



Automating changes: Cha-Q search-and-replace

Presenter: Coen De Roover



The Ekeko/X Program Transformation Tool
Coen De Roover and Katsuro Inoue
14th IEEE International Working Conference on
Source Code Analysis and Manipulation (SCAM 2014)



Vrije
Universiteit
Brussel

Universiteit
Antwerpen

Problem: automating changes within a program

many

before

```
public class BreakStatement extends Statement {  
    @EntityProperty(value = SimpleName.class)  
    private EntityIdentifier label;  
  
    public EntityIdentifier getLabel() {  
        return label;  
    }  
  
    public void setLabel(EntityIdentifier label) {  
        this.label = label;  
    }  
}
```

value of annotation

after

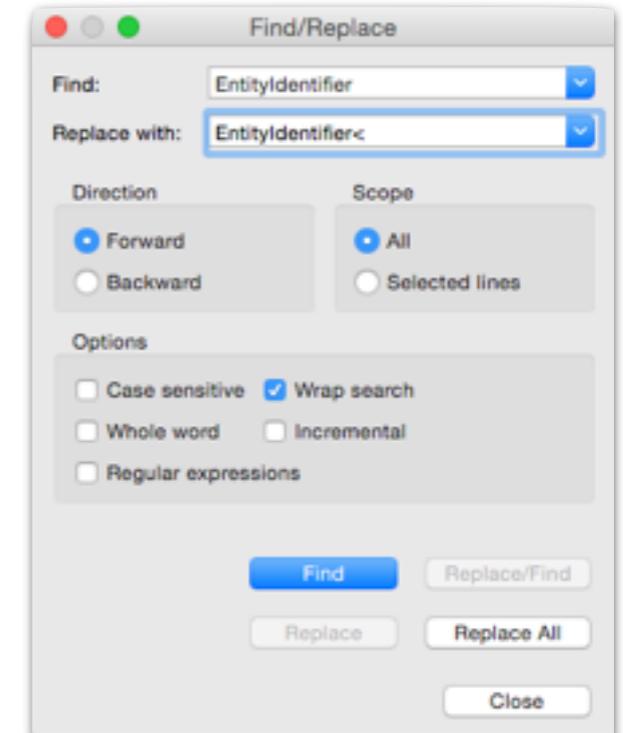
```
public class BreakStatement extends Statement {  
    @EntityProperty(value = SimpleName.class)  
    private EntityIdentifier<SimpleName> label;  
  
    public EntityIdentifier<SimpleName> getLabel() {  
        return label;  
    }  
  
    public void setLabel(EntityIdentifier<SimpleName> label) {  
        this.label = label;  
    }  
}
```

