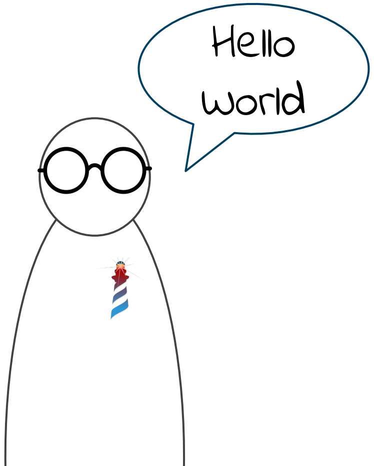
Visualizing models

Benoît Verhaeghe
badetitou@gmail.com

badetitou.github.io



Merci !!!

Ces diapos sont inspirés de ceux de

- Anne Etien
- Nicolas Anquetil

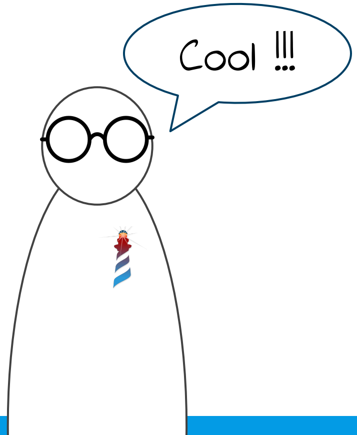
Plan

- Pharo ?
- Visualisation (Mondrian)
- Famix (Moose)
- Exemple - Qualité Logicielle
- Exemple - Interface Graphique
- Bonus - Stage

Pharo ?

kesako

Only Objects



Messages

- Unary
  - Sans argument

```
42 factorial
```

- Binary
  - Avec un seul argument

```
41 + 1
```

- Keywords
  - Avec un ou plusieurs arguments

```
kill: ‘Cora’ with: aKnife
```

From Java to Pharo

```
String function(int param1, int param2){

    this.otherFunction(param1, param2);

    return “Hello World”;

}
```

From Java to Pharo

Enlever parenthèses, points, virgule, point-virgule

```
String function int param1 int param2
    this otherFunction param1 param2
    return “Hello World”
```

From Java to Pharo

Enlever les indicateurs des types

```
function param1 param2

    this otherFunction param1 param2

    return “Hello World”
```

From Java to Pharo

Remplacer par la nouvelle syntaxe

this → self
return → ^
“” → ‘’

```
function param1 param2

    self otherFunction param1 param2

    ^ ‘Hello World’
```

From Java to Pharo

Ajouter mot-clefs pour les fonctions
et les points en fin de lignes

```
function: param1 with:param2

    self otherFunction: param1 with: param2.

    ^ ‘Hello World’
```

Résultat

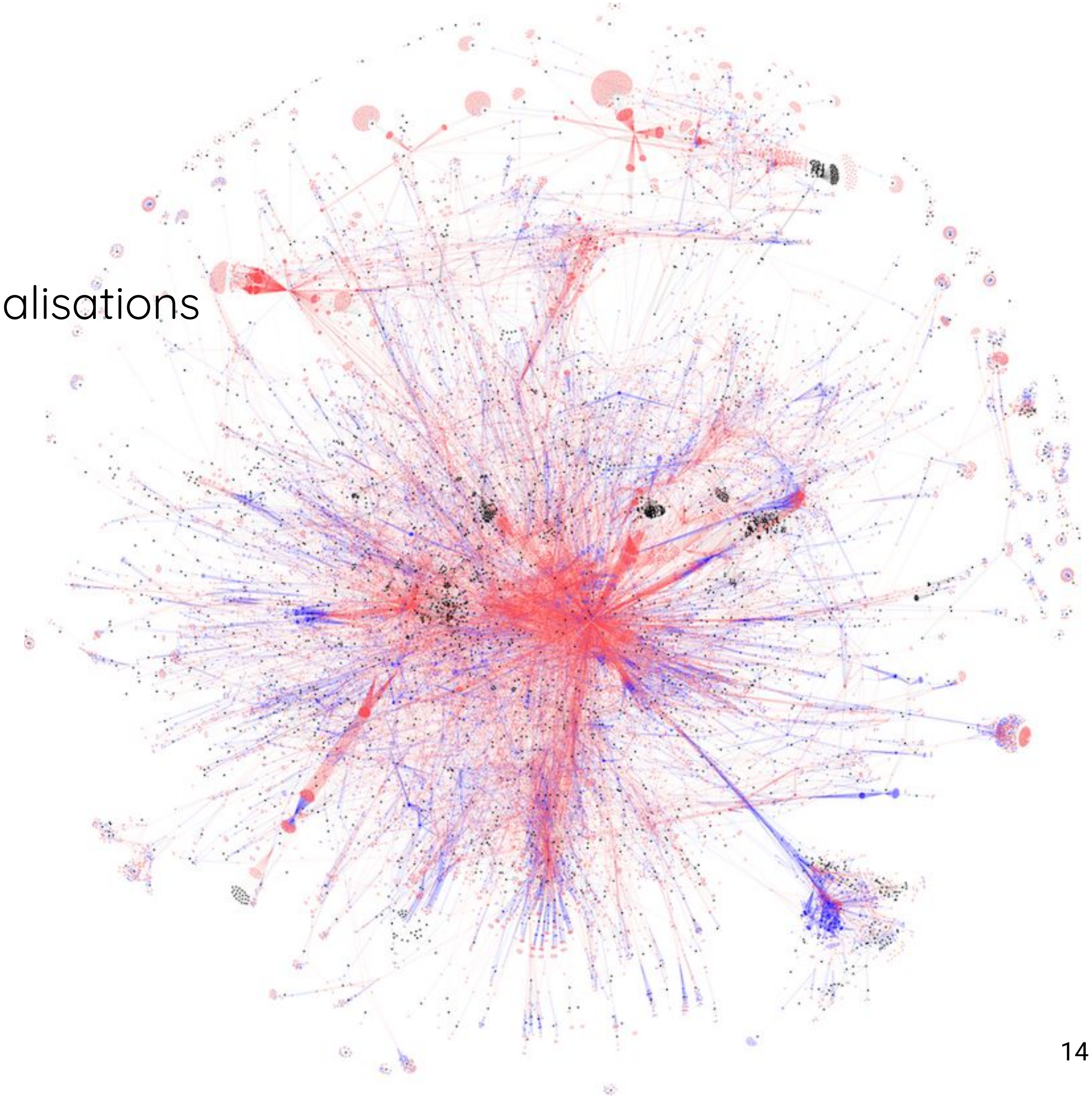
```
String function(int param1, int param2){
    this.otherFunction(param1, param2);
    return “Hello World”;
}

function: param1 with:param2
    self otherFunction: param1 with: param2.
    ^ ‘Hello World’
```

