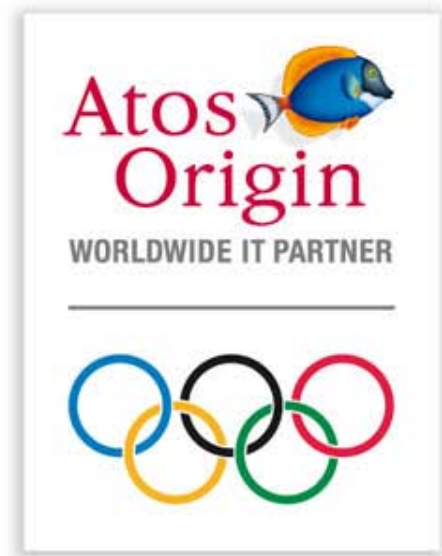




- >> AMELIORER LA PERFORMANCE
- >> AUGMENTER LA SOUPLESSE
- >> ASSURER LA TRANSPARENCE
- >> REDUIRE LES COUTS
- >> AMELIORER LA RELATION CLIENT
- >> ACCELERER LA MISE SUR LE MARCHÉ
- >> INNOVER
- >> AMELIORER L'EFFICACITE
- >> AMELIORER L'ADAPTABILITE
- >> GARANTIR LA CONFORMITE



# ATL Workshop : UML Quality Analysis presentation

## UML models measurements with ATL

**AtlanMod / Atos Origin Integration project**  
**Mathieu Vénisse**

[mathieu.venisse@gmail.com](mailto:mathieu.venisse@gmail.com)

# Contents

- UML models measurements
- Goals
- General process
- Sub processes
- Example
- Conclusion

# Contents

- UML models measurements
- Goals
- General process
- Sub processes
- Example
- Conclusion

# UML models measurements

- Formal software design described with UML is a good practice for the objects oriented application development ;
- Standard rules are developed to assist in the achievement of reliable, structured, scalable and maintainable software ;
- UML metrics are defined to help the analysts and the designers to make well-constructed softwares ;
- Most of the available tools are design to works on the implementation code.

# Contents

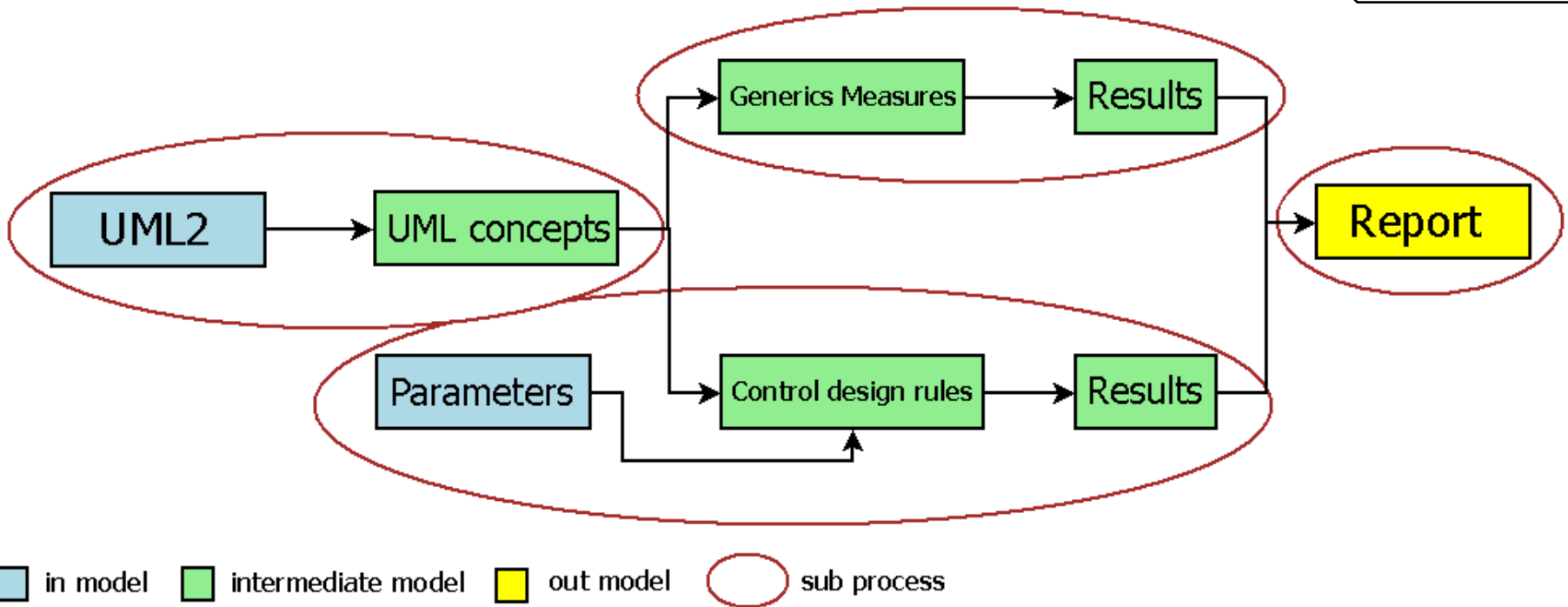
- UML models measurements
- **Goals**
- General process
- Sub processes
- Example
- Conclusion

# Goals

- This project have to be independant of implementation languages ;
- Compute generic metrics
  - Chidamber & Kemerer metrics,
  - Brito e Abreu metrics ;
- Generic control design rules, with possibility to configure them, used to check if a project respects specific design ;
- Automatic generation of reports (PDF and HTML format)
- The input model is conform to UML 2 metamodel (extension .uml).

