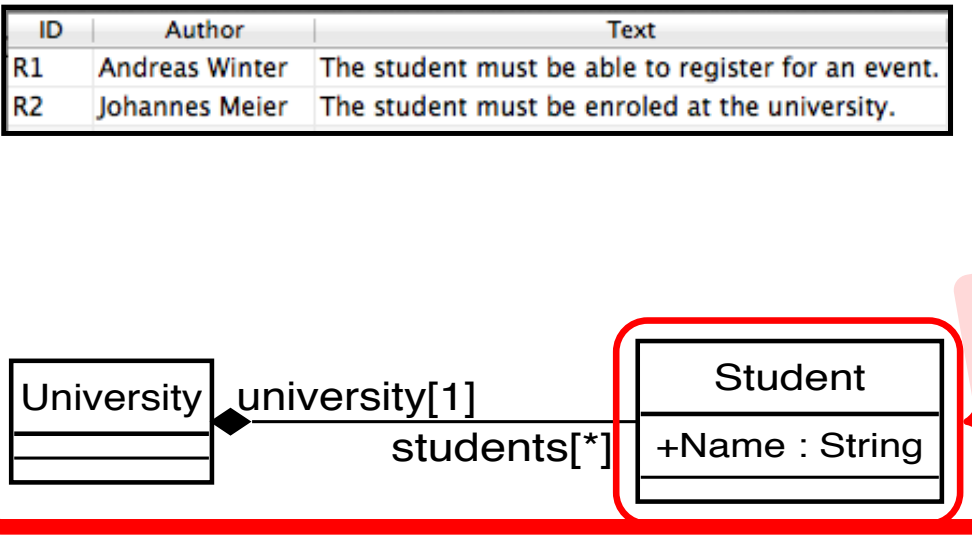


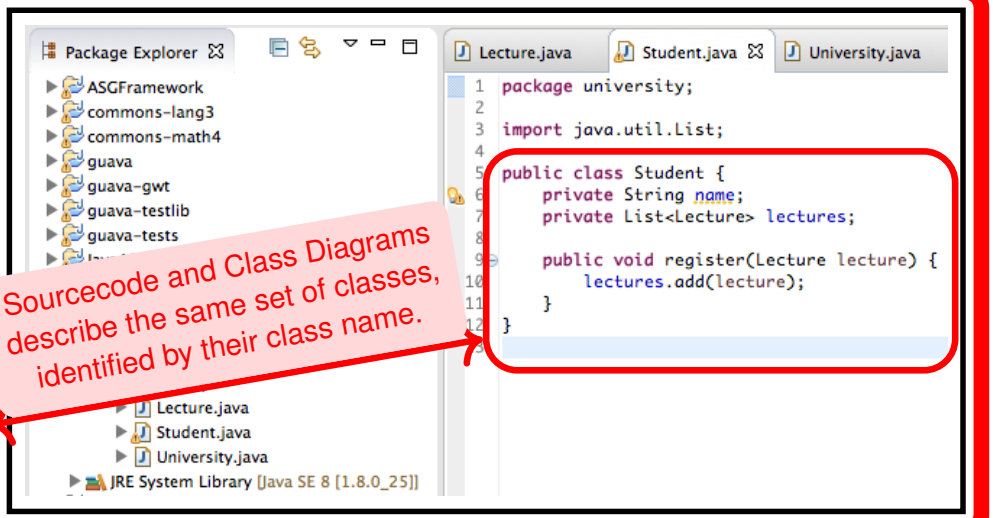



# Challenges

1. Formalize Consistency Rules

| ID | Author | Text |
|---|---|---|
| R1 | Andreas Winter | The student must be able to register for an event. |
| R2 | Johannes Meier | The student must be enroled at the university. |

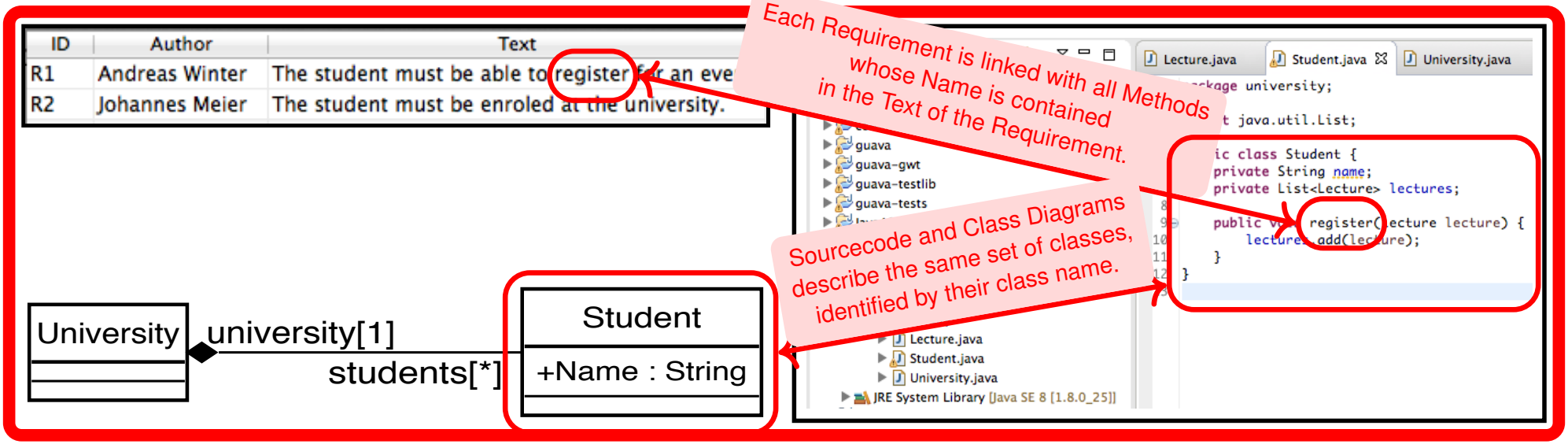



Sourcecode and Class Diagrams describe the same set of classes, identified by their class name.

# Challenges

1. Formalize Consistency Rules



# Challenges

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

# Challenges

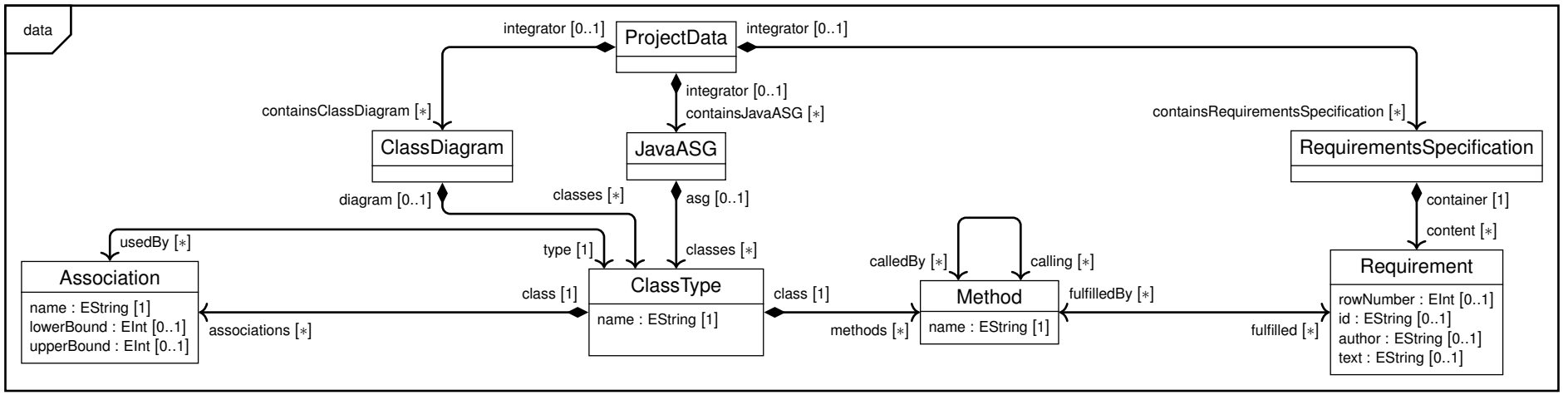
1. Formalize Consistency Rules
2. Create explicit SUM(M)

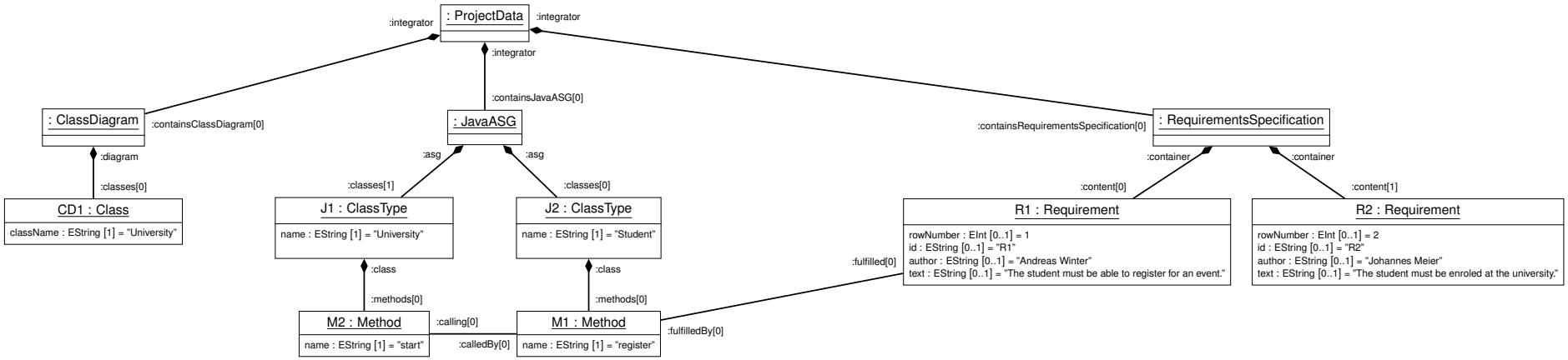
- reuse Model Techniques which work only with one Model
- used as single Point of Truth
- Single Underlying Model [ASB09]
- SUMM and SUM are explicit

# 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



# 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

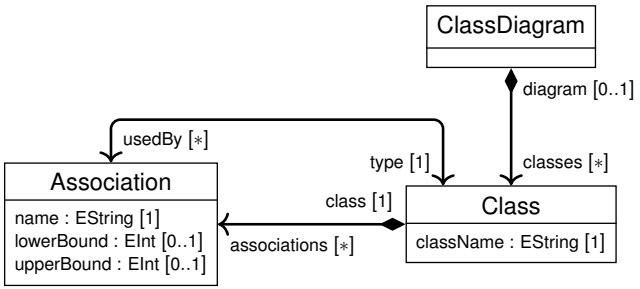
1. Formalize Consistency Rules
2. Create explicit SUM(M)
3. Support initial (Meta)Models:
    a. Reuse initial Models

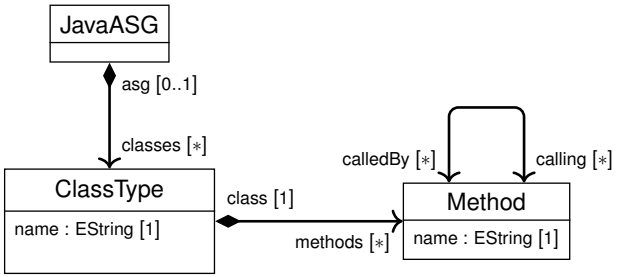
- existing Metamodels:
  DSLs, Environments, Tools, ...
- existing Models:
  ongoing projects, legacy data, ...

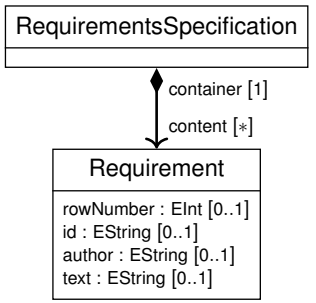
# Challenges

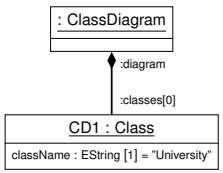
1. Formalize Consistency Rules
2. Create explicit SUM(M)
3. Support initial (Meta)Models:
   a. Reuse initial Models

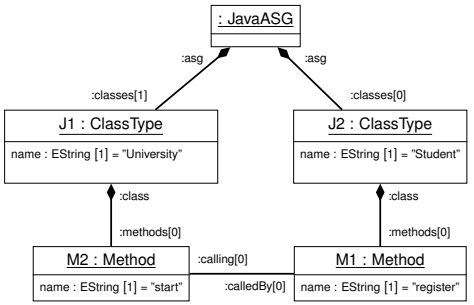
- existing Metamodels:
  DSLs, Environments, Tools, . . .
- existing Models:
  ongoing projects, legacy data, . . .

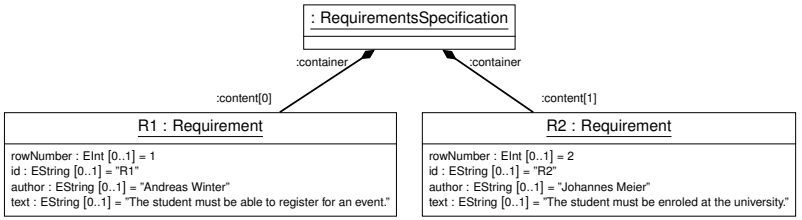


# Challenges

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 inititial 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 inititial Models
4. Ensure Model Consistency

# Related Work

| | synthetic | projectional | | |
|---|---|---|---|---|
| | | OSM | Vitruvius | GEMOC |
| 1. Formalize Consistency Rules | ✓ | ✓ | ✓ | ✓ |
| 2. Create explicit SUM(M) | ✗ | ✓ | ✗ | ✗ |
| 3a. Reuse initial Models | ✓ | ✗ | ✓ | ✓ |
| 3b. Fix initial Inconsistencies | ✓ | – | ✗ | ? |
| 3c. Consistent initial Models | ✓ | – | ✓ | ✓ |
| 4. Ensure Model Consistency | ✓ | ✓ | ✓ | ✓ |

- 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 inititial Models
4. Ensure Model Consistency

**Activities**

# Metamodel Integration

**Challenges**

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

**Activities**

1. Configuration of Operators

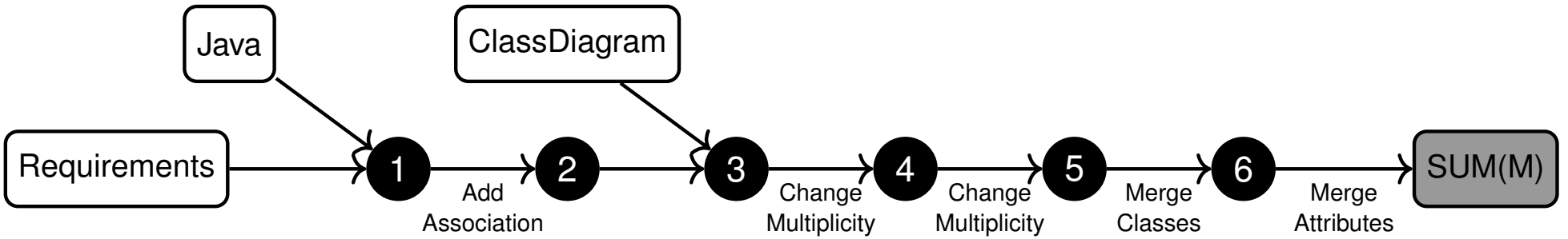
SUMM

3. Support initial (Meta)Models:
   a. Reuse initial Models
   b. Fix initial Inconsistencies
   c. Consistent inititial Models
4. Ensure Model Consistency

# Metamodel Integration

**Challenges** **Activities**

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



# Metamodel Integration

**Challenges**

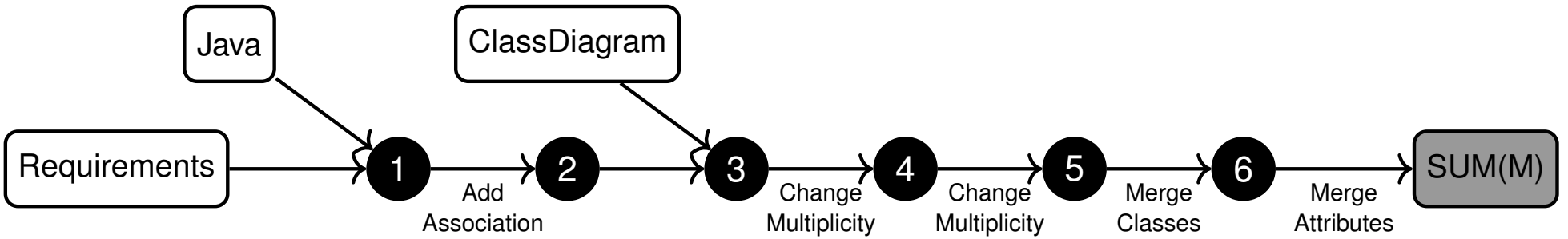
1. Formalize Consistency Rules
2. Create explicit SUM(M)
3. Support initial (Meta)Models:
   a. Reuse initial Models
   b. Fix initial Inconsistencies
   c. Consistent inititial Models
4. Ensure Model Consistency

**Activities**

1. Configuration of Operators
2. Initialization of SUM

SUMM

SUM



# Metamodel Integration

**Challenges**

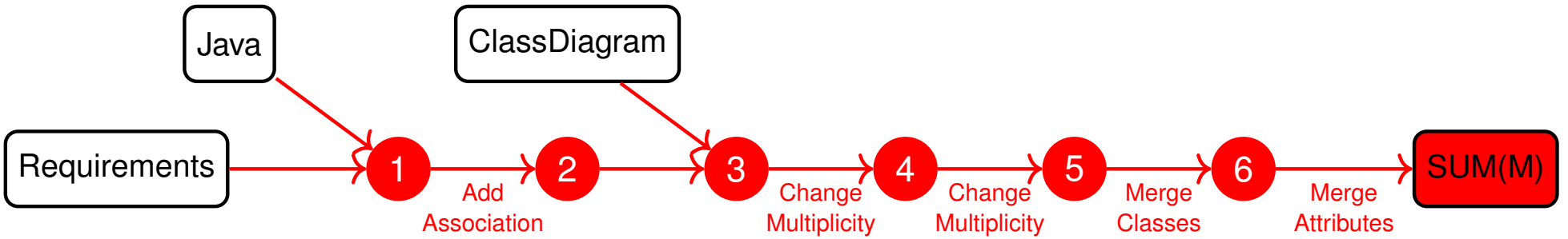
1. Formalize Consistency Rules
2. Create explicit SUM(M)
3. Support initial (Meta)Models:
   a. Reuse initial Models
   b. Fix initial Inconsistencies
   c. Consistent inititial Models
4. Ensure Model Consistency

**Activities**

1. Configuration of Operators
2. Initialization of SUM

SUMM

SUM



# Metamodel Integration

**Challenges**

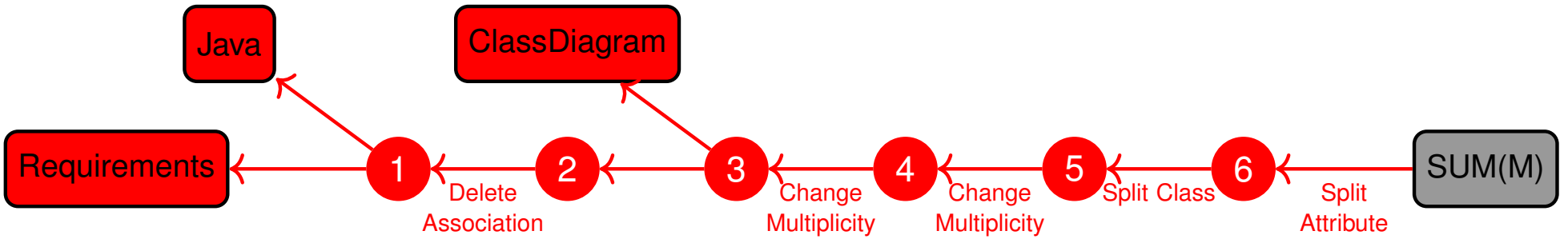
1. Formalize Consistency Rules
2. Create explicit SUM(M)
3. Support initial (Meta)Models:
   a. Reuse initial Models
   b. Fix initial Inconsistencies
   c. Consistent inititial Models
4. Ensure Model Consistency