Visualisation

L’outil Mondrian

Mondrian

- DSL
- Permet de créer des visualisations
- Simple à utiliser

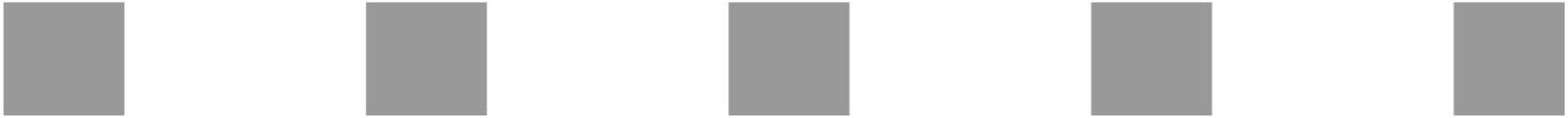


Créer une view

```
view := RTMondrian new.
^ view
```

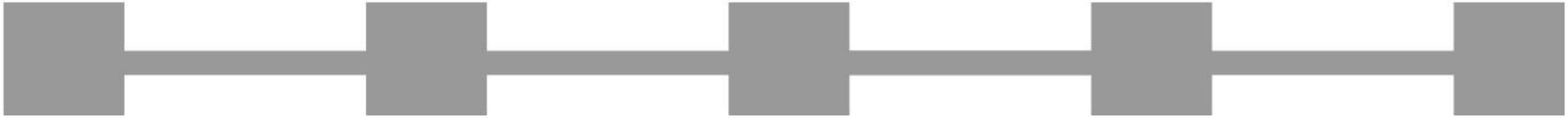
Ajouter des noeuds

```
view := RTMondrian new.
view nodes: { 1. 2. 3. 4. 5. }.
^ view
```

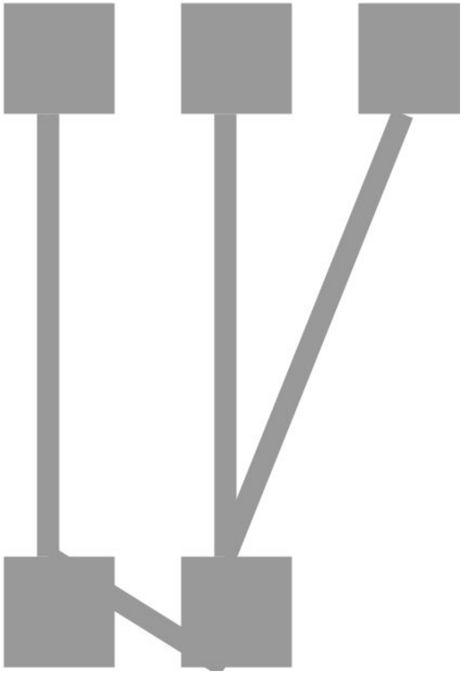


Ajouter des liens

```
view := RTMondrian new.
view nodes: { 1. 2. 3. 4. 5. }.
view edges useAssociations: { 1->2. 3->2. 4->3. 5->3 }.
^ view
```