# Contents

- UML models measurements
- Goals
- **General process**
- Sub processes
- Example
- Conclusion



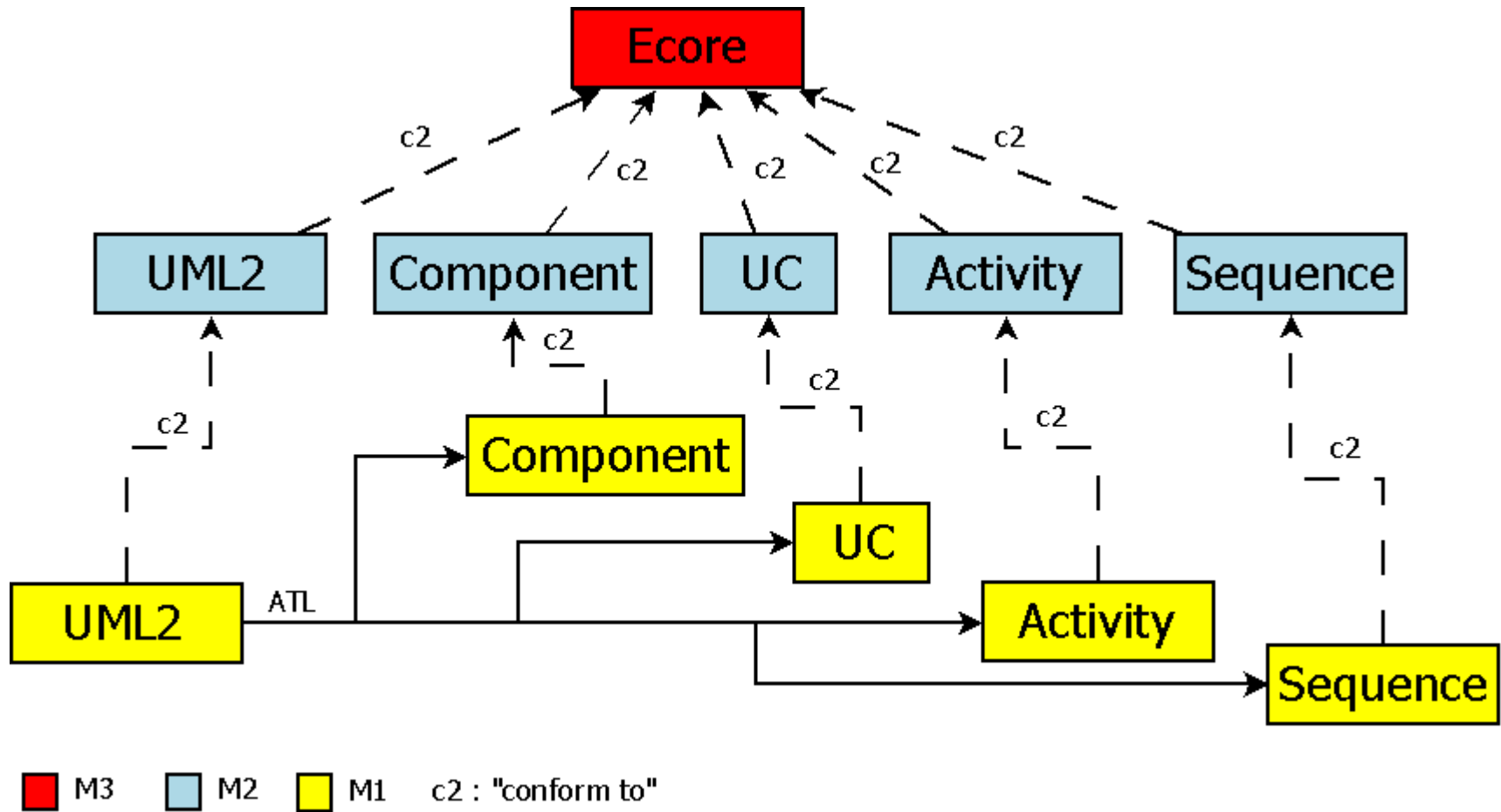
- Four sub processes :
  - Uml representation, generic measurements, control design rules and report generation



# Contents

- UML models measurements
- Objectives
- General process
- Sub processes
  - UML representation to inner representation
  - Generic measurements
  - Control design rules
  - Operating results
- Example
- Conclusion