to be used as type parameter

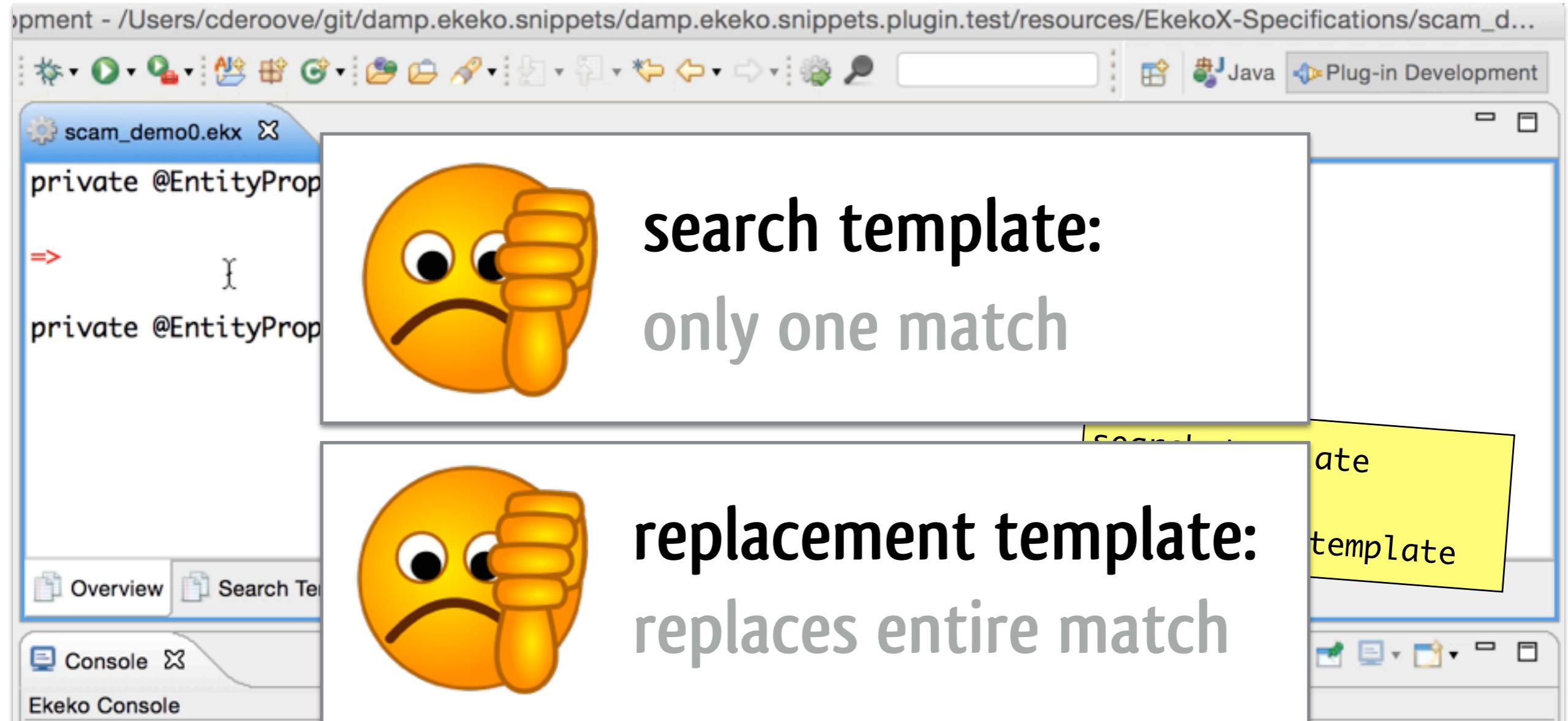
be.ac.chaq.model.ast.java
AbstractTypeDeclaration.java
Annotation.java
AnnotationTypeDeclaration.java
AnnotationTypeMemberDeclaration.java
AnonymousClassDeclaration.java
ArrayAccess.java
ArrayCreation.java
ArrayInitialization.java
ArrayType.java
AssertStatement.java
Assignment.java
ASTIdentifier.java
ASTNode.java
Block.java
BlockComment.java
BodyDeclaration.java
BooleanLiteral.java
BreakStatement.java
CastExpression.java
CatchClause.java
CharacterLiteral.java
ClassInstanceCreation.java
Comment.java
CompilationUnit.java
ConditionalExpression.java
ConstructorInvocation.java
ContinueStatement.java
DoStatement.java
EmptyStatement.java
EnhancedForStatement.java
EnumConstantDeclaration.java
EnumDeclaration.java
Expression.java
ExpressionStatement.java
FieldAccess.java
FieldDeclaration.java
ForStatement.java
IDocElement.java
IfStatement.java
ImportDeclaration.java
InfixExpression.java
Initializer.java
InstanceOfExpression.java
Javadoc.java
LabeledStatement.java
LineComment.java
MarkerAnnotation.java
MemberRef.java
MemberValuePair.java
MethodDeclaration.java
MethodInvocation.java

State of the art: two extremes

```
;relies on functions for program manipulation
;;can be invoked directly
;;following fixes all at once
(for [[field fieldtype annotype]
    ;;subjects of change
    (ekeko [?field ?fieldtype ?annotype]
        (fresh [?anno ?annotypeliteral]
            (field-declaration!incorrect ?field)
            (has :type ?field ?fieldtype)
            (fielddeclaration-annotation ?field ?anno)
            (annotation!ep-type!literal ?anno ?annotypeliteral)
            (has :type ?annotypeliteral ?annotype))))]
    ;actual changes
    (let [fieldtype-copy (copy-astnode fieldtype)
        annotype-copy (copy-astnode annotype)
        new-type (create-parameterized-type fieldtype-copy)]
        (change-property-node field :type new-type)
        (add-node (current-rewrite-for-cu (.getRoot field)) new-type :typeArguments annotype-copy 0)))
    ;;have scheduled changes
    ;;still need to apply them
    (apply-and-reset-rewrites))
```



Cha-Q search-and-replace



The screenshot shows a Java IDE interface with a search results window. The window has two entries, each featuring a large yellow emoji of a sad face with a raised fist. The first entry is labeled "search template: only one match" and the second is labeled "replacement template: replaces entire match". A yellow callout box points to the second entry with the text "ate template".

