


Generating rules for API Evolution

David Wettstein, 17.05.16

API Evolution

- APIs change over time
- Changes need to be updated in your code manually

labelledInteraction := RTLabelled new
labelledInteraction := RTLabelled new



Project goal

- API changes can cause problems in projects using an older version of the API
- Get automatically detected and packaged rules, such that you know what changed

APIEvolutionMiner (AEM)

- APIEvolutionMiner¹
 - extract rules by monitoring API changes
 - mined at revision level in code history

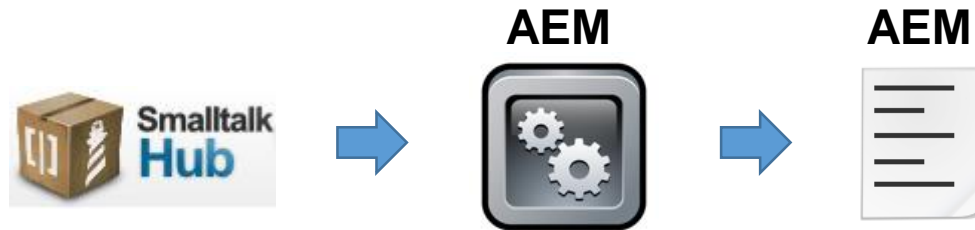
¹ [Hora A., Etien A., Anquetil N., Ducasse S., Valente M. T. APIEvolutionMiner: Keeping API evolution under control. In proceedings of CSMR-WCRE 2014.](#)

APIEvolutionMiner (AEM)

- Investigated
 - Jenkins CI jobs from Inria RMoD
 - AEM source code

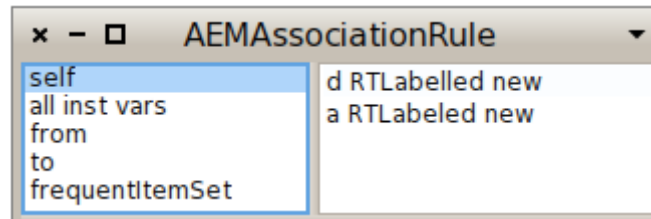
APIEvolutionMiner (AEM)

- Result: We can run it for any project per script



Problems with AEM

- Custom rule format:



- Wanted:

- ProtoObject
- Object
- RBLintRule
- **RBTransformationRule**

Problems with AEM

- AEM does not create RBLintRule!
- Main task:



Our approach

- Focus on automatically discovered changes of type “one to one”
- Generate subclasses of `RBTransformationRule` within code
 - using `subclass:` and `compile:`

Difficulties

- Differences in change pattern:
 - RTLabelled new => RTLabeled new
 - (?) builder: => ? addedInBuilder:
 - ? translateBy: => ? translate to:

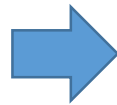
RBTransformationRule

- Uses `RBParseTreeRewriter` for doing a search & replace in the parse tree
- Works also with placeholders (e.g. ``@object)

Solution

- RTLabelled new => RTLabeled new

```
x - □ AEMAssociationRule
self
all inst vars
from
to
frequentItemSet
d RTLabelled new
a RTLabeled new
```



```
x - □ AEMTransformRule_d_RTLabelled_new
self
all inst vars
superclass
methodDict
self superclass RBTransformationRule
AEMTransformRule_d_RTLabelled_new_
_a_RTLabelled_new_
```

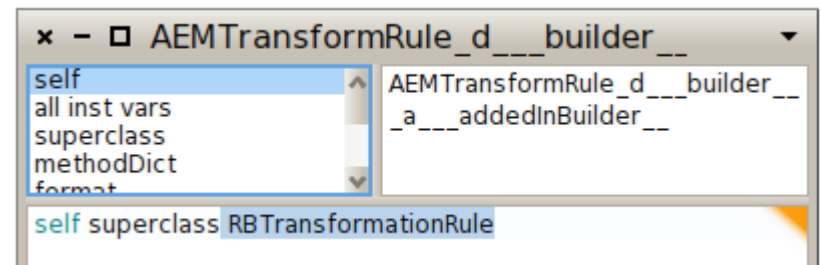
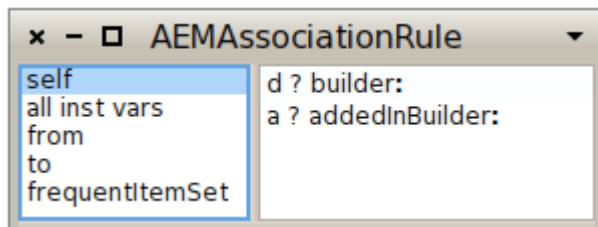
- This one is easy 😊

initialize

```
super initialize.
self rewriteRule
  replace: ' RTLabelled new '
  with: ' RTLabeled new '
```

Solution

- ? builder: => ? addedInBuilder:

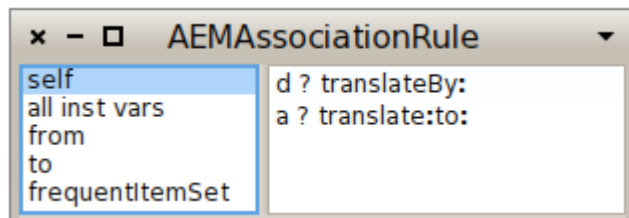


initialize

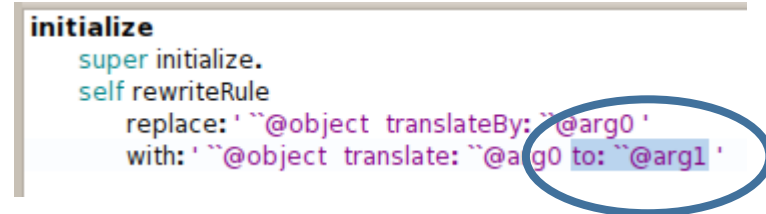
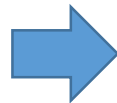
```
super initialize.  
self rewriteRule  
replace: ``@object builder: ``@arg0 '  
with: ' ``@object addedInBuilder. ``@arg0 '
```

What about this?