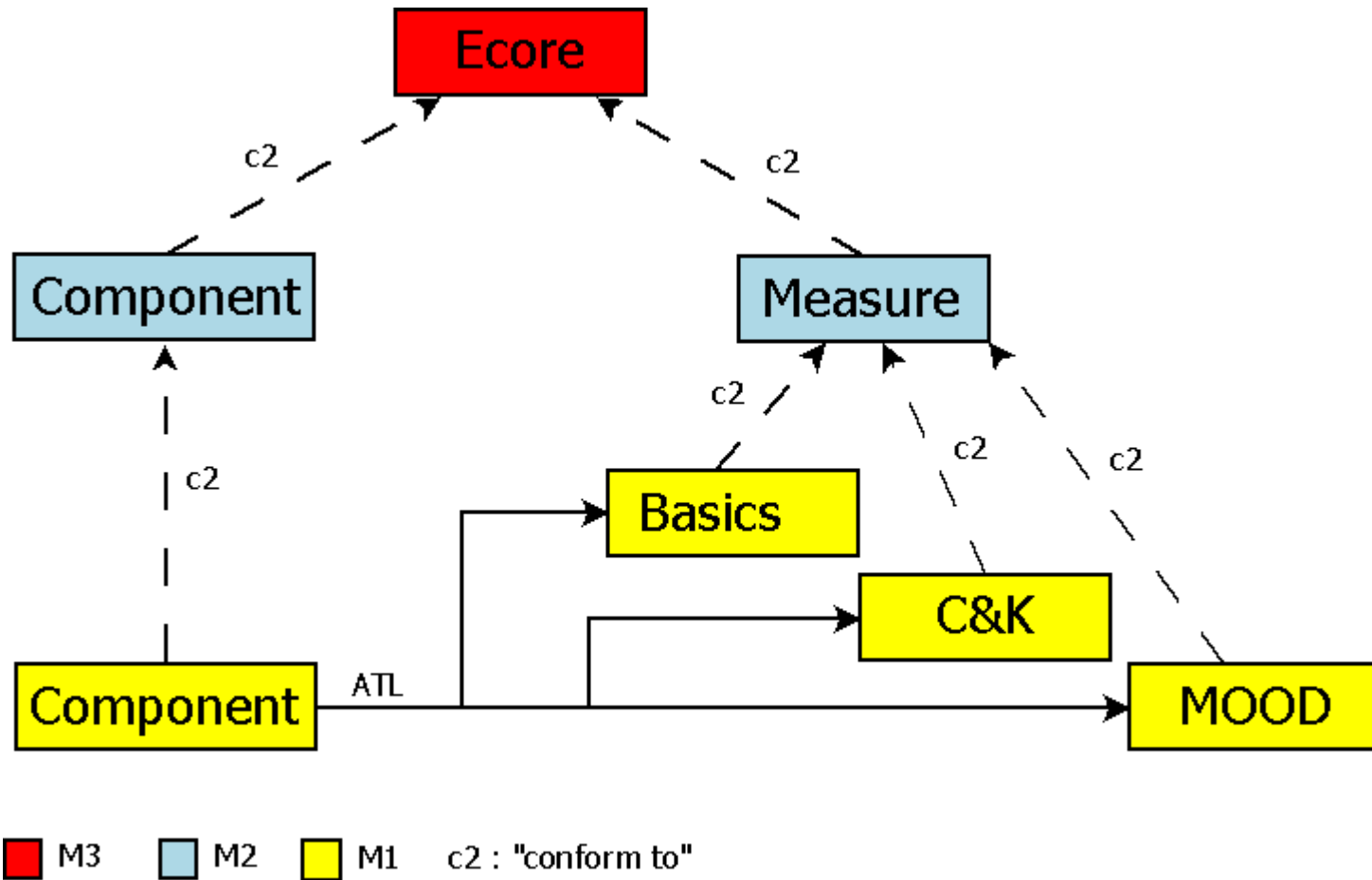
# UML to inner representation



## UML to inner representation (continued)

- Input UML models can be directly created by an UML modeler conforming to the UML 2 metamodel (*TopCased* for example).
- **Advantages :**
  - Separation of concerns :
    - The internal representations separate the original model in several models that depend on the measurements that will be made thereafter ;
    - Small models : ATL transformation are easier to write and faster to compute ;
  - Scalability :
    - If the UML metamodel evolves, just a few transformations must be build to date
- **Inconvenients :**
  - Add one more step in the general process

# Generics measurements

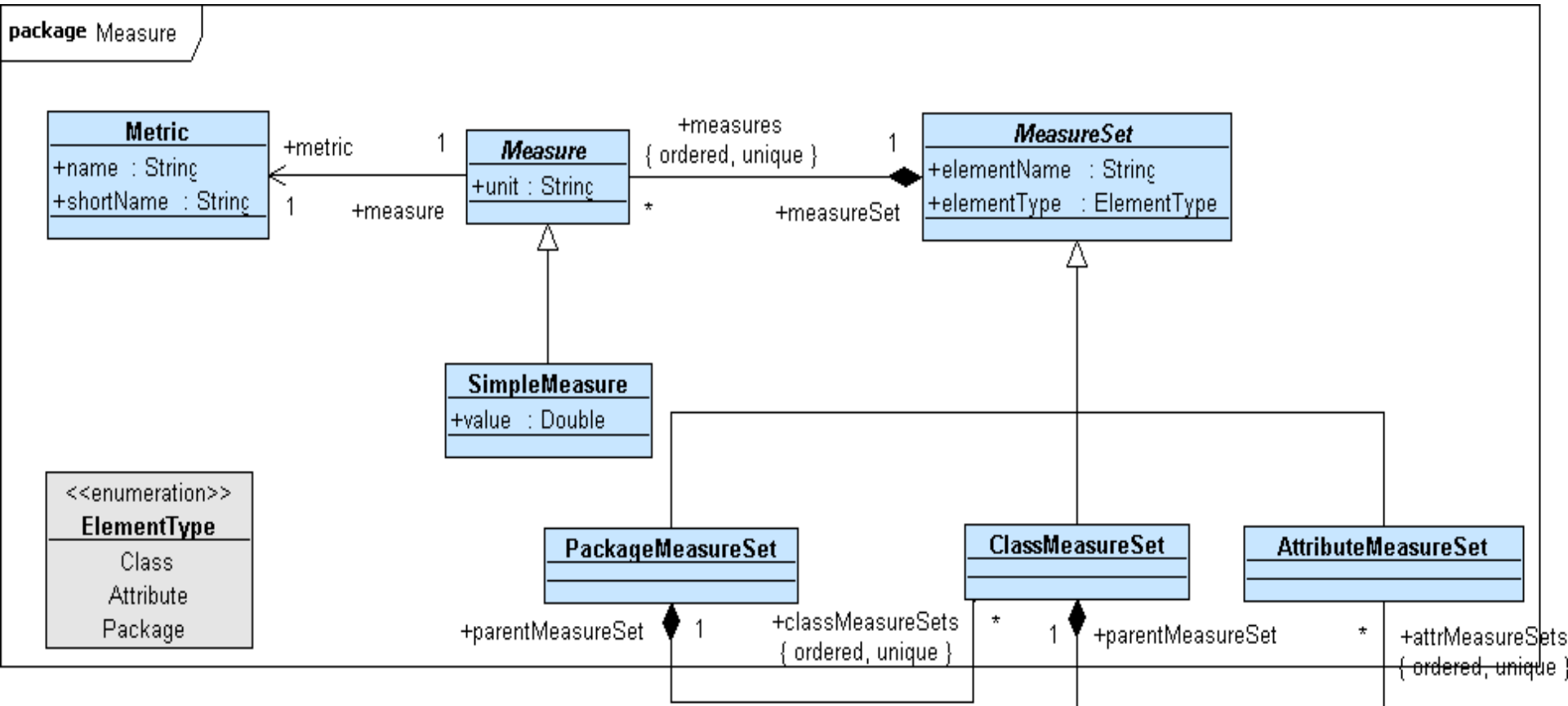


# Contents

- UML models measurements
- Objectives
- General process
- Sub processes
  - UML representation to inner representation
  - **Generic measurements**
  - Control design rules
  - Operating results
- Example
- Conclusion

