

# Supporting co-evolution of design and source code

Andy Kellens  
[akellens@vub.ac.be](mailto:akellens@vub.ac.be)



# Various things

- IntensiVE
- Program querying
- Regularity mining
- Smart annotations
- Analyzing Cobol with Cognac

# Various things

- IntensiVE
- Program querying
- Regularity mining
- **Smart annotations**
- **Analyzing Cobol with Cognac**

# Smart Annotations



# Java annotations

```
public class AccessorExample {
    public Integer my_field;
    public Integer my_field2;

    @Getter
    public Integer getMy_field() {
        return my_field;
    }

    public Integer getMy_field2() {
        return my_field2;
    }

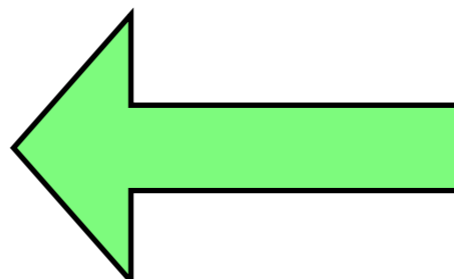
    @Getter
    public Integer returnField2()
    {
        return my_field2;
    }

    @Getter
    public Integer getNothing()
    {
        return null;
    }
}
```

# Java annotations

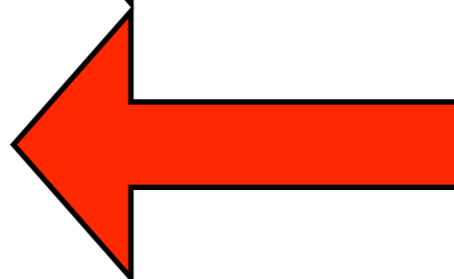
```
public class AccessorExample {  
    public Integer my_field;  
    public Integer my_field2;
```

```
@Getter  
public Integer getMy_field() {  
    return my_field;  
}
```



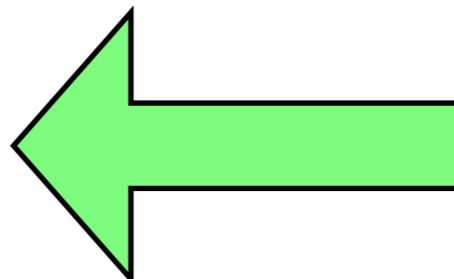
OK

```
public Integer getMy_field2() {  
    return my_field2;  
}
```



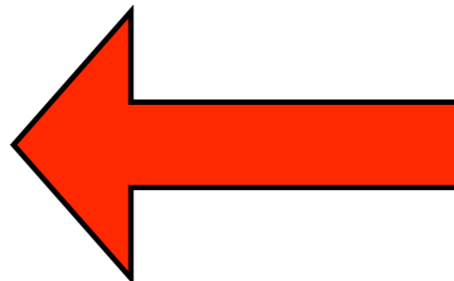
Missing!!!

```
@Getter  
public Integer returnField2()  
{  
    return my_field2;  
}
```



OK

```
@Getter  
public Integer getNothing()  
{  
    return null;  
}
```



Incorrect!!!

```
}
```

# Rules

- **RULE1:Accessors should start with the prefix 'get'**
- **RULE2:Accessors should return the value of a field**

# Smart annotation

```
@Target(ElementType.METHOD)
public @interface Getter {
    @Necessary
    public static final String RULE
        = "?item methodDeclarationHasName: ?name, " +
          "['get*' match: ?name asString]";

    @Sufficient
    public static final String RULE2 =
        "?item isMethodDeclaration, ?class definesMethod: ?item, " +
        "?class definesVariable: ?variable, " +
        "?item returns: ?variable";
}
```



# Smart annotation

Annotation => Rule

```
@Target(ElementType.METHOD)
public @interface Getter {
    @Necessary
    public static final String RULE
        = "?item methodDeclarationHasName: ?name, " +
          "['get*' match: ?name asString]";
}
```

