

# 1 Architecture

## 1.1 Matching

Aufgabe eines *Matchers* ist es, die korrespondierenden Elemente aus Modell A und Modell B, also die Elemente, die in beiden Modellen übereinstimmen, zu identifizieren. Dabei ist das Ergebnis vor allem davon abhängig anhand welcher Kriterien der Matcher eine Übereinstimmung festlegt. Hier wird unter anderem unterschieden zwischen *ID*-, *signatur*- und *ähnlichkeitsbasierten* Verfahren.

Das Framework definiert einen Extension Point (siehe 3.1), um durch entsprechende Matcher erweitert werden zu können. Die konzeptuelle Struktur des Ergebnisses, also eines *Matchings*, ist durch das in Abbildung 1 dargestellte, EMF-basierte Metamodell definiert.

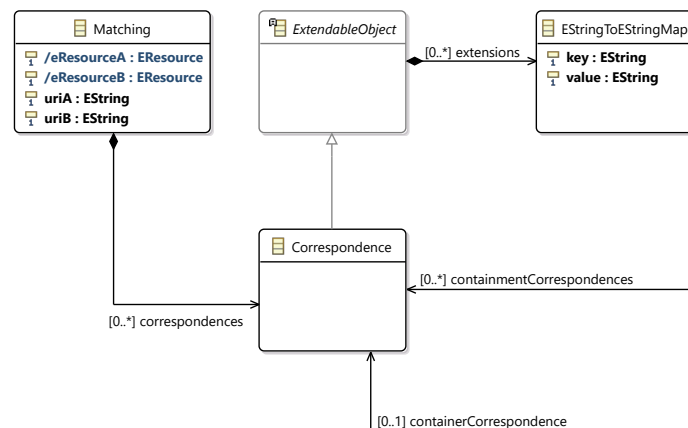


Abbildung 1: Abstract Syntax of a model matching

TODO: Beschreibung der abstrakten Syntax

## 1.2 Difference Derivation

Ausgehend von den gefunden Korrespondenzen berechnet der *Difference Derivator* eine technische Differenz (*low-level difference*) der Modelle. Alle Objekte und Referenzen, für die keine Korrespondenz existiert, müssen demnach entweder in Modell B hinzugefügt, oder aus Modell A entfernt worden sein. Durch die Verwendung eines *Technical Difference Builders* können Modellelemente von der Differenzberechnung ausgeschlossen werden. Dieser kann ebenfalls über einen Extension Point (siehe 3.2) erweitert und für die jeweilige Domäne angepasst werden.

### 1.3 Lifting Engine

Aufgabe der *Lifting Engine* ist es, die zuvor berechnete technische Differenz semantisch zu liften. Eine technische Differenz enthält alle Änderungen auf Basis des Metamodells, also der abstrakten Syntax. Beim *semantischen liften* werden die einzelnen Änderungen zu sogenannten *Semantic Change Sets* gruppiert, welche Editieroperation auf Ebene der konkreten Syntax repräsentieren.

Die konzeptuelle Struktur einer gelifteten, symmetrischen Differenz ist durch das in Abbildung 2 dargestellte, EMF-basierte Metamodell definiert.