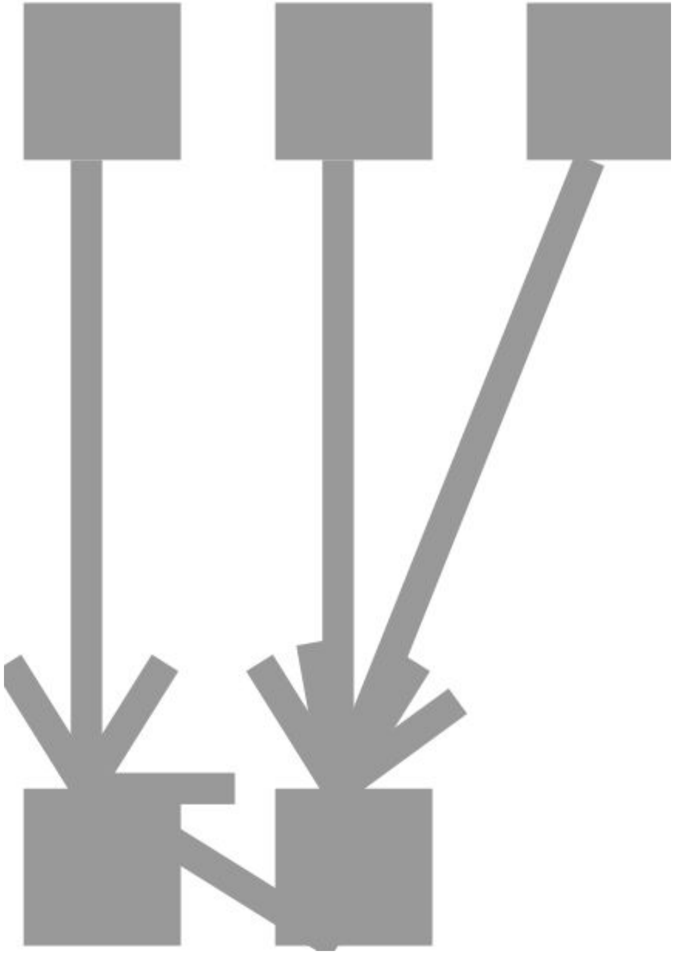


Disposer les éléments



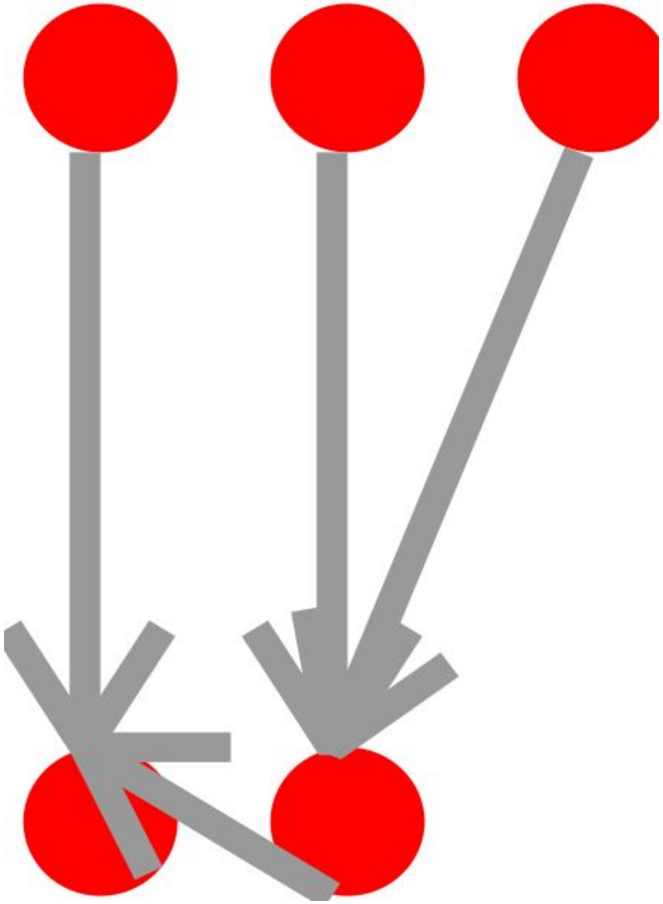
```
view := RTMondrian new.
view nodes: { 1. 2. 3. 4. 5. }.
view edges useAssociations: { 1->2. 3->2. 4->3. 5->3 }.
view layout tree.
^ view
```

Forme des liens



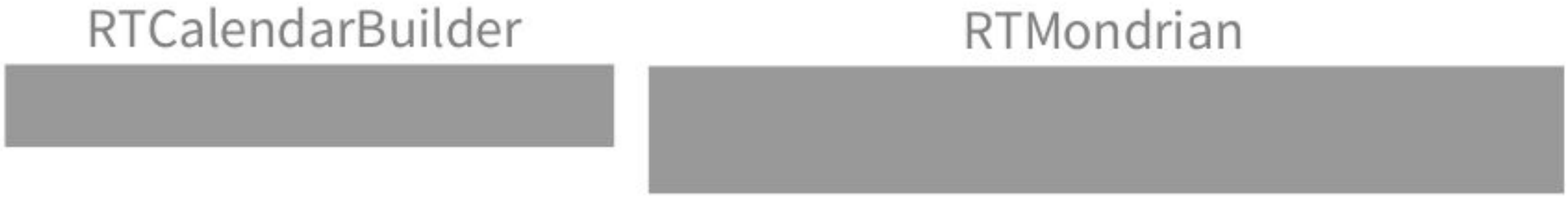
```
view := RTMondrian new.
view nodes: { 1. 2. 3. 4. 5. }.
view shape line arrowedLine shape head baseSize: 5.0; size: 8.
view edges useAssociations: { 1->2. 3->2. 4->3. 5->3 }.
view layout tree.
^ view
```

Forme des noeuds



```
view := RTMondrian new.
view shape circle color: Color red.
view nodes: { 1. 2. 3. 4. 5. }.
view shape line arrowedLine shape head baseSize: 5.0; size: 8.
view edges useAssociations: { 1->2. 3->2. 4->3. 5->3 }.
view layout tree.
^ view
```

Forme des noeuds

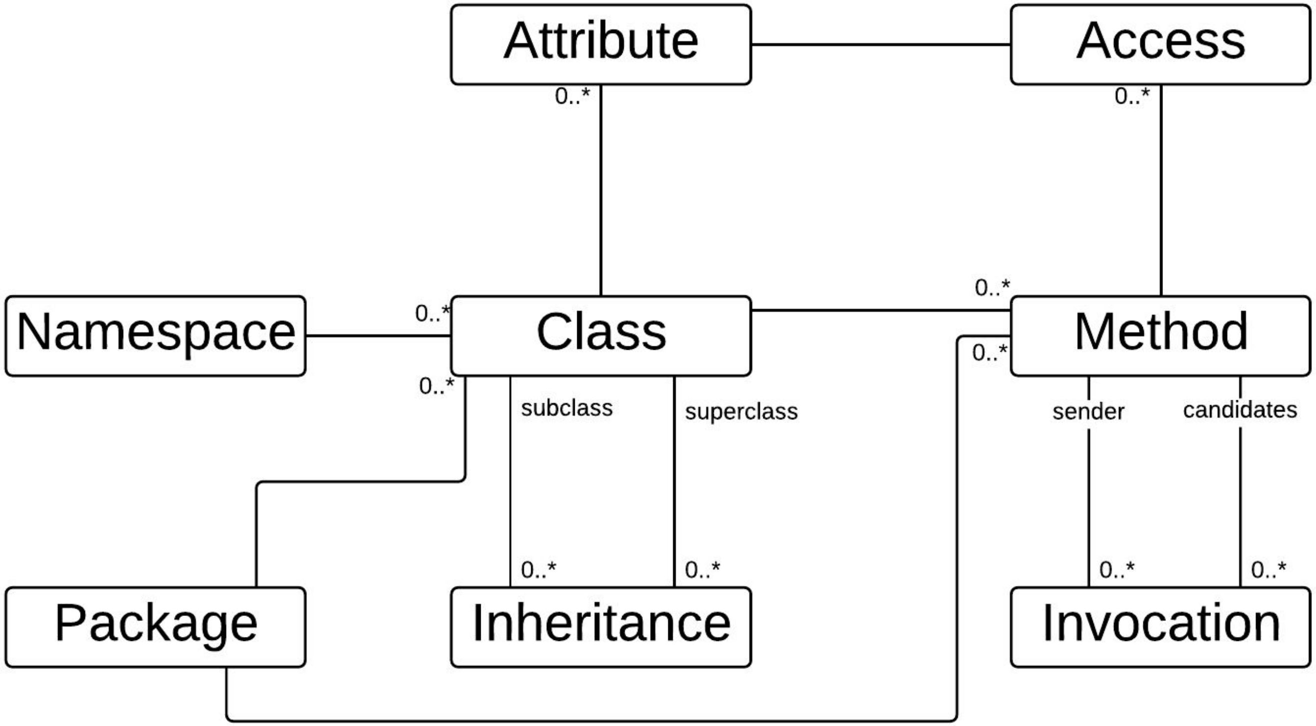


```
view := RTMondrian new.
view shape rectangle;
    height: [:class | class methods size];
    width: [:class | class methods sum: #numberOfLinesOfCode];
    withTextAbove: #name.
view nodes: {RTCalendarBuilder. RTMondrian.}.
^ view
```