Annotation <=> Rule

```
@Sufficient
public static final String RULE2 =
    "?item isMethodDeclaration, ?class definesMethod: ?item, " +
    "?class definesVariable: ?variable, " +
    "?item returns: ?variable";
}
```

# Eclipse integration



```
package example;

public class AccessorExample {
    public Integer my_field;
    public Integer my_field2;

    @Getter
    public Integer getMy_field() {
        return my_field;
    }

    public Integer getMy_field2() {
        return my_field2;
    }

    @Getter
    public Integer returnField2()
    {
        return my_field2;
    }

    @Getter
    public Integer getNothing()
    {
        return null;
    }
}
```

0 errors, 3 warnings, 0 others

Description	Resource	Path	Location
⚠ Getter	AccessorExample.java	Smart Annotation Example/	line 12
⚠ Getter	AccessorExample.java	Smart Annotation Example/	line 16
⚠ Getter	AccessorExample.java	Smart Annotation Example/	line 22

# Exceptions

```
public class AccessorExample {  
    public Integer my_field;  
    public Integer my_field2;  
  
    @Getter  
    public Integer getMy_field() {  
        return my_field;  
    }  
  
    @DoesNotApply(Getter.class)  
    public Integer getMy_field2() {  
        return my_field2;  
    }  
  
    @Getter  
    public Integer returnField2()  
    {  
        return my_field2;  
    }  
  
    @Getter  
    @DoesApply(Getter.class)  
    public Integer getNothing()  
    {  
        return null;  
    }  
}
```

# Exceptions

```
public class AccessorExample {  
    public Integer my_field;  
    public Integer my_field2;  
  
    @Getter  
    public Integer getMy_field() {  
        return my_field;  
    }  
  
    @DoesNotApply(Getter.class)  
    public Integer getMy_field2() {  
        return my_field2;  
    }  
  
    @Getter  
    public Integer returnField2()  
    {  
        return my_field2;  
    }  
  
    @Getter  
    @DoesApply(Getter.class)  
    public Integer getNothing()  
    {  
        return null;  
    }  
}
```

Explicit exceptions



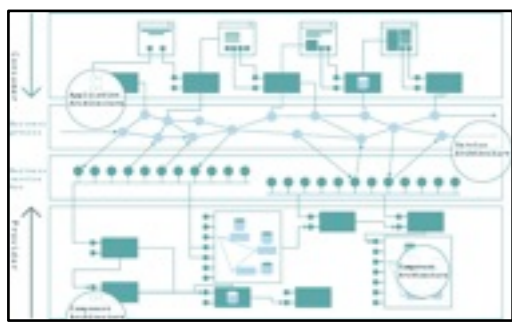
# Cognac

# Context

- Industrial case study with inno.com
- Average-sized Cobol system (500 KLoc)
- Verify design

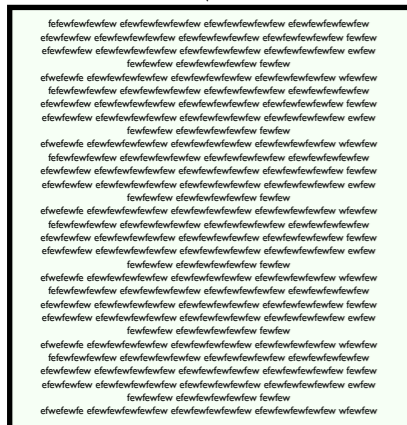
# Context

- Industrial case study with inno.com
- Average-sized Cobol system (500 KLoc)
- Verify design