# Generics measurements (continued)

## Measure metamodel



# Generics measurements (continued)

Three kinds of measurements :

## Basics metrics

Acronym	Description	Context
TNA	Get the total number of packages	M
TNC	Get the total number of classes	P
TNA	Get the total number of attributes	P/C
TNAI	Get the total number of attributes inherited	P/C
TNM	Get the total number of methods	P/C
TNMI	Get the total number of methods inherited	P/C
TNAs	Get the total number of associations	P/C
TNAsI	Get the total number of associations inherited	P/C

- These metrics are implemented in ATL libraries (as helpers) and reused in others transformations

# Generics measurements (continued)

Three kinds of measurements :

## Chidamber & Kemerer metrics

Acronym	Description	Context
WMC	Get the weighted methods per class	C
DIT	Get the Depth inheritance tree	P/C
NOC	Get the number of children	P/C
CBO	Get the couplage between objects	P/C



# Generics measurements (continued)

Three kinds of measurements :

## Chidamber & Kemerer metrics

Acronym	Description	Context
WMC	Get the weighted methods per class	C
DIT	Get the Depth inheritance tree	P/C
NOC	Get the number of children	P/C
CBO	Get the couplage between objects	P/C

- WMC predicts the duration and the additional work needed to develop a class and gives an indication on its reusability.
- DIT estimates the complexity of a class and its reusability.
- NOC shows the influence of the class on the global system : the class should be tested more precisely.
- CBO estimates the modularity, the reusability and the maintainability effort.

# Generics measurements (continued)

Three types of measurements :

## Brito e Abreu metrics

Acronym	Description	Context
AIF	Get the attributes inheritance factor	P/C
MIF	Get the methods inheritance factor	P/C
AHF	Get the attributes hiding factor	P/C
MHF	Get the methods hiding factor	P/C

## Generics measurements (continued)

Three types of measurements :

### Brito e Abreu metrics

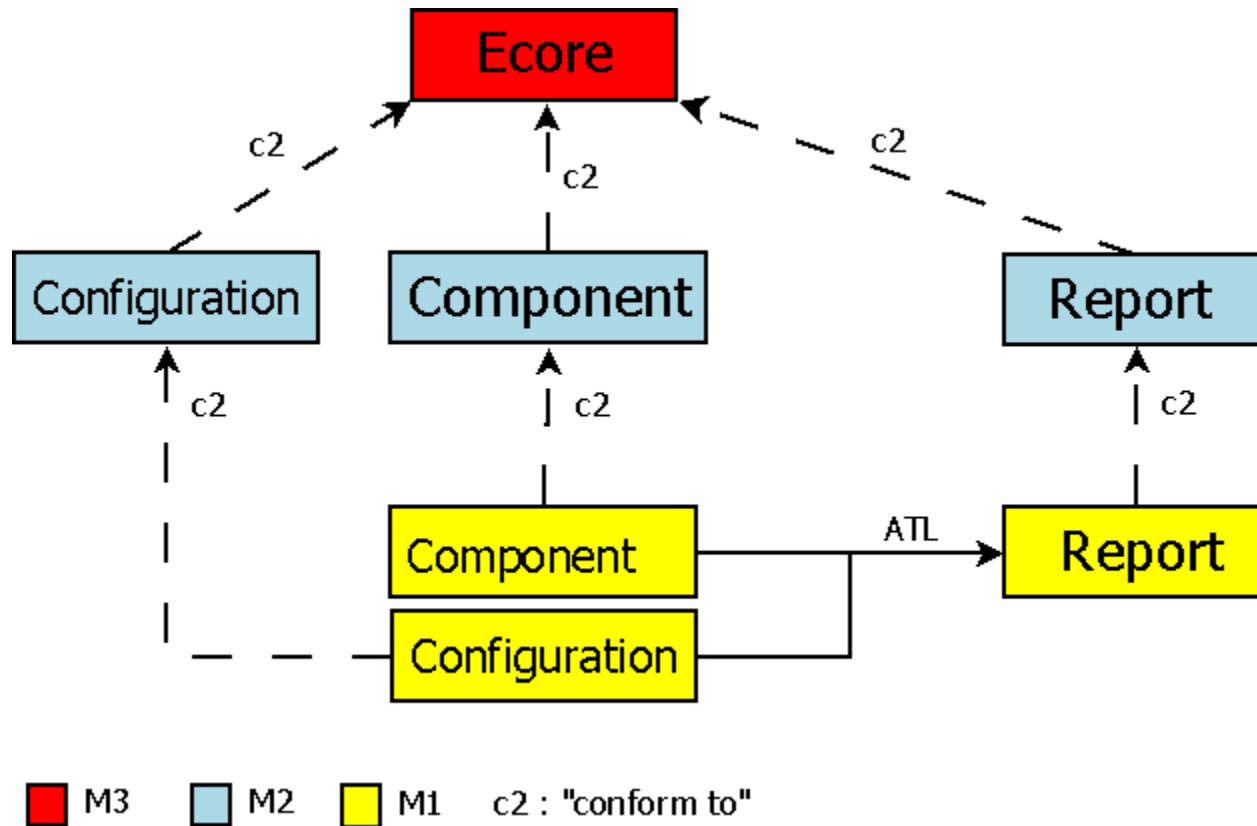
Acronym	Description	Context
AIF	Get the attributes inheritance factor	P/C
MIF	Get the methods inheritance factor	P/C
AHF	Get the attributes hiding factor	P/C
MHF	Get the methods hiding factor	P/C

- AIF and the MIF estimate the abstraction of a class.
- The AHF and the MHF estimate the encapsulation.

