

Refactoring Long Method Smell



ALEX MALMANN BECKER

PROJETO FINAL

TÉCNICAS AVANÇADAS DE MODERNIZAÇÃO DE SISTEMAS

PROF. DR. VALTER VIEIRA DE CAMARGO

MESTRADO CIÊNCIA DA COMPUTAÇÃO

2016/02

Tool Refactoring Long Method

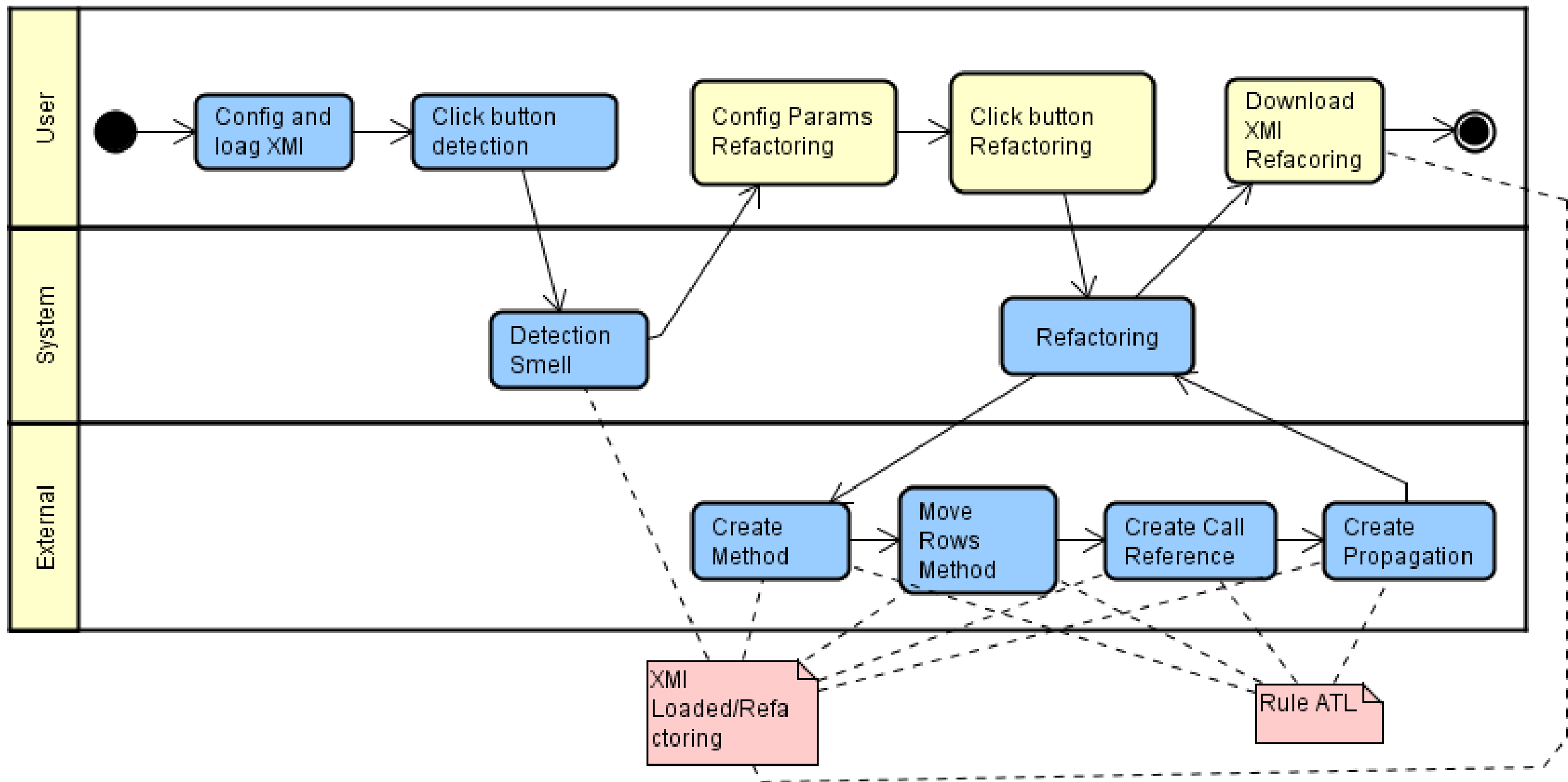
Java Web platform

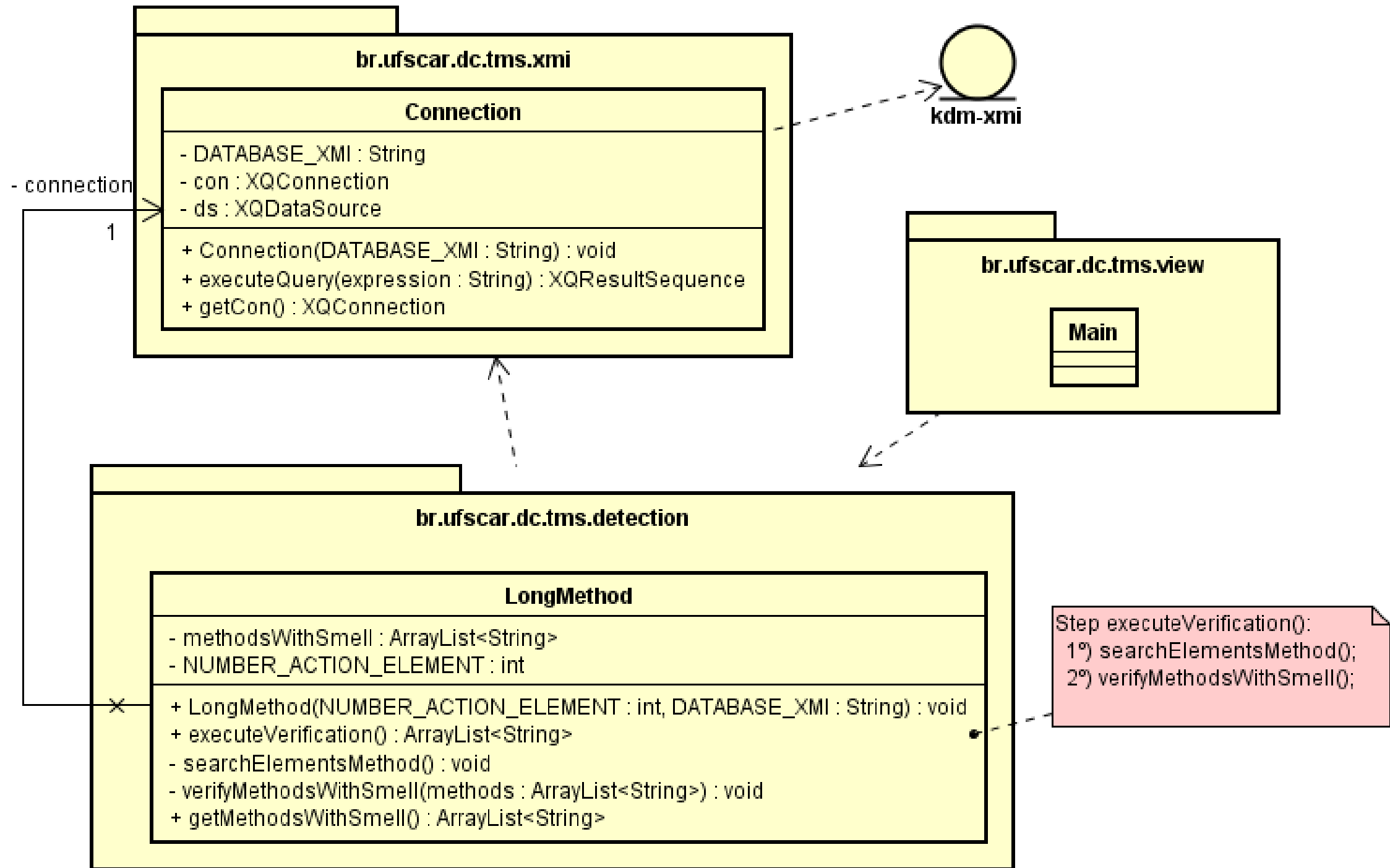
JSF framework (Java Server Faces)