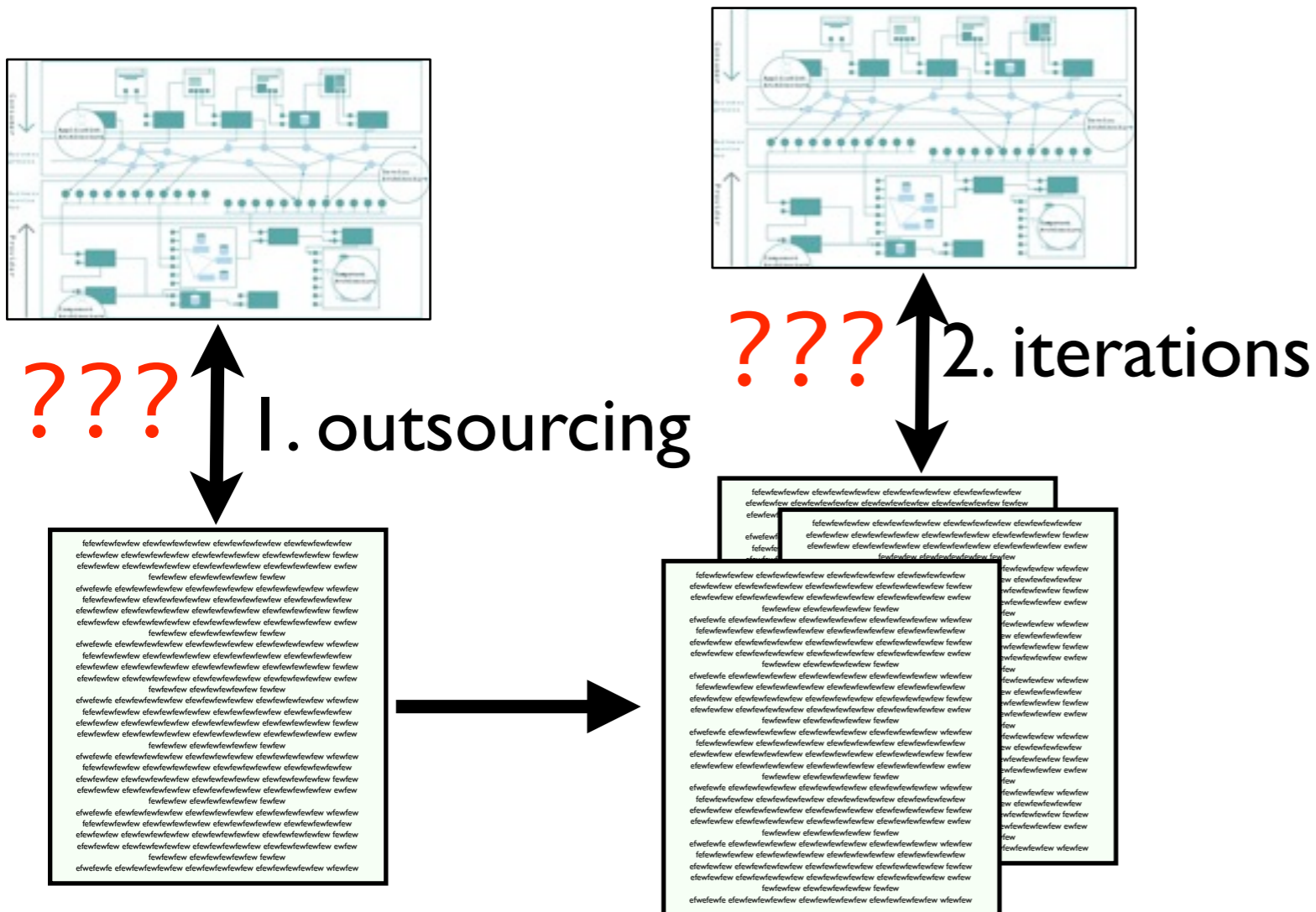
???