# Contents

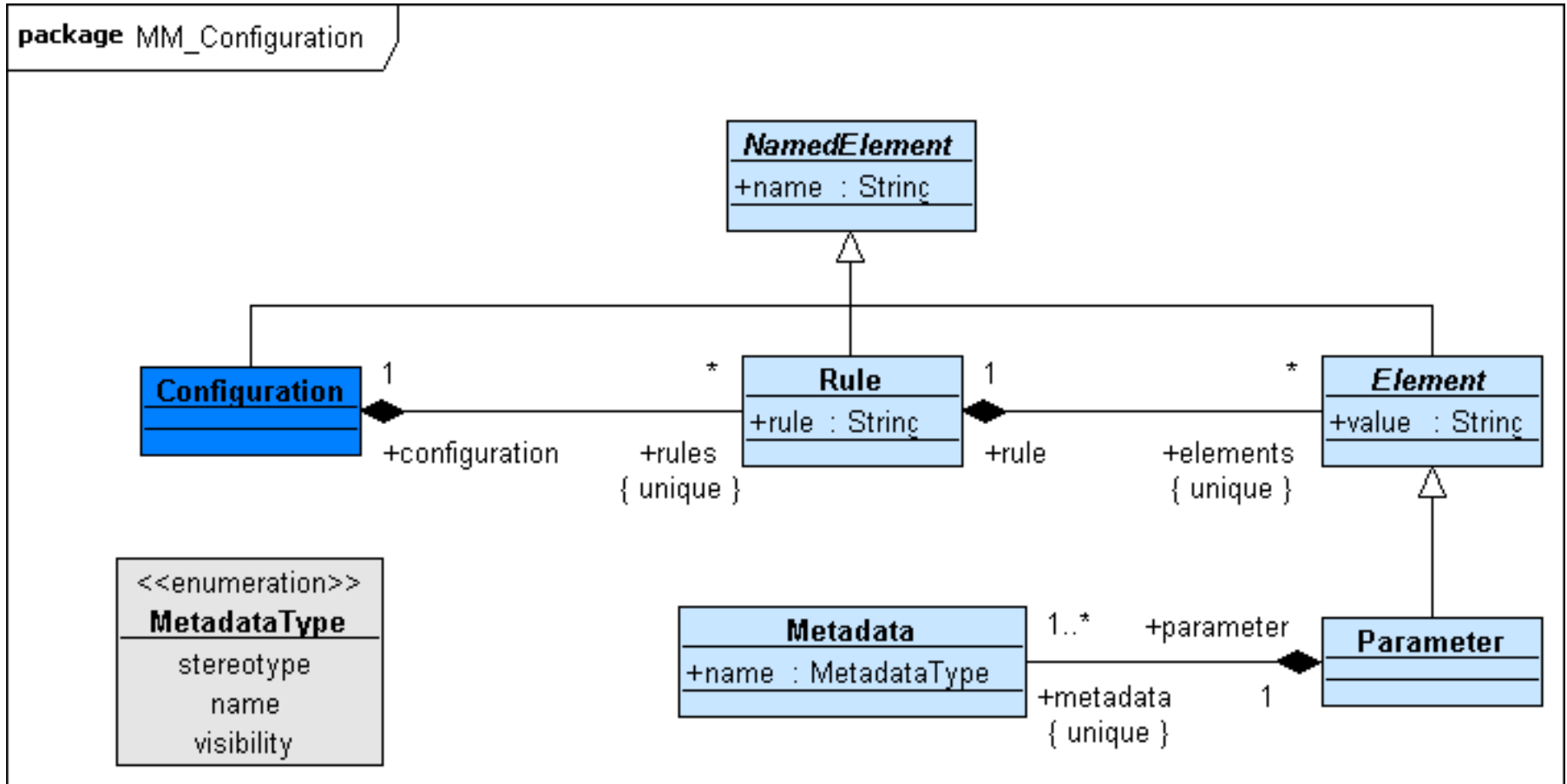
- UML models measurements
- Objectives
- General process
- Sub processes
  - UML representation to inner representation
  - Generic measurements
  - **Control design rules**
  - Operating results
- Example
- Conclusion