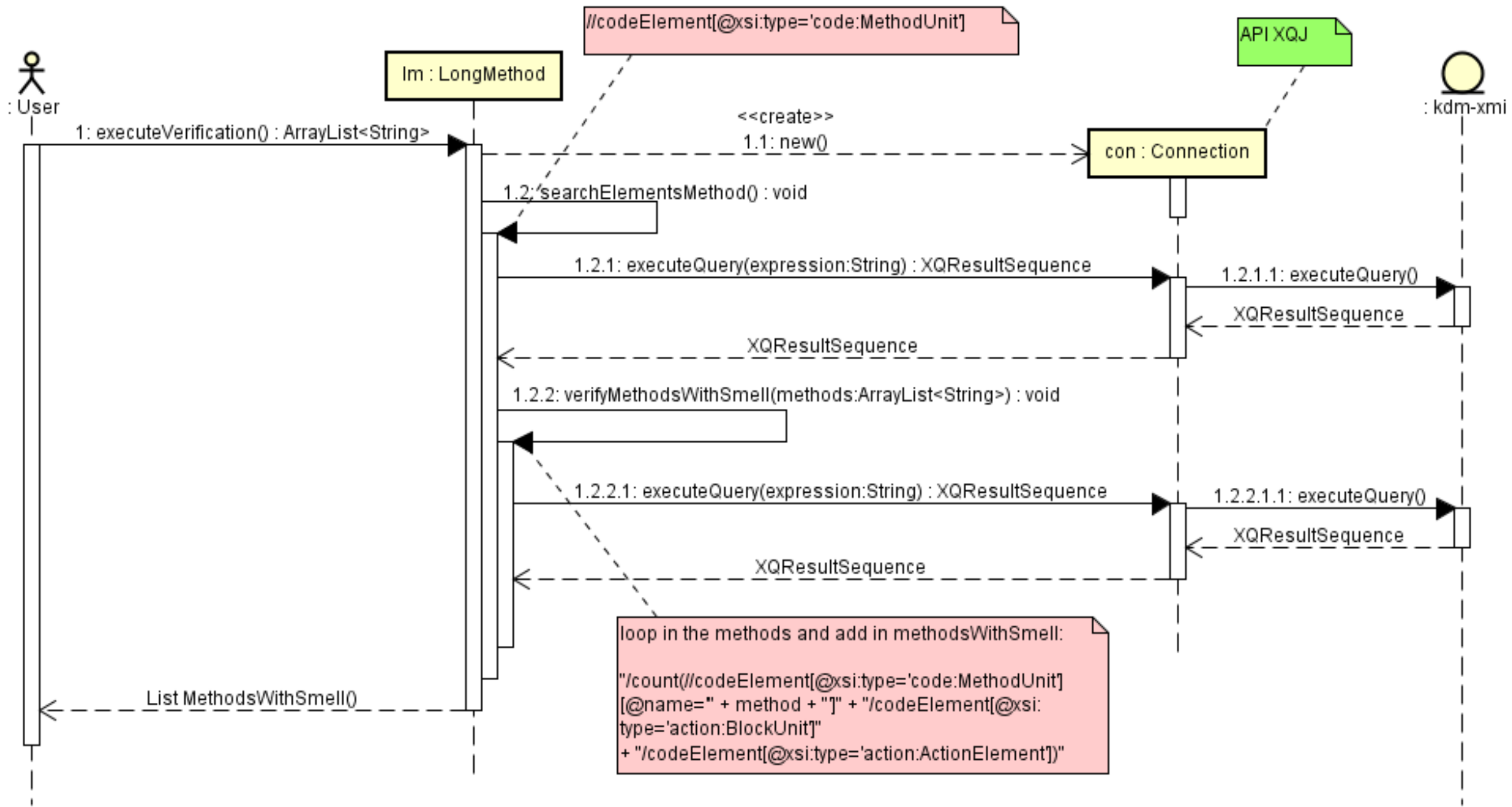
MVC architecture (Model-View-Controller)

API XQJ (Xquery – XPath)