↑ ↓ I. outsourcing



# Context

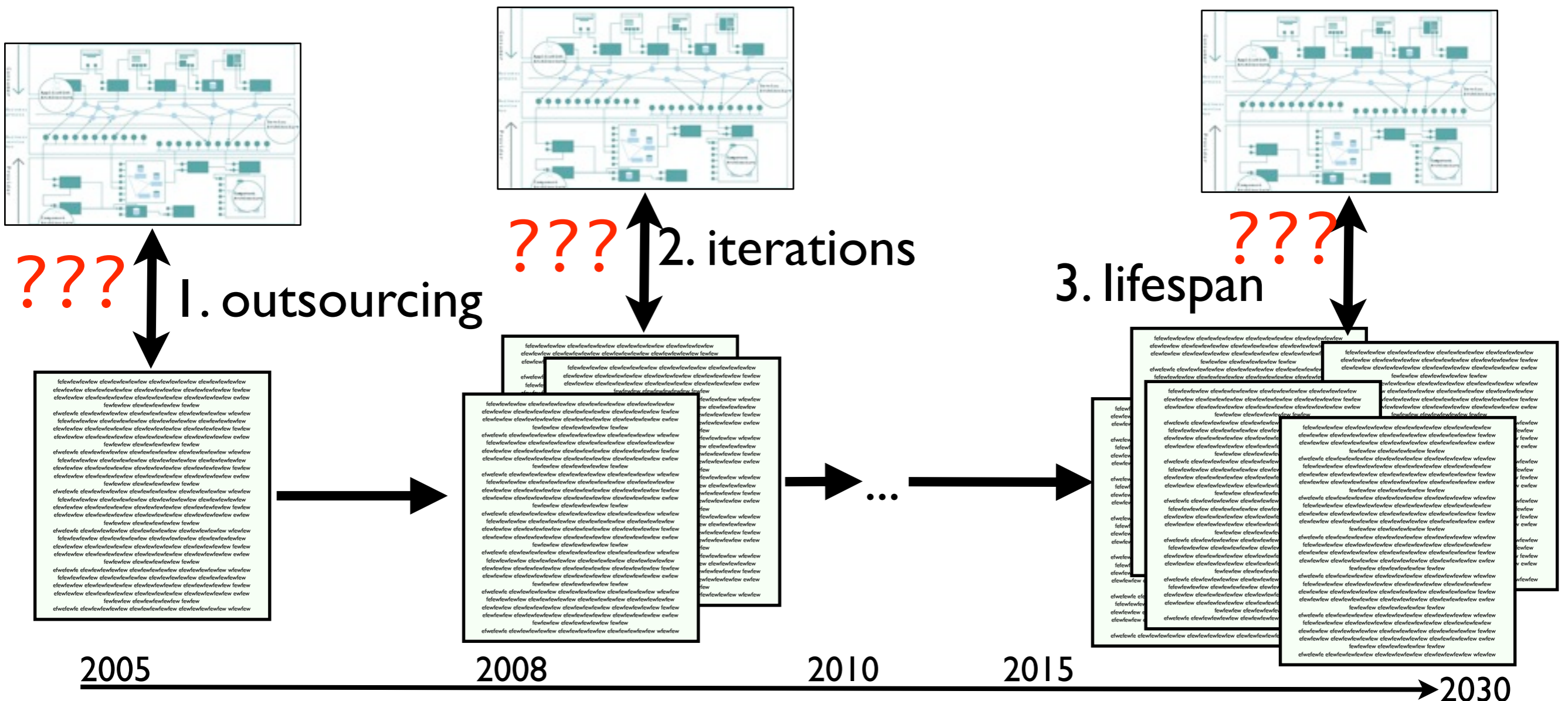
- Industrial case study with inno.com
- Average-sized Cobol system (500 KLoc)
- Verify design





# Context

- Industrial case study with inno.com
- Average-sized Cobol system (500 KLoc)
- Verify design



# Challenges

- Analyzing Cobol code
  - Parsing
  - More than structural information
- Performance
- Extensibility
  - Various kinds of design rules

# Cognac

- Declarative meta programming to verify design wrt. Cobol code
- Built on top of IntensiVE
- Technically:
  - Island-based parser
  - Library of logic predicates
  - Basic static analyses





# Example: Copybook-linkage

PROGRAM5

```
list of placeholder text for PROGRAM5
```

CALL 'Program10' using **PARAM**

PROGRAM10

```
list of placeholder text for PROGRAM10
```

???Same data definition???

