



Automating Systematic Edits with Cha-Q search-and-replace

Presenters: Tim Molderez, Nick Nuyts

Cha-Q



Federale
Overheidsdienst
FINANCIEN

FOD Financiën pilot study

- Calcul IPP project: Simulation personal taxes

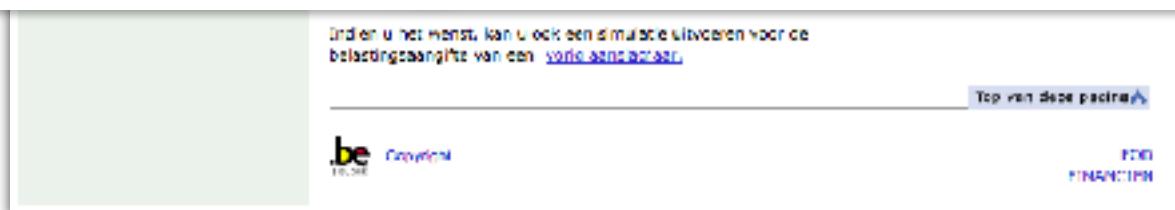
The screenshot shows the homepage of the Tax-Calc website. At the top, there are language links (NL, FR, DE) and a banner stating 'Al 3 miljoen gebruikers... En jij?' with a link to 'TAX-Calc www.taxonweb.be'. The page features a large image of a computer monitor displaying a family photo, with the text 'Welkom bij Tax-Calc, de berekeningsmodule van de Personenbelasting'. Below this, there's information about the tax year 2015 and a 'Start Tax-Calc' button. A sidebar on the right contains links to 'MEER OVER DIT Onderwerp?' and other related topics. The footer includes the 'be Overheid' logo and 'FOD FINANCIËN'.

FOD Financiën pilot study

- Calcul IPP project: Simulation personal taxes



- 2987 Java source files; 274883 LOC
- Examined development activity of Sep. 2014 - May 2016
(751 commits)



Introducing fields + usages

```
public class ReportCalcSummary implements Serializable {  
    ...  
    + private Long impotReferRegulPosVla;  
    + public Long getImpotReferRegulPosVla() {  
        +     return impotReferRegulPosVla;}  
    + public void setImpotReferRegulPosVla(Long impotReferRegulPosVla) {  
        +     this.impotReferRegulPosVla = impotReferRegulPosVla;}  
    ...  
    +     .append("impotReferRegulPosVla = ").append(this.impotReferRegulPosVla).append(TAB)  
    ...  
    +     dto.setImpotReferRegulPosVla(new Long(0));  
    ...  
    +     if (g8653 > 0) {dto.setImpotReferRegulPosVla(a(dto.getImpotReferRegulPosVla(), g8653));}  
    ...  
    +     vo.setImpotReferRegulPosVla(dto.getImpotReferRegulPosVla());  
    ...
```

Introducing fields + usages

```
public class ReportCalcSummary {  
    ...  
    + private Long impotReferRegulPosVla;  
    + public Long getImpotReferRegulPosVla() {  
        +     return impotReferRegulPosVla;  
    + public void setImpotReferRegulPosVla(Long impotReferRegulPosVla) {  
        +     this.impotReferRegulPosVla = impotReferRegulPosVla;  
    ...  
    +     .append("impotReferRegulPosVla = ").append(this.impotReferRegulPosVla).append(TAB)  
    ...  
    +     dto.setImpotReferRegulPosVla(new Long(0));  
    ...  
    +     if (g8653 > 0) {dto.setImpotReferRegulPosVla(a(dto.getImpotReferRegulPosVla(), g8653));}  
    ...  
    +     vo.setImpotReferRegulPosVla(dto.getImpotReferRegulPosVla());  
    ...  
}
```

A new field

Introducing fields + usages

```
public class ReportCalcSummary {  
    ...  
    + private Long impotReferRegulPosVla;  
    + public Long getImpotReferRegulPosVla() {  
        +     return impotReferRegulPosVla;  
    }  
    + public void setImpotReferRegulPosVla(Long impotReferRegulPosVla) {  
        +     this.impotReferRegulPosVla = impotReferRegulPosVla;  
    }  
    ...  
    +     .append("impotReferRegulPosVla = ").append(this.impotReferRegulPosVla).append(TAB)  
    ...  
    +     dto.setImpotReferRegulPosVla(new Long(0));  
    ...  
    +     if (g8653 > 0) {dto.setImpotReferRegulPosVla(a(dto.getImpotReferRegulPosVla(), g8653));}  
    ...  
    +     vo.setImpotReferRegulPosVla(dto.getImpotReferRegulPosVla());  
    ...
```