Project ATL







Refactoring Long Method Smell



Upload XMI

Number Lines Smell

Number Lines Smell

xmilnicial.xmi

Next

2016 Refactoring Long Method Smell. By: Alex Malmann Becker

Refactoring Long Method Smell



Detection Smell

Method With Smell

(Pag. 1/1 - 1 records)



#

Name Method

1 longMethodDoModel

(Pag. 1/1 - 1 records)



Content of the selected method

Previous

Next

ATL RULES


```
-- @nsURI MM=http://www.eclipse.org/ModelDisco/kdm/code
```

```
module create_new_method;  
create OUT : MM refining IN : MM;
```

```
rule createNewMethod{  
  from  
    source : MM!ClassUnit (source.name = 'Pessoa')  
  to  
    target: MM!ClassUnit (  
      codeElement <- source.codeElement->including(newMethod)  
    ),  
    newMethod: MM!MethodUnit (  
      name <- 'newMethod',  
      export <- 'public',  
      kind <- 'method',  
      codeElement <- newSignature  
    ),  
    newSignature: MM!Signature (  
      name <- 'newMethod'  
    )  
}
```

```
-- @nsURI MM=http://www.eclipse.org/MoDisco/kdm/code  
-- @nsURI MM=http://www.eclipse.org/MoDisco/kdm/action
```

```
module move_rows_method;  
create OUT: MM refining IN: MM;
```

```
rule moveRowsMethod {  
  from  
    source: MM!MethodUnit (  
      source.name = 'newMethod'  
    )  
  to  
    target: MM!MethodUnit (  
      codeElement <- blockUnit  
    ),  
    blockUnit: MM!BlockUnit (  
      codeElement <- thisModule.getRows(  
        thisModule.getBlockUnitMethod(  
          thisModule.getMethodUnit(  
            thisModule.getClassUnit('ControllerPessoa')  
            , 'longMethodDoModel')  
          )  
        , 1, 10)  
      )  
    )  
}  
  
helper def: getClassUnit(className: String): MM!ClassUnit =  
  MM!ClassUnit.allInstances() -> any(e | e.name = className);  
  
helper def: getMethodUnit(classUnitToGetMethod: MM!ClassUnit, nameMethod: String):  
  MM!MethodUnit =  
    classUnitToGetMethod.codeElement -> any(e | e.name.equalsIgnoreCase(nameMethod));  
  
helper def: getBlockUnitMethod(methodFrom: MM!MethodUnit):  
  MM!BlockUnit =  
    methodFrom.codeElement->any(e | e.oclIsTypeOf(MM!BlockUnit));  
  
helper def: getRows(blockUnit: MM!BlockUnit, startLine: Integer, endLine: Integer) : MM!ActionElement =  