Famix Model

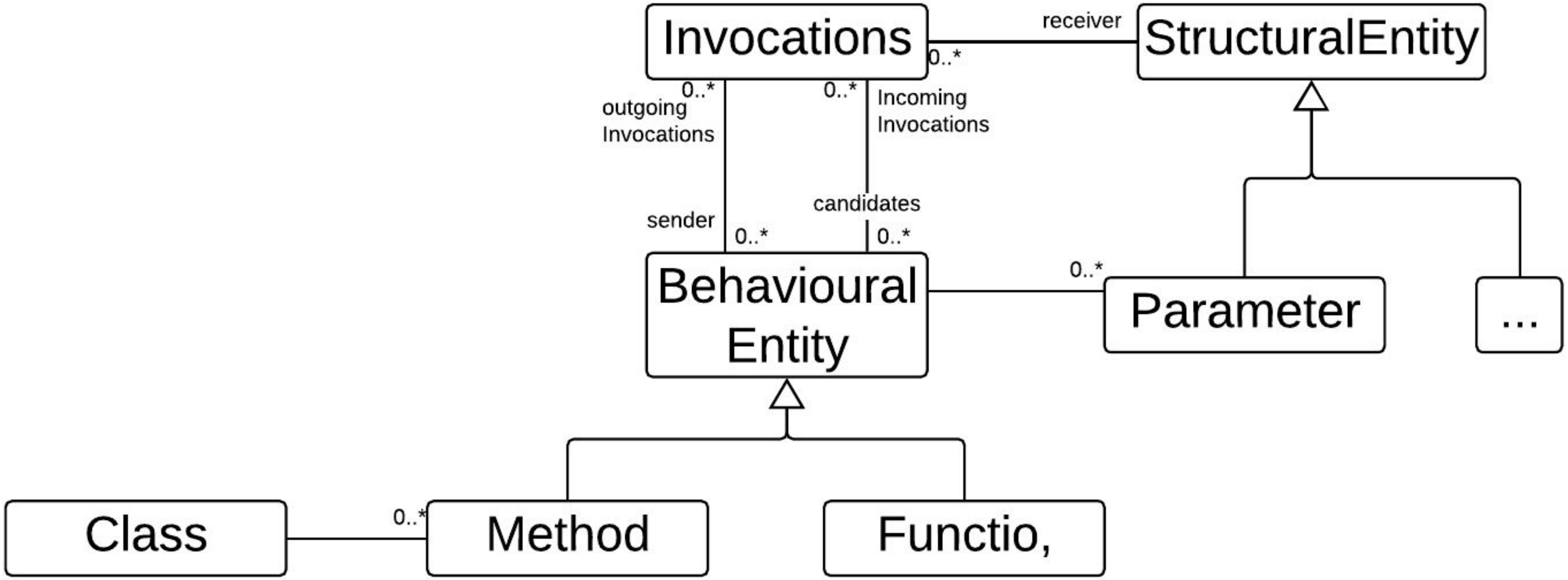
Java

Main model



source: http://themoosebook.org

Les invocations



source: http://themoosebook.org

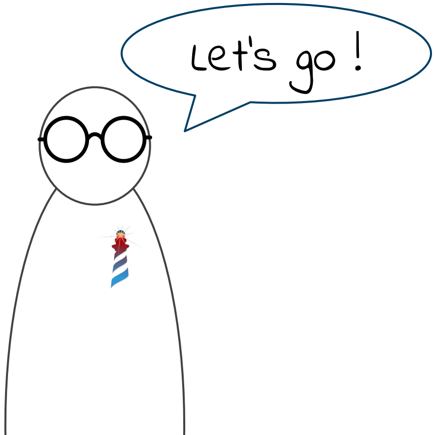
Qualité Logicielle

RCA

Que pouvons nous étudier ?