A new field

Its getter

Introducing fields + usages

```
public class ReportCalcSummary {  
    ...  
    + private Long impotReferRegulPosVla;  
    + public Long getImpotReferRegulPosVla() {  
        +     return impotReferRegulPosVla;  
    }  
    + public void setImpotReferRegulPosVla(Long impotReferRegulPosVla) {  
        +     this.impotReferRegulPosVla = impotReferRegulPosVla;  
    }  
    ...  
    +     .append("impotReferRegulPosVla = ").append(this.impotReferRegulPosVla).append(TAB)  
    ...  
    +     dto.setImpotReferRegulPosVla(new Long(0));  
    ...  
    +     if (g8653 > 0) {dto.setImpotReferRegulPosVla(a(dto.getImpotReferRegulPosVla(), g8653));}  
    ...  
    +     vo.setImpotReferRegulPosVla(dto.getImpotReferRegulPosVla());  
    ...
```

A new field

Its getter

Its setter

Introducing fields + usages

```
public class ReportCalcSummary {
```

```
...
```

```
+ private Long impotReferRegulPosVla;
```

```
+ public Long getImpotReferRegulPosVla() {  
+     return impotReferRegulPosVla;}
```

```
+ public void setImpotReferRegulPosVla(Long impotReferRegulPosVla) {  
+     this.impotReferRegulPosVla = impotReferRegulPosVla;}
```

A new field

```
    ...
```

Its getter

Its setter

```
+     .append("impotReferRegulPosVla = ").append(this.impotReferRegulPosVla).append(TAB)
```

```
...
```

```
+     dto.setImpotReferRegulPosVla(new Long(0));
```

```
...
```

```
+     if (g8653 > 0) {dto.setImpotReferRegulPosVla(a(dto.getImpotReferRegulPosVla(), g8653));}
```

```
...
```

```
+     vo.setImpotReferRegulPosVla(dto.getImpotReferRegulPosVla());
```

```
...
```

Systematic edits overview

Type	# sys. edits	avg. instances
<i>Adding similar call statements</i>	38	53
<i>Adding/removing similar fields</i>	33	96.26
<i>Adding similar if cases</i>	13	21.92
<i>Adding similar test cases</i>	12	51.67
<i>Generating similar classes</i>	5	8.2
<i>Formatting changes</i>	5	63.6
<i>Miscellaneous</i>	14	33.5
Total	120	

Systematic edits overview

Type	# sys. edits	avg. instances
Adding similar call statements	29	52

- ~16% of commits containing systematic edits
- ~7% of commits are systematic edits related to tax law changes → Likely to reoccur

Formatting changes	5	63.6
Miscellaneous	14	33.5
Total	120	

Cha-Q search-and-replace

Cha-Q search-and-replace

- Systematic edits are unavoidable

Cha-Q search-and-replace

- Systematic edits are unavoidable
- Repetitive and error-prone if done manually

Cha-Q search-and-replace

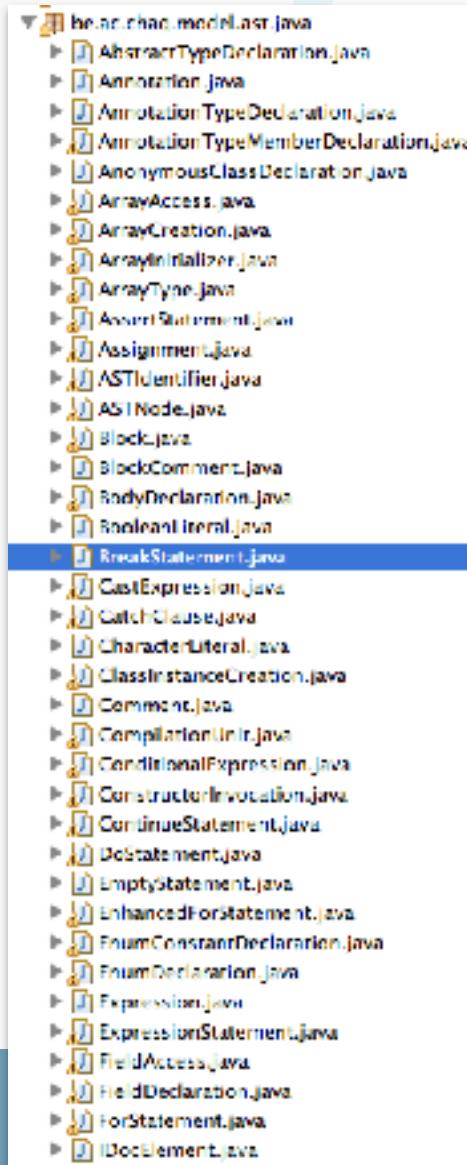
- Systematic edits are unavoidable
- Repetitive and error-prone if done manually
- **Automate it!**
 - Only describe the similar change once
 - Tool can apply it to all locations

Example systematic edit

```
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;  
    }  
}
```

Example systematic edit

```
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;  
    }  
}
```



A file tree diagram showing a large number of Java files under the package 'be.arachne.model.ast'. The files listed include: be.arachne.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, ASNode.java, Block.java, BlockComment.java, BodyDeclaration.java, BooleanLiteral.java, BreakStatement.java (which is highlighted in blue), CastExpression.java, CatchClause.java, CharacterLiteral.java, ClassInstanceCreation.java, Comment.java, CompilationUnit.java, ConditionalExpression.java, ConstructorInvocation.java, ContinueStatement.java, DoStatement.java, EmptyStatement.java, EnhancedForStatement.java, EnumConstraintDeclaration.java, EnumDeclaration.java, Expression.java, ExpressionStatement.java, FieldAccess.java, FieldDeclaration.java, ForStatement.java, and Document.java.

