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

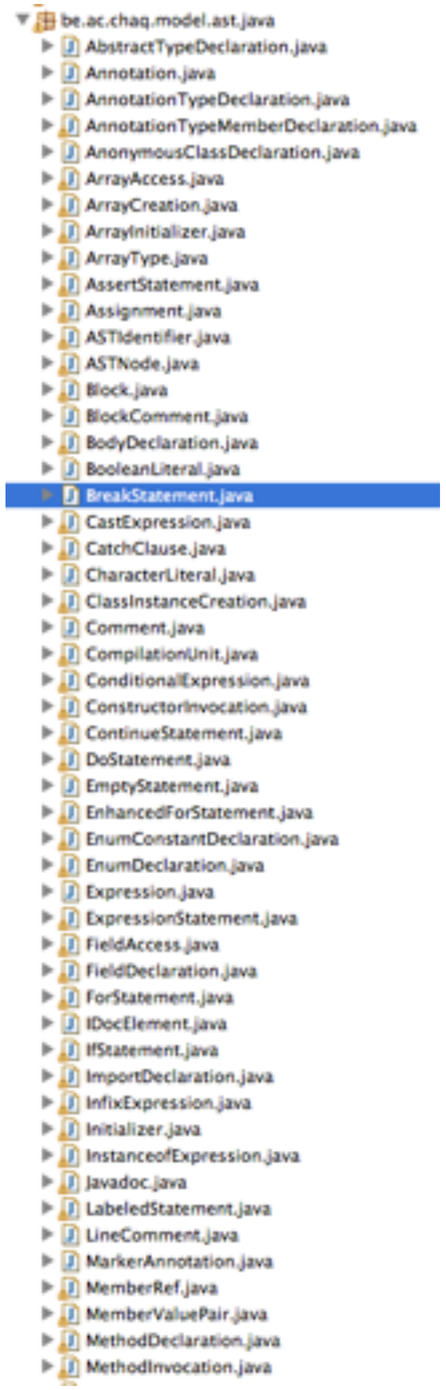
```
public class BreakStatement extends Statement {
    @EntityProperty(value = SimpleName.class)
    private EntityIdentifier label;

    public EntityIdentifier getLabel() {
        return label;
    }

    public void setLabel(EntityIdentifier label) {
        this.label = label;
    }
}
```

value of annotation

after

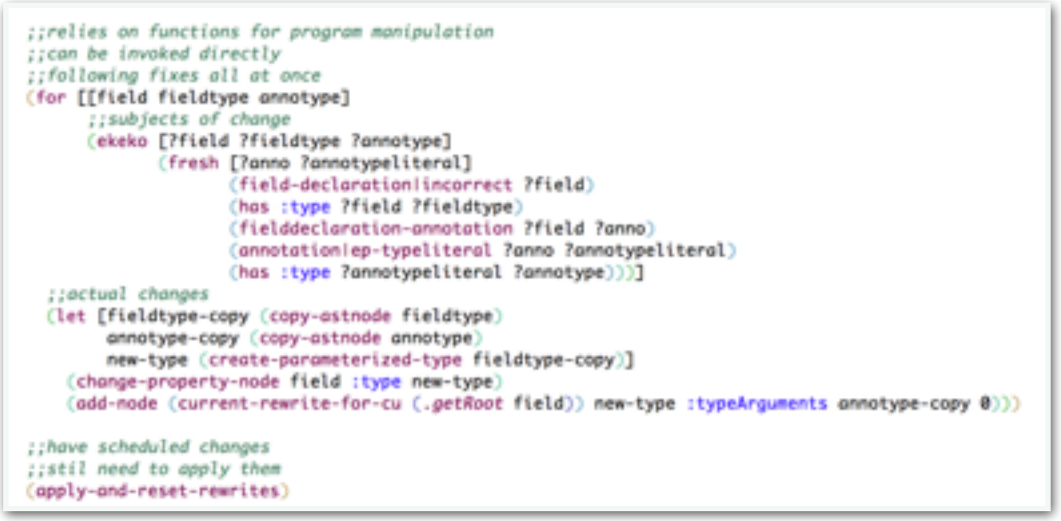
```
public class BreakStatement extends Statement {
    @EntityProperty(value = SimpleName.class)
    private EntityIdentifier<SimpleName> label;

    public EntityIdentifier<SimpleName> getLabel() {
        return label;
    }

    public void setLabel(EntityIdentifier<SimpleName> label) {
        this.label = label;
    }
}
```