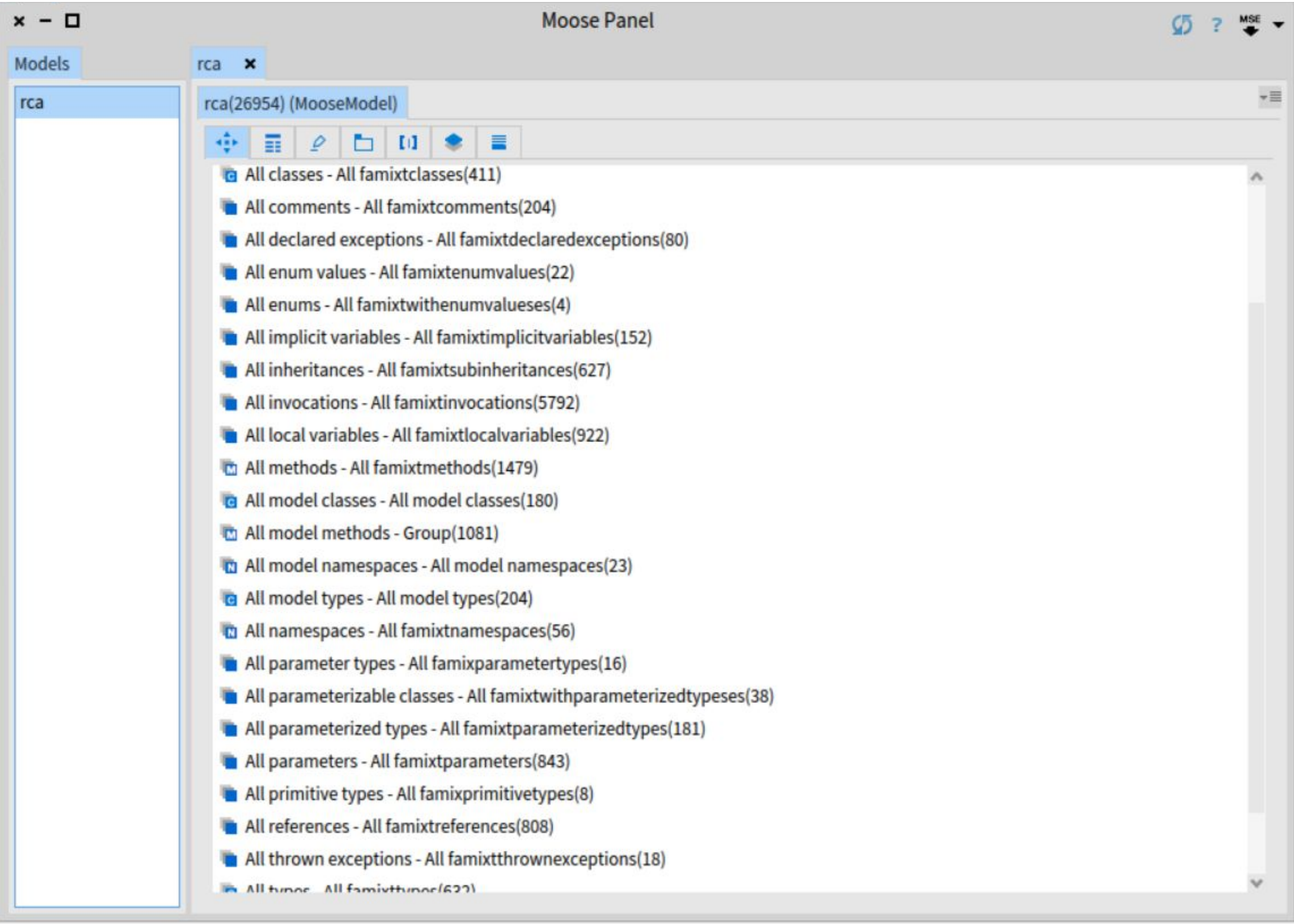
- Nb de packages
- Nb de classes
- Nb de méthodes
- Nb d’attributs
- Code mort
- Tests
- Méthodes dépréciées
- Complexité cyclomatique

- Hiérarchie de paquetages
- Diagramme de classes
- Cycles entre paquetages
- Répartitions classes/paquetages
- God classes (LOC et nb de méthodes)
- Lazy classes (LOC et nb de méthodes)



Import

```
mooseModel := MooseModel importFromMSEStream:
    msePath asFileReference readStream.
mooseModel rootFolder: rootFolderPath.
mooseModel name: 'rca'.
mooseModel install.
```



All classes - All famixtclasses(411)
All comments - All famixtcomments(204)
All declared exceptions - All famixtdeclaredexceptions(80)
All enum values - All famixtenumvalues(22)
All enums - All famixtwithenumvalueses(4)
All implicit variables - All famixtimplicitvariables(152)
All inheritances - All famixtsubinheritances(627)
All invocations - All famixtinvocations(5792)
All local variables - All famixtlocalvariables(922)
All methods - All famixtmethods(1479)
All model classes - All model classes(180)
All model methods - Group(1081)
All model namespaces - All model namespaces(23)
All model types - All model types(204)
All namespaces - All famixtnamespaces(56)
All parameter types - All famixparametertypes(16)
All parameterizable classes - All famixtwithparameterizedtypeses(38)
All parameterized types - All famixtparameterizedtypes(181)
All parameters - All famixtparameters(843)
All primitive types - All famixprimitivetypes(8)
All references - All famixtreferences(808)
All thrown exceptions - All famixtthrownexceptions(18)
All types - All famixttypes(632)

Paquetages/Classes/Méthodes

```
mooseModel allModelNamespaces
mooseModel allModelClasses
mooseModel allModelMethods
```

Dans le projet RCA

```
mooseModel allX
```

Dans le projet RCA
+
Les dépendances extérieures

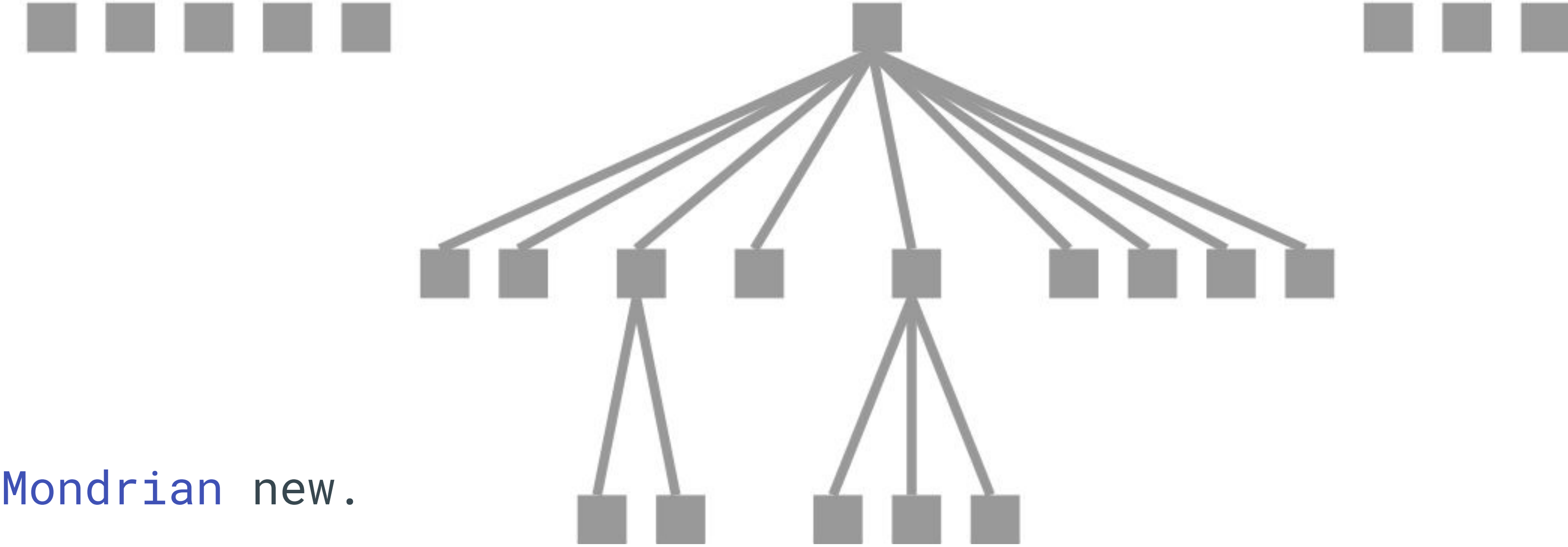
Compter le nombre d’éléments

```
“Nombre de classes”
mooseModel allModelClasses size. “180”

“Nombre de lignes de code”
mooseModel allModelClasses sum: #numberOfLinesOfCode. “11613”

“Nombre de lignes de code - aussi”
mooseModel allModelClasses sum:
    [ :class | class numberOfLinesOfCode]. “11613”
```