- be.arachne.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
- ASNode.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
- EnumConstraintDeclaration.java
- EnumDeclaration.java
- Expression.java
- ExpressionStatement.java
- FieldAccess.java
- FieldDeclaration.java
- ForStatement.java
- Document.java

Example systematic edit

A field of type EntityIdentifier

```
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;  
    }  
}
```

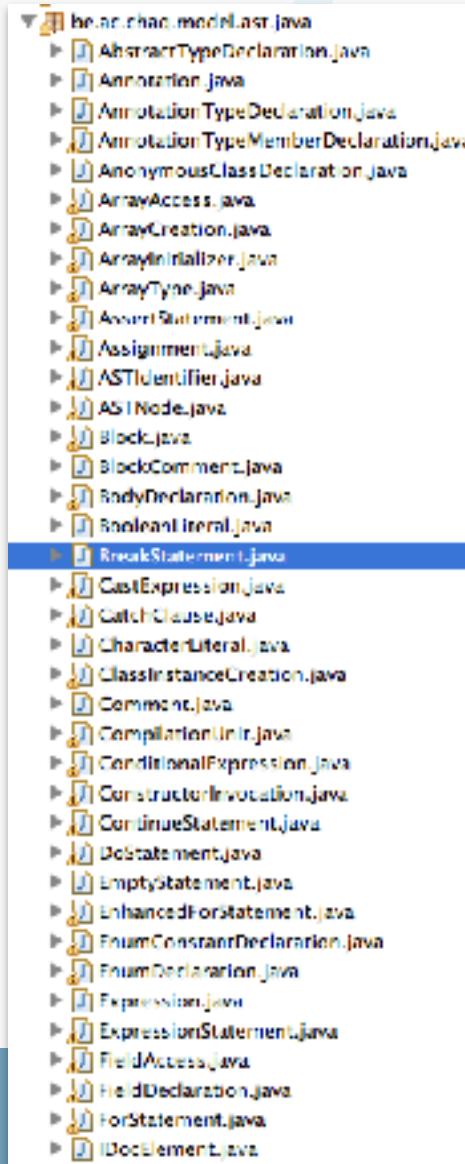
↳ breakStatement.java
↳ AbstractTypeDeclaration.java
↳ Annotation.java
↳ AnnotationTypeDeclaration.java
↳ AnnotationTypeMemberDeclaration.java
↳ AnonymousClassDeclaration.java
↳ ArrayAccess.java
↳ ArrayCreation.java
↳ ArrayInitializer.java
↳ ArrayType.java
↳ AssertStatement.java
↳ Assignment.java
↳ ASTIdentifier.java
↳ ASINode.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
↳ EnumConstraintDeclaration.java
↳ EnumDeclaration.java
↳ Expression.java
↳ ExpressionStatement.java
↳ FieldAccess.java
↳ FieldDeclaration.java
↳ ForStatement.java
↳ Document.java

Ex

An identifier of an entity .. but what kind of entity?

A field of type EntityIdentifier

```
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;  
    }  
}
```



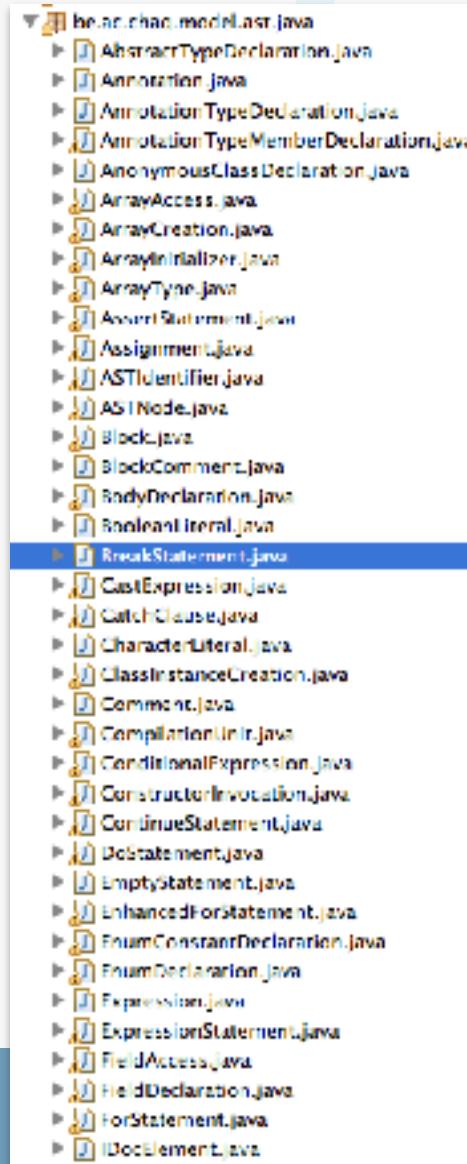
↳ be.arachne.model.ast.java
↳ AbstractTypeDeclaration.java
↳ Annotation.java
↳ AnnotationTypeDeclaration.java
↳ AnnotationTypeMemberDeclaration.java
↳ AnonymousClassDeclaration.java
↳ ArrayAccess.java
↳ ArrayCreation.java
↳ ArrayInitializer.java
↳ ArrayType.java
↳ AssertStatement.java
↳ Assignment.java
↳ ASTIdentifier.java
↳ ASINode.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
↳ EnumConstraintDeclaration.java
↳ EnumDeclaration.java
↳ Expression.java
↳ ExpressionStatement.java
↳ FieldAccess.java
↳ FieldDeclaration.java
↳ ForStatement.java
↳ Document.java