to be used as type parameter

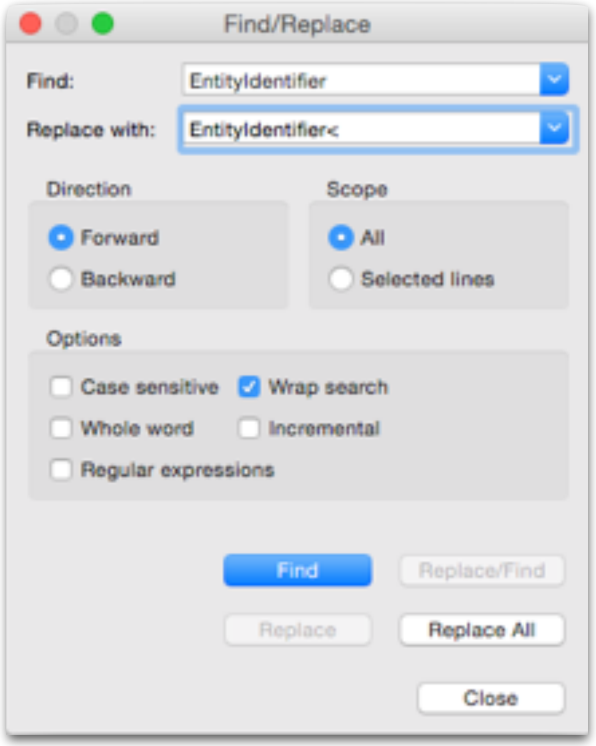


```
▼ be.ac.chaq.model.ast.java
    ▶ AbstractTypeDeclaration.java
    ▶ Annotation.java
    ▶ AnnotationTypeDeclaration.java
    ▶ AnnotationTypeMemberDeclaration.java
    ▶ AnonymousClassDeclaration.java
    ▶ ArrayAccess.java
    ▶ ArrayCreation.java
    ▶ ArrayInitializer.java
    ▶ ArrayType.java
    ▶ 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-typeliteral ?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
;;stil need to apply them
(apply-and-reset-rewrites)
```