Development - /Users/cderoove/git/damp.ekeko.snippets/damp.ekeko.snippets.plugin.test/resources/EkekoX-Specifications/scam_d...

scam_demo0.ekx

```
private @EntityProp  
=>  
private @EntityProp
```

Overview Search Te...

Console X

Ekeko Console

Cleaning the Ekeko model.

Populating Ekeko project models for queried projects.

Adding a new project to the Ekeko model: P/TestCase-TypeParameters

Populating JavaProjectModel for: TestCase-TypeParameters

Gathered information from JDT compilation units in 30ms

Adding a new project to the Ekeko model: P/TestCase-TypeParameters

Populating JavaProjectModel for: TestCase-TypeParameters

Gathered information from JDT compilation units in 18ms

Updating a project in the model: P/TestCase-TypeParameters

Processing Java Project Delta: Changed ICompilationUnit

Populating Ekeko project models for queried projects.

Adding a new project to the Ekeko model: P/TestCase-TypeParameters

Populating JavaProjectModel for: TestCase-TypeParameters

Gathered information from JDT compilation units in 30ms

Adding a new project to the Ekeko model: P/TestCase-TypeParameters

Populating JavaProjectModel for: TestCase-TypeParameters

Gathered information from JDT compilation units in 18ms

Updating a project in the model: P/TestCase-TypeParameters

Processing Java Project Delta: Changed ICompilationUnit

Plug-in Development - /Users/cderooeve/git/damp.ekeko.snippets/damp.ekeko.snippets.plugin.test/resources/EkekoX-Specifica...d...

scam_demo0.ekx scam_demo1chaq.ekx

```
@EntityProperty(value=?annoType.class) private EntityIdentifier ...;  
=>  
private @EntityProperty(value=Expression.class) EntityIdentifier<Expression> array;
```



1/ template = code + wildcards & vars

2/ directives for control over matching

3/ directives for control over replacing

Query Stats

Table Columns Tree

?FieldDeclaration86202

△ private @EntityProperty(value=Expression.class) EntityIdentifier array;

Plug-in Development - /Users/cderooove/git/damp.ekeko.snippets/damp.ekeko.snippets.plugin.test/resources/EkekoX-Specifica... d...

scam_demo1chaq.ekx scam_demo2chaq.ekx

```
[@EntityProperty(value=?annoType.class) private]@match[set] EntityIdentifier ...;
=> [<code>]@[<directive>]
private @EntityProperty(value=Expression.class) EntityIdentifier<Expression> array;
```



1/ template = code + wildcards & vars

2/ directives for control over matching

3/ directives for control over replacing

Mark Results

Query Stats

?annoType	?FieldDeclaration1
Type	△ @EntityProp...
Expression	△ @EntityProp...
ArrayInitializer	△ @EntityProp...
Expression	△ @EntityProp...
ArrayType	△ @EntityProp...
Name	△ @EntityProp...

Table Columns Tree

Package X Plug-ins

- Chaqeko-test.ekt
- Chaqeko-test2.ekk
- Chaqeko-test3.ekx
- emptybody.ekt
- invokedby.ekt
- invokes.ekt
- m1.ekt
- m2.ekt
- match-set.ekt
- orimplicit.ekt
- orsimple.ekt
- overrides.ekt
- referredby.ekt
- regexp_oneormore.ekt
- regexp.ekt
- scam_demo0.ekx
- scam_demo1.ekx
- scam_demo1ch.ekx
- scam_demo2.ekx
- scam_demo2ch.ekx
- scam_demo3.ekx
- scam_demo3.ekx
- scam_demo3.ekx
- scm_matchedmodifiers.ekt
- wildcard.ekt