Ex

An identifier of an entity .. but what kind of entity?

A field of type EntityIdentifier

```
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;  
    }  
}
```



↳ be.arachne.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
↳ ASINode.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
↳ EnumConstraintDeclaration.java
↳ EnumDeclaration.java
↳ Expression.java
↳ ExpressionStatement.java
↳ FieldAccess.java
↳ FieldDeclaration.java
↳ ForStatement.java
↳ Document.java

Ex

An identifier of an entity .. but what kind of entity?

A field of type EntityIdentifier

```
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;  
    }  
}  
  
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;  
    }  
}
```



↳ breakStatement.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
↳ EnumConstraintDeclaration.java
↳ EnumDeclaration.java
↳ Expression.java
↳ ExpressionStatement.java
↳ FieldAccess.java
↳ FieldDeclaration.java
↳ ForStatement.java
↳ Document.java

Ex

An identifier of an entity .. but what kind of entity?

A field of type EntityIdentifier

```
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;  
    }  
}
```

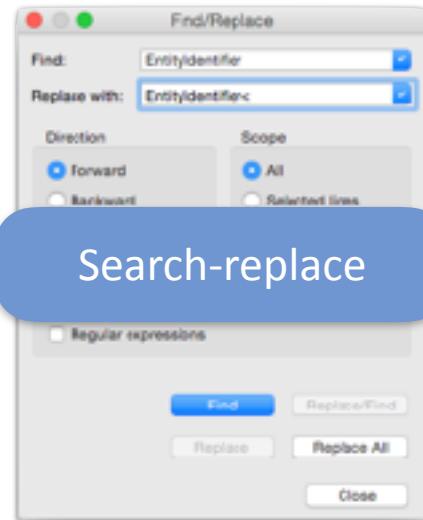
Used as type parameter

```
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;  
    }  
}
```



↳ breakStatement.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
↳ EnumConstraintDeclaration.java
↳ EnumDeclaration.java
↳ Expression.java
↳ ExpressionStatement.java
↳ FieldAccess.java
↳ FieldDeclaration.java
↳ ForStatement.java
↳ Document.java

State of the art: 2 extremes



Search-replace

```
;entities or functions for program manipulation
;can be invoked directly
;following fixes all at once
(for [(field [fieldtype annotype]
           ;subjects of change
           (execo [(field [fieldtype annotype]
                         (fresh [funnie [annotype literal]])]
                   ;field annotations for execution, fix it!)]))]

;executed
;Get [rte]
;    annot;ps-copy (copy-astnode annotype)
;    new-type (create-parameterized-type fieldtype-copy)
;    Change-property-node field-type new-type
;    Old-node (current-rewrite-for-cu (getRoot field)) set-type typeArguments annotype-copy @)))))

;have scheduled changes
;still need to apply them
;apply and re-reschedules
```

Transformation languages



Cha-Q search-and-replace



Demo 1

The screenshot shows a Java IDE interface with the following details:

- Title Bar:** Java - /Users/soft/Documents/Github/damp.ekeko.snippets/damp.ekeko.snippets.plugin.test/resources/EkekoX-Specifi...
- Toolbar:** Standard Java IDE toolbar with icons for file operations, search, and navigation.
- Project Explorer:** Shows a project named "chaq-demo1.ekx".
- Code Editor:** Displays two lines of Java code:

```
private @EntityProperty(value=Expression.class) EntityIdentifier array;  
{  
private @EntityProperty(value=Expression.class) EntityIdentifier<Expression> array;
```
- Search Bar:** Located at the top of the code editor.
- Toolbars:** Overview, Search Templates, Replacement Templates.
- Bottom Navigation:** Problems, Javadoc, Declaration, Console, Progress, Call Hierarchy.
- Ekeko Console:** Shows log output:

```
Adding a new project to the Ekeko model: P/TestCase-TypeParameters  
Populating JavcProjectModel for: TestCase-TypeParameters  
Gathered information from JDT compilation units in 36ms
```

Cha-Q search-and-replace



Demo 1

The screenshot shows a Java code editor window with the following details:

- Title Bar:** Java - /Users/soft/Documents/Github/damp.ekeko.snippets/damp.ekeko.snippets.plugin.test/resources/EkekoX-Specifi...
- Toolbar:** Standard Java IDE toolbar with icons for file operations, search, and navigation.
- Project Explorer:** Shows a project named "chac-demo1.ekx".
- Code Editor:** Displays two lines of Java code:

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

A pink rectangular box highlights the first line of code.