Find/Replace

Find: EntityIdentifier

Replace with: EntityIdentifier<

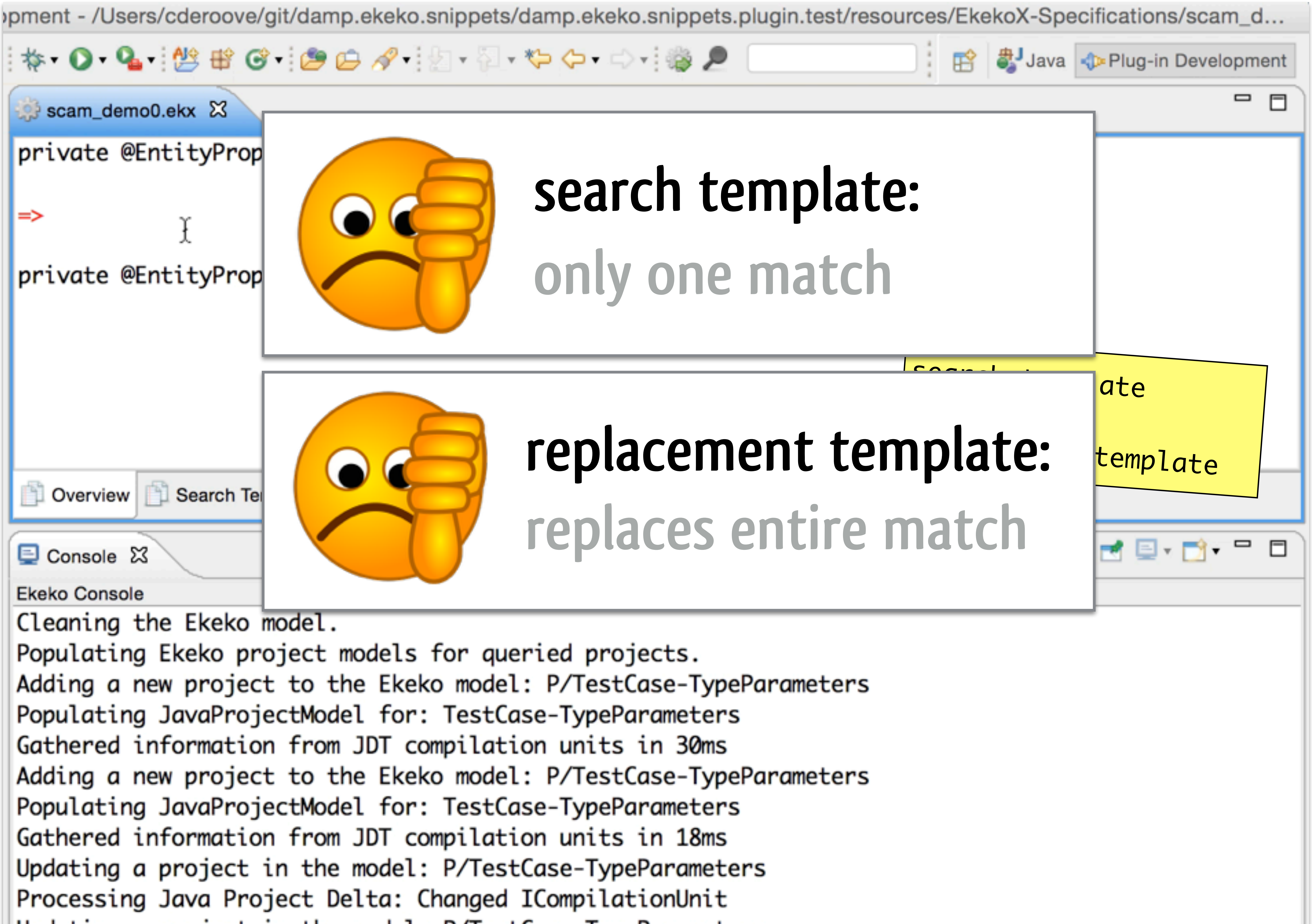
Direction: ● Forward ○ Backward

Scope: ● All ○ Selected lines

Options: □ Case sensitive ☑ Wrap search □ Whole word □ Incremental □ Regular expressions

Cha-Q search-and-replace

Demo 1


pment - /Users/cderoove/git/damp.ekeko.snippets/damp.ekeko.snippets.plugin.test/resources/EkekoX-Specifications/scam_d...
Java
Plug-in Development
scam_demo0.ekx
Overview
Search Te
ate
template
Console
Ekeko Console

```
private @EntityProp

=>

private @EntityProp
```

search template:

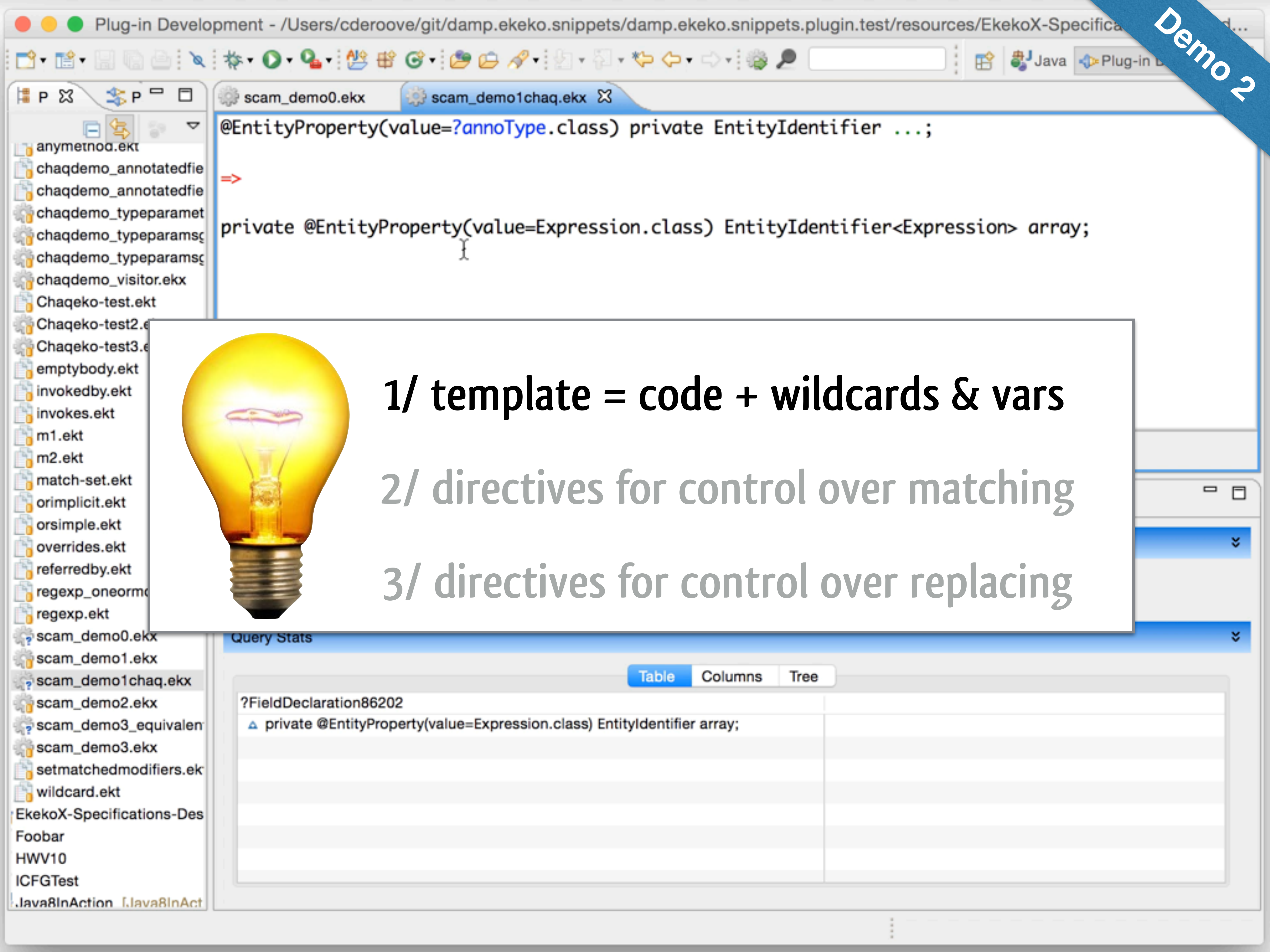
only one match

replacement template:

replaces entire match

```
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
```