COPY 'COPYBOOK1'

COPY 'COPYBOOK1'

COPYBOOK1

```
01 PARAM
.....
```

# Example: Copybook-linkage

PROGRAM5

```

efewfiew efewfiew efewfiew efewfiew efewfiew
efewfiew efewfiew efewfiew efewfiew efewfiew
efewfiew efewfiew efewfiew efewfiew efewfiew
efewfiew efewfiew efewfiew efewfiew efewfiew
efewfiew efewfiew efewfiew efewfiew efewfiew
efewfiew efewfiew efewfiew efewfiew efewfiew
efewfiew efewfiew efewfiew efewfiew efewfiew
efewfiew efewfiew efewfiew efewfiew efewfiew
efewfiew efewfiew efewfiew efewfiew efewfiew
efewfiew efewfiew efewfiew efewfiew efewfiew
efewfiew efewfiew efewfiew efewfiew efewfiew
efewfiew efewfiew efewfiew efewfiew efewfiew
efewfiew efewfiew efewfiew efewfiew efewfiew
efewfiew efewfiew efewfiew efewfiew efewfiew
efewfiew efewfiew efewfiew efewfiew efewfiew
efewfiew efewfiew efewfiew efewfiew efewfiew
efewfiew efewfiew efewfiew efewfiew efewfiew
efewfiew efewfiew efewfiew efewfiew efewfiew
efewfiew efewfiew efewfiew efewfiew efewfiew
efewfiew efewfiew efewfiew efewfiew efewfiew
efewfiew efewfiew efewfiew efewfiew efewfiew
efewfiew efewfiew efewfiew efewfiew efewfiew

```

CALL 'Program10' using **PARAM**

??Same data definition??

PROGRAM10

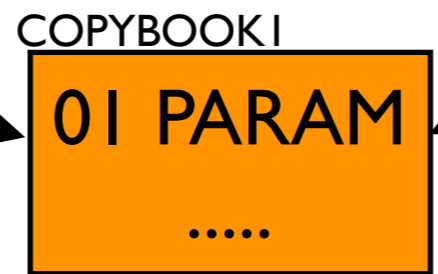
```

efewfiew efewfiew efewfiew efewfiew efewfiew
efewfiew efewfiew efewfiew efewfiew efewfiew
efewfiew efewfiew efewfiew efewfiew efewfiew
efewfiew efewfiew efewfiew efewfiew efewfiew
efewfiew efewfiew efewfiew efewfiew efewfiew
efewfiew efewfiew efewfiew efewfiew efewfiew
efewfiew efewfiew efewfiew efewfiew efewfiew
efewfiew efewfiew efewfiew efewfiew efewfiew
efewfiew efewfiew efewfiew efewfiew efewfiew
efewfiew efewfiew efewfiew efewfiew efewfiew
efewfiew efewfiew efewfiew efewfiew efewfiew
efewfiew efewfiew efewfiew efewfiew efewfiew
efewfiew efewfiew efewfiew efewfiew efewfiew
efewfiew efewfiew efewfiew efewfiew efewfiew
efewfiew efewfiew efewfiew efewfiew efewfiew
efewfiew efewfiew efewfiew efewfiew efewfiew
efewfiew efewfiew efewfiew efewfiew efewfiew
efewfiew efewfiew efewfiew efewfiew efewfiew
efewfiew efewfiew efewfiew efewfiew efewfiew
efewfiew efewfiew efewfiew efewfiew efewfiew
efewfiew efewfiew efewfiew efewfiew efewfiew
efewfiew efewfiew efewfiew efewfiew efewfiew

```

COPY 'COPYBOOK1'