An orange callout bubble labeled "Search template" points to the first line of code.

```
private @EntityProperty(value=Expression.class) EntityIdentifier<Expression> array;
```
- Search Templates Tab:** A tab bar at the bottom of the code editor includes "Overview", "Search Templates" (which is selected), and "Replacement Templates".
- Console Tab:** A tab bar at the bottom of the window includes "Problems", "Javadoc", "Declaration", "Console" (selected), "Progress", and "Call Hierarchy".
- Ekeko Console:** A text area showing log output:

```
Ekeko Console
Adding a new project to the Ekeko model: P/TestCase-TypeParameters
Populating JavcProjectModel for: TestCase-TypeParameters
Gathered information from JDT compilation units in 36ms
```

Cha-Q search-and-replace



Demo 1

The screenshot shows the Cha-Q search-and-replace feature integrated into an IDE. The code editor displays two occurrences of a template:

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

This template is highlighted with a pink rectangle and has an orange callout labeled "Search template".

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

This template is also highlighted with a pink rectangle and has an orange callout labeled "Replacement template".

The IDE interface includes:

- Toolbar with various icons.
- Project Explorer on the left showing a file named "chac-demo1.ekx".
- Search Templates tab in the bottom navigation bar.
- Replacement Templates tab in the bottom navigation bar.
- Ekeko Console tab in the bottom navigation bar.
- Ekeko Console output:
 - Adding a new project to the Ekeko model: P/TestCase-TypeParameters
 - Populating JavcProjectModel for: TestCase-TypeParameters
 - Gathered information from JDT compilation units in 36ms

Cha-Q search-and-replace



Demo 1

The screenshot shows a Java IDE interface with the following details:

- Title Bar:** Java - /Users/soft/Documents/Github/damp.ekeko.snippets/damp.ekeko.snippets.plugin.test/resources/EkekoX-Specifi...
- Toolbar:** Standard Java IDE toolbar with icons for file operations, search, and navigation.
- Project Explorer:** Shows a project named "chaq-demo1.ekx".
- Code Editor:** Displays two lines of Java code:

```
private @EntityProperty(value=Expression.class) EntityIdentifier array;  
{  
private @EntityProperty(value=Expression.class) EntityIdentifier<Expression> array;
```
- Search Bar:** Located above the code editor, showing the placeholder "Search for...".
- Toolbars:** Overview, Search Templates, Replacement Templates.
- Bottom Navigation:** Problems, Javadoc, Declaration, Console, Progress, Call Hierarchy.
- Ekeko Console:** Shows log output:

```
Adding a new project to the Ekeko model: P/TestCase-TypeParameters  
Populating JavcProjectModel for: TestCase-TypeParameters  
Gathered information from JDT compilation units in 36ms
```



Demo 2

chaq-demo2.ekx

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

=>
}

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

Overview

Search Templates

Replacement Templates

Problems

@ Javadoc

Declaration

Console

Progress

Call Hierarchy

Ekeko Console

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

Populating JavaProjectModel for: TestCase-TypeParameters

Gathered information from JDT compilation units in 36ms

Placeholder variable



Demo 2

chaq-demo2.ekx

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

Overview

Search Templates

Replacement Templates

Problems

Javadoc

Declaration

Console

Progress

Call Hierarchy

Ekeko Console

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

Populating JavaProjectModel for: TestCase-TypeParameters

Gathered information from JDT compilation units in 36ms

 Demo 2

Placeholder variable

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

Wildcard

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

Overview

Search Templates

Replacement Templates

Problems

@ Javadoc

Declaration

Console

Progress

Call Hierarchy

Ekeko Console

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

Populating JavaProjectModel for: TestCase-TypeParameters

Gathered information from JDT compilation units in 36ms

Java - /Users/soft/Documents/Github/damp.ekeko.snippets/damp.ekeko.snippets.plugin.test/resources/EkekoV_Specifi...

Quick Access

Demo 2

chaq-demo2.ekx

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

Overview Search Templates Replacement Templates

Problems Javadoc Declaration Console Progress Call Hierarchy

Ekeko Console

Adding a new project to the Ekeko model: P/TestCase-TypeParameters
Populating JavaProjectModel for: TestCase-TypeParameters
Gathered information from JDT compilation units in 36ms



Demo 2

chaq-demo2.ekx

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

Overview

Problems

Ekoyo Console
Adding a new
Populating Ja
Gathered info

ArrayAccess.java X

```
package be.ac.chaq.model.ast.java;  
+ import y.EntityIdentifier;[]  
  
public class ArrayAccess extends Expression {  
    private @EntityProperty(value = Expression.class)  
    EntityIdentifier array;
```

This field was not found!