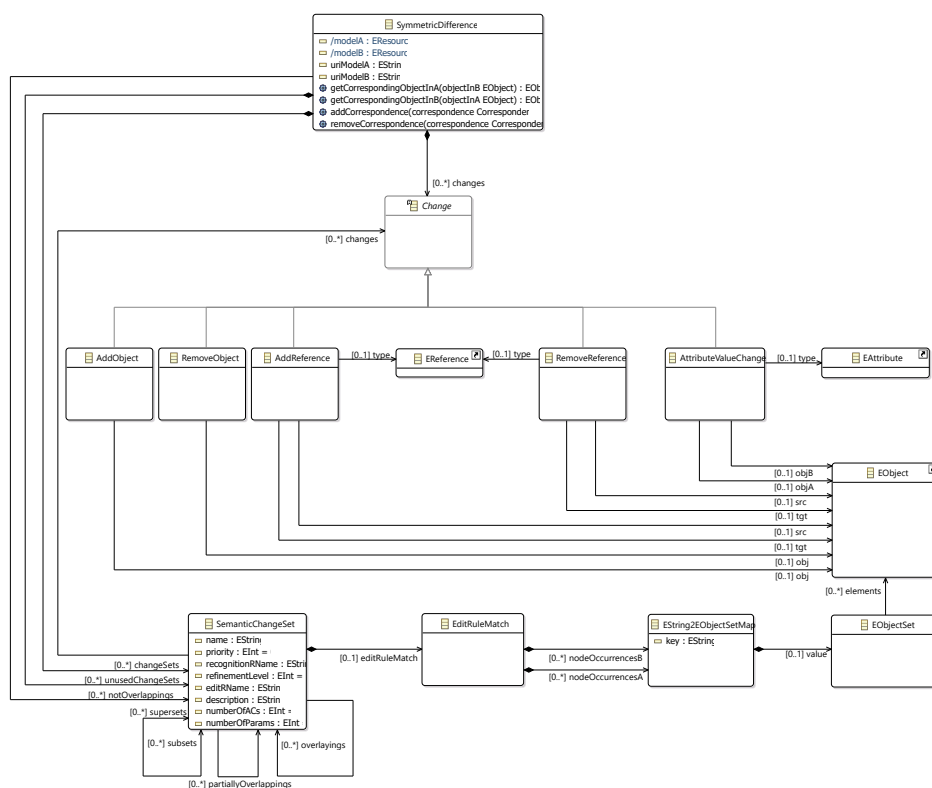


Abbildung 2: Abstract Syntax of a descriptive model difference

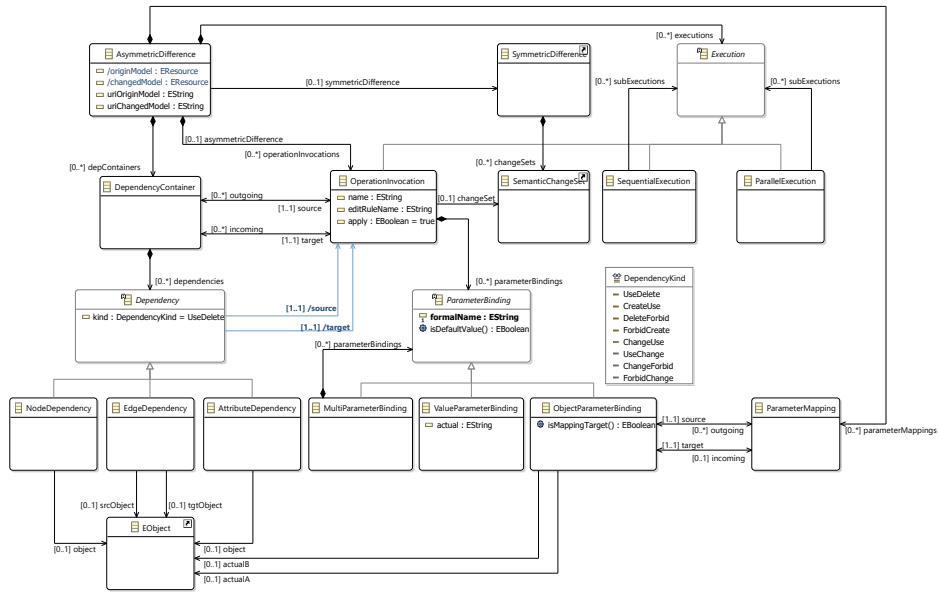


Abbildung 3: Abstract Syntax of a prescriptive model difference

## 1.4 Rule Base

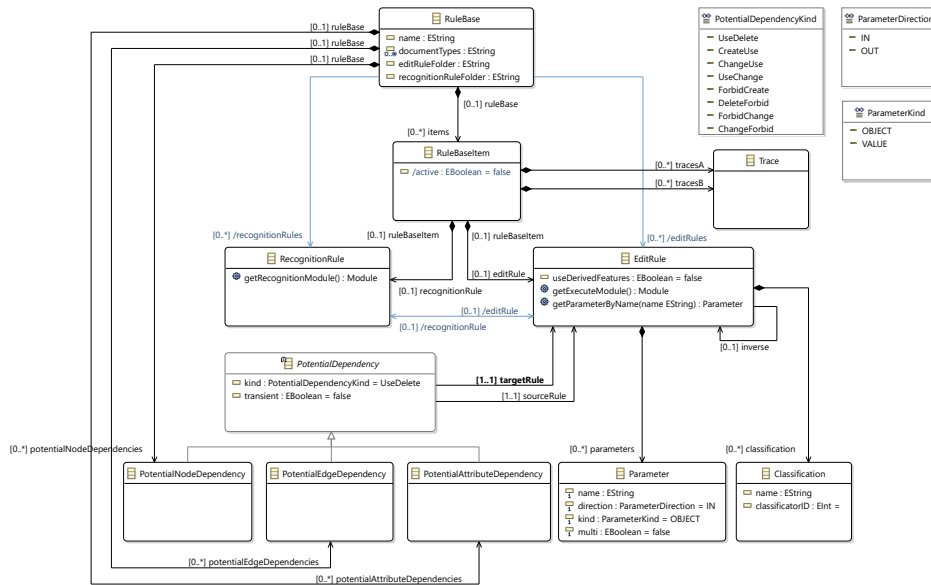


Abbildung 4: Conceptual structure of a rule base

## 2 Application Programming Interface

**Plug-in ID:** org.sidiff.matching.api

**Extension Points:** none

**Extensions:** none

**Description:**

Die Benutzerschnittstelle stellt Methoden zur Verfügung, um Korrespondenzen zwischen mehreren Modellen zu berechnen und zu serialisieren.

**Plug-in ID:** org.sidiff.difference.technical.api

**Extension Points:** none

**Extensions:** none

**Description:**

Die Benutzerschnittstelle stellt Methoden zur Verfügung, um eine technische Differenz zwischen zwei Modellen zu berechnen und zu serialisieren.

**Plug-in ID:** org.sidiff.difference.lifting.api

**Extension Points:** none

**Extensions:** none

**Description:**

Die Benutzerschnittstelle stellt Methoden zur Verfügung, um eine technische Differenz semantisch zu liften und zu serialisieren.

**Plug-in ID:** org.sidiff.difference.asymmetric.api

**Extension Points:** none

**Extensions:** none

**Description:**

Die Benutzerschnittstelle stellt Methoden zur Verfügung, um eine geliftete ausführbare asymmetrische Differenz zu berechnen und zu serialisieren.

## 3 Extension Points

### 3.1 Matching Engine

**Plug-in ID:** org.sidiff.correspondences

**Extension Points:** org.sidiff.correspondences.extensionpoint

**Extensions:** none

**Description:**

TODO

**Plug-in ID:** org.sidiff.candidates

**Extension Points:** org.sidiff.candidates.extensionpoint

**Extensions:** none

**Description:**

TODO