# Control design rules



# Control design rules (continued)

## Configuration metamodel



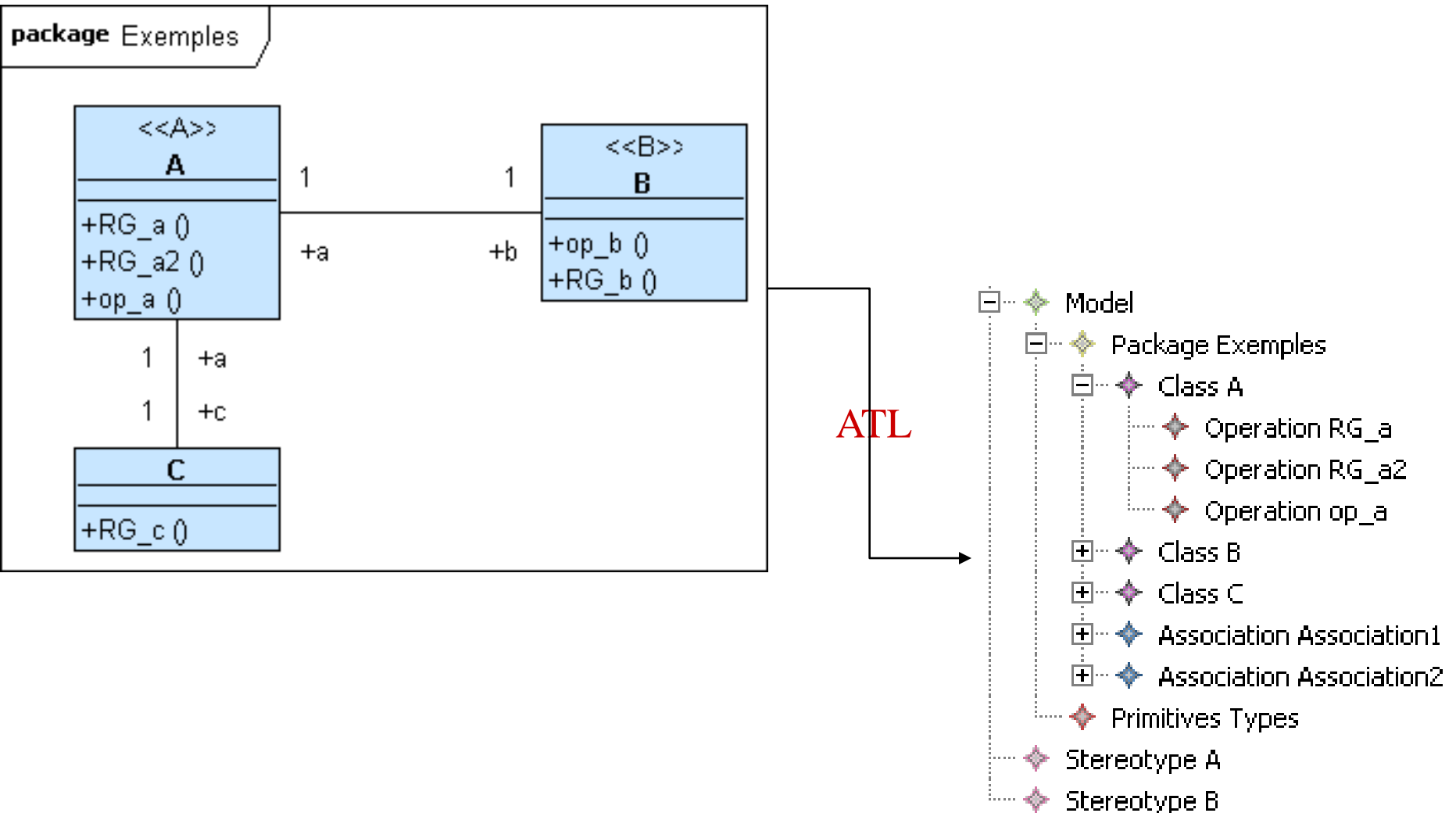
# Control design rules (continued)

## Generic control design rules

Name	Description
LinkedClasses	Find classes that are linked together
NotLinkedClasses	Find classes that are not linked together
OperationInClass	Find operation that are in class
OperationNotInClass	Find operation that are not in class
ExistInterface	Find interface that are implemented by classes
ClassWithoutInterface	Find classes that not implement interface
ClassWithSpecificInterface	Find classes that implements interface
ClassInPackage	Find classes that are in package