**Activities**

1. Configuration of Operators
2. Initialization of SUM

SUMM

SUM



# Metamodel Integration

**Challenges**

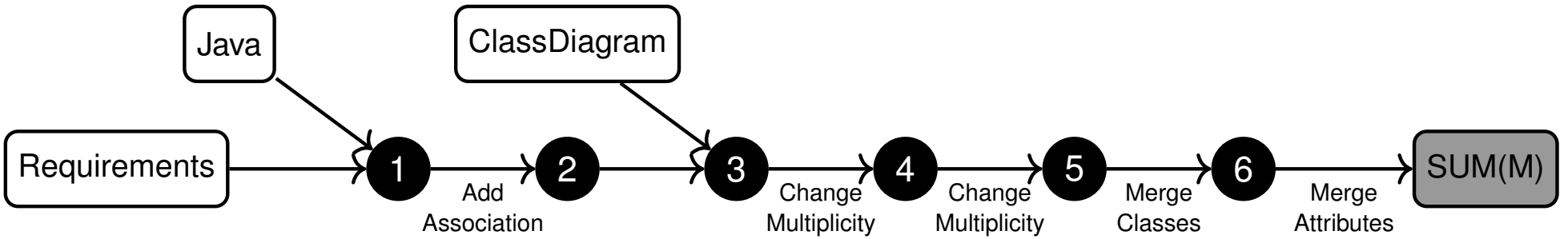
1. Formalize Consistency Rules
2. Create explicit SUM(M)
3. Support initial (Meta)Models:
   a. Reuse initial Models
   b. Fix initial Inconsistencies
   c. Consistent inititial Models
4. Ensure Model Consistency

**Activities**

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

SUMM

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 inititial Models
4. Ensure Model Consistency

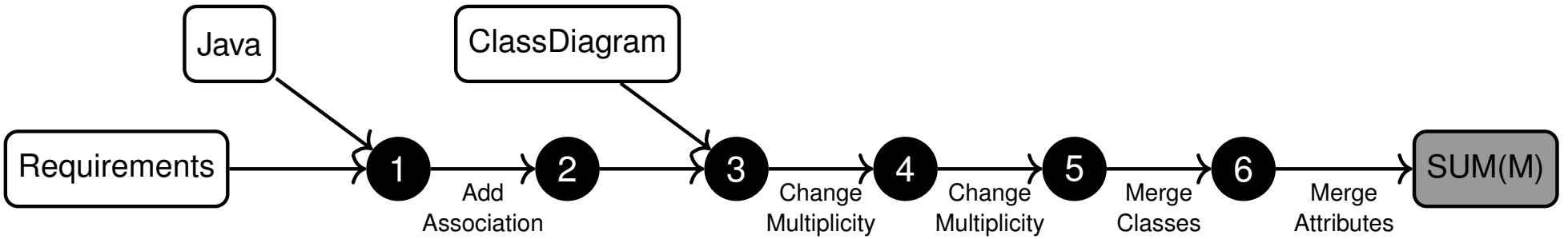
**Activities**

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

SUMM

SUM

1x Methodologist



# 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 inititial Models
4. Ensure Model Consistency

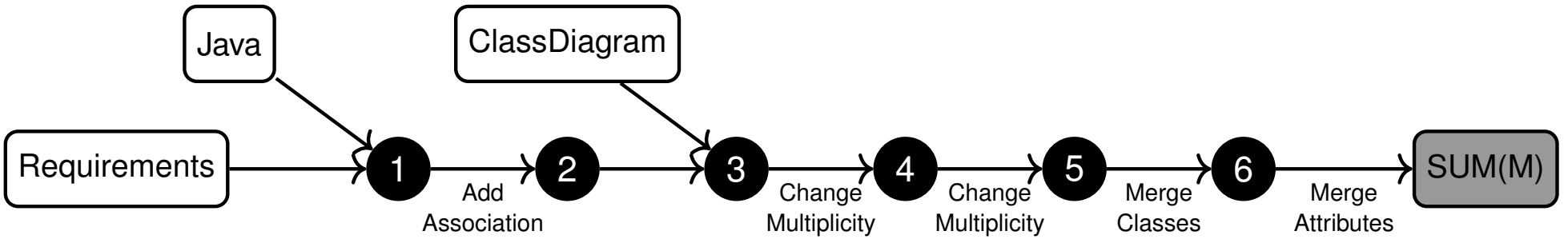
**Activities**

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

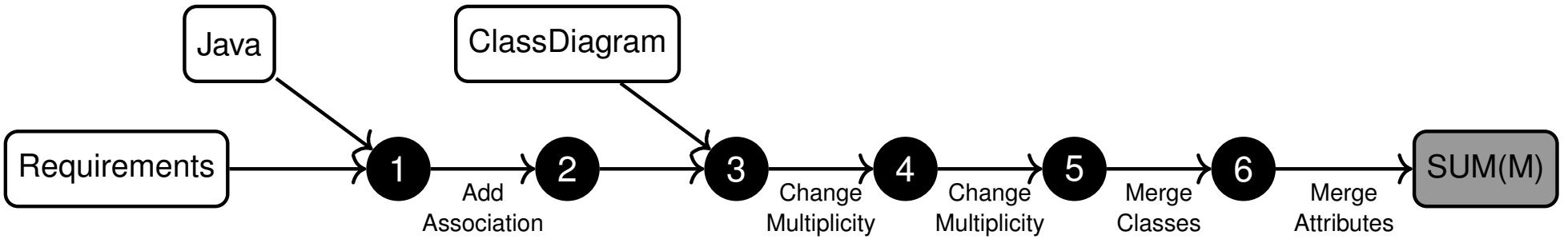
SUMM

SUM

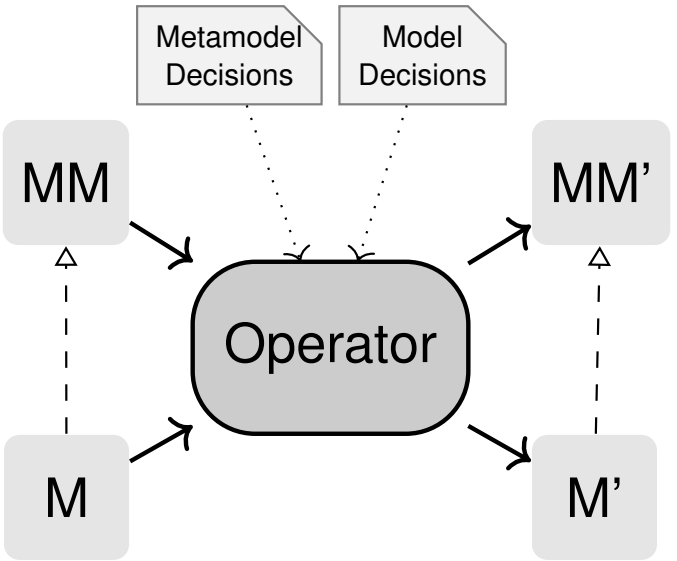
*n x* User

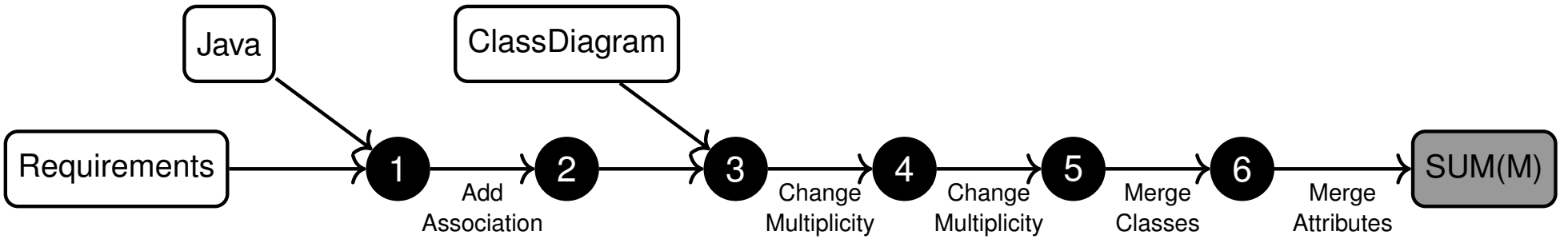


# 1. Configuration of Operators

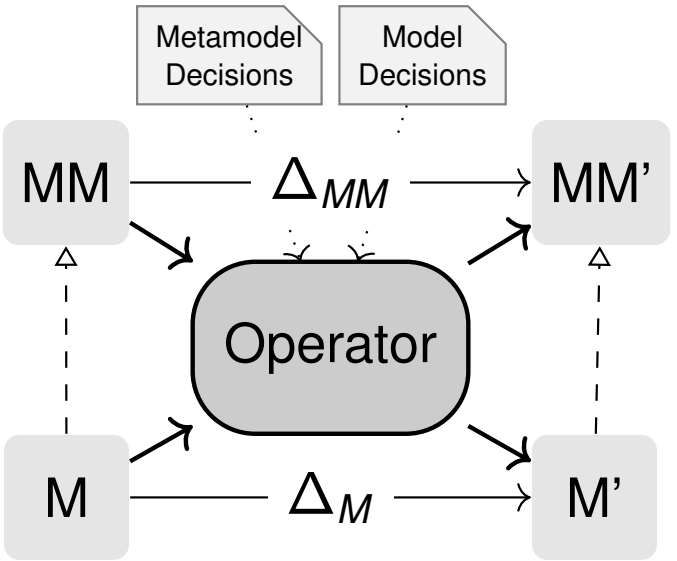


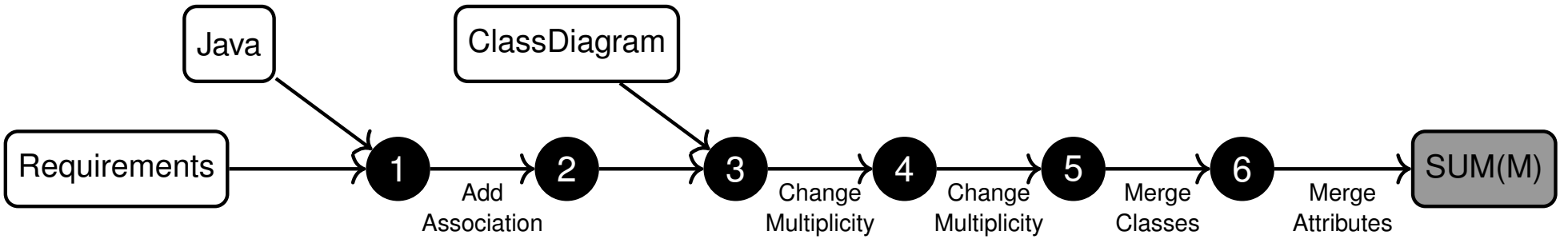
# 1. Configuration of Operators



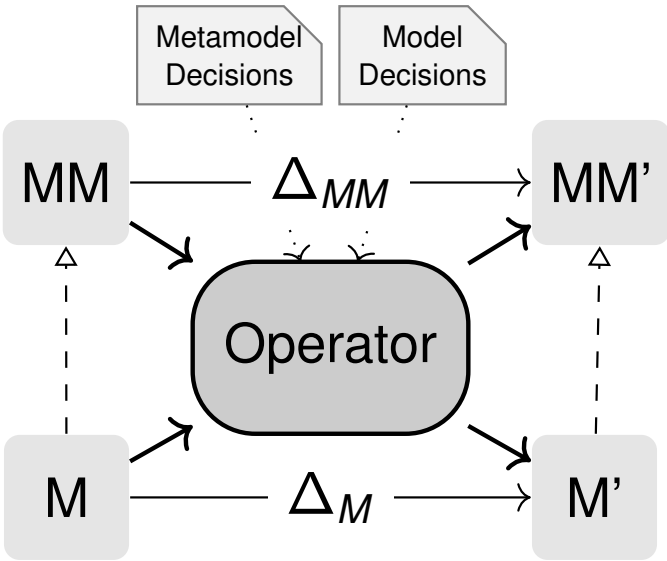



# 1. Configuration of Operators



# 1. Configuration of Operators

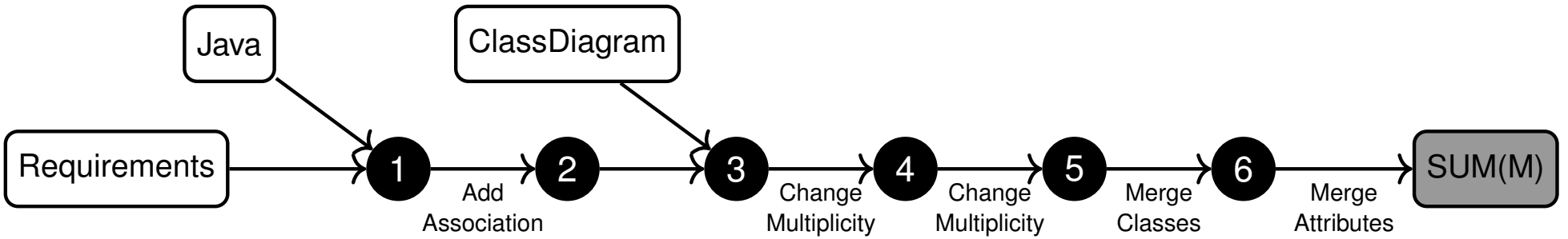



Chechik, Nejati, Sabetzadeh:
**A Relationship-Based Approach to Model Integration** (2012)