EkekoX-Specifications-DesignPatterns [dar]

Foobar

HWV10

ICFGTest

Java8InAction [Java8InAction master]

JavaTemplatesUnitTests

jEdit

JHotDraw51

juzu

LANSimulation_rev107

The screenshot shows the Eclipse Platform interface with the following components:

- Top Bar:** Shows various icons for file operations, search, and help.
- Toolbar:** Includes icons for plug-ins, search, and Java development.
- Left Sidebar:** Lists files and projects: `ore.ekt`, `x`, `x`, `aq.ekx`, `x`, `aqa.ekx`, `aqb.ekx`, `equivalentcomposition`, `x`, `ifiers.ekt`, and `ons-DesignPatterns [dan]`.
- Central Editor:** Displays code snippets for `scam_demo2chaqa.ekx` and `*scam_demo2chaqb.ekx`. The code includes annotations like `@EntityProperty`, `private`, `@match|set`, `[EntityIdentifier]@[(equals ?fieldType)]`, and `[EntityIdentifier<?annoType>]@[replace ?fieldType]`.
- Lightbulb Icon:** A large yellow lightbulb icon is positioned on the left side of the central editor area.
- Text Labels:** Three text labels are overlaid on the screen:
 - 1/ template = code + wildcards & vars**
 - 2/ directives for control over matching**
 - 3/ directives for control over replacing**
- Bottom Panel:** A "Query Stats" panel with a table showing data for `?FieldDeclaration1`, `?fieldType`, and `?annoType`. The table rows include:

?FieldDeclaration1	?fieldType	?annoType
△ @EntityProp...	EntityIdentifier	Type
△ @EntityProp...	EntityIdentifier	Expression
△ @EntityProp...	EntityIdentifier	ArrayInitializer
△ @EntityProp...	EntityIdentifier	Expression
△ @EntityProp...	EntityIdentifier	ArrayType
△ @EntityProp...	EntityIdentifier	Name

Plug-in Development - /Users/cderooeve/git/damp.ekeko.snippets/damp.ekeko.snippets.plugin.test/resources/EkekoX-Specification_demo3.ekx

Demo 3

The screenshot shows an IDE interface with two open files:

- scam_demo3.ekx**: An Ekeko X Specification file containing Java code snippets. A red arrow points from the word "return" in the code to a target icon in the editor's bottom panel.
- ArrayCreation.java**: A Java class definition.

The Java code includes imports for `java.util.List`, `be.ac.chaq.model.entity.EntityIdentifier`, `be.ac.chaq.model.entity.EntityListProperty`, and `be.ac.chaq.model.entity.EntityProperty`. It defines a class `ArrayCreation` extending `Expression`.

A large text overlay "automated changes!" is positioned in the center of the screen, with three red arrows pointing towards the target icon in the Ekeko X editor.

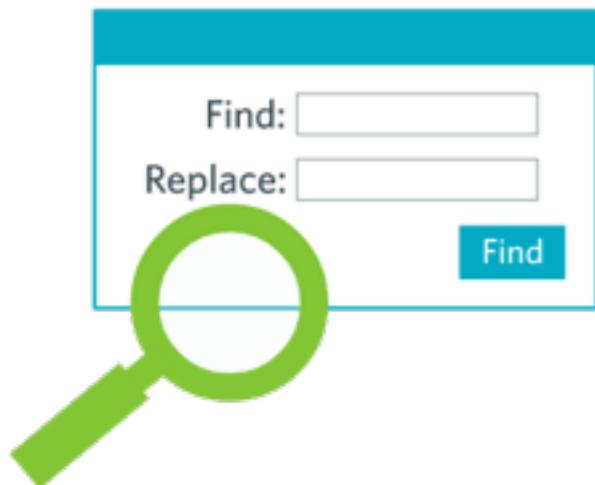
Bottom navigation bar:

- Overview
- Search Templates
- Console
- Ekeko Query Results

Research on automating changes

Cha-Q search-and-replace

we are excited



truly template-driven
match control and data flow variants

we are looking for cases



find and fix all
occurrences of
the same bug



evolve existing
code to newer
Java versions



update existing
clients to new
version of API



co-maintain
multiple system
variants