  blockUnit.codeElement.subSequence(startLine, endLine);
```

```
-- @nsURI MM=http://www.eclipse.org/MoDisco/kdm/code
-- @nsURI MM=http://www.eclipse.org/MoDisco/kdm/action
```

```
module create_call_reference;
create OUT: MM refining IN: MM;
```

```
rule createCallReference {
  from
    source: MM!BlockUnit (
      source.refImmediateComposite().oclIsTypeOf(MM!MethodUnit) and
      source.refImmediateComposite().name = 'longMethodDoModel'
    )
  to
    target: MM!BlockUnit (
      codeElement <- addCall
    ),
    addCall : MM!ActionElement(
      name <- 'expression statement',
      kind <- 'expression statement',
      codeElement <- ae
    ),
    ae : MM!ActionElement (
      name <- 'method invocation',
      kind <- 'method invocation',
      actionRelation <- call
    ),
    call : MM!Calls (
      to <- thisModule.getMethodUnit(
        thisModule.getClassUnit('Pessoa')
          , 'newMethod'),
      from <- ae
    )
}
```

```
helper def: getClassUnit(className: String): MM!ClassUnit =
  MM!ClassUnit.allInstances() -> any(e | e.name = className);
```

```
helper def: getMethodUnit(classUnitToGetMethod: MM!ClassUnit, nameMethod: String):
  MM!MethodUnit =
```

```
  classUnitToGetMethod.codeElement -> any(e | e.name.equalsIgnoreCase(nameMethod));
```

```

-- @nsURI MM=http://www.eclipse.org/MoDisco/kdm/code
-- @nsURI MMC=http://www.eclipse.org/MoDisco/kdm/core
-- @nsURI MMA=http://www.eclipse.org/MoDisco/kdm/action
-- @nsURI MMS=http://www.eclipse.org/MoDisco/kdm/structure

module create_propagation;
create OUT: MMC, OUT2: MMA, OUT3: MMS, OUT4: MM
refining IN: MMC, IN2: MMA, IN3: MMS, IN4: MM;

rule createPropagation {
  from
    source: MMC!AggregatedRelationship(
      (source.to = thisModule.getToFrom('model')) and
      (source.from = thisModule.getToFrom('controller'))
    )
  to
    target: MMC!AggregatedRelationship(
      density <- source.density + 1,
      relation <- thisModule.findCall(thisModule.getBlockUnitMethod(
        thisModule.getMethodUnit(
          thisModule.getClassUnit('ControllerPessoa')
        , 'longMethodDoModel')
      ))
    )
}

helper def: getToFrom(nameStructure: String):
  MMS!Layer =