The screenshot shows the Eclipse IDE interface with the following details:

- Title Bar:** Shows the title "Java - /Users/soft/Documents/Github/damp.ekeko.snippets/damp.ekeko.snippets.plugin.test/resources/EkekoX-Specifi...".
- Toolbar:** Standard Eclipse toolbar icons.
- Project Explorer:** Shows a project named "chaq-demo3.ekx".
- Code Editor:** Displays Java code with annotations. The code includes:

```
[@EntityProperty(value=?annoType.class) private]@match|set EntityIdentifier ...;
}
private @EntityProperty(value=Expression.class) EntityIdentifier<Expression> array;
```
- Toolbars:** "Overview", "Search Templates", and "Replacement Templates" are visible at the bottom of the editor area.
- Bottom Navigation:** Includes tabs for "Problems", "Javadoc", "Declaration", "Console" (selected), "Progress", and "Call Hierarchy".
- Ekeko Console:** A separate window titled "Ekeko Console" showing log output:

```
Adding a new project to the Ekeko model: P/TestCase-TypeParameters
Populating JavaProjectModel for: TestCase-TypeParameters
Gathered information from JDT compilation units in 36ms
```
- Right Sidebar:** Shows icons for "File", "New", "Open", "Save", "Import", "Export", "Properties", and "Help".

Demo 3

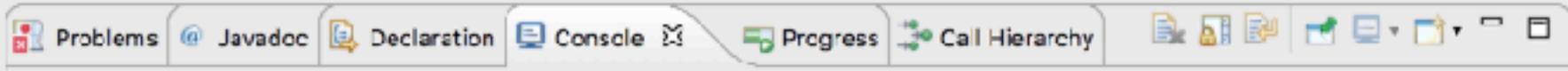


Demo 3

The screenshot shows the Eclipse IDE interface with the following details:

- Project:** chaq-demo3.ekx
- Code Snippet:** A Java code fragment is shown:

```
[@EntityProperty(value=?annoType.class) private]@[match|set] EntityIdentifier ...;
```
- Completion Suggestion:** The word "match" is highlighted with a pink rectangle.
- Tooltips:** An orange callout bubble points to the "match" suggestion with the text "Directive, to add more control to searches".
- Toolbar:** Standard Eclipse toolbar icons are visible at the top.
- Bottom Bar:** Includes tabs for Overview, Search Templates, and Replacement Templates.



Ekeko Console

Adding a new project to the Ekeko model: P/TestCase-TypeParameters
Populating JavaProjectModel for: TestCase-TypeParameters
Gathered information from JDT compilation units in 36ms

Java - /Users/soft/Documents/Github/damp.ekeko.snippets/damp.ekeko.snippets.plugin.test/resources/EkekoX-Specifi...

Quick Access

Demo 3

chaq-demo3.ekx

```
@EntityProperty[value=?annoType.class] private EntityIdentifier ...;
```

[match|set]

Attached to this piece of code

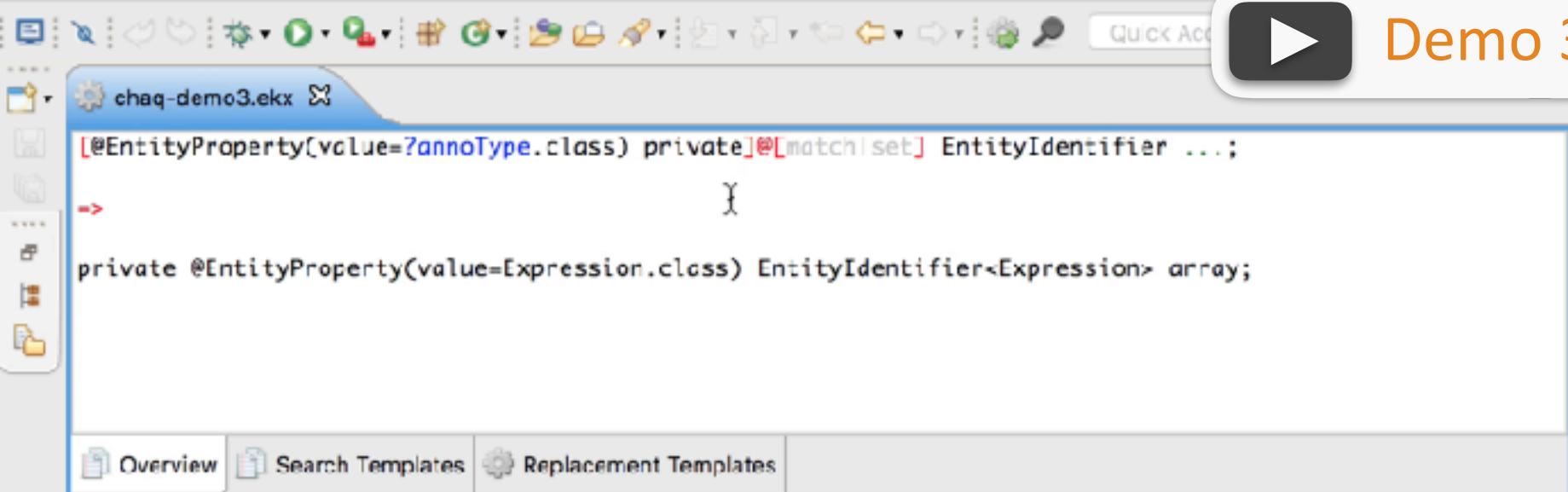
Directive, to add more control to searches