# Control design rules (continued)

## Simple example





# Control design rules (continued)

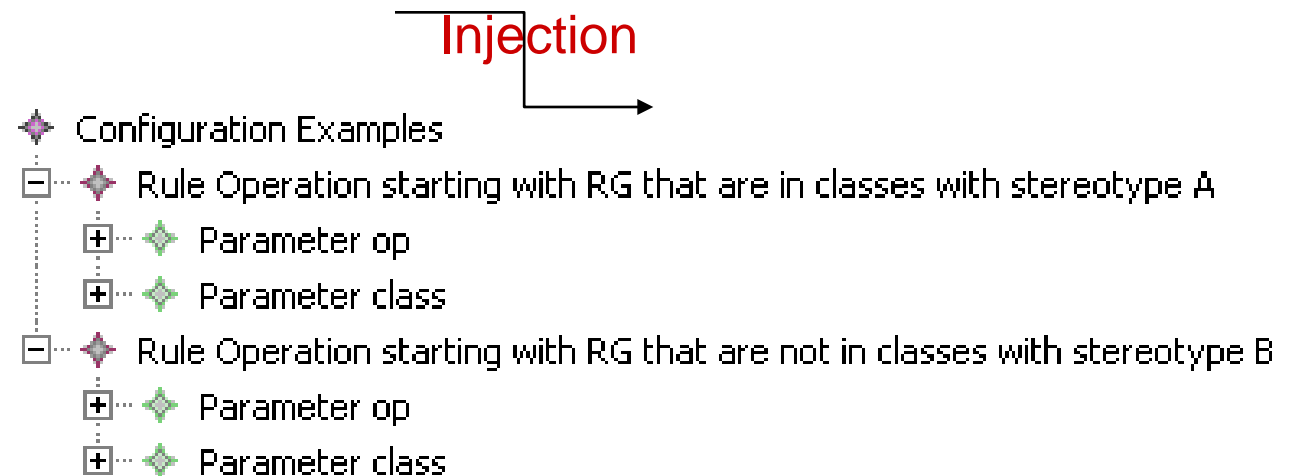
## Simple example - Configuration

```

Configuration Examples {
  rule OperationsInClass "Operation starting with RG that are in classes with stereotype A" {
    op : metadata <- (name), value <- RG;
    "class" : metadata <- (stereotype), value <- A;
  }

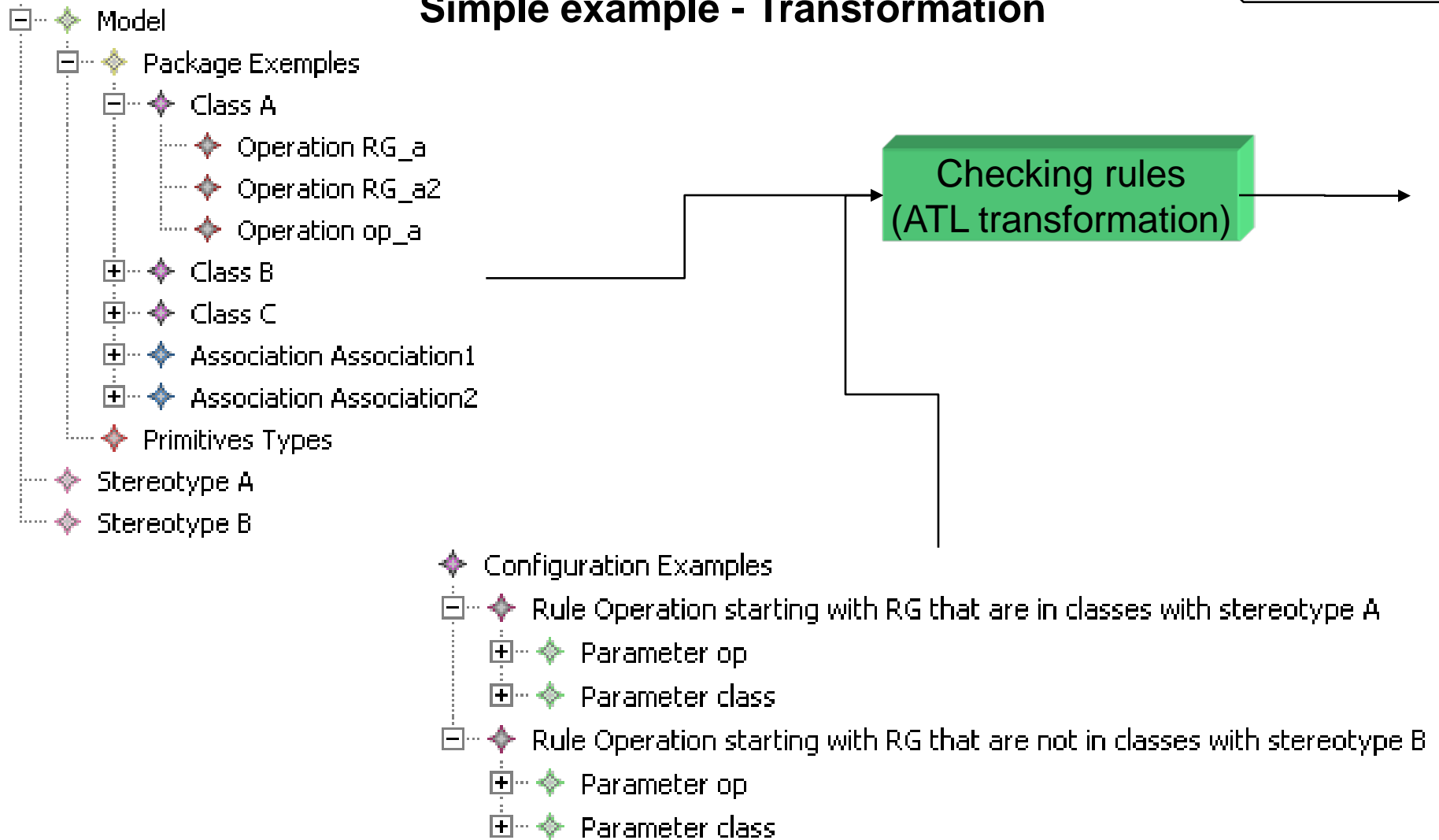
  rule OperationsNotInClass "Operation starting with RG that are not in classes with stereotype B" {
    op : metadata <- (name), value <- RG;
    "class" : metadata <- (stereotype), value <- B;
  }
}

```



# Control design rules (continued)

## Simple example - Transformation



# Control design rules (continued)

## Simple example - Transformation

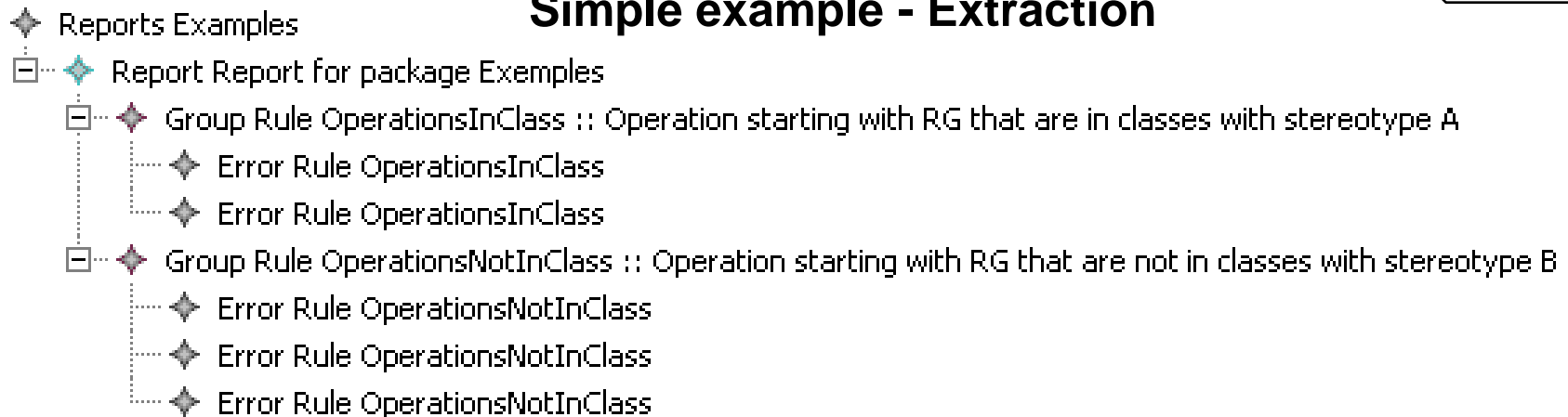
Checking rules  
(ATL transformation)