Chercher des annotations

```
“Sélectionner les classes avec une annotation”
mooseModel allModelClasses
    select: [:class | class annotationInstances isNotEmpty ].

“Sélectionner les classes dépréciées”
(mooseModel allModelClasses select: [:class |
    class annotationInstances isNotEmpty
      and: [ class annotationTypes anySatisfay:
        [:type | type name = ‘Deprecated’ ]]).
```

Hiérarchies de paquetages



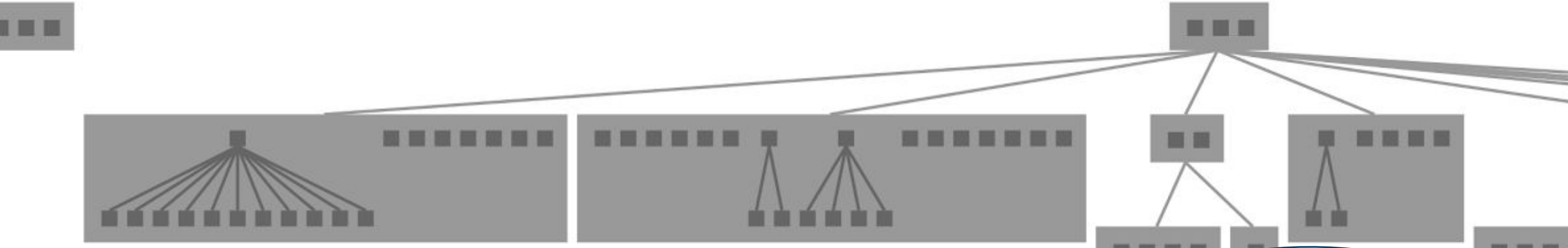
```
| b |
b := RTMondrian new.
b nodes: mooseModel allModelNamespaces.
b edgesFrom: #parentScope.
b layout tree.
b build.
b view
```

Relations classes/paquetages

```
| b |
b := RTMondrian new.
b shape rectangle.
b nodes: model allModelNamespaces forEach: [ :p |
    b nodes: p classes.
    b edges connectFrom: #superclass.
    b layout tree ].
b edgesFrom: #parentScope.
b layout tree.
b build.
^ b view
```

Relations classes/paquetages

```
| b |
b := RTMondrian new.
b shape rectangle.
```



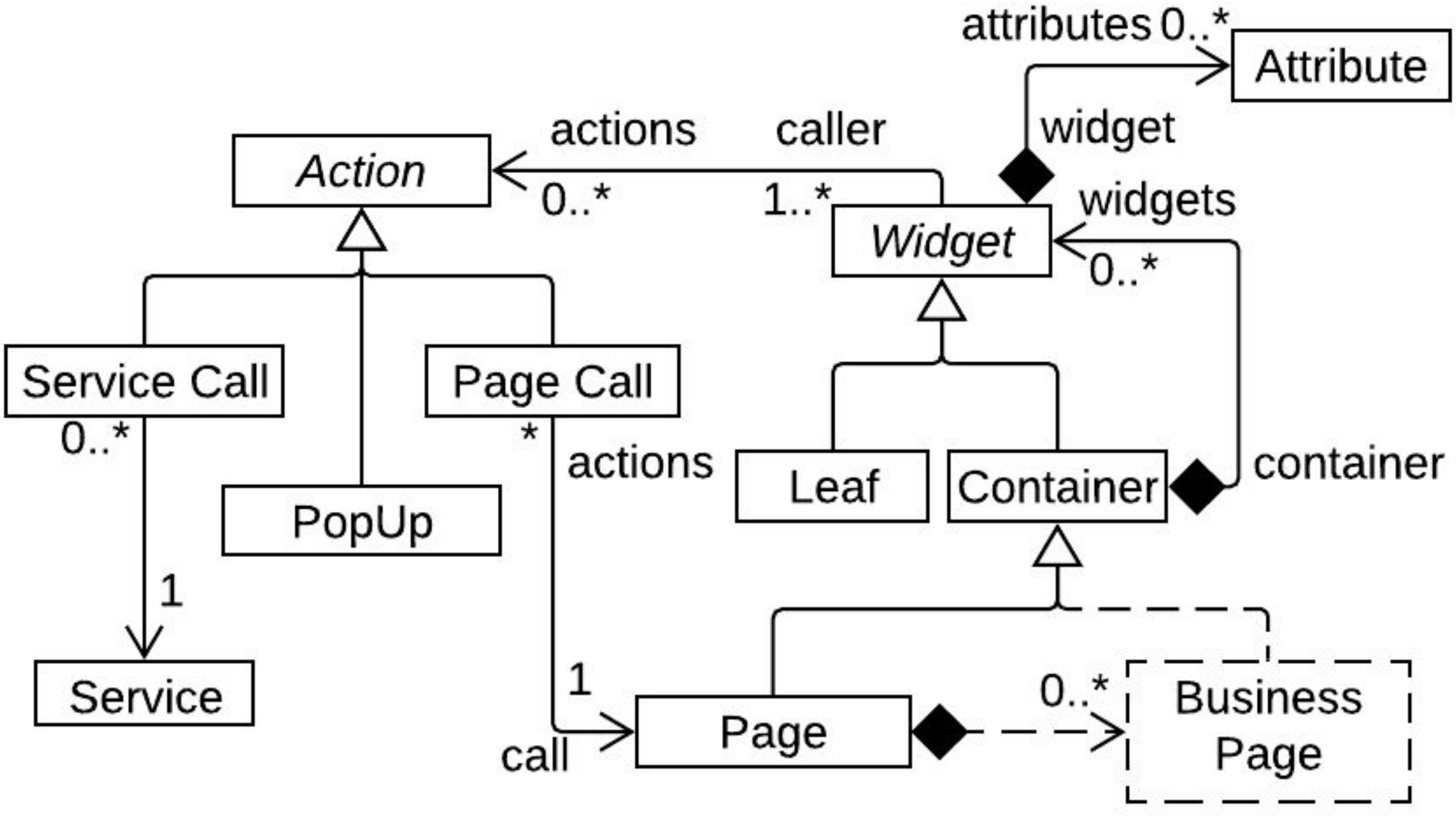
```
b layout tree.
b build.
^ b view
```

GUI Example

(oui oui, c’est mon travail)

Comment représenter une Interface Graphique ?