Demo 2



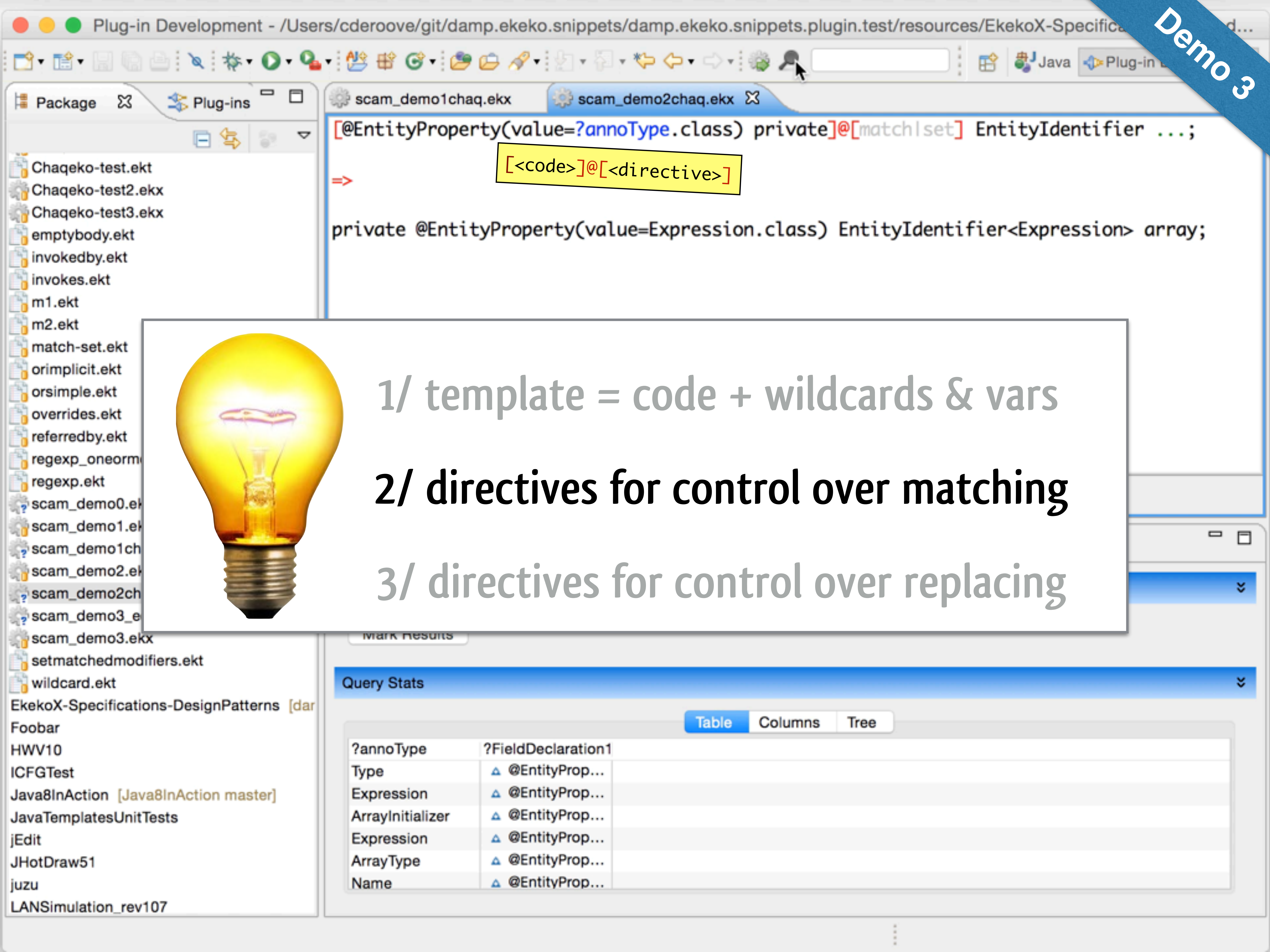
```
@EntityProperty(value=?annoType.class) private EntityIdentifier ...;

=>

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

?FieldDeclaration86202
private @EntityProperty(value=Expression.class) EntityIdentifier array;



```
[@EntityProperty(value=?annoType.class) private]@[match|set] EntityIdentifier ...;