Overview Search Templates Replacement Templates

Problems Javadoc Declaration Console Progress Call Hierarchy

Ekeko Console

Adding a new project to the Ekeko model: P/TestCase-TypeParameters
Populating JavaProjectModel for: TestCase-TypeParameters
Gathered information from JDT compilation units in 36ms

 Demo 3

The screenshot shows the Eclipse IDE interface. The top bar includes standard icons for file operations, search, and help, along with a "Quick Access" button. A large video player icon with a play button is overlaid on the top right. The main workspace displays Java code in a class named "chaq-demo3.ekx". The code includes annotations like `@EntityProperty` and `EntityIdentifier<Expression>`. Below the code editor are three tabs: "Overview", "Search Templates", and "Replacement Templates", with "Overview" currently selected. The bottom part of the interface features a toolbar with various icons for navigating and managing projects, and a "Ekeko Console" window.

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

Problems Javadoc Declaration Console Progress Call Hierarchy

Ekeko Console

Adding a new project to the Ekeko model: P/TestCase-TypeParameters
Populating JavaProjectModel for: TestCase-TypeParameters
Gathered information from JDT compilation units in 36ms

The screenshot shows an IDE interface with several windows and toolbars.

Top Bar: Contains standard icons for file operations (New, Open, Save, Print, etc.), a search bar, and a "Quick" button. To the right is a large orange "Demo 4" button with a play icon.

EkekoX Editor: A window titled "chaq-demo4.eck" displays EkekoX code. It includes annotations like `@EntityProperty`, `private`, `matchiset`, `EntityIdentifier`, and `equals`. Below the editor are tabs for "Overview", "Search Templates", and "Replacement Templates".

Code Editor: A window titled "BreakStatement.java" shows Java code for a class named `BreakStatement`. The code includes imports for `be.ac.chaq.model.entity.EntityIdentifier` and `SimpleName`. It defines a private field `label` of type `EntityIdentifier` and provides methods to get and set its value.

Bottom Bar: Includes tabs for "Problems", "Javadoc", "Declaration", "Console" (with a count of 23), "Progress", and "Call Hierarchy". There are also various status icons and a toolbar with icons for file operations.

This piece of code needs to change



Demo 4

```
chaq-demo4.ecc
```

```
[@EntityProperty(value=?connType.class) private]@matchiset EntityIdentifier [equals ?FieldType] ...;
```

```
=>
```

```
[EntityIdentifier<?arntoType>]@[replace ?FieldType]
```

```
Overview Search Templates Replacement Templates
```

```
BreakStatement.java
```

```
import be.ac.chaq.model.entity.EntityIdentifier;
```

```
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;  
    }  
}
```

Problems

Javadoc

Declaration

Console

Progress

Call Hierarchy

Ekeko Console

This piece of code needs to change

Demo 4

```
chaq-demo4.eck
[@EntityProperty(value=?connType.class) private]@([match]set) EntityIdentifier<?annoType> @([replace ?fieldType]) ...;

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

Overview Search Templates Replacement Templates

BreakStatement.java

```
import be.ac.chaq.model.entity.EntityIdentifier;

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;
    }
}
```

Problems Javadoc Declaration Console Progress Call Hierarchy

Ekeko Console

The screenshot shows a Java development environment with two main panes. The top pane displays EkekoX code for a template named `chaq-demo4.eck`. The code includes annotations like `@EntityProperty` and `@matchiset`. The bottom pane displays Java code for a class named `BreakStatement.java`, which extends `Statement` and has methods for getting and setting a label. The Java code editor includes standard navigation and search tools.

Top Pane (EkekoX Editor):

```
[@EntityProperty(value=?cnnctType.class) private]@matchiset [EntityIdentifier]@[(equals ?fieldType)] ...;  
=>  
[EntityIdentifier<?arntoType>]@[(replace ?fieldType)]
```

Bottom Pane (Java Editor):

```
import be.ac.chaq.model.entity.EntityIdentifier;  
  
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;  
    }  
}
```

Demo 4

The screenshot shows an IDE interface with several tabs and toolbars. The main editor window displays EkekoX code. A tooltip with an orange arrow points to a specific directive in the code:

```
[@EntityProperty(value=?cnnctType.class) private]@matchiset [EntityIdentifier]@[(equals ?fieldType)] ...;  
=>  
[EntityIdentifier<?arnoType>]@[(replace ?fieldType)]
```

The directive `[EntityIdentifier<?arnoType>]@[(replace ?fieldType)]` is highlighted with a red rectangular box.

The tooltip text is: "Rewriting directive: replace the value of ?fieldType ..".

Problems Javadoc Declaration Console Progress Call Hierarchy

Ekeko Console

Demo 4



Demo 4

chaq-demo4.eck

[@EntityProperty(value=?cnnctType.class) private]@matchiset [EntityIdentifier]@[(equals ?fieldType)] ...;

>>

[EntityIdentifier<?arntoType>]@[(replace ?fieldType)]

.. with this new piece of code.

Rewriting directive: replace the value of ?fieldType ..