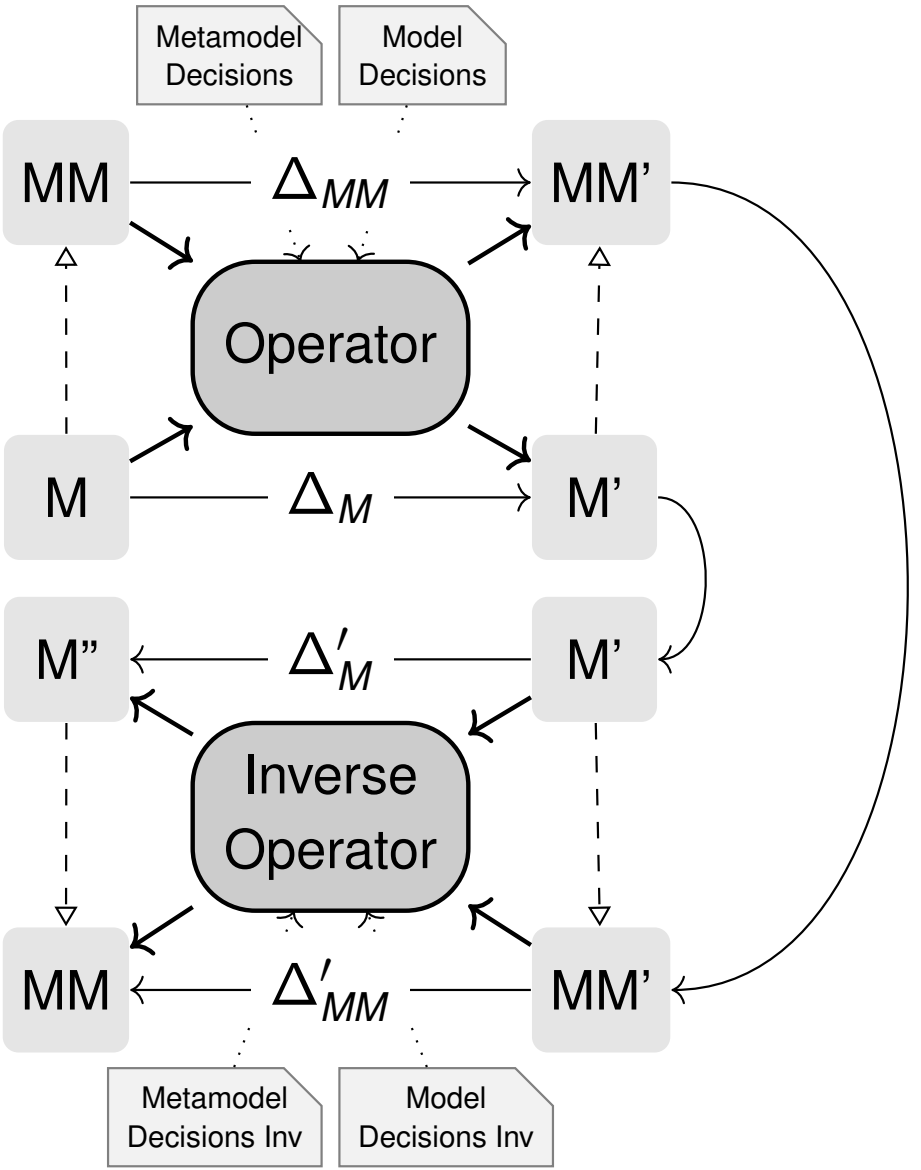
→ merge, composition, weaving

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

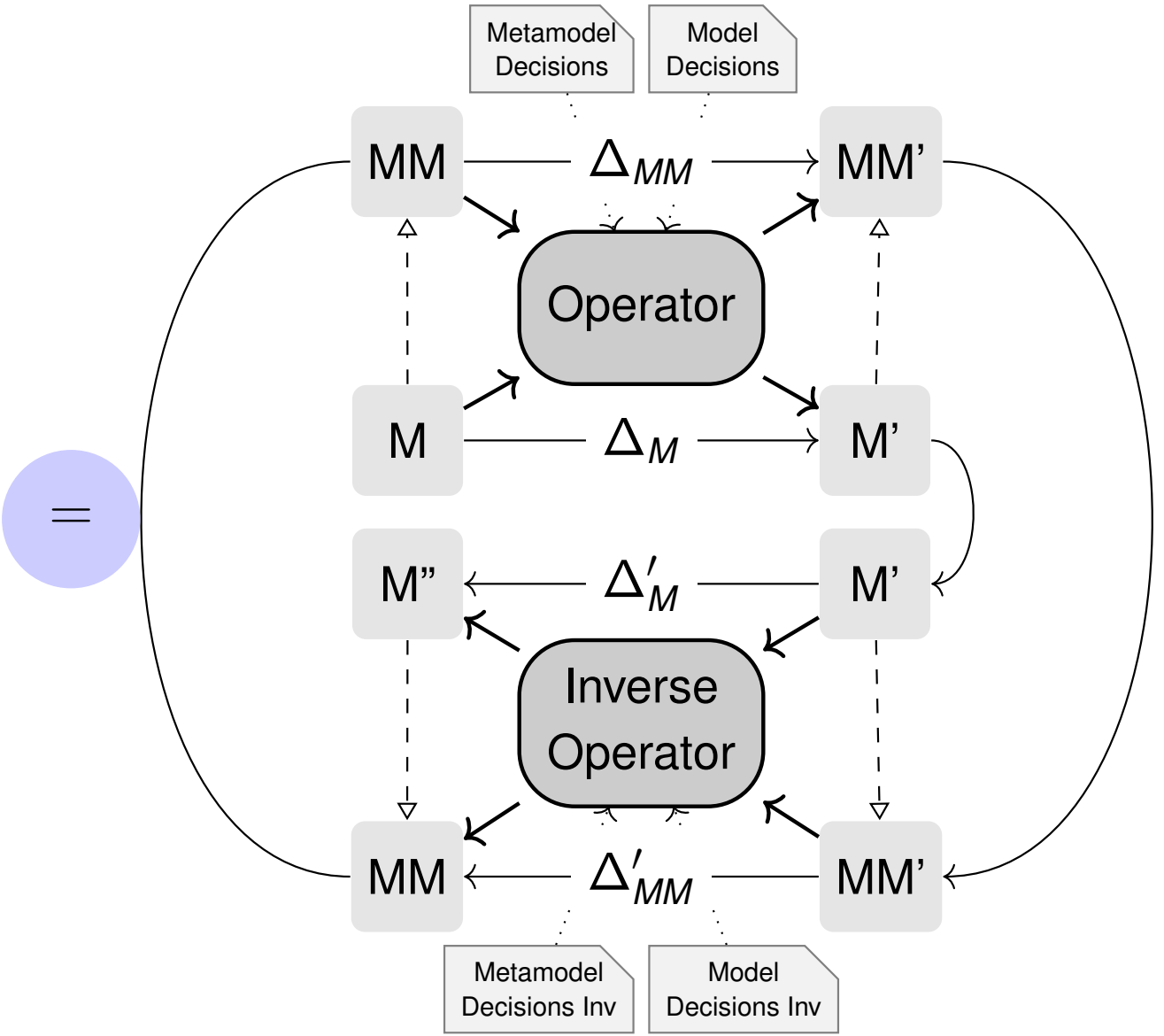
→ extended Coupled Operators



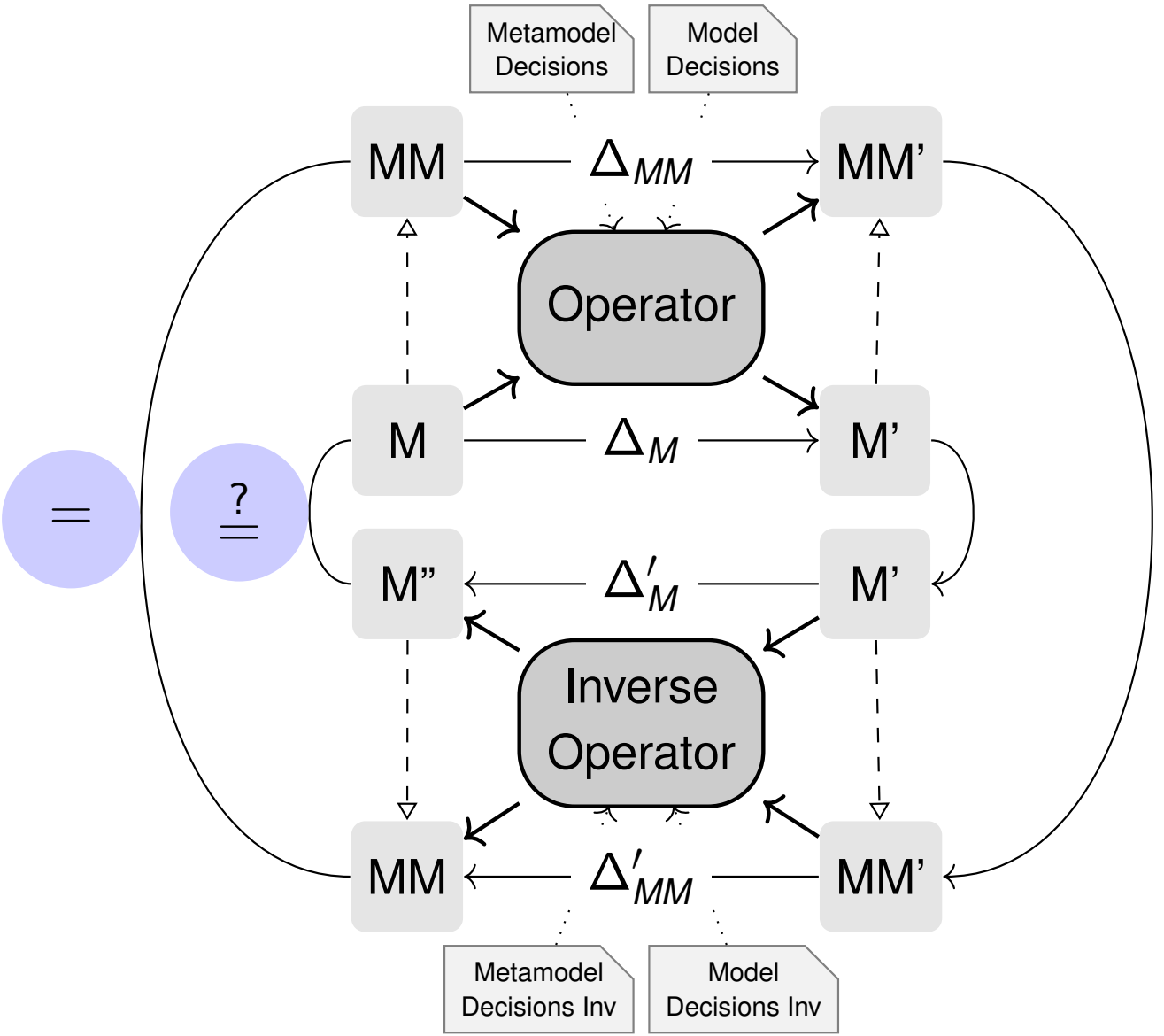
# 1. Configuration of Operators



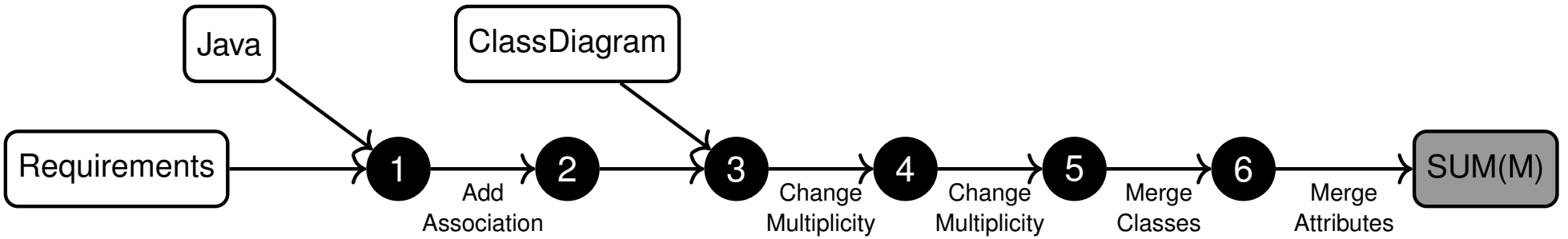
# 1. Configuration of Operators



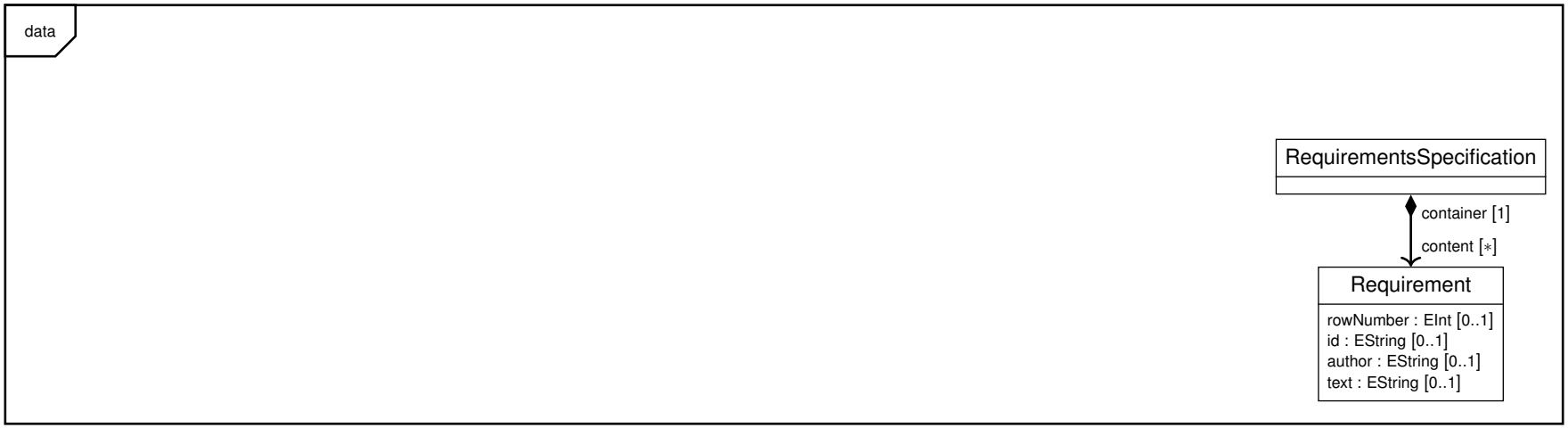
# 1. Configuration of Operators

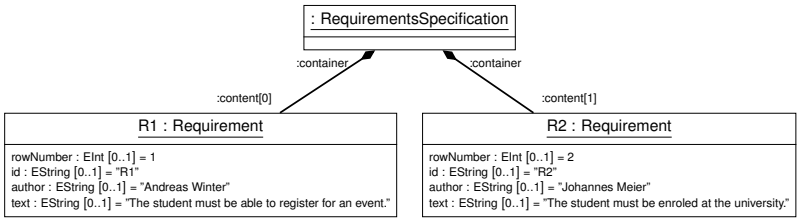


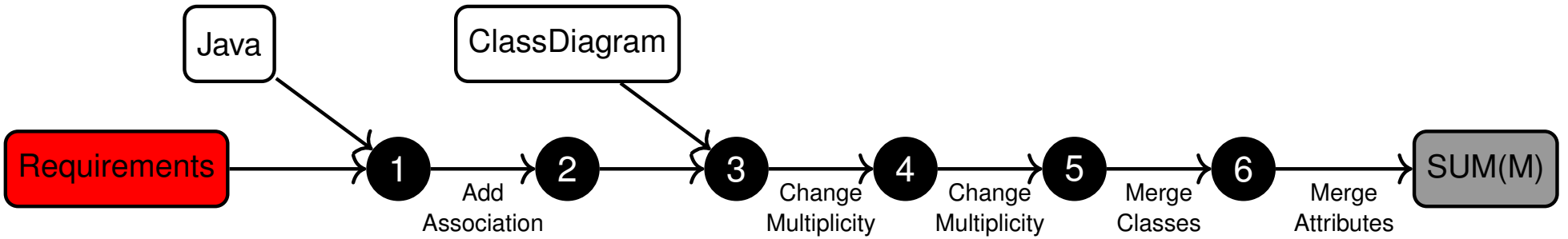
# 2. Initialization of SUM: Overview



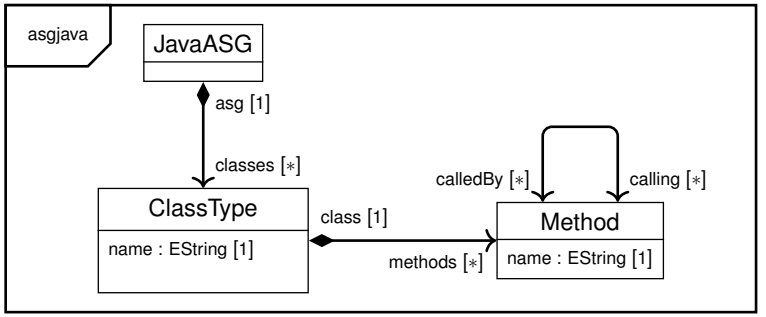
# 2. Initialization of SUM: Overview

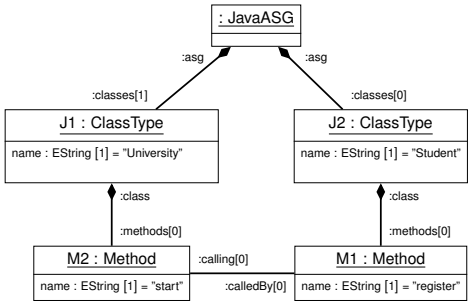


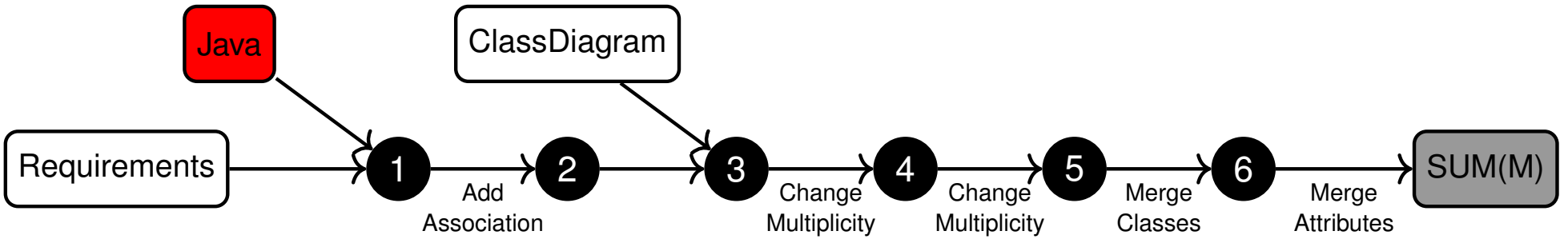



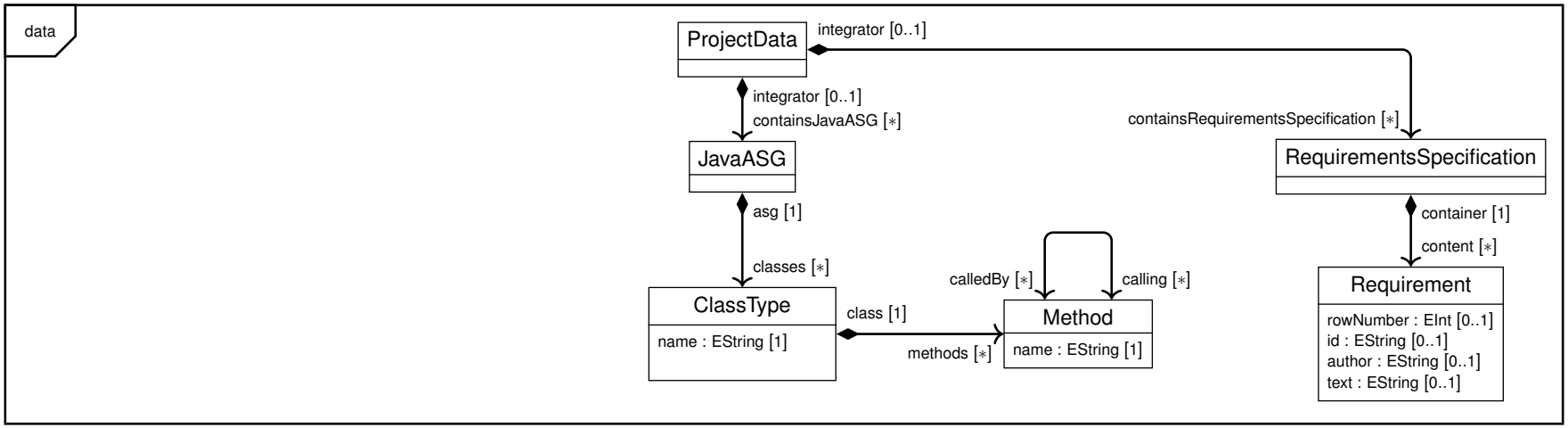
# 2. Initialization of SUM: Overview

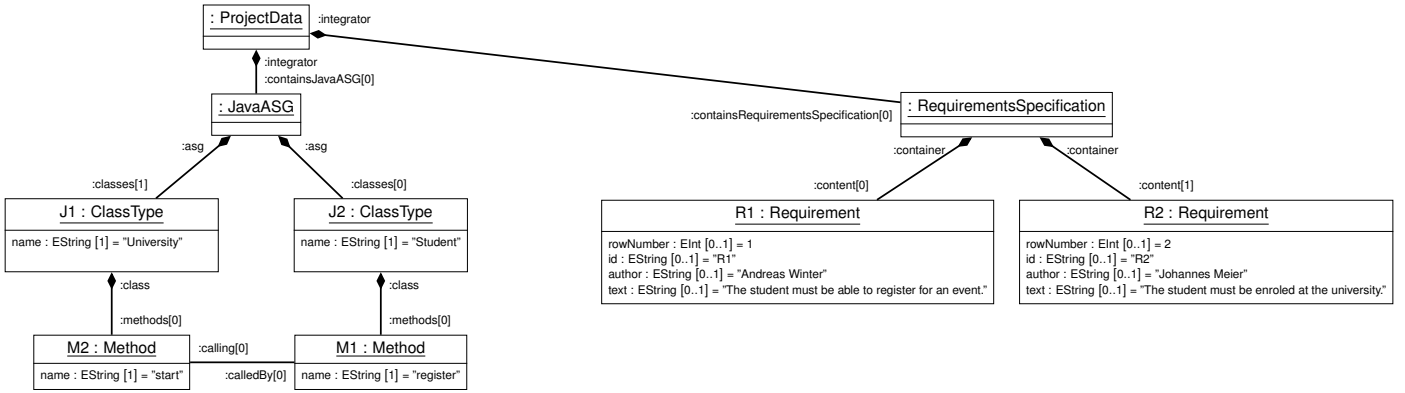


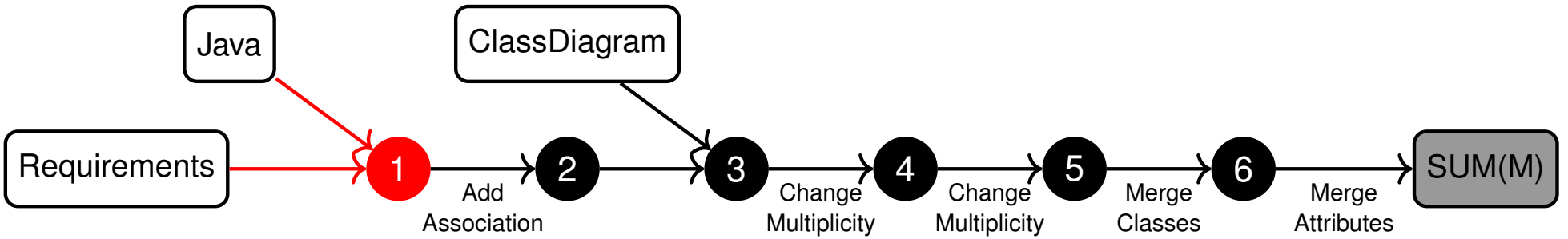



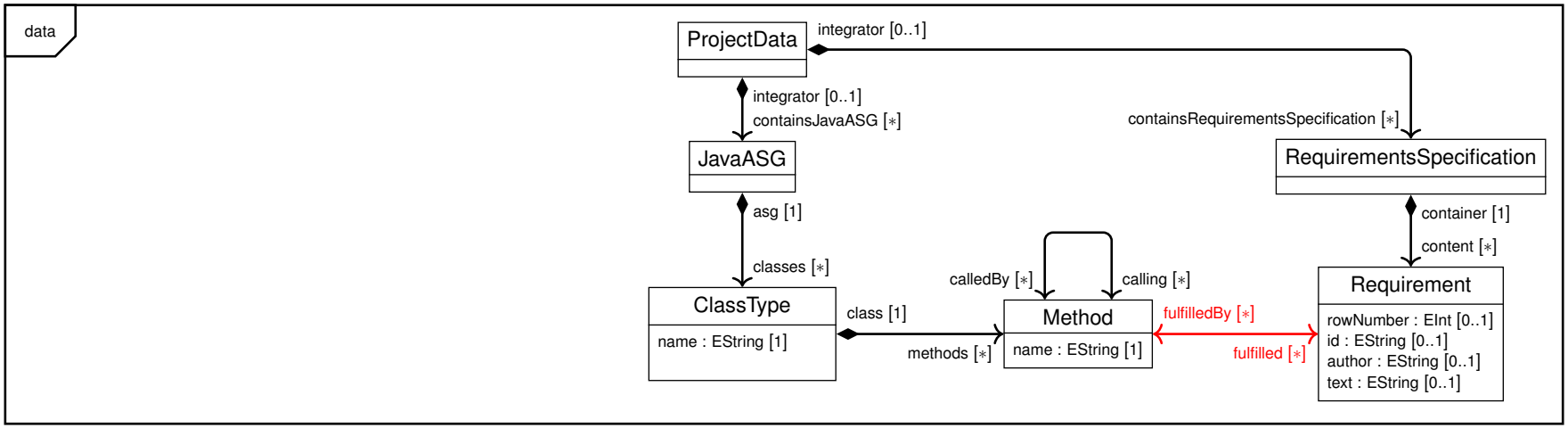
# 2. Initialization of SUM: Overview

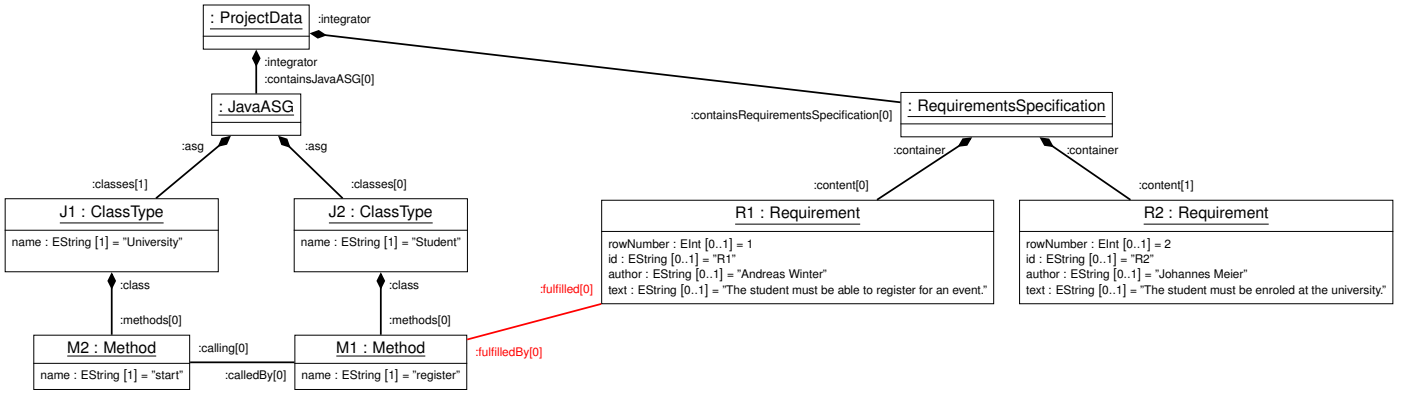


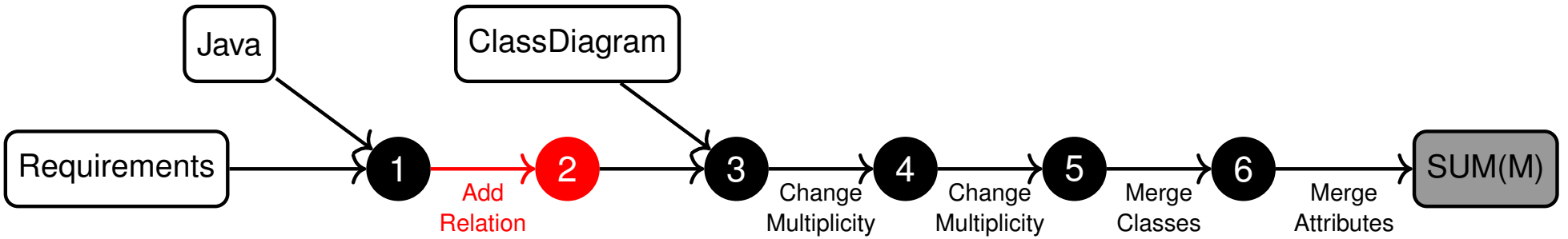



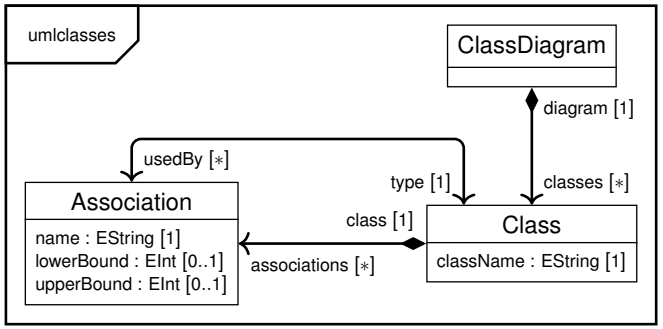