**Plug-in ID:** org.sidiff.matcher

**Extension Points:** org.sidiff.matcher.extensionpoint

**Extensions:** none

**Description:**

This extension point is used to add new matchers. A plugin that adds this extension point has to implement the IMatcher interface.

**Plug-in ID:** org.sidiff.similarities

**Extension Points:** org.sidiff.similarities.extensionpoint

**Extensions:** none

**Description:**

TODO

## 3.2 Difference Derivator

**Plug-in ID:** org.sidiff.difference.technical

**Extension Points:** org.sidiff.difference.technical.technical\_difference\_builder\_extension

**Extensions:** none

**Description:**

This extension point is used to add a technical difference builder to the lifting engine. A plugin that adds this extension point has to implement the ITechnicalDifferenceBuilder interface. This interface offers the following information:

- The document type the technical difference builder was implemented for.

### 3.3 Lifting Engine

**Plug-in ID:** org.sidiff.difference.rulebase

**Extension Points:** org.sidiff.difference.rulebase.rulebase\_extension

**Extensions:** org.eclipse.emf.ecore.generated\_package

**Description:**

This extension point is used to add new rulebases to the recognition engine. A plugin that adds this extension point had to implement the IRuleBase interface. This interface offers the following information to the recognition engine:

- A description name of the rulebase.
- The document type the rulebase was generated for. In other words, the meta model the edit rules were implemented for.
- A list containing all rulebase recognition rules which will be applied by the recognition engine.

### 3.4 Merging Engine

**Plug-in ID:** org.sidiff.conflicts.modifieddetector

**Extension Points:** org.sidiff.conflicts.modifieddetector

**Extensions:** None

**Description:**

TODO