COPY 'COPYBOOK1'



Design rule: A *caller* should include the *copybook* used in the *linkage* section of the *callee*

# Copybook -linkage

## *Called programs*

```
?program programCallsProgram: ?calledProgram
```

## *Program with copybook*

```
?program programWithCopyStatement: ?copy,  
?copy copyStatementInLinkageSection,  
?copy copyStatementIncludesCopybook: ?copybook
```



# Copybook -linkage

## *Called programs*

?program programCallsProgram: ?calledProgram



## *Program with copybook*

∃

?program programWithCopyStatement: ?copy,  
?copy copyStatementInLinkageSection,  
?copy copyStatementIncludesCopybook: ?copybook

- ∇ ?caller ∈ Called programs :
- ∃ ?corresponding ∈ Program with copybook :
- ?caller.calledProgram equals: ?corresponding.program,
- ?caller.program programIncludesCopybook: ?corresponding.copybook

# Copybook -linkage

if a program calls another program, it should also include the corresponding copybook of that program()

Domain: 656 out of 698 (93.9828%) Range: 116 out of 201 (57.7114%)

if a program calls another program, it should also include the corresponding copybook of that program

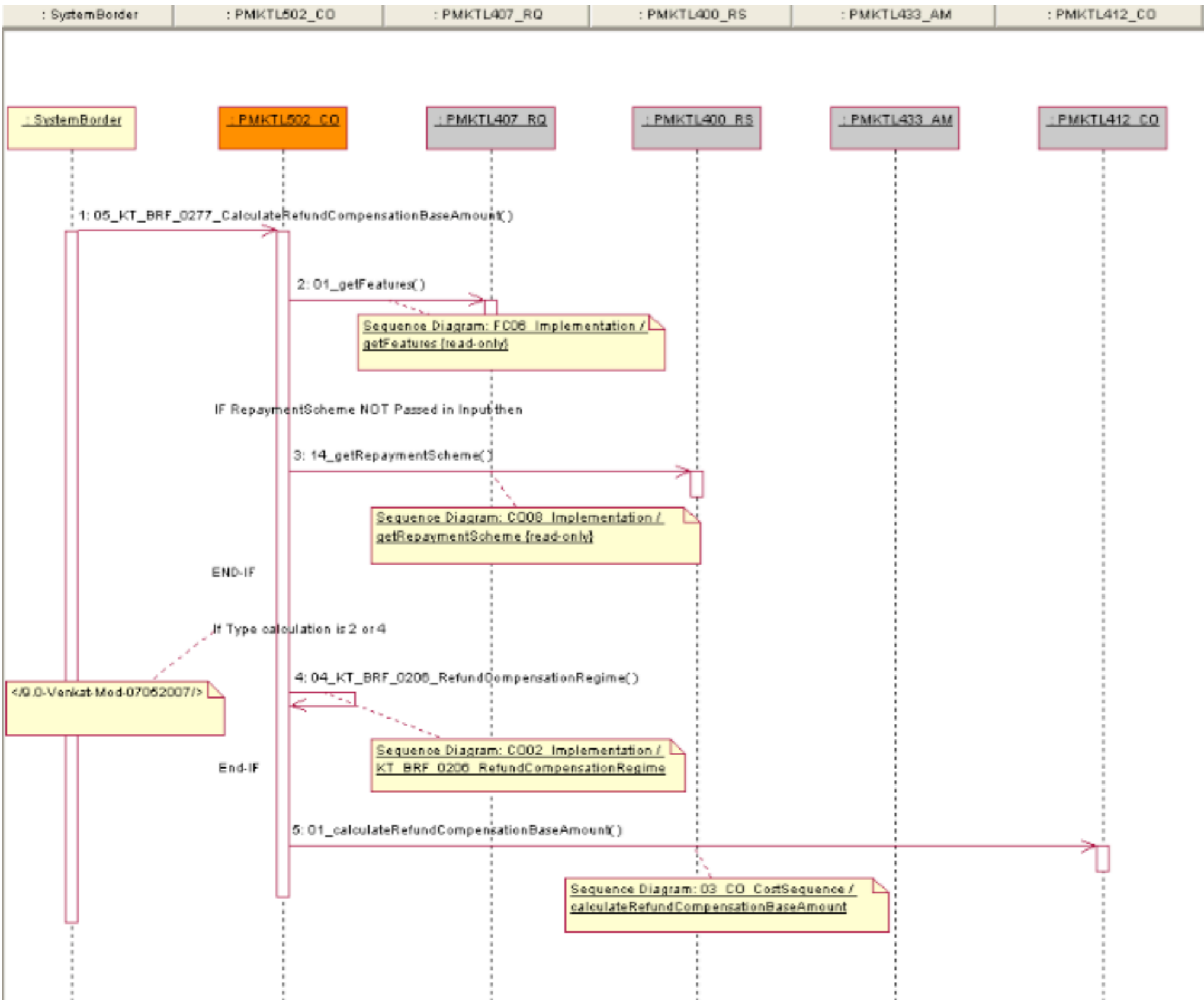
program -> Program PMKTR342 calledProgram -> Program PMKTR439	program -> Program PMKTR439 copybook -> RMKTY439
program -> Program PMKTR342 calledProgram -> Program PMKTR439	program -> Program PMKTR439 copybook -> RMKTY439
program -> Program PMKTR534 calledProgram -> Program PMKTR439	program -> Program PMKTR439 copybook -> RMKTY439
program -> Program PMKTR534 calledProgram -> Program PMKTR439	program -> Program PMKTR439 copybook -> RMKTY439
program -> Program PMKTD365 calledProgram -> Program PMKTR459	program -> Program PMKTR459 copybook -> RMKTY459
program -> Program PMKTD365 calledProgram -> Program PMKTR459	program -> Program PMKTR459 copybook -> RMKTY459
program -> Program PMKTR404 calledProgram -> Program PMKTR504	program -> Program PMKTR504 copybook -> RMKTY504
program -> Program PMKTR404 calledProgram -> Program PMKTR504	program -> Program PMKTR504 copybook -> RMKTY504
program -> Program PMKTM307 calledProgram -> Program PMKTR434	program -> Program PMKTR434 copybook -> RMKTY434