# 2. Initialization of SUM: Overview

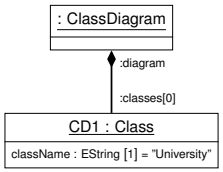


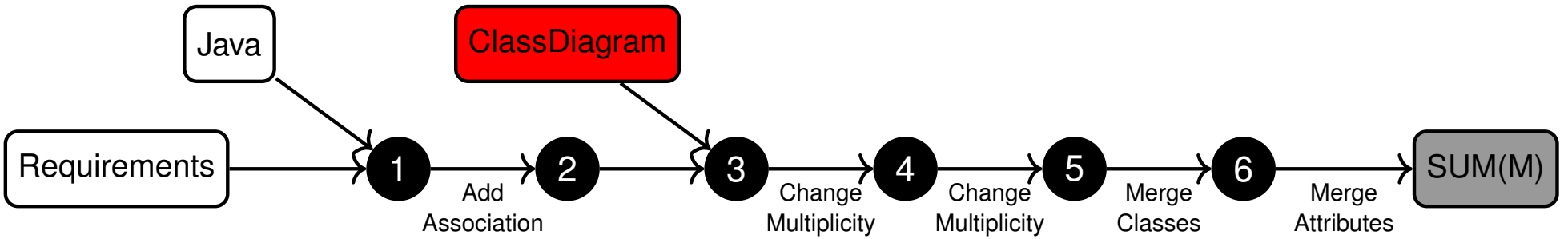



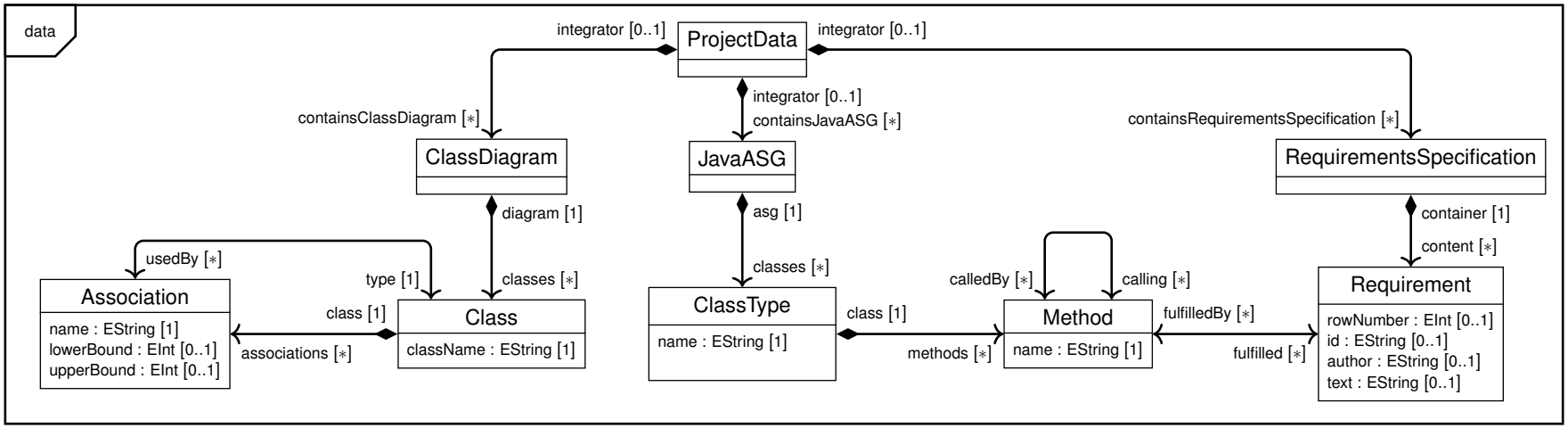
# 2. Initialization of SUM: Overview

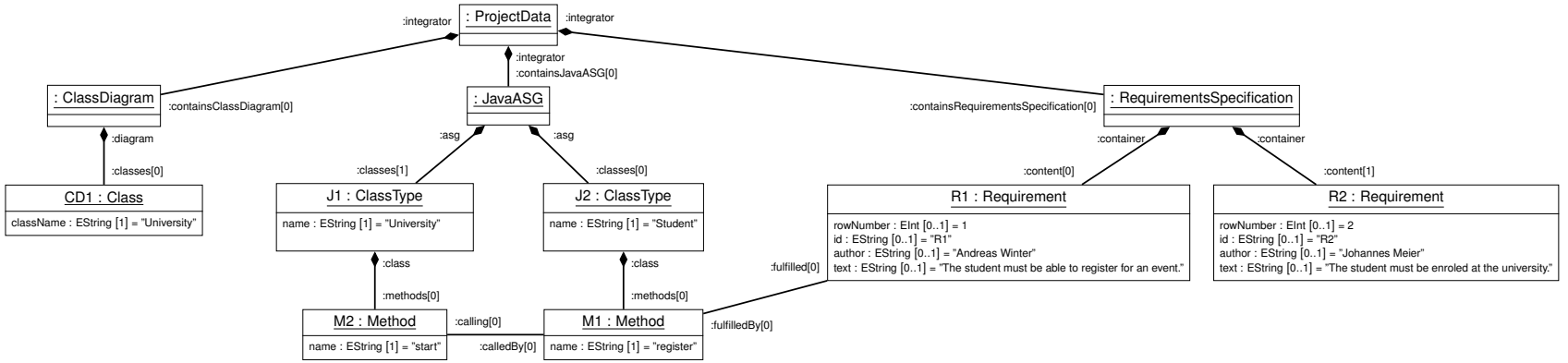


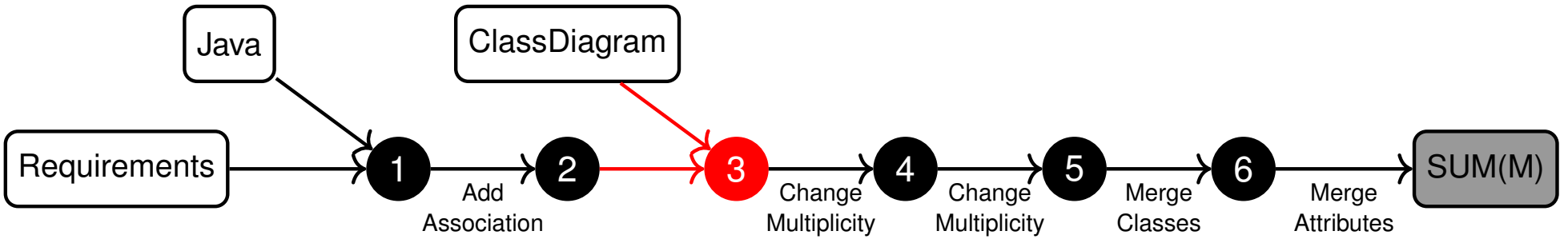



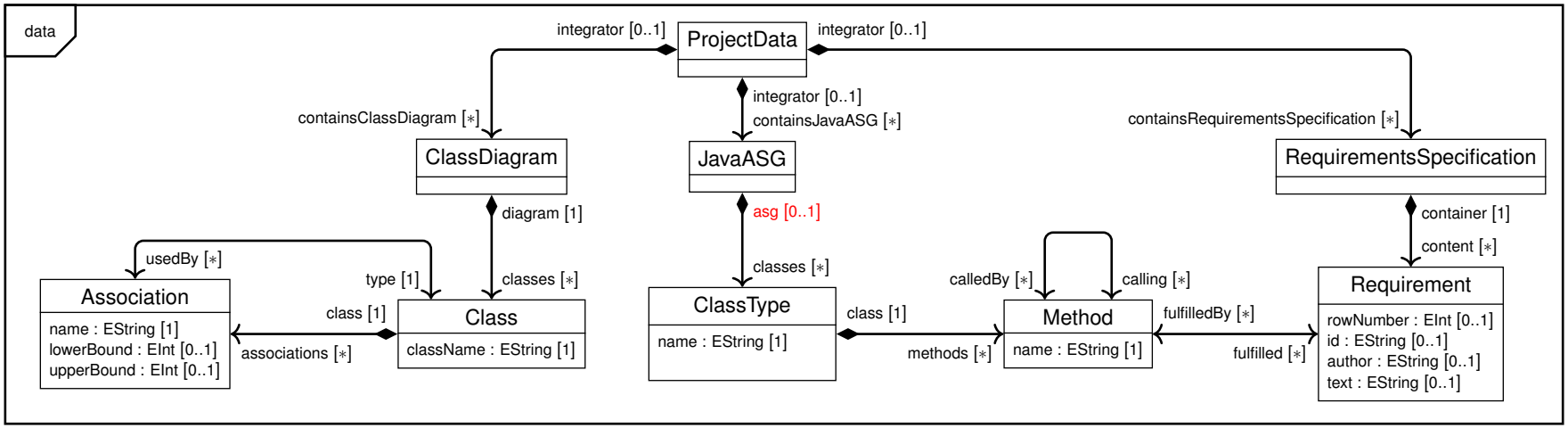
# 2. Initialization of SUM: Overview

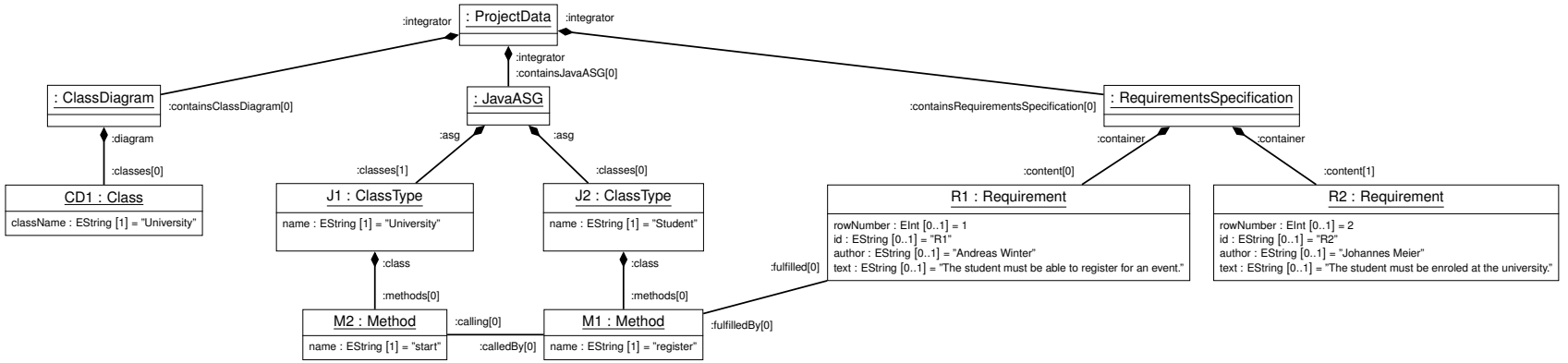


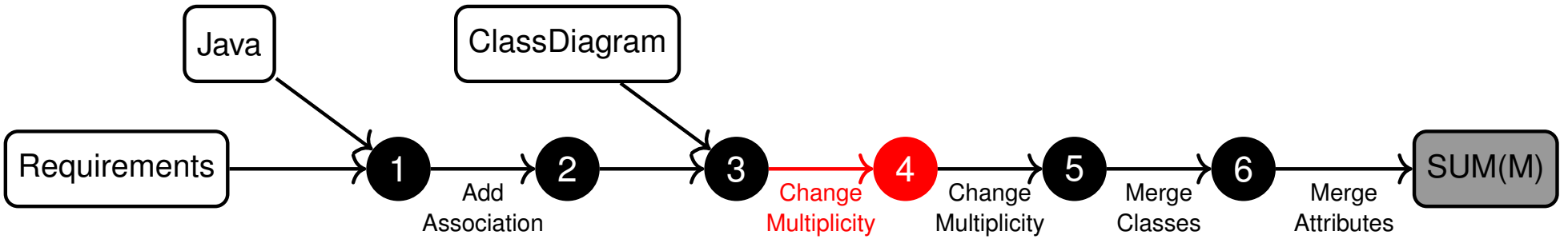



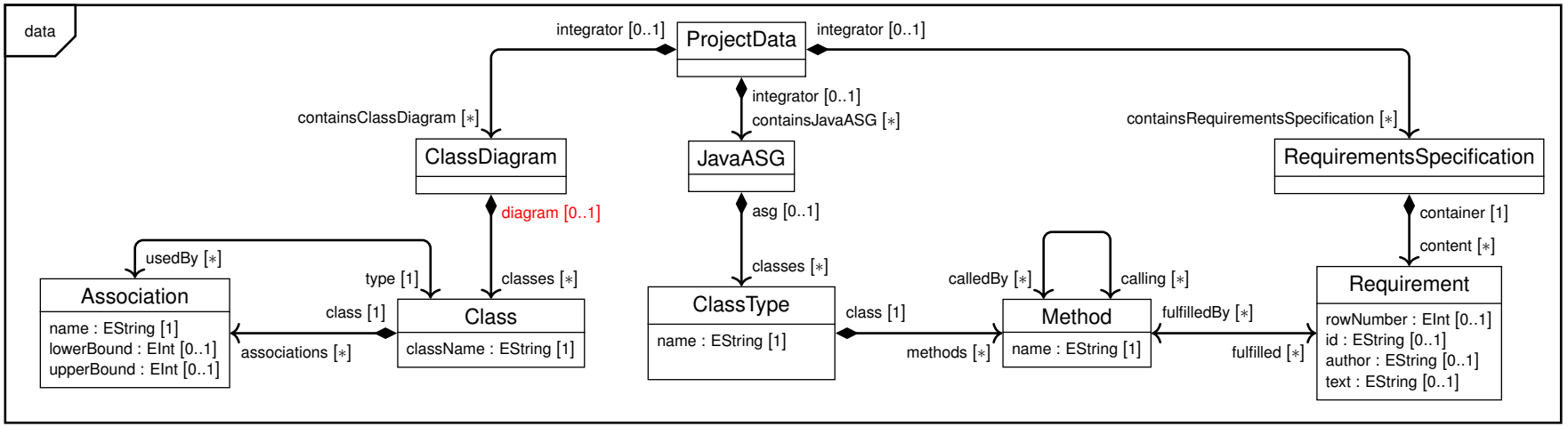
# 2. Initialization of SUM: Overview

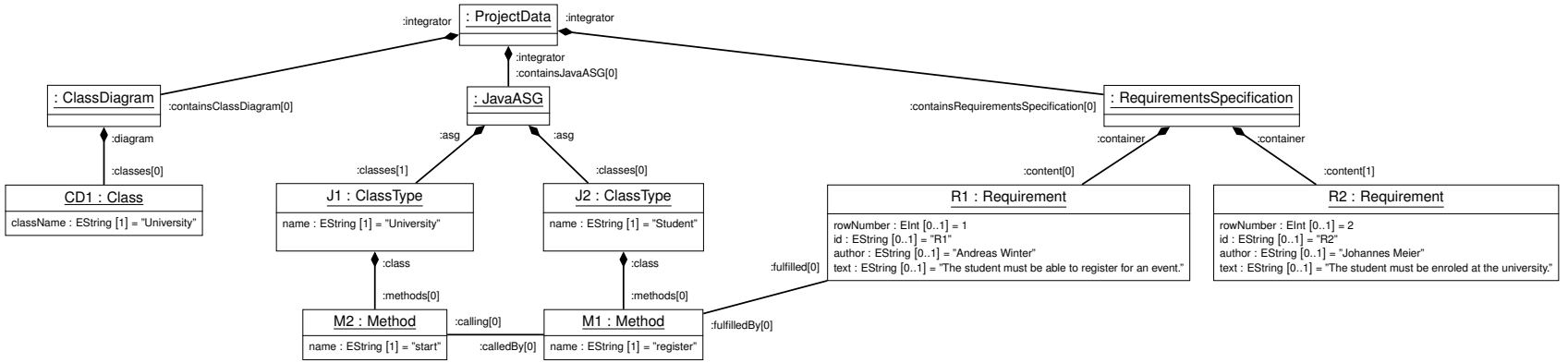


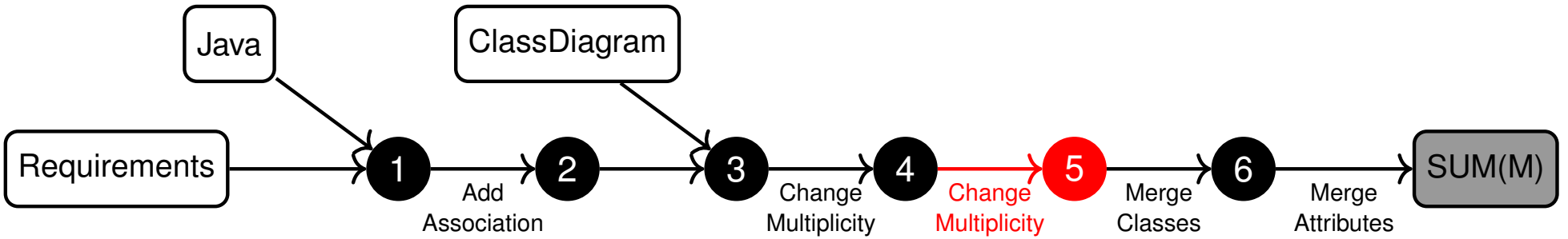



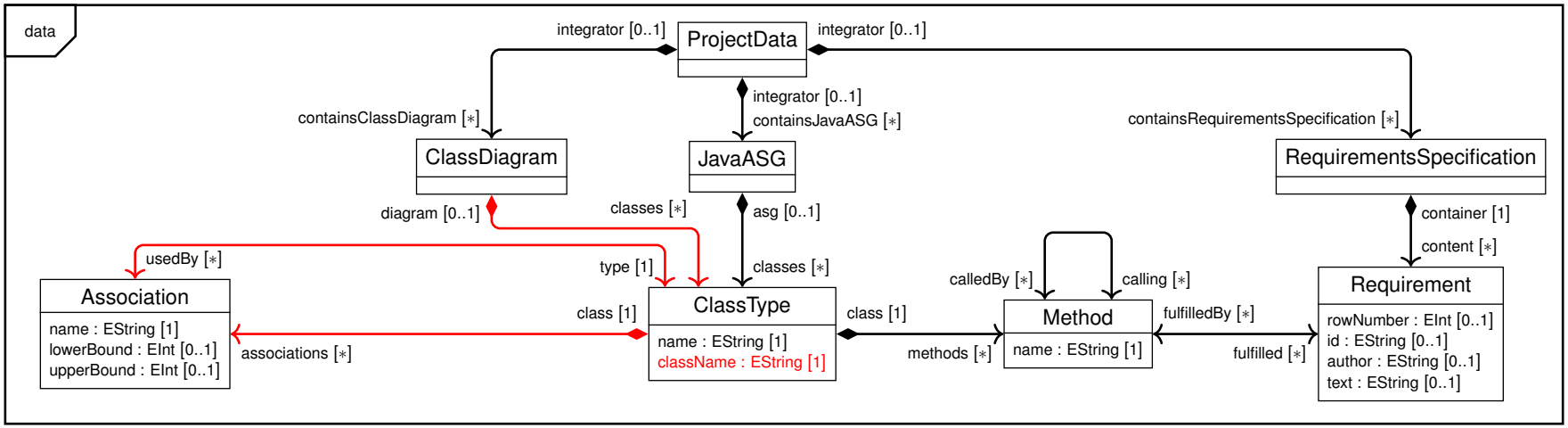
# 2. Initialization of SUM: Overview

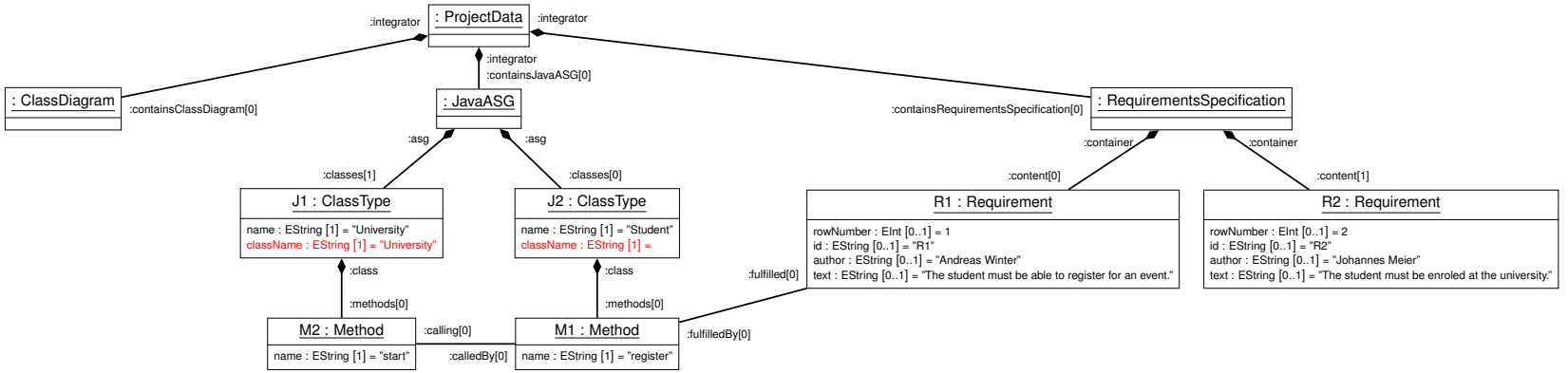


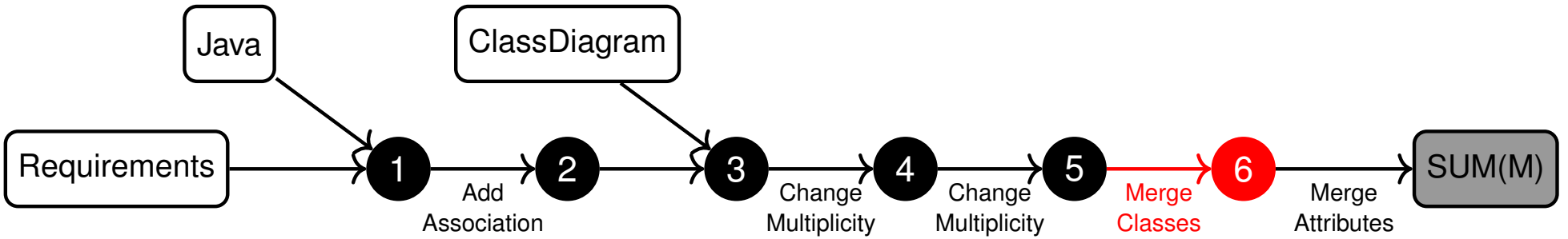



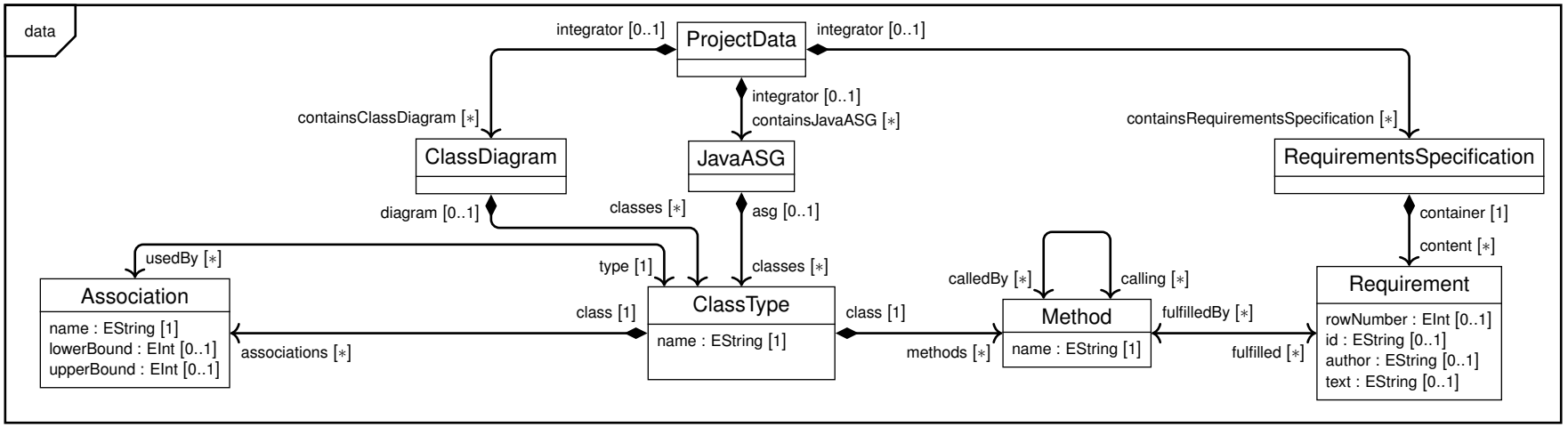
# 2. Initialization of SUM: Overview

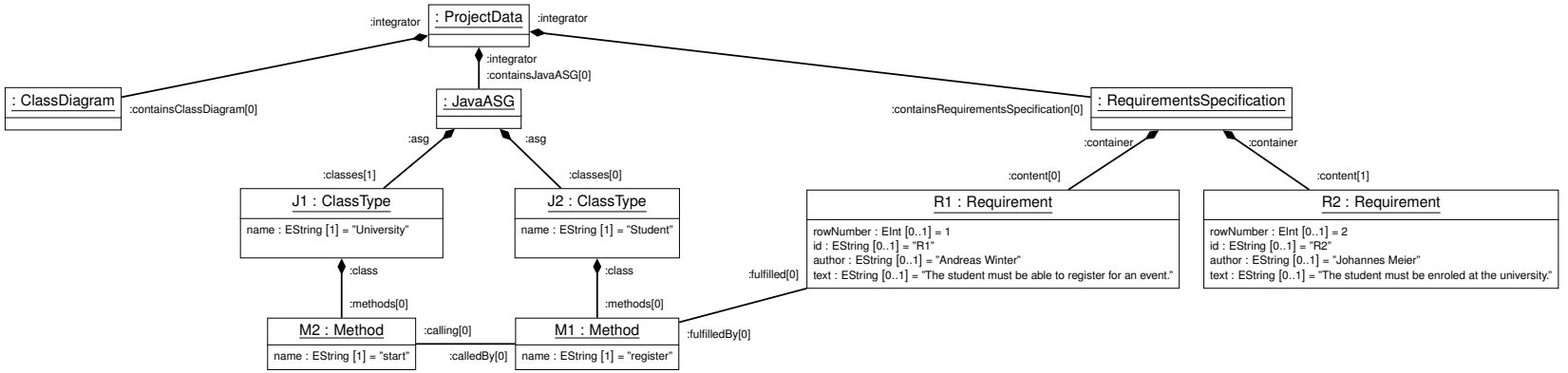


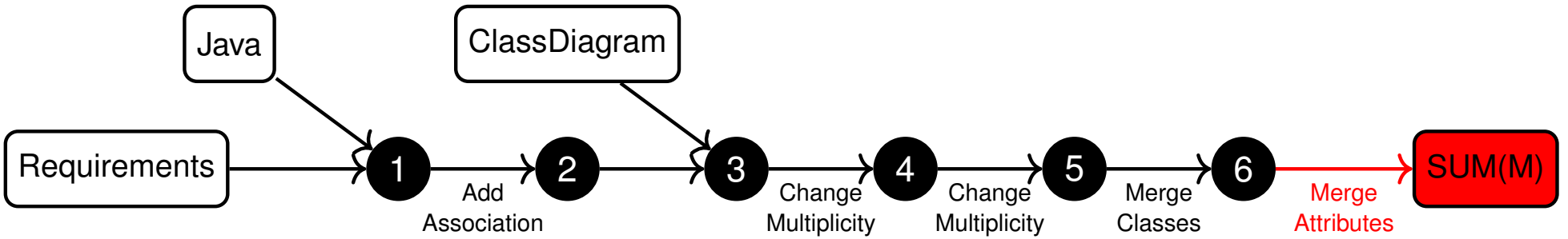



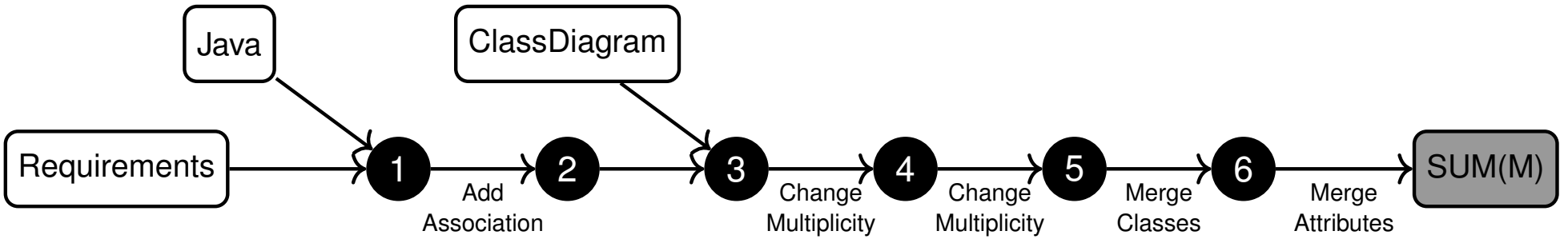
# 2. Initialization of SUM: Overview



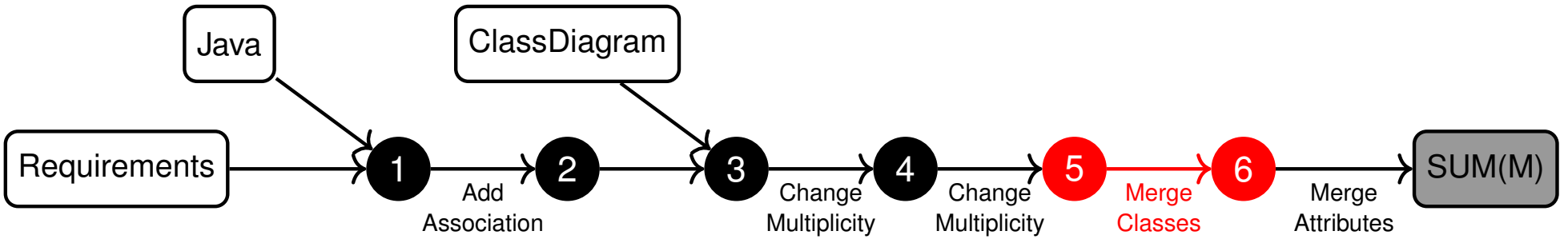



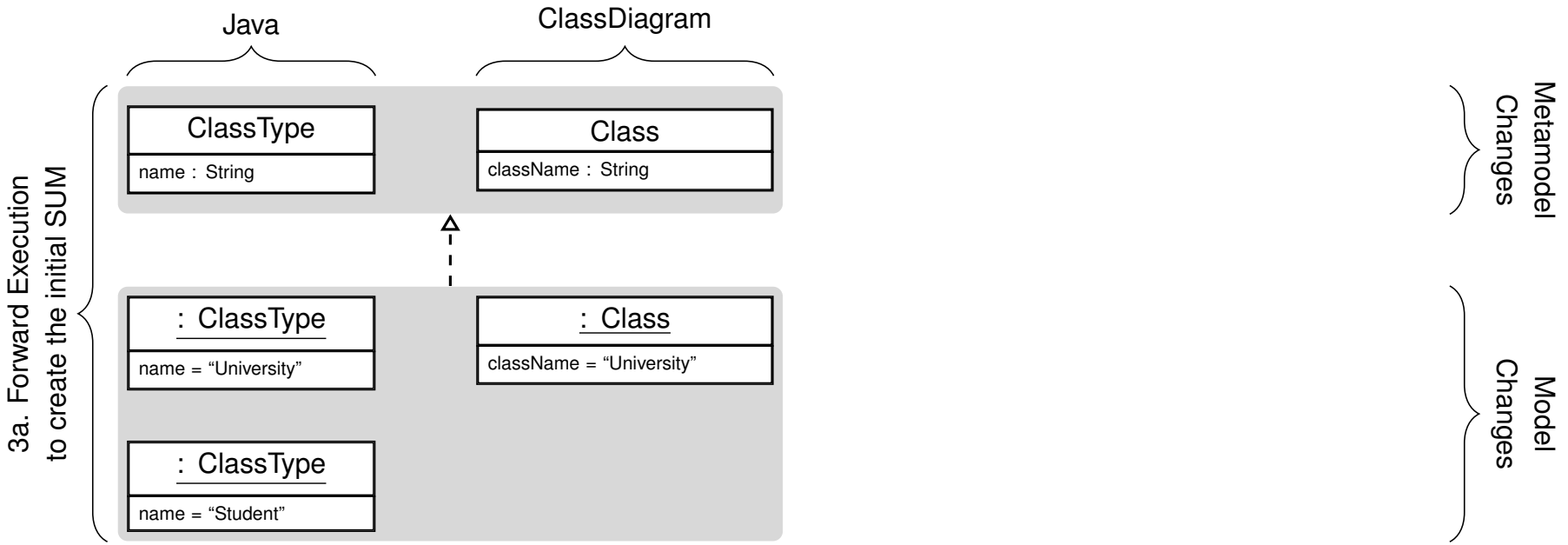