```
+ import be.ac.chaq.model.entity.EntityIdentifier;

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;
    }
}
```

Problems

Javadoc

Declaration

Console

Progress

Call Hierarchy

Ekeko Console

The screenshot shows an IDE interface with several windows and toolbars.

Top Bar: Contains standard icons for file operations (New, Open, Save, Print, etc.), a search bar, and a "Quick" button. To the right is a large orange "Demo 4" button with a play icon.

EkekoX Editor: A window titled "chaq-demo4.eck" displays EkekoX code. It includes annotations like `@EntityProperty`, `private`, `matchiset`, `EntityIdentifier`, and `equals`. Below the editor are tabs for "Overview", "Search Templates", and "Replacement Templates".

Code Editor: A window titled "BreakStatement.java" shows Java code for a class named `BreakStatement`. The code includes imports for `be.ac.chaq.model.entity.EntityIdentifier` and `SimpleName`. It defines a private field `label` of type `EntityIdentifier` and provides methods to get and set its value.

Bottom Bar: Includes tabs for "Problems", "Javadoc", "Declaration", "Console" (with a count of 23), "Progress", and "Call Hierarchy". There are also various status icons and a toolbar with icons for file operations.

Replace field declaration, getter and setter

Demo 5



scam_demo3.ekx

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

→

[EntityIdentifier<?annoType>]@[repl
[EntityIdentifier<?annoType>]@[repl
[EntityIdentifier<?annoType>]@[repl
```

ArrayCreation.java

```
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 {
  @EntityProperty(value = ArrayType.class)
  private EntityIdentifier type;

  @EntityListProperty(value = Expression.class)
  private List<EntityIdentifier> dimensions;

  @EntityProperty(value = ArrayInitializer.class)
  private EntityIdentifier initializer;

  public EntityIdentifier getType() {
    return type;
```

src

- be.ac.chaq.change
- be.ac.chaq.model.ast.java
- AbstractTypeDeclaration.java
- Annotation.java
- AnnotationTypeDeclaration
- AnnotationTypeMemberDeclaration
- AnonymousClassDeclaration
- 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
- CatchCause.java
- CharacterLiteral.java
- ClassInstanceCreation.java
- Comment.java
- CompilationUnit.java
- ConditionalExpression.java
- ConstructorInvocation.java
- ContinueStatement.java
- DoStatement.java
- EmptyStatement.java

Console Ekeko Query Results

Replace field declaration, getter and setter

Demo 5



scam_demo3.ekx

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

→

[EntityIdentifier<?annoType>]@[repl
[EntityIdentifier<?annoType>]@[repl
[EntityIdentifier<?annoType>]@[repl
```

ArrayCreation.java

```
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 {
  @EntityProperty(value = ArrayType.class)
  private EntityIdentifier type;

  @EntityListProperty(value = Expression.class)
  private List<EntityIdentifier> dimensions;

  @EntityProperty(value = ArrayInitializer.class)
  private EntityIdentifier initializer;

  public EntityIdentifier getType() {
    return type;
```

src

- be.ac.chaq.change
- be.ac.chaq.model.ast.java
- AbstractTypeDeclaration.java
- Annotation.java
- AnnotationTypeDeclaration
- AnnotationTypeMemberDeclaration
- AnonymousClassDeclaration
- 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
- CatchCause.java
- CharacterLiteral.java
- ClassInstanceCreation.java
- Comment.java
- CompilationUnit.java
- ConditionalExpression.java
- ConstructorInvocation.java
- ContinueStatement.java
- DoStatement.java
- EmptyStatement.java

Console Ekeko Query Results

Cha-Q search-and-replace

Cha-Q search-and-replace

- FOD Financiën pilot study:
 - ~62% of systematic edits can be fully automated
 - Remainder can be partially automated

Cha-Q search-and-replace

- FOD Financiën pilot study:
 - ~62% of systematic edits can be fully automated
 - Remainder can be partially automated
- What can Cha-Q search-and-replace do for *you*?



*find and fix all occurrences of
the same bug*



*upgrade existing clients to a
newer API version*



create custom refactorings



*co-maintain multiple system
variants*



*evolve existing code to newer
versions of Java*



*automate repeating
maintenance tasks*

Cha-Q search-and-replace

- FOD Financiën pilot study:
 - ~62% of systematic edits can be fully automated
 - Remainder can be partially automated
- What can Cha-Q search-and-replace do for *you*?



*find and fix all occurrences of
the same bug*



*upgrade existing clients to a
newer API version*



create custom refactorings



*co-maintain multiple system
variants*



*evolve existing code to newer
versions of Java*



*automate repeating
maintenance tasks*