=>

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

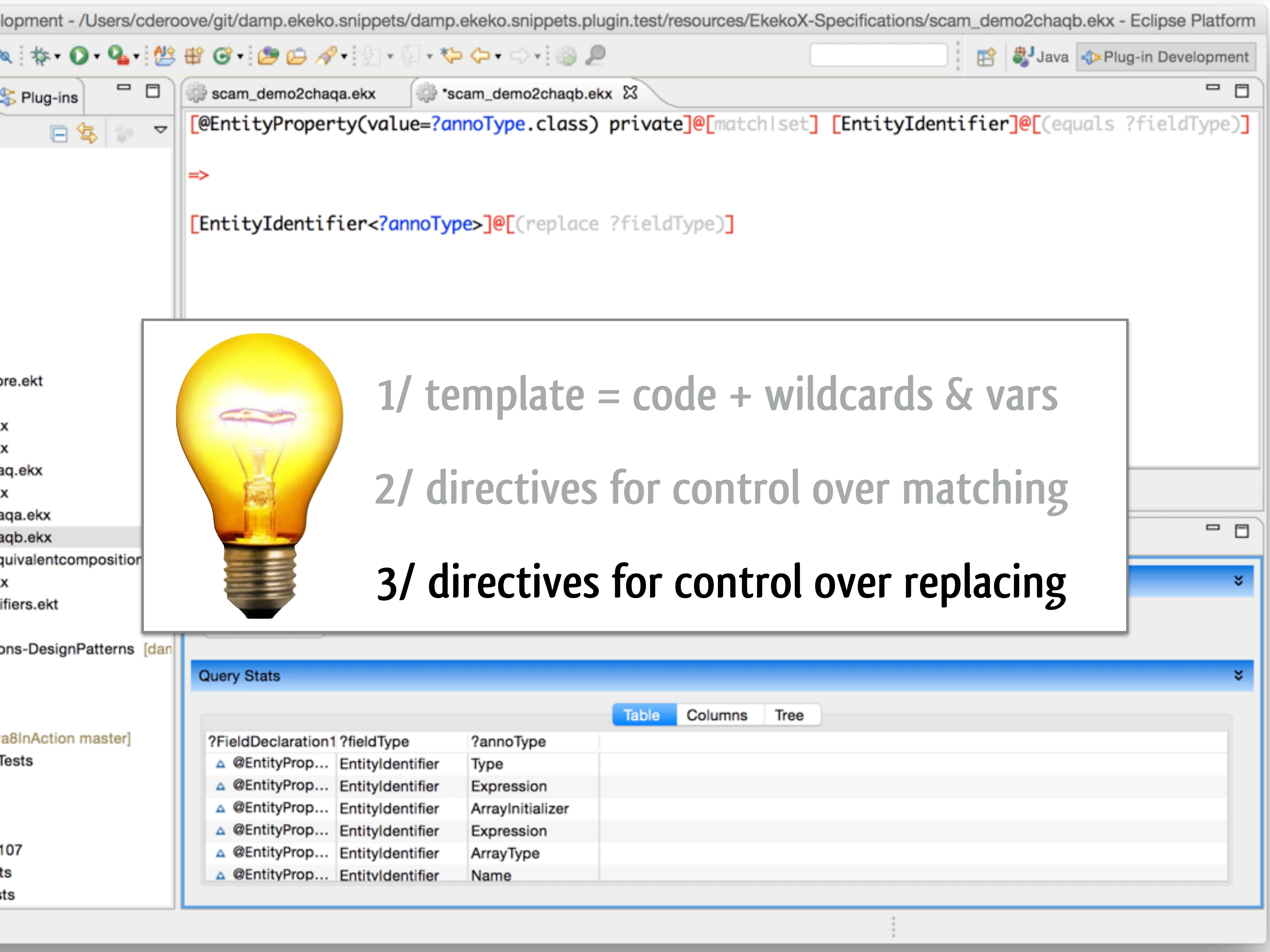
[<code>]@[<directive>]