# 2. Initialization of SUM: Details

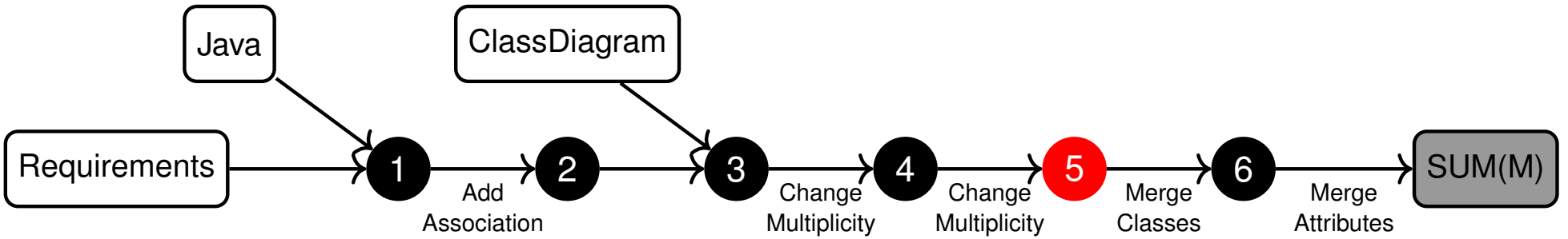


# 2. Initialization of SUM: Details

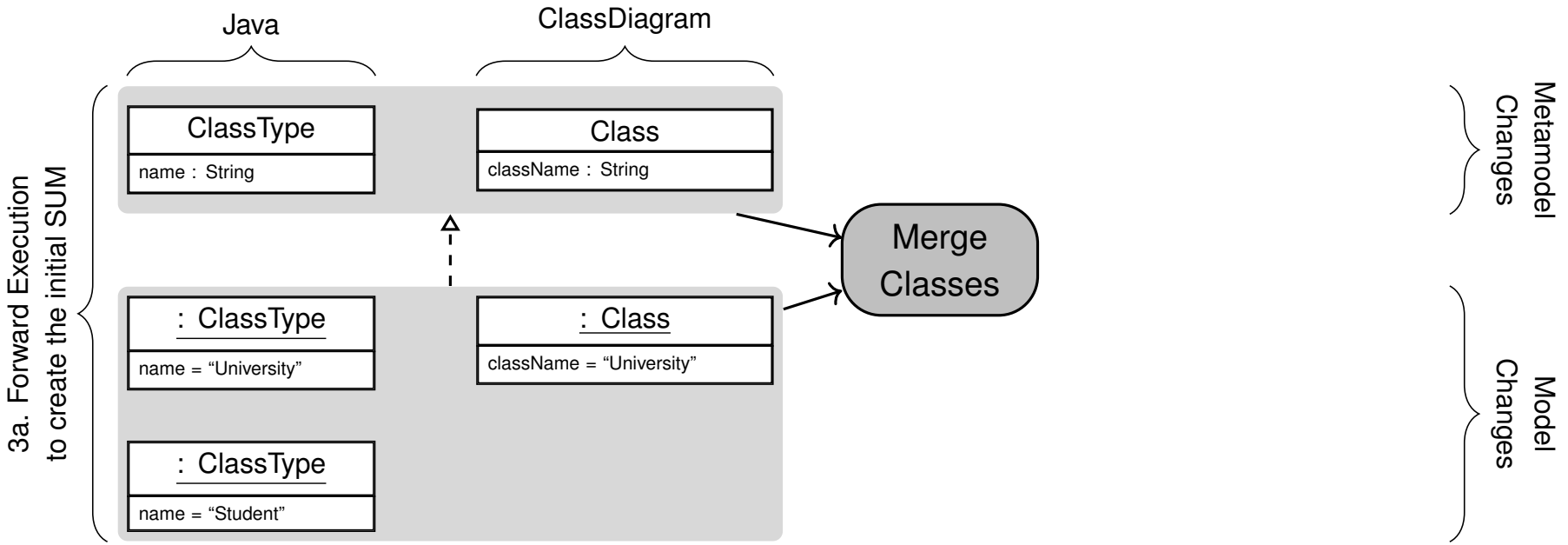


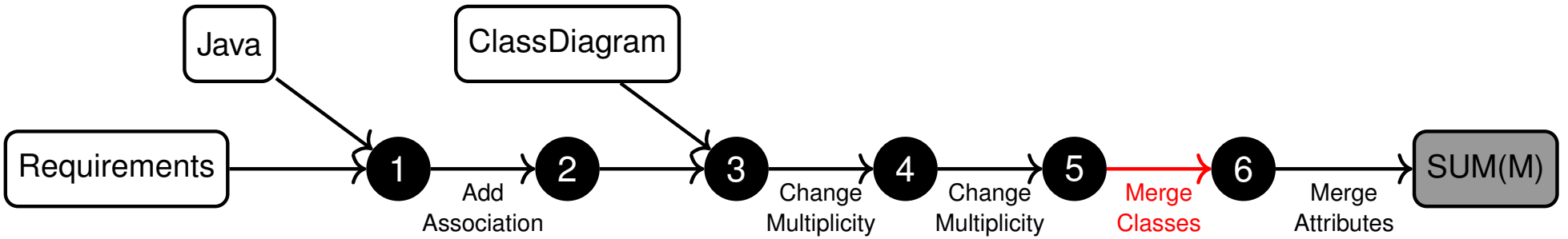
# 2. Initialization of SUM: Details



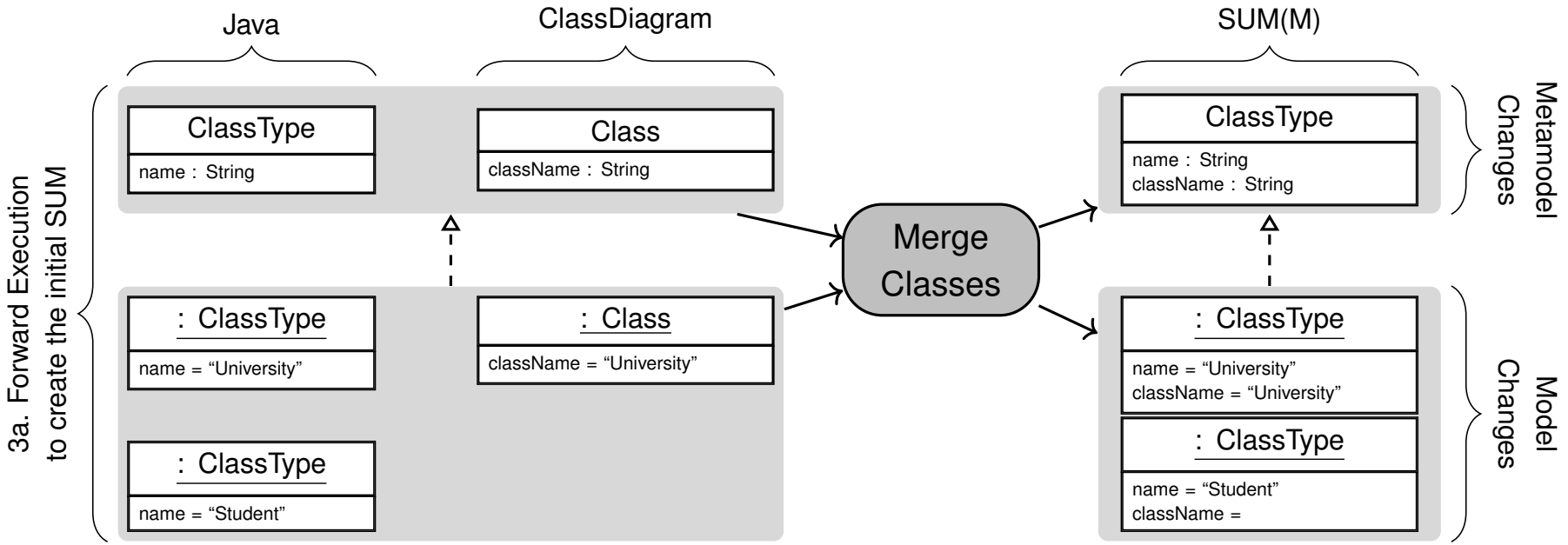



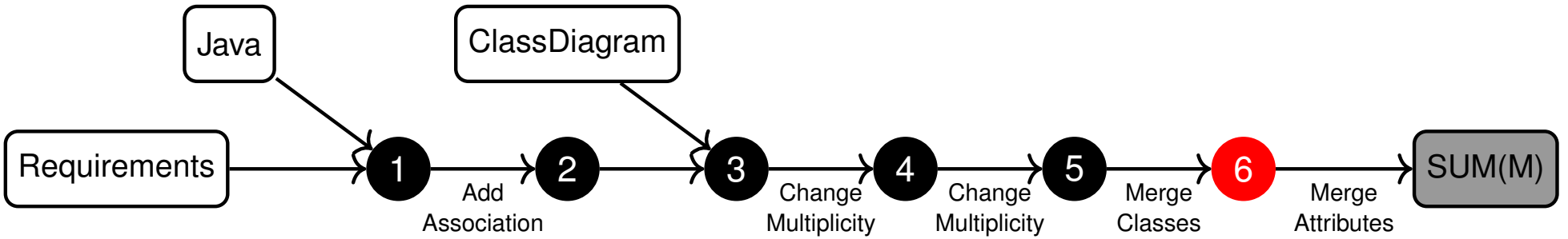
# 2. Initialization of SUM: Details



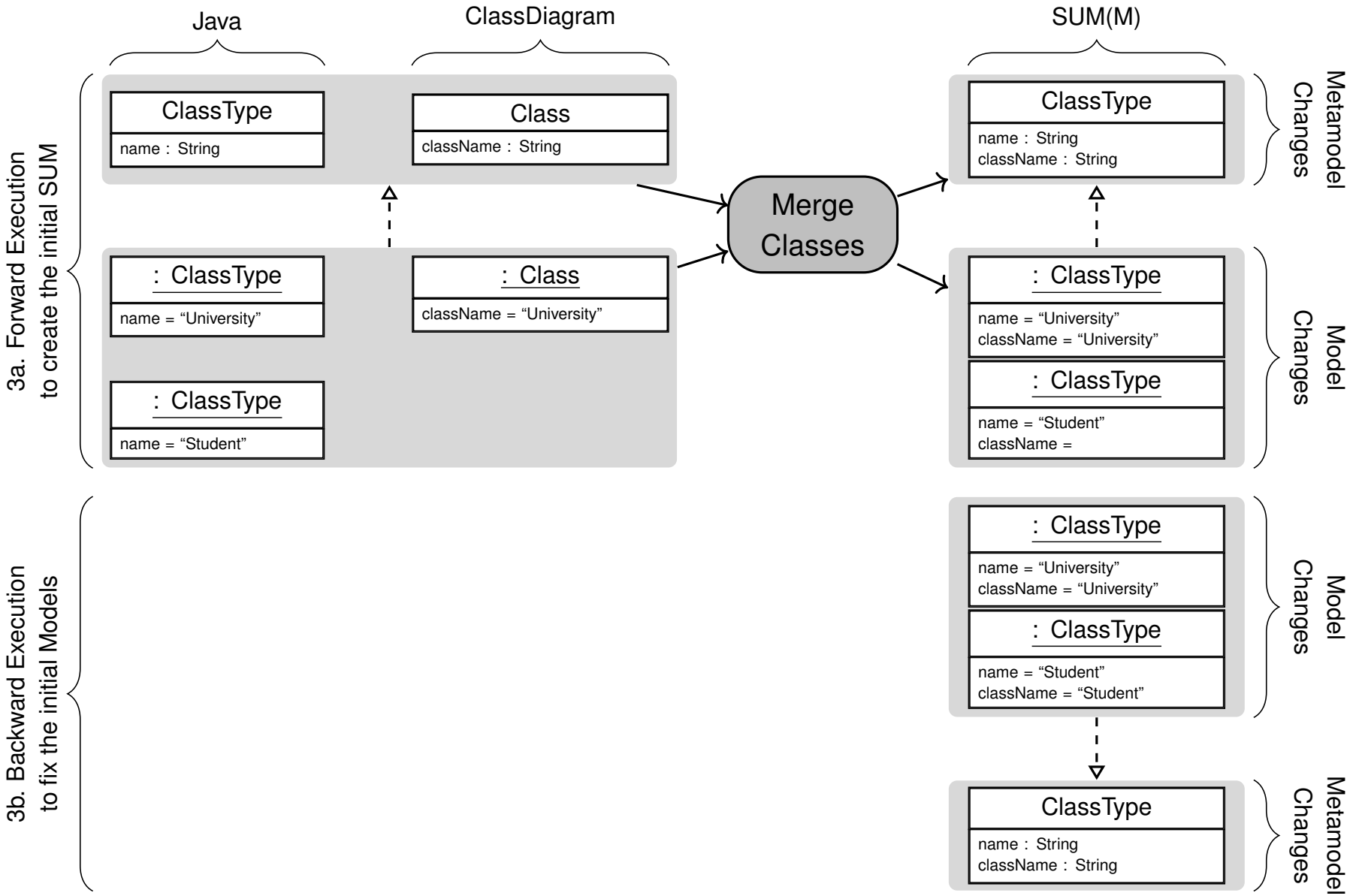



# 2. Initialization of SUM: Details

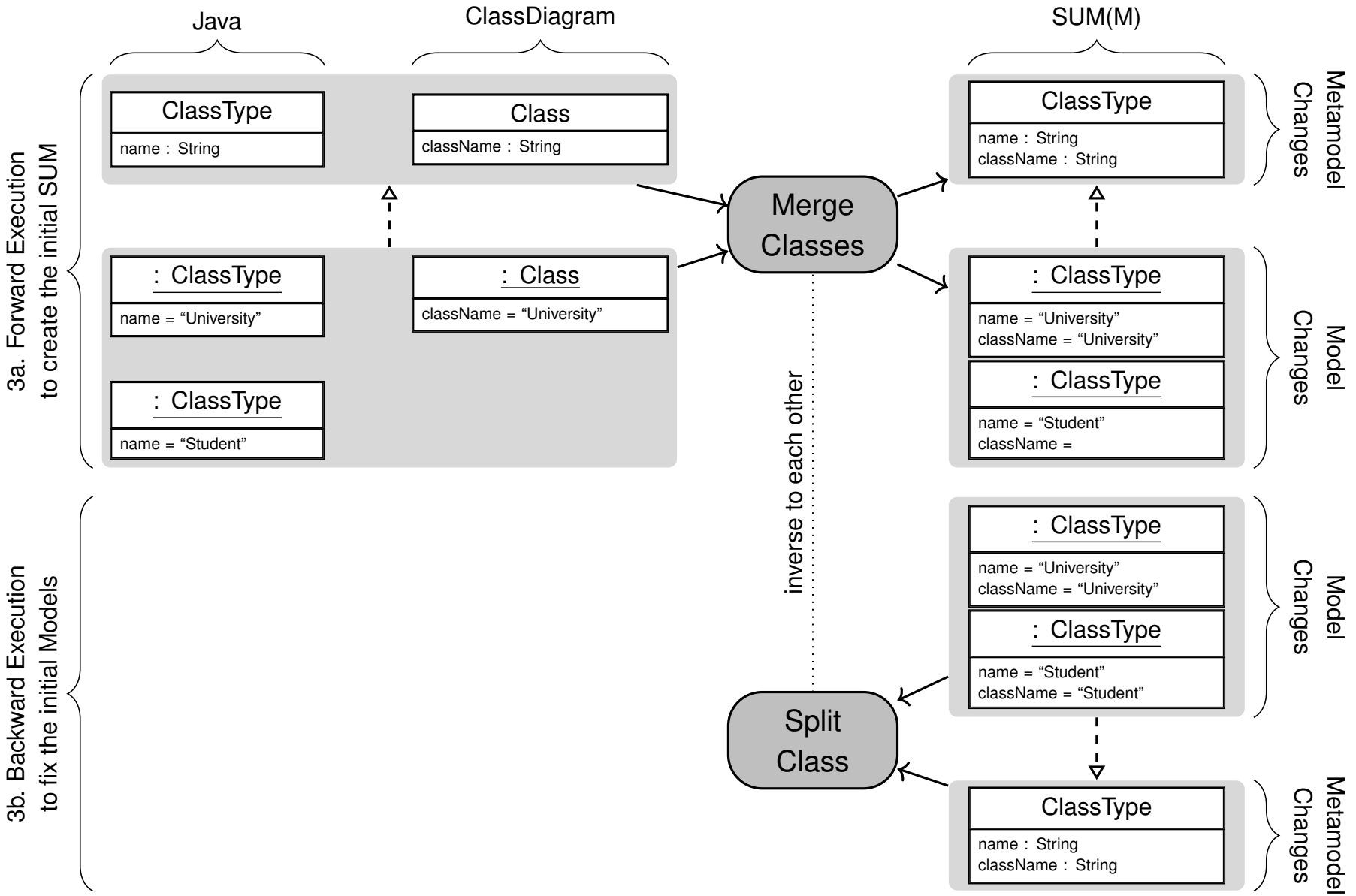


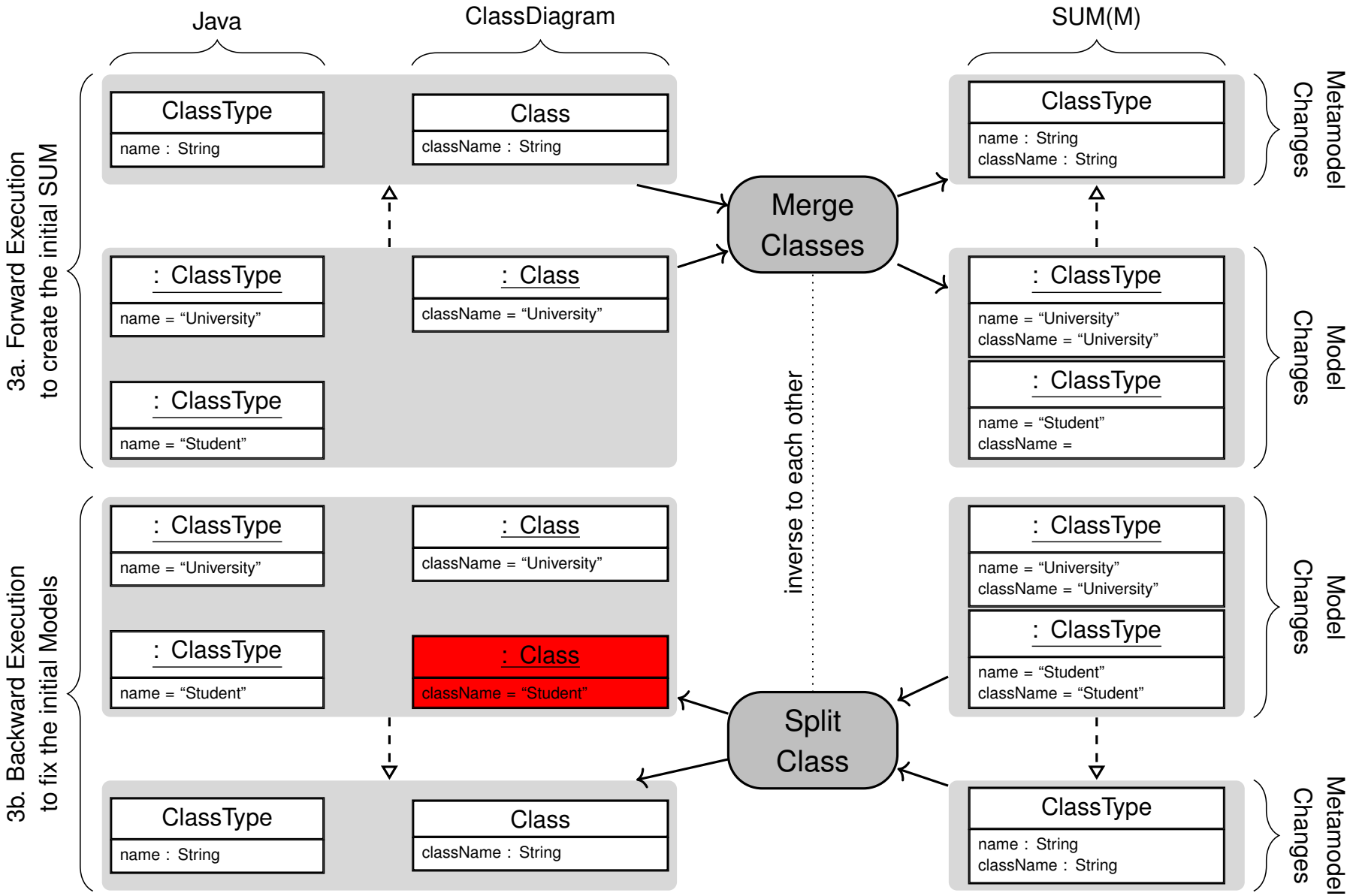
# 2. Initialization of SUM: Details



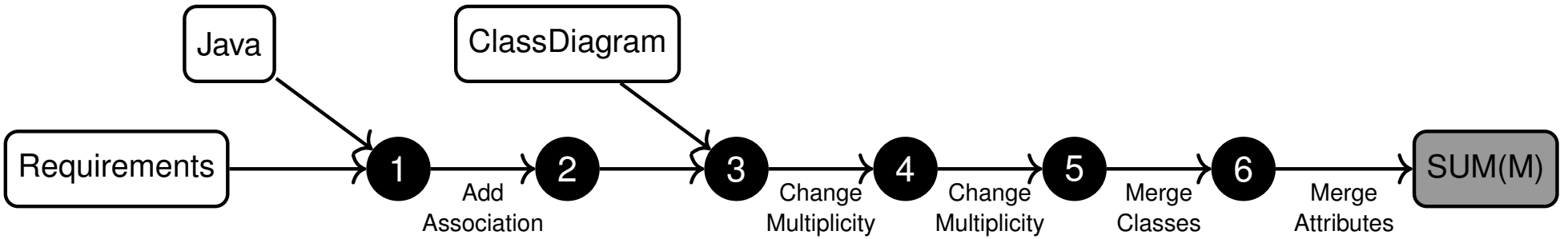
# 2. Initialization of SUM: Details



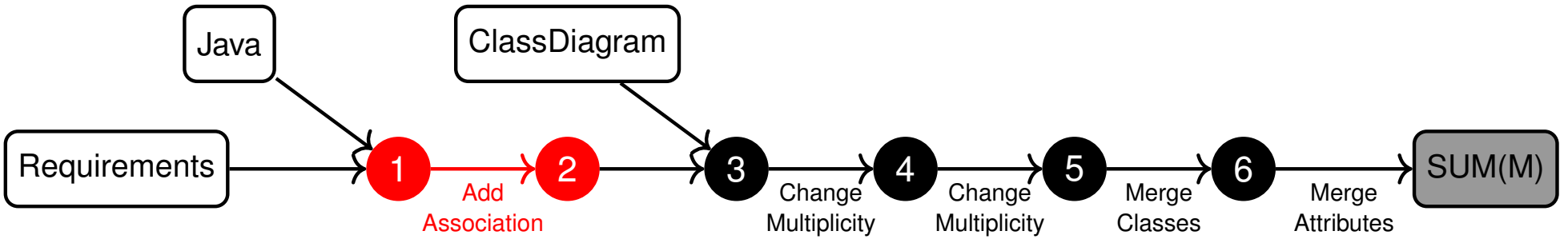
# 2. Initialization of SUM: Details



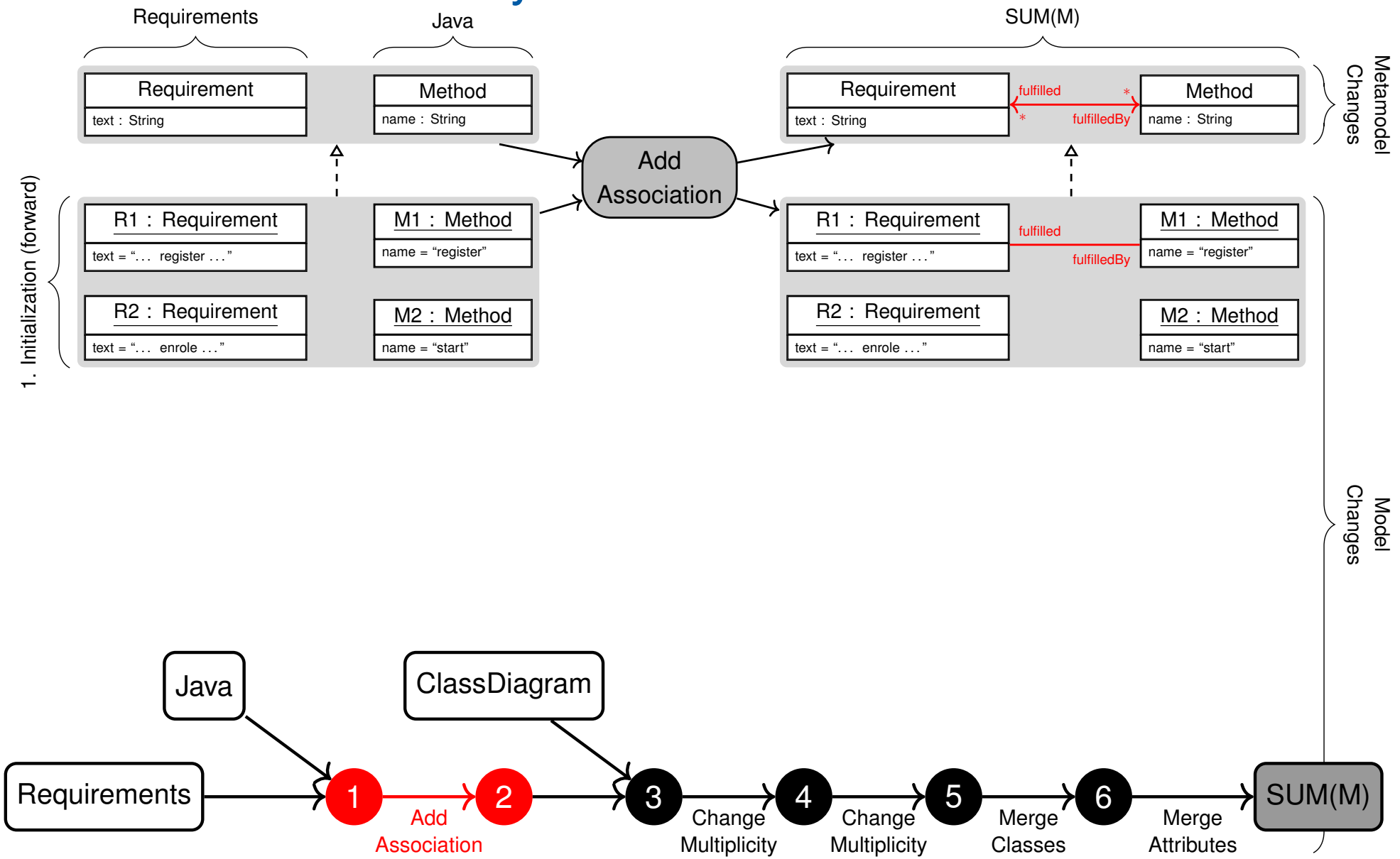
# 3. Consistency Assurance: Details



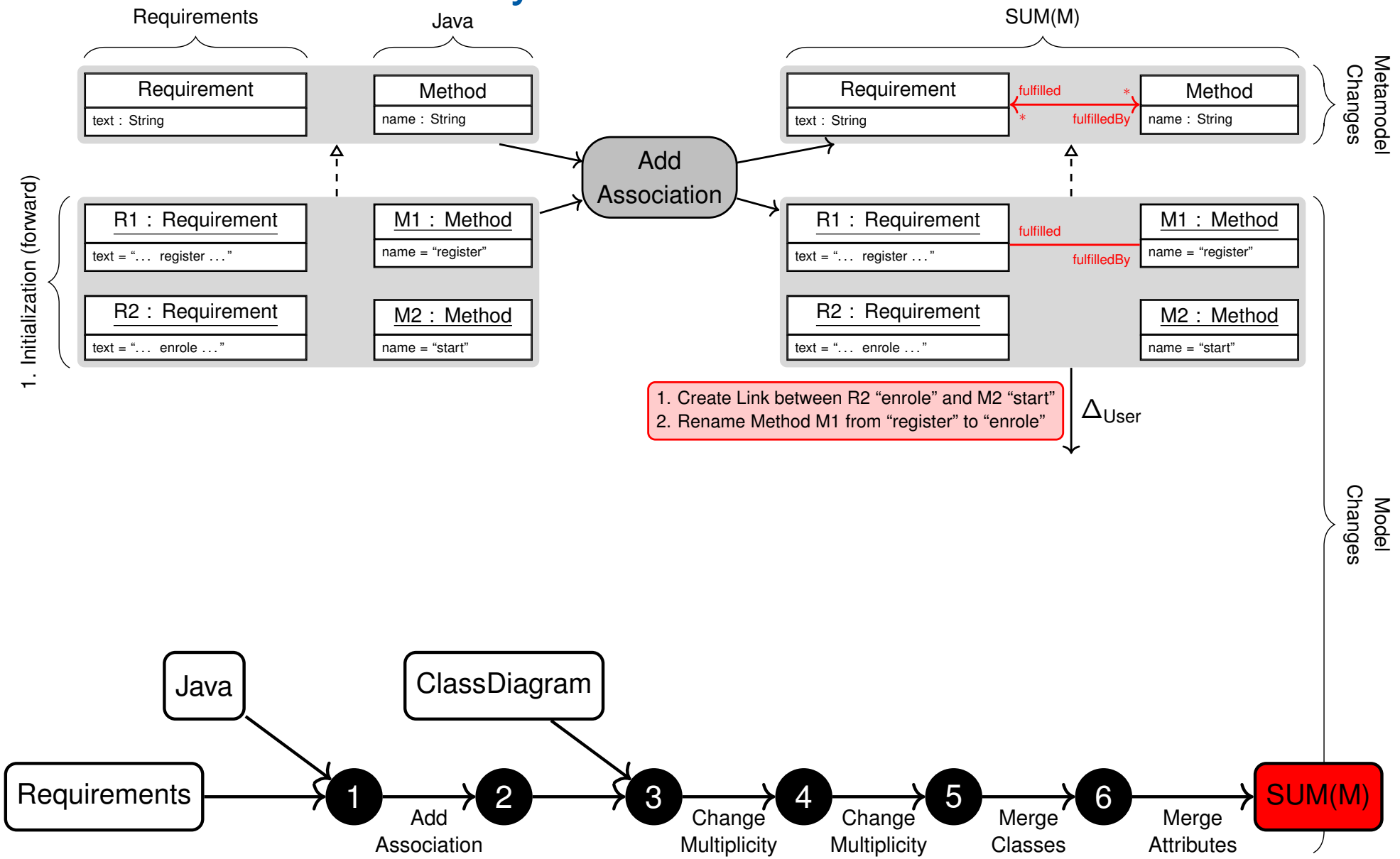
# 3. Consistency Assurance: Details



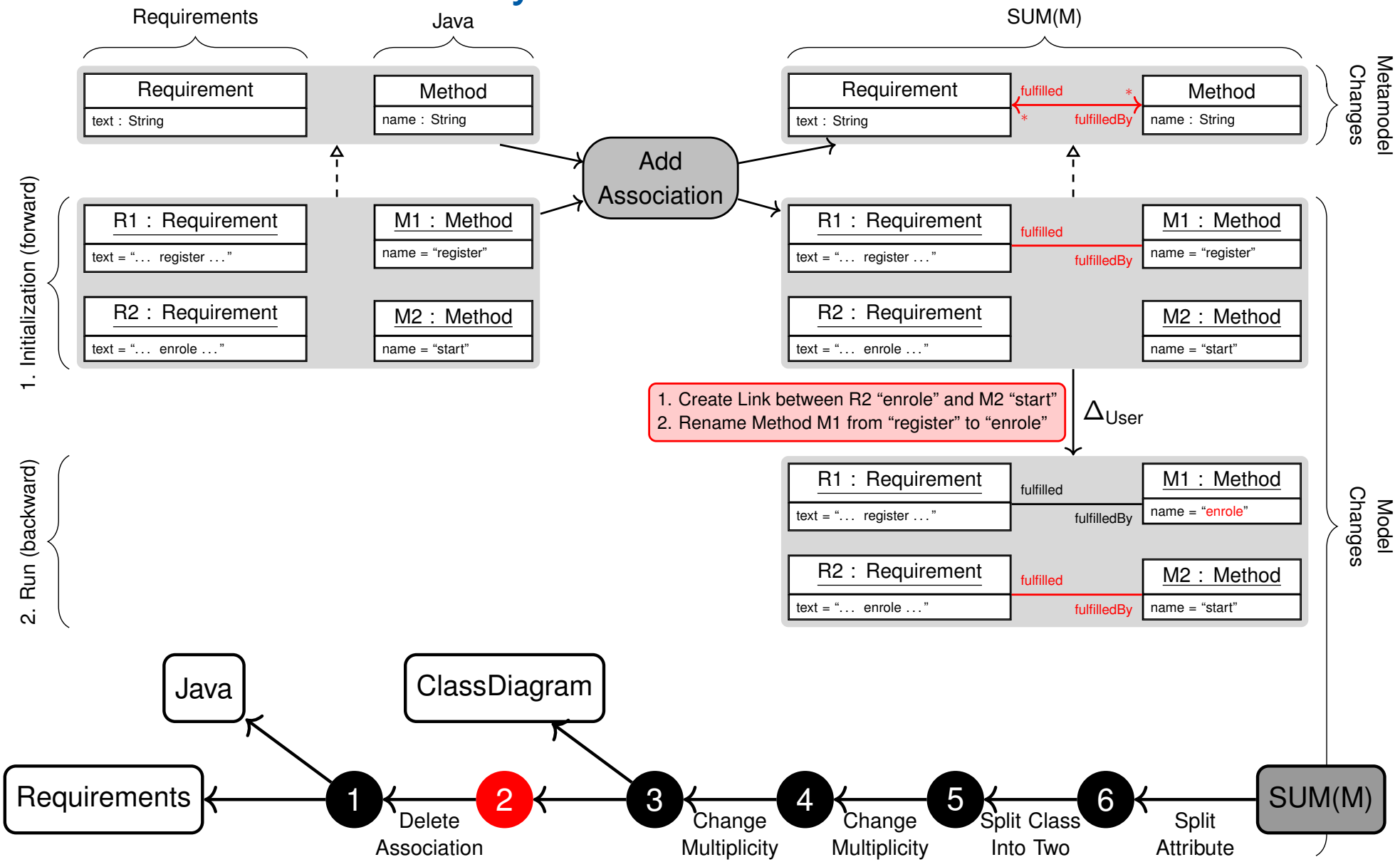
# 3. Consistency Assurance: Details



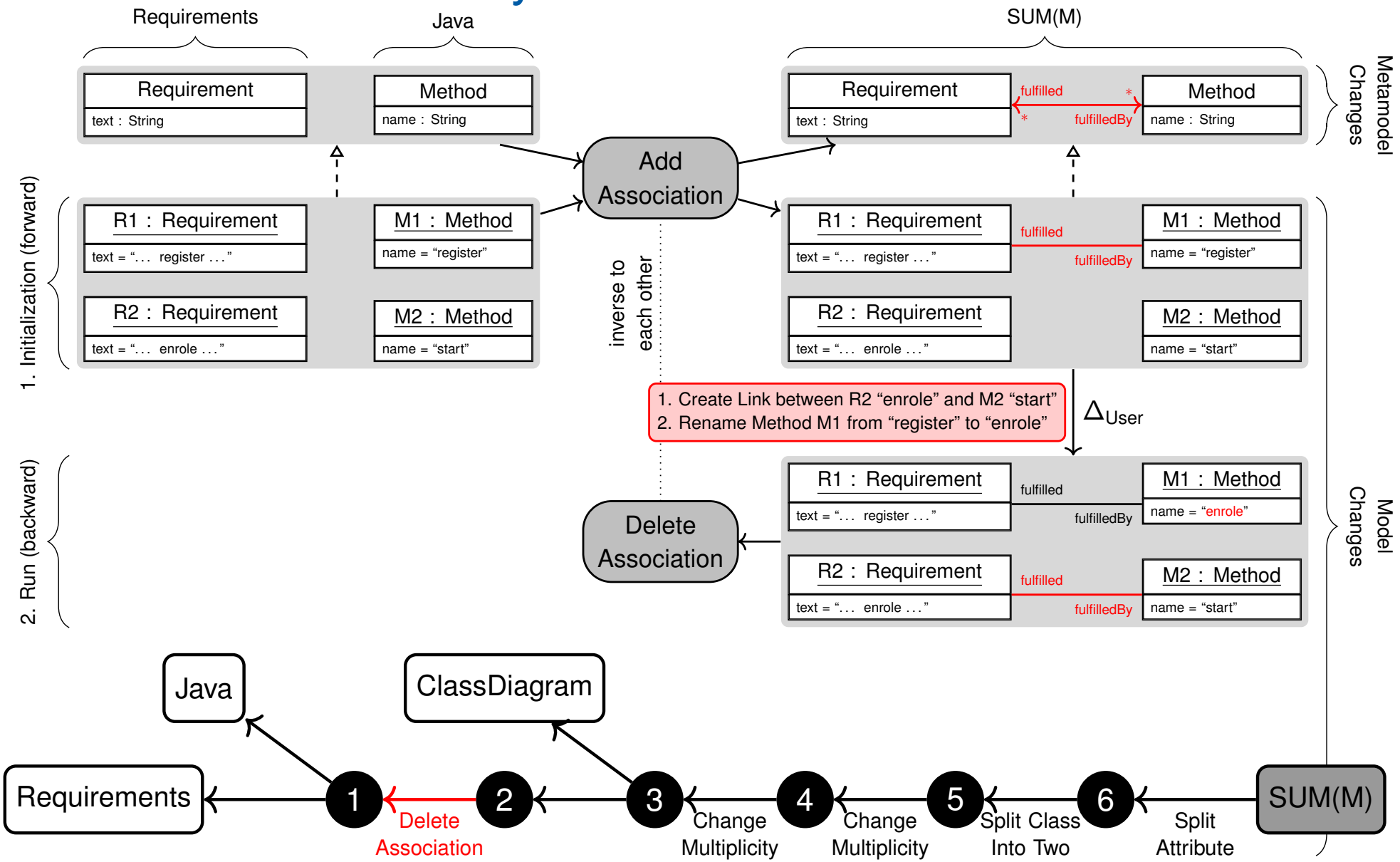
# 3. Consistency Assurance: Details



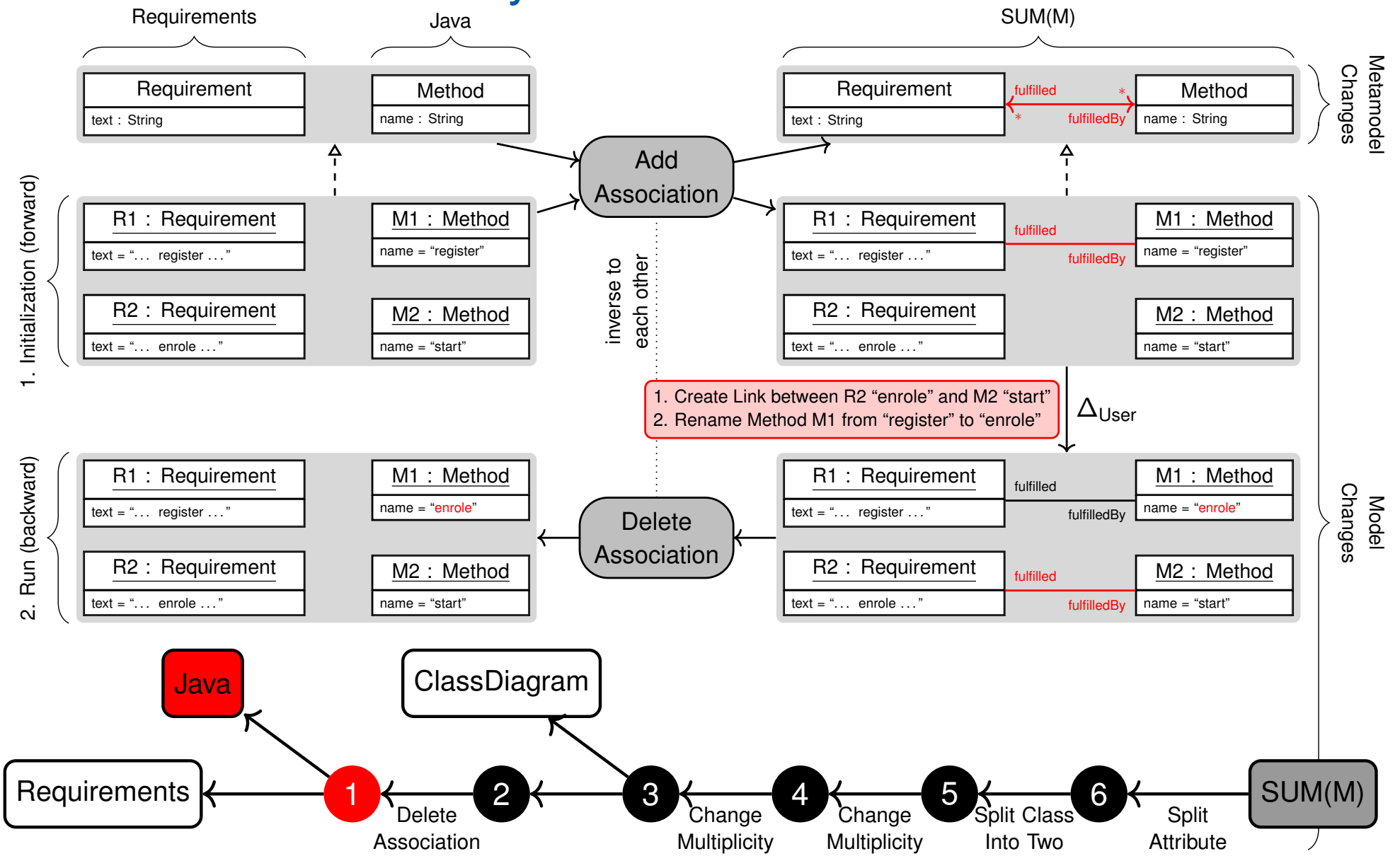