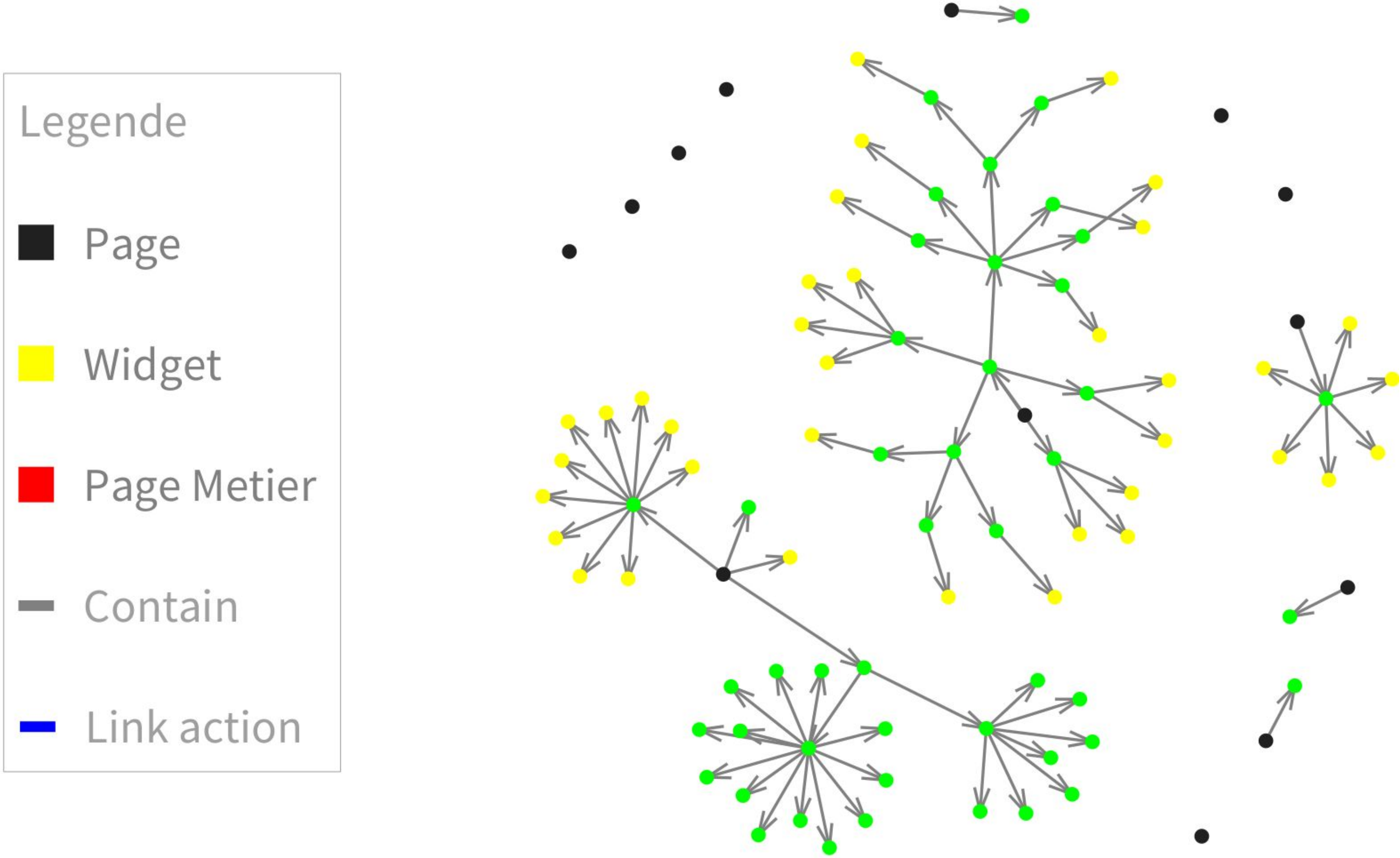
Comment représenter une Interface Graphique ?



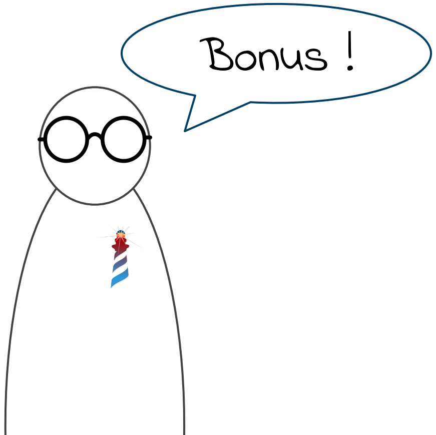
General visu

```
| b |
b := RTMondrian new.
b shape circle color: [ :cl | cl color ].
b nodes: (aBLModel allBLWidget , aBLModel allBLService) asSet
    asOrderedCollection.
(b shape line arrowedLine
   headOffset: 1.0;) shape head
   baseSize: 5.0;
   size: 8.
b edges useAssociations: (aBLModel allBLWidget collect:
#allAssociations) flatten.
b layout force strength: 0.4.
b view pushBackEdges.
^ b
```