program -> Program PMKTM307 calledProgram -> Program PMKTR434	program -> Program PMKTR434 copybook -> RMKTY434
program -> Program PMKTR470 calledProgram -> Program PMKTR570	program -> Program PMKTR570 copybook -> RMKTY570
program -> Program PMKTR470 calledProgram -> Program PMKTR570	program -> Program PMKTR570 copybook -> RMKTY570
program -> Program PMKTD398	program -> Program PMKTR889

Not in domain Not in range

program -> Program PMKTR447, calledProgram -> Program PMAZR019	program -> Program PMKTC334, copybook -> RMKTI334
program -> Program PMKTR454, calledProgram -> Program PMFFR400	program -> Program PMKTC391, copybook -> RMKTI391
program -> Program PMKTR454, calledProgram -> Program PMFMR001	program -> Program PMKTL319, copybook -> RMKTY319
program -> Program PMKTR476, calledProgram -> Program PMFPR016	program -> Program PMKTC323, copybook -> RMKTI323
program -> Program PMKTR447, calledProgram -> Program PMAMR230	program -> Program PMKTL369, copybook -> RMKTY369
program -> Program PMKTR460, calledProgram -> Program PMRBR010	program -> Program PMKTC389, copybook -> RMKTI389
program -> Program PMKTR447, calledProgram -> Program PMAMR004	program -> Program PMKTL329, copybook -> RMKTY329
program -> Program PMKTR573, calledProgram -> Program PMKBR704	program -> Program PMKTC321, copybook -> RMKTI321
program -> Program PMKTR004, calledProgram -> Program PMMLR037	program -> Program PMKTC330, copybook -> RMKTI330
program -> Program PMKTR573, calledProgram -> Program PMYPL470	program -> Program PMKTL348, copybook -> RMKTY348
program -> Program PMKTR448, calledProgram -> Program PMFFR113	program -> Program PMKTL308, copybook -> RMKTY308
program -> Program PMKTR447, calledProgram -> Program PMAMR052	program -> Program PMKTL337, copybook -> RMKTY337
program -> Program PMKTR459, calledProgram -> Program PMPYR001	program -> Program PMKTC311, copybook -> RMKTD311
program -> Program PMKTR461, calledProgram -> Program PMHAR091	program -> Program PMKTC345, copybook -> RMKTI345
program -> Program PMKTR448, calledProgram -> Program PMFFR130	program -> Program PMKTC336, copybook -> RMKTI336
program -> Program PMKTR572, calledProgram -> Program PMYPL470	program -> Program PMKTL335, copybook -> RMKTY335