- ? translateBy: => ? translate:to:



```
x - □ AEMAssociationRule
self
all inst vars
from
to
frequentItemSet
d ? translateBy:
a ? translate:to:
```



```
initialize
super initialize.
self rewriteRule
  replace: '@object translateBy: '@arg0 '
  with: '@object translate: '@arg0 to: '@arg1 '
```

- Different number of arguments! Which one to take?

Example: Roassal2

- 68 automatic rules by AEM
- 57 convertible into RBTransformationRule

Results in Critic Browser

- Load Roassal2 version 1.4
- 706 critics found by 37 of 57 rules



```
x - □
▼ Selected Rules (To sort: 706, ToDo: 0, Wrong: 0)
  ▼ generatedAEMTransformRules (To sort: 706, ToDo: 0, Wrong: 0)
    d ? addClassesFromRegExp: => a ? addClassesMatching: (To sort: 0, ToDo: 0, Wrong: 0)
    d ? addPlugin: => a ? addDecorator: (To sort: 0, ToDo: 0, Wrong: 0)
    d ? asGroup => a RTGroup new (To sort: 7, ToDo: 0, Wrong: 0)
    d ? axisFrom:to: => a ? axisXFrom:to: (To sort: 14, ToDo: 0, Wrong: 0)
    d ? builder: => a ? addedInBuilder: (To sort: 17, ToDo: 0, Wrong: 0)
    d ? count => a ? identityHash (To sort: 0, ToDo: 0, Wrong: 0)
```


Results in Critic Browser

- ? builder: => ? addedInBuilder:

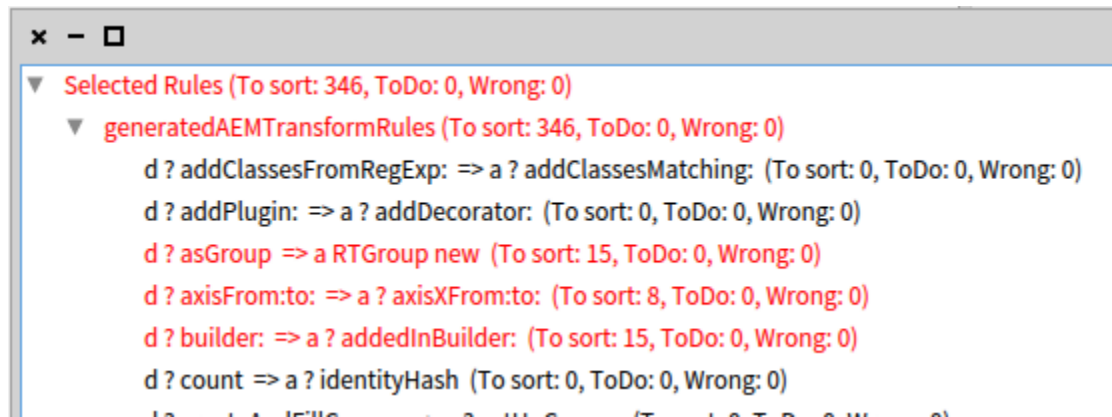
The screenshot shows the Critic Browser interface with the following components:

- Window Title:** d ? builder: => a ? addedInBuilder:
- Selected Rules (To sort: 706, ToDo: 0, Wrong: 0):**
 - generatedAEMTransformRules (To sort: 706, ToDo: 0, Wrong: 0)
 - d ? addClassesFromRegExp: => a ? addClassesMatching: (To sort: 0, ToDo: 0, Wrong: 0)
 - d ? addPlugin: => a ? addDecorator: (To sort: 0, ToDo: 0, Wrong: 0)
 - d ? asGroup => a RTGroup new (To sort: 7, ToDo: 0, Wrong: 0)
 - d ? axisFrom:to: => a ? axisXFrom:to: (To sort: 14, ToDo: 0, Wrong: 0)
 - d ? builder: => a ? addedInBuilder: (To sort: 17, ToDo: 0, Wrong: 0)**
- Rule List (Right Panel):**
 - RTCharterBuilder>>#resetAxisConfiguration (Roassal2-Builder-Charter)
 - RTDoubleBarBuilder>>#bottomValue:titled: (Roassal2-Builder-Charter)
 - RTDoubleBarBuilder>>#topValue:titled: (Roassal2-Builder-Charter)
 - RTDoubleGrapherBuilder>>#addRight: (Roassal2-Builder-Charter)
 - RTExploraBuilder>>#buildNode:parent: (Roassal2-Builder-Explora)
 - RTKiviatBuilder>>#axis: (Roassal2-Builder-Kiviat)
 - RTSunhurstBuilder>>#stratum: (Roassal2-Builder-Sunhurst)
- Buttons:** Run new configuration, Save Critics, Browse, Transform, Mark as wrong
- Rule Detail View (Bottom Panel):**

```
resetAxisConfiguration
axisConfiguration := RTAxisConfiguration new.
axisConfiguration addedInBuilder: self
axisConfiguration builder: self
```

Results in Critic Browser

- Load Roassal2 version 1.28 (13.05.16)
- 346 critics found by 20 of 57 rules



Future work

- Integrate the code into a CI service
- AEM still runs on Pharo 2

Summary

- APIs change over time
- APIEvolutionMiner cannot update changes automatically
- Our extension can do this