# 3. Consistency Assurance: Details



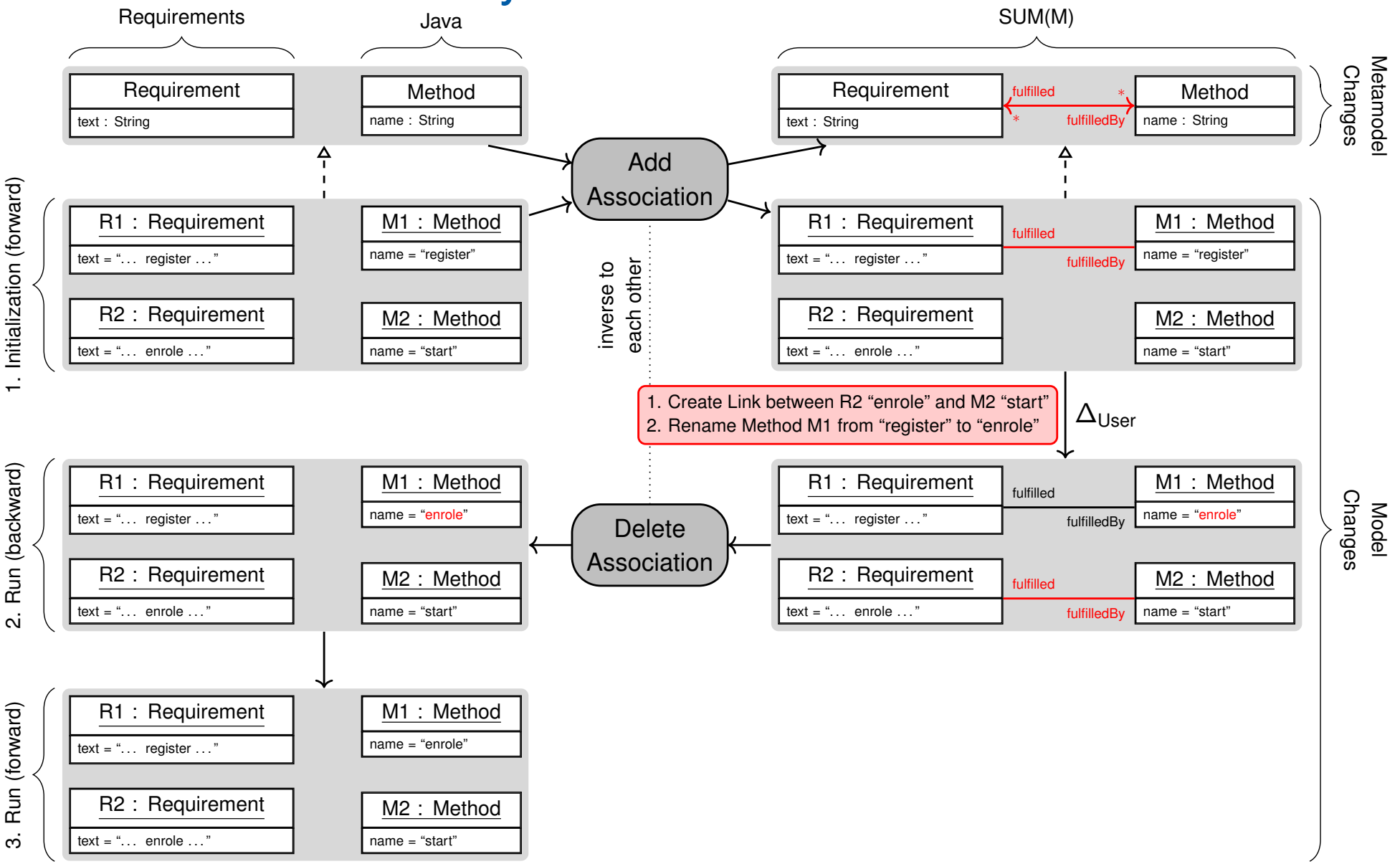
# 3. Consistency Assurance: Details



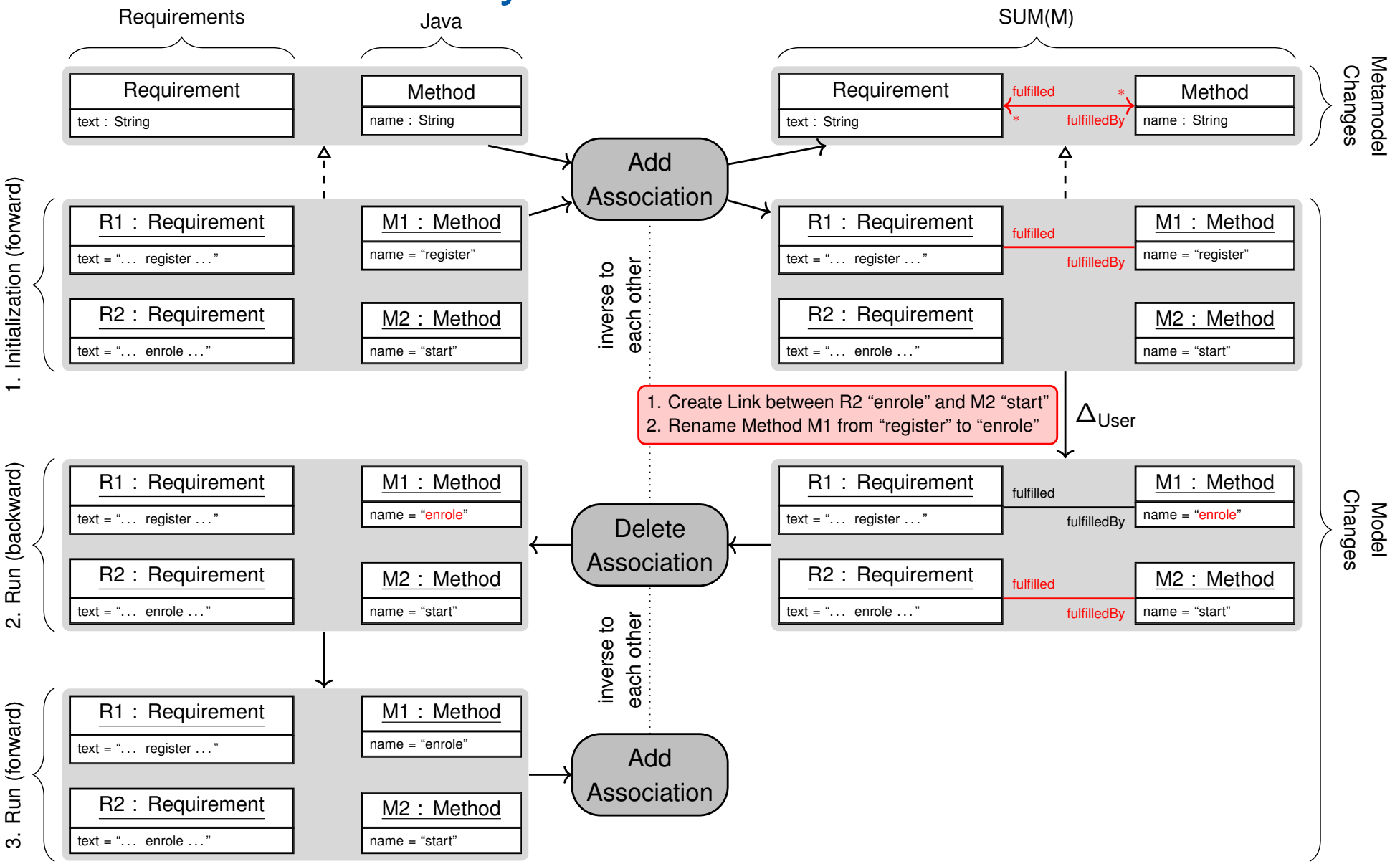
# 3. Consistency Assurance: Details



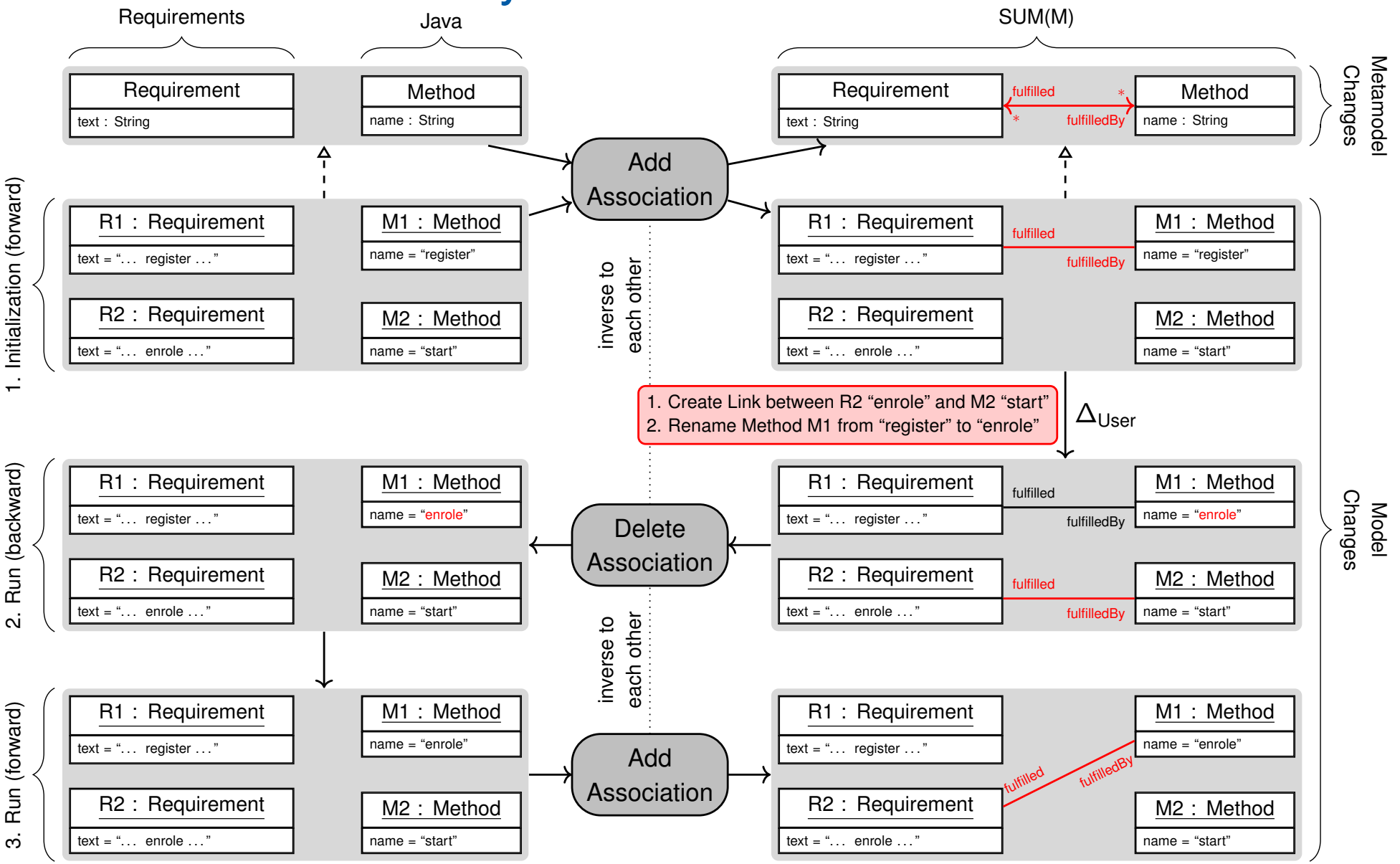
# 3. Consistency Assurance: Details



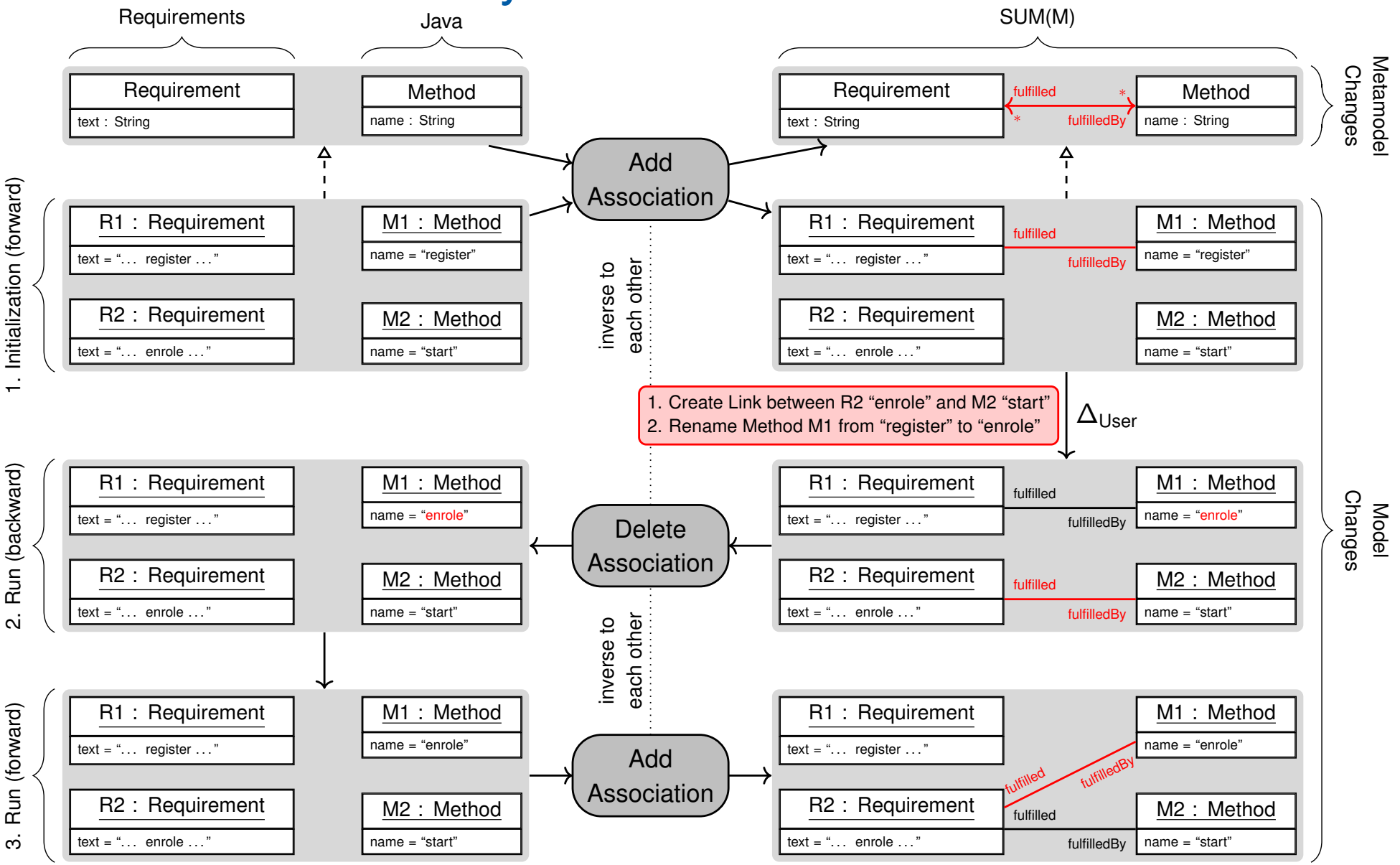
# 3. Consistency Assurance: Details



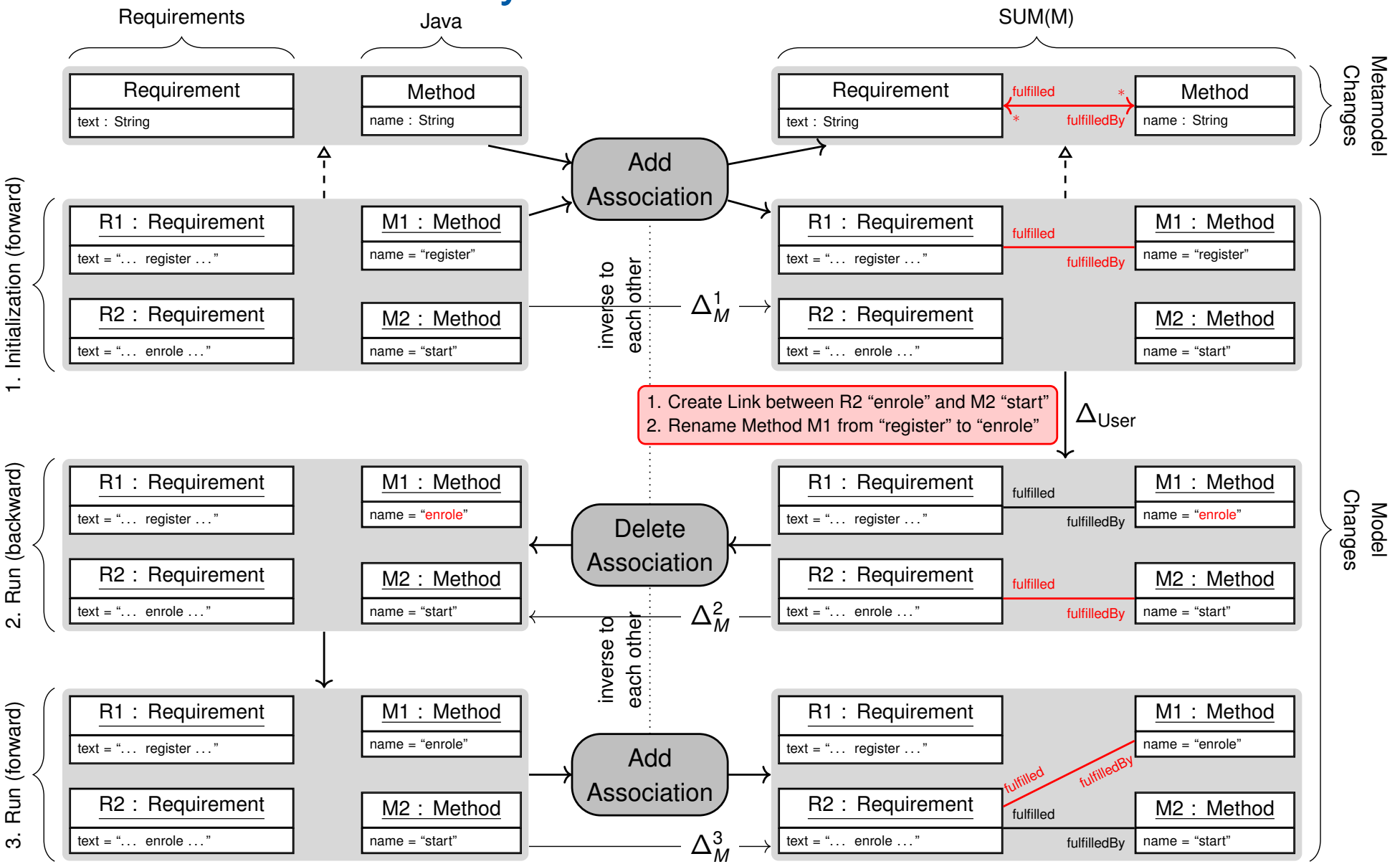
# 3. Consistency Assurance: Details



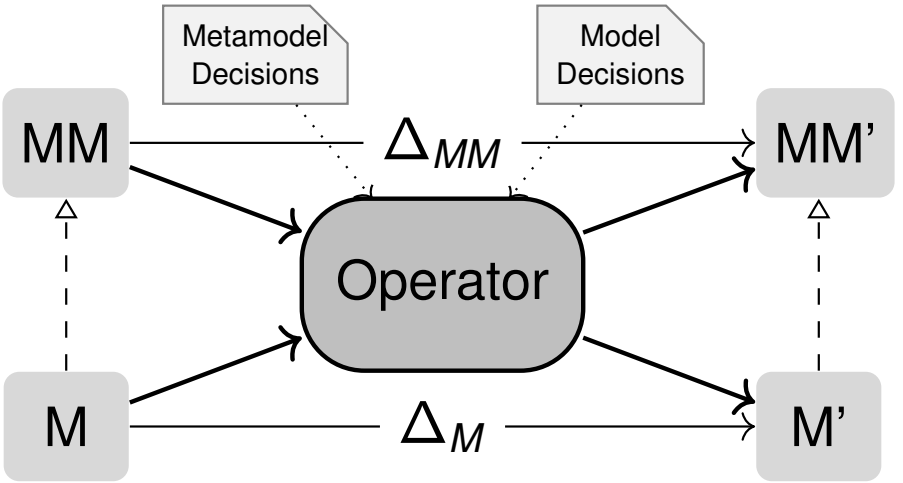
# 3. Consistency Assurance: Details



# 3. Consistency Assurance: Details



# Operators: Summary

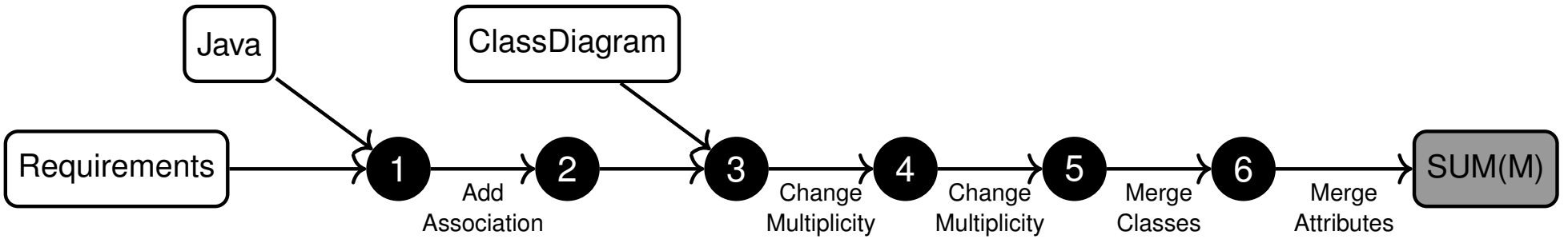



- **Metamodel Change** $\Delta_{MM}$:
  small Change in the Metamodel
- **Model Change** $\Delta_M$:
  handle Model-Co-Evolution (Coupled Operators [HVW11])
- Configurations by the Methodologist:
  - **Metamodel Decisions**: set Properties for wanted Metamodel Changes
  - **Model Decisions**: describe 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
- $\rightarrow$ separated Models are migrated to projectional Views on the SUM