1/ template = code + wildcards & vars

2/ directives for control over matching

3/ directives for control over replacing

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



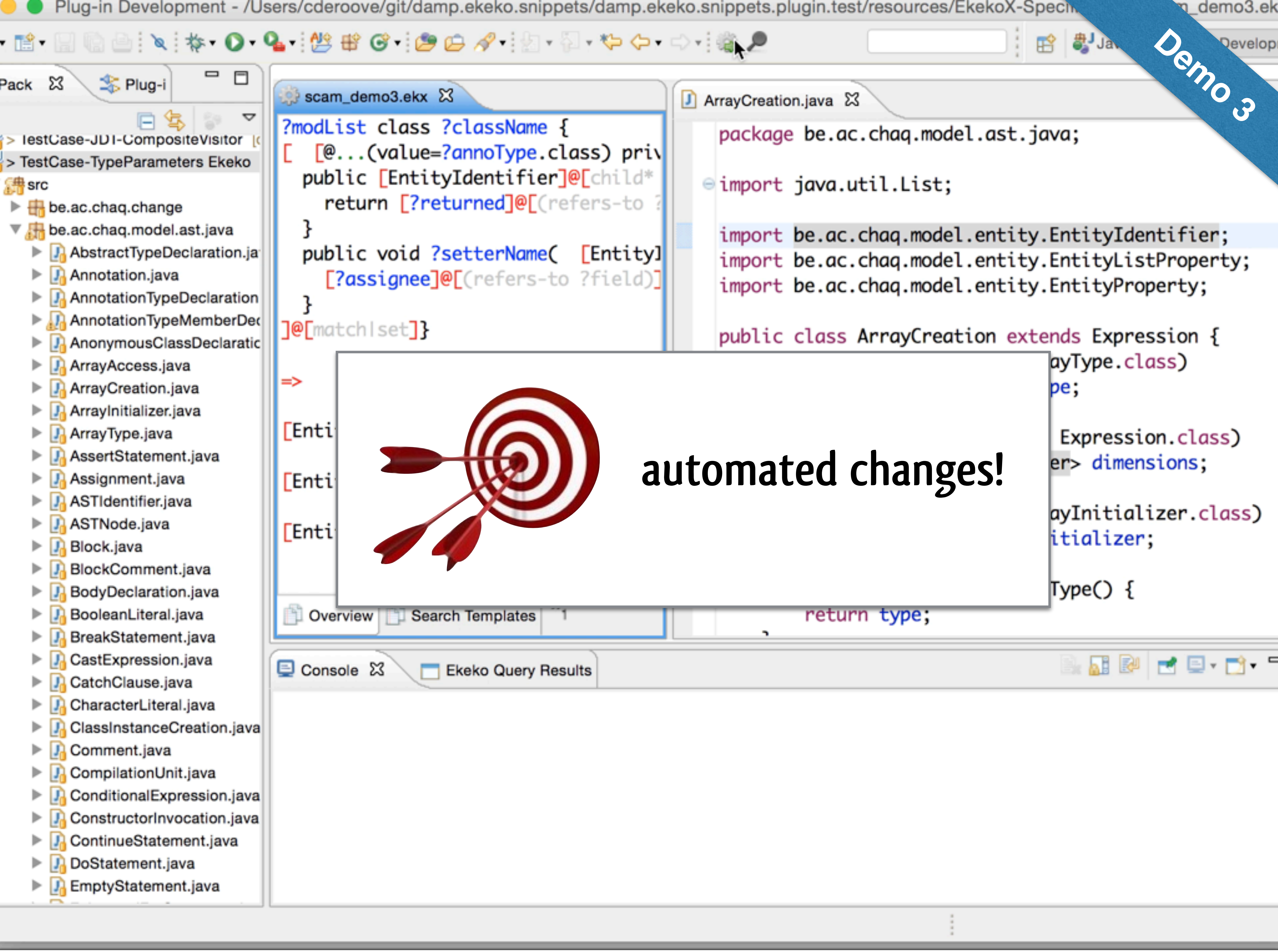
```
[@EntityProperty(value=?annoType.class) private]@[match|set] [EntityIdentifier]@[(equals ?fieldType)]

=>

[EntityIdentifier<?annoType>]@[(replace ?fieldType)]
```

1/ template = code + wildcards & vars

2/ directives for control over matching

3/ directives for control over replacing

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



```
?modList class ?className {
[  [@...(value=?annoType.class) priv
  public [EntityIdentifier]@[child*
    return [?returned]@[(refers-to ?
  }
  public void ?setterName(  [Entity]
    [?assignee]@[(refers-to ?field)]
  }
]@[match|set]}

=>

[Enti

[Enti

[Enti
```

```
package be.ac.chaq.model.ast.java;

import java.util.List;

import be.ac.chaq.model.entity.EntityIdentifier;
import be.ac.chaq.model.entity.EntityListProperty;
import be.ac.chaq.model.entity.EntityProperty;

public class ArrayCreation extends Expression {
                                        ayType.class)
                                        pe;

                                         Expression.class)
                                        er> dimensions;

                                        ayInitializer.class)
                                        itializer;

                                        Type() {
        return type;
```

automated changes!

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