

FirstInstance Changes

Overview

- **3 change groups:**
 - **Group Toolbox:** About similarity checking operations
 - **Group SimChecker:** About refining roles of SimilarityX classes
 - **Group Structure:** About extracting re-usable parts

Group - Toolbox

- **ISimilarityToolbox** (short: Toolbox)
- **ISimilarityRequest** (short: Request)
- **ISimilarityRequestHandler** (short: Handler)

Group – Toolbox (Problems)

- **Mutual operations not encapsulated properly**
 - Such operations and their parameters are not a unit
 - \Rightarrow Analysing data flow becomes more difficult
- **Access to mutual operations not centralised properly**
 - No central, non-singleton access point for such operations
 - \Rightarrow Modifying such operations becomes harder (especially during runtime)
 - \Rightarrow Hard to recognise potential relations between operations
- **No dynamic control over similarity checking**
 - No mechanisms to establish control during runtime
 - \Rightarrow Testing and implementing stateful operations becomes more challenging

Group – Toolbox (Concepts)

- **Modified Chain of Responsibility**
- **Operations requiring «outside» parameters as Requests**
- **Functionalities corresponding to Requests as Handlers**
- **Assign & match Handler(s) for Request class(es)**

Group – Toolbox (Concepts)

- **Requests contain parameters, Handlers return result**
- **Handlers can send out further requests**
 - Other Handler(s) (or Toolbox) as class member, if needed
- **Dynamic control over similarity checking**
- **Requests & Handlers implementation & context dependent**

Group – Toolbox (Toolbox)

- **Toolbox stores (Request class, Handler) pairs**
- **Toolbox itself can serve as Handler**
 - Implements the Handler interface
 - Forwards Requests to appropriate Handler
- **Toolbox receives Requests from outside**
- **Toolbox determines the appropriate Handler**
 - Decision mechanism can be extracted in future (if needed)

Group - Toolbox

- **ISimilarityToolboxBuilder** (short: Builder)
- **ISimilarityToolboxFactory** (short: Factory)

Group – Toolbox (Problems)

- **Some operations require «outside» parameters**
 - The parameters are dragged throughout the control flow (ex: normalization)
 - \Rightarrow Parameter, attribute and even constructor pollution
- **All tools (and their parameters) are always included**
 - Forced to construct unnecessary tools/mechanisms (ex: PCMSimChecker)
 - \Rightarrow Redundant dependencies, unnecessary attributes/parameters
- **No stepwise tool construction**
 - Parameters required for the tools have to be piled up and dragged along
 - \Rightarrow Creates unnecessary chaos and confusion

Group – Toolbox (Concepts)

- **Builder + Factory to initialise and instantiate Toolbox**
- **Builder: Mitigates parameter/attribute chaos**
- **Factory: Lets builder abstract from Toolbox instantiation**

Group – SimChecking

- **SimilarityChecker** (short: Checker)
- **SimilarityComparer** (short: Comparer)
- **SimilaritySwitch** (short: Switch)
- **Inner switches**

Group – SimChecking (Problems)

- **Checker does too much:**
 - 1) Contains a sizable chunk of similarity checking logic
 - 2) Actively partakes in similarity checking (checkStatementPosition flag)
 - 3) Initialises and manages normalization constructs
 - 4) Acts as a Facade for outside classes
- **Switch is a god-class and knows «outside» parameters**
- **No way to customise inner switches inside Switch**
- **Inner switches are inaccessible from tests**

Group – SimChecking (Checker)

- **Acts as a Facade for outside accessors**
- **Takes a Toolbox as constructor parameter**
- **Forwards work (via Requests) to the Comparer it constructs**
- **No longer (directly) responsible for Switch construction**
- **Only one Checker per similarity check**

Group – SimChecking (Comparer)

- **A new layer of indirection after Checker**
- **Takes a Toolbox as constructor parameter**
- **Forwards the work from Checker to its Toolbox**
- **No longer (directly) responsible for Switch construction**
- **Only one Comparer built per Checker**

Group – SimChecking (Switch)

- **Takes an I...Handler instance as constructor parameter**
 - Currently, the Toolbox of the Handler that constructed the Switch
- **Forwards Requests from its inner switches to its Toolbox**
- **Inner switches relieved of most attributes**
 - Exceptions: The «main» Switch and checkStatementPosition (CSP) flag
 - Switch required for forwarding requests
 - CSP flag required due to the way similarity checking works (endless recursions, important attribute)

Group - Structure

Extracted re-usable packages:

(from `org.splevo.jamopp.diffing`)

- **...similarity.base**
- **...similarity.base.ecore**

Group - Structure

- **The resulting structure contains many re-usable elements:**
 - 1) (Request class, Handler) pairing and matching concept
 - 2) Dynamic toolset construction
 - 3) Dynamically changable pairs
 - 4) Numerous interfaces with default methods
 - 5) Adaptations to EMFtext across similarity checking classes

Group - Structure (Base)

- **Contains the structure from before as abstract classes/interfaces**
- **Various helper methods/interfaces (also useful for testing)**
- **Without a certain context (Java, PCM, etc.)**
- **Without a certain external mechanism (EMFtext, etc.)**
- **Concrete implementations of ToolboxFactory & (empty) Toolbox**

Group - Structure (Base.Ecore)

- **Complements the Base Package with ComposedSwitch**
- **Extends abstract classes to conform ComposedSwitch**
- **Introduces new interfaces for working with inner switches**
- **Contains helper methods/interfaces to avoid code duplication**
 - Examples: In JavaSimilaritySwitch & PCMRepositorySwitch
 - Benefits tests as well
- **Exemplary Requests & Handlers for hierarchical EObject comparing**