Copybook

# Rational Rose doc.



# Verifying the doc.

Sequence Diagrams of Computational Components

file:///Users/akellens/Documents/VUB/images/Cobol/Reeboc%20HTML%20Report/Reeboc7.html

Seaside Cartoons News Software VUB Research Fun Gmail Fortisbanking ING Exploration

## Sequence Diagrams of Computational Components

*Verification of the Rational Rose sequence diagrams in the Computational Components*

Number of inconsistent instances:7  
Total number of instances:11

### Regularities instances

**Regularity instance** with parameter SequenceDiagramsOfComputationalComponents(Program PMKTR503,C:\Documents and Settings\Administrator\Desktop\Reeboc Delivery 1.0\Technical\_Design.mdl>Logical View>Technical Design>02\_Business\_Services (= Functional Components)>01\_Computational\_Components>CC01\_CalculationMgr>CC01\_Implementations>CC01\_Implementation>01\_RS\_RepaymentSchemeCalculationSequence>KT\_BRF\_0037\_GetFirstDueDate,#(Program PMKTR703))

### Intensional relation Sequence diagram verification

*All calls in the sequence diagram should be found in the code in the correct order.*

Elements from source intensional view SequenceDiagramViewGenerator not in relation: 1/1

Inconsistent entities

```
diagram->C:\Documents and Settings\Administrator\Desktop\Reeboc Delivery 1.0\Technical_Design.mdl>Logical View>Technical Design>02_Business_Services (= Functional Components)>01_Computational_Components>CC01_CalculationMgr>CC01_Implementations>CC01_Implementation>01_RS_RepaymentSchemeCalculationSequence>KT_BRF_0037_GetFirstDueDate  
program->Program PMKTR503  
sequence->#(Program PMKTR703)
```

---

**Regularity instance** with parameter SequenceDiagramsOfComputationalComponents(Program PMKTR503,C:\Documents and Settings\Administrator\Desktop\Reeboc Delivery 1.0\Technical\_Design.mdl>Logical View>Technical Design>02\_Business\_Services (= Functional Components)>01\_Computational\_Components>CC01\_CalculationMgr>CC01\_Implementations>CC01\_Implementation>03\_CO\_CostSequence>calculateRefundCompensationBaseAmount,#(Program PMKTR433 Program PMKTR513))

### Intensional relation Sequence diagram verification

*All calls in the sequence diagram should be found in the code in the correct order.*

Elements from source intensional view SequenceDiagramViewGenerator not in relation: 1/1

Inconsistent entities

```
diagram->C:\Documents and Settings\Administrator\Desktop\Reeboc Delivery 1.0\Technical_Design.mdl>Logical View>Technical Design>02_Business_Services (= Functional Components)>01_Computational_Components>CC01_CalculationMgr>CC01_Implementations>CC01_Implementation>03_CO_CostSequence>calculateRefundCompensationBaseAmount
```