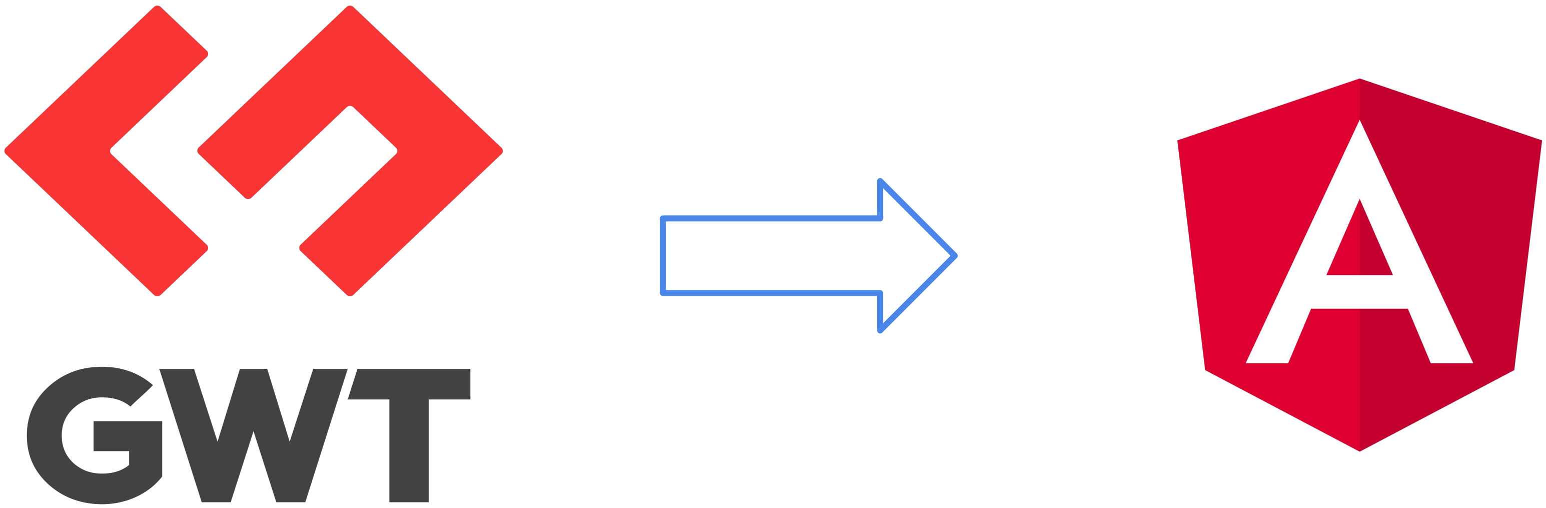
Et pour RCA ?



Stage

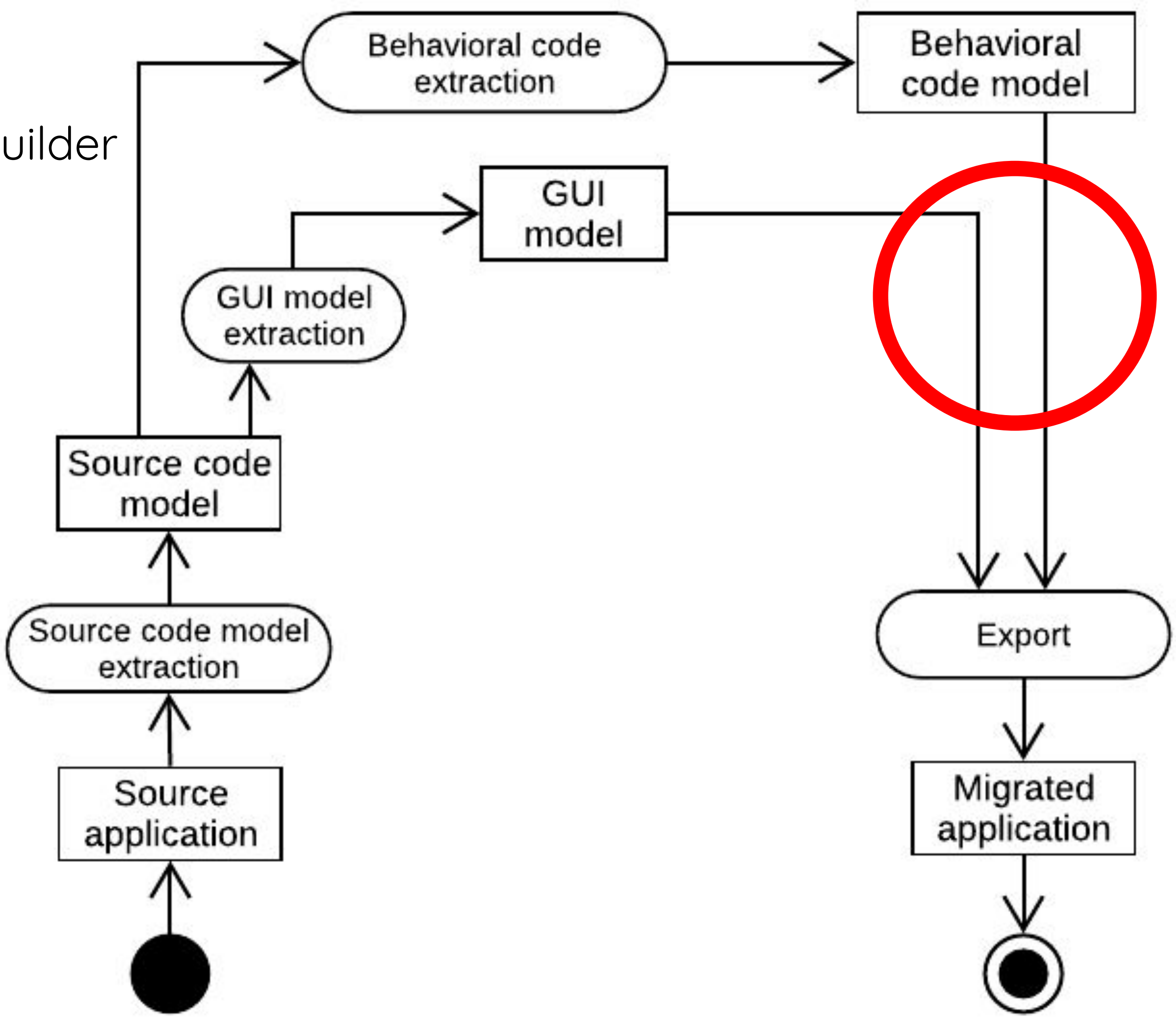


Objectif



Stage

Création d’un GUI Builder


Behavioral code extraction
Behavioral code model
GUI model
GUI model extraction
Source code model
Source code model extraction
Source application
Export
Migrated application