    MMS!Layer.allInstances() -> any(e | e.implementation.first().name = nameStructure);

helper def: getClassUnit(className: String): MM!ClassUnit =
  MM!ClassUnit.allInstances() -> any(e | e.name = className);

helper def: getMethodUnit(classUnitToGetMethod: MM!ClassUnit, nameMethod: String):
  MM!MethodUnit =
    classUnitToGetMethod.codeElement -> any(e | e.name.equalsIgnoreCase(nameMethod));

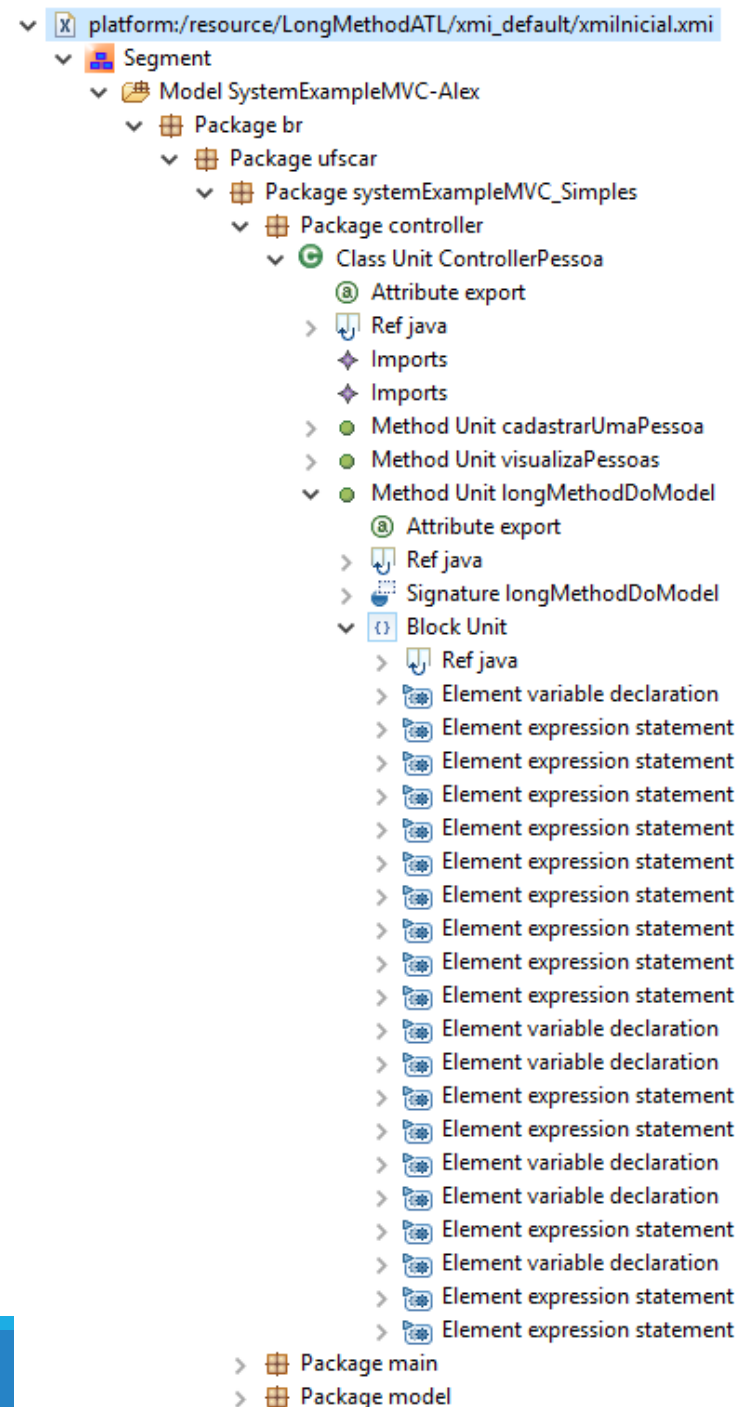
helper def: getBlockUnitMethod(methodFrom: MM!MethodUnit):
  MM!BlockUnit =
    methodFrom.codeElement->any(e | e.oclIsTypeOf(MM!BlockUnit));

helper def: findCall(block: MM!BlockUnit):
  MMA!Calls =
    block.codeElement->any(e | (e.oclIsTypeOf(MMA!ActionElement)) and
      ( thisModule.getActionElement( ) ) );

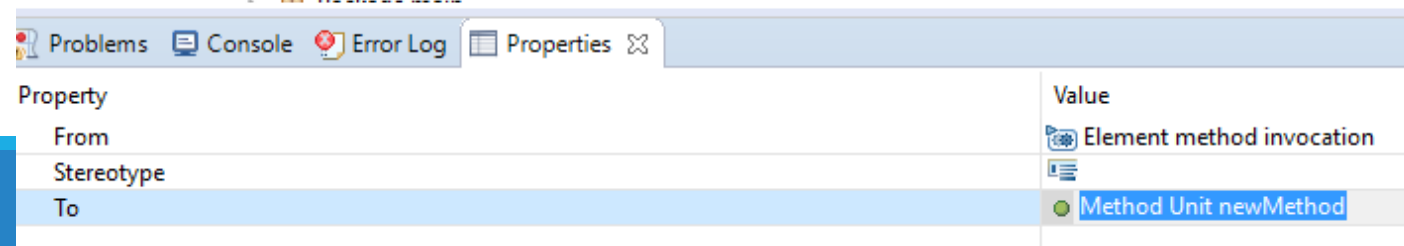
helper def: getActionElement(ae: MMA!ActionElement):
  Boolean = ( (ae.actionRelation.first().getTo() = thisModule.getMethodUnit(thisModule.getClassUnit('Pessoa'), 'newMethod'))
    and (ae.actionRelation.first().getFrom().name = 'method invocation') );

```

Result XMI



- > Element expression statement
- > Element expression statement
- > Element variable declaration
- > Element variable declaration
- > Element expression statement
- > Element expression statement
- > Element variable declaration
- > Element variable declaration
- > Element expression statement
- > Element variable declaration
- > Element expression statement
- > Element expression statement
- > Element expression statement
- > Package main
- ▼ Package model
 - > Class Unit BancoDeDadosPessoa
 - ▼ Class Unit Pessoa
 - Attribute export
 - > Ref java
 - > Storable Unit id
 - > Storable Unit nome
 - > Storable Unit idade
 - > Storable Unit profissao
 - > Method Unit Pessoa
 - > Method Unit Pessoa
 - > Method Unit getNome
 - > Method Unit setNome
 - > Method Unit getIdade
 - > Method Unit setIdade
 - > Method Unit getProfissao
 - > Method Unit setProfissao
 - > Method Unit toString
 - > Package view
- > Language Unit Common Java datatypes
- > Model externals
- > Inventory Model source references
- ▼ Model
 - > Layer main
 - > Layer view
 - ▼ Layer controller
 - ◆ Aggregated Relationship 11
 - > Layer model



Thanks!