# Literature I

[ASB09] Colin Atkinson, Dietmar Stoll, and Philipp Bostan. Supporting View-Based Development through Orthographic Software Modeling. Evaluation of Novel Approaches to Software Engineering (ENASE), pages 71–86, 2009.

[BHK+14] Erik Burger, Jörg Henss, Martin Küster, Steffen Kruse, and Lucia Happe. View-based model-driven software development with ModelJoin. Software & Systems Modeling, 2014.

[CNS12] Marsha Chechik, Shiva Nejati, and Mehrdad Sabetzadeh. A Relationship-Based Approach to Model Integration. Innovations Syst Softw Eng, 8(123):3–18, 2012.

[EEC+14] Mahmoud El Hamlaoui, Sophie Ebersold, Bernard Coulette, Mahmoud Nassar, and Adil Anwar. Heterogeneous models matching for consistency management. In 2014 IEEE Eighth International Conference on Research Challenges in Information Science (RCIS), pages 1–12. IEEE, may 2014.

[HVW11] Markus Herrmannsdoerfer, Sander D. Vermolen, and Guido Wachsmuth. An Extensive Catalog of Operators for the Coupled Evolution of Metamodels and Models. Software Language Engineering, LNCS 6563:163–182, 2011.

# Literature II

[IEE11] IEEE. ISO/IEC/IEEE 42010:2011 - Systems and software engineering - Architecture description. 2011(March):1–46, 2011.

[KBL13] Max E Kramer, Erik Burger, and Michael Langhammer. View-centric engineering with synchronized heterogeneous models. Proceedings of the 1st Workshop on View-Based, Aspect-Oriented and Orthographic Software Modelling - VAO '13, pages 1–6, 2013.

[LDC18] Manuel Leduc, Thomas Degueule, and Benoit Combemale. Modular Language Composition for the Masses. SLE 2018 - 11th ACM SIGPLAN International Conference on Software Language Engineering, 2018.

[MW18] Johannes Meier and Andreas Winter. Towards Evolution Scenarios of Integrated Software Artifacts. Softwaretechnik-Trends, 38(2):63–64, 2018.

[RJV09] J. R. Romero, Juan Ignacio Jaén, and Antonio Vallecillo. Realizing correspondences in multi-viewpoint specifications. Proceedings - 13th IEEE International Enterprise Distributed Object Computing Conference, EDOC 2009, pages 163–172, 2009.

[SK08] Andy Schürr and Felix Klar. 15 Years of triple graph grammars: Research challenges, new contributions, open problems. Lecture Notes in Computer Science, 5214 LNCS:411–425, 2008.