- ◆ Reports Examples
  - ▢ ◆ Report Report for package Exemples
    - ▢ ◆ Group Rule OperationsInClass :: Operation starting with RG that are in classes with stereotype A
      - ◆ Error Rule OperationsInClass
      - ◆ Error Rule OperationsInClass
    - ▢ ◆ Group Rule OperationsNotInClass :: Operation starting with RG that are not in classes with stereotype B
      - ◆ Error Rule OperationsNotInClass
      - ◆ Error Rule OperationsNotInClass
      - ◆ Error Rule OperationsNotInClass

# Control design rules (continued)

## Simple example - Extraction



Extraction



**Reports** Examples

**Report** "Report for package Exemples"

**Error group** "Rule OperationsInClass :: Operation starting with RG that are in classes with stereotype A" :

**Error error** "Rule OperationsInClass" **context** "RG\_a in Exemples.A"

**Error error** "Rule OperationsInClass" **context** "RG\_a2 in Exemples.A"

**Error group** "Rule OperationsNotInClass :: Operation starting with RG that are not in classes with stereotype B" :

**Error error** "Rule OperationsNotInClass" **context** "RG\_a in Exemples.A with stereotype : A - "

**Error error** "Rule OperationsNotInClass" **context** "RG\_a2 in Exemples.A with stereotype : A - "

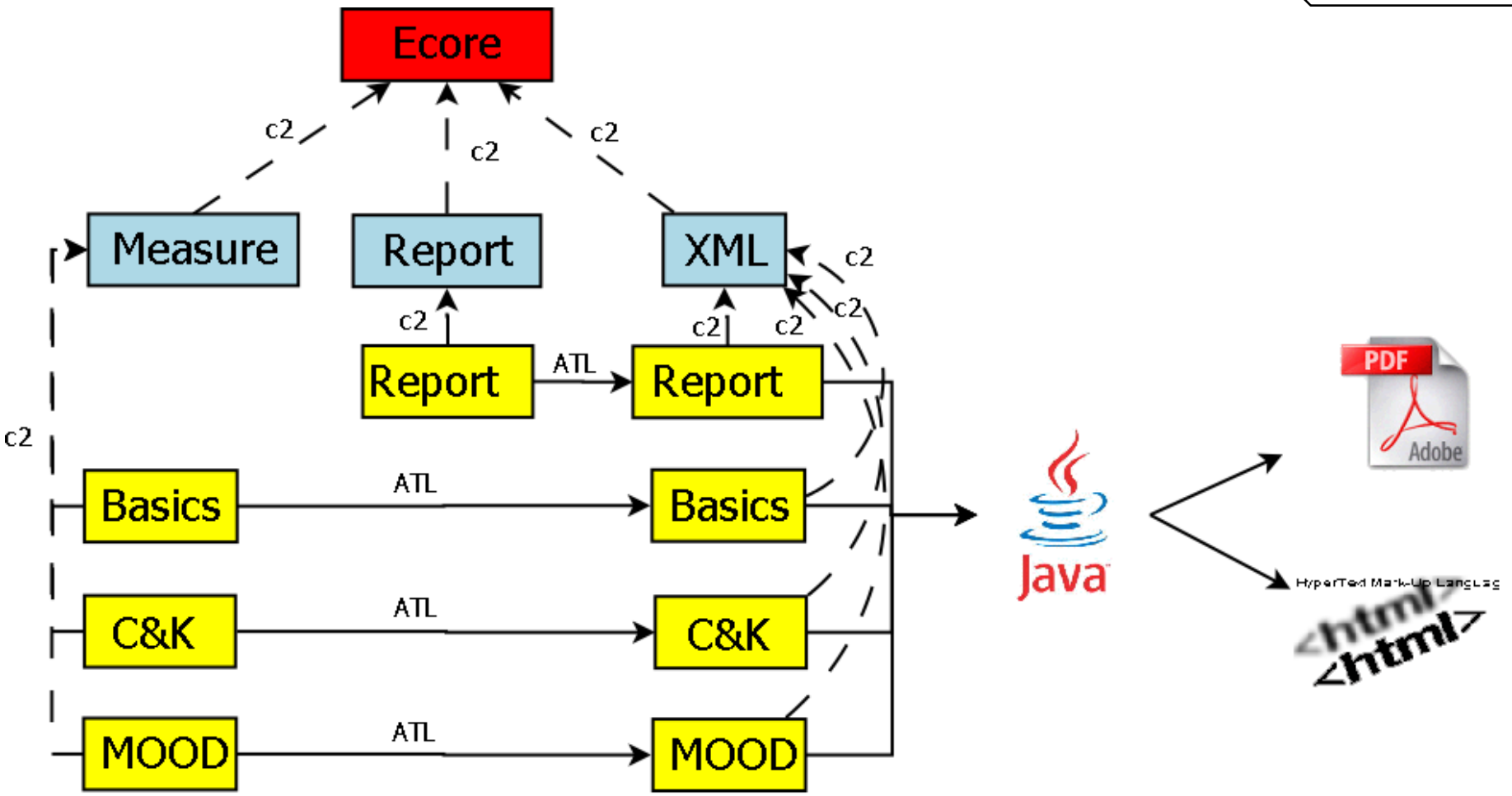
**Error error** "Rule OperationsNotInClass" **context** "RG\_c in Exemples.C with stereotype : no"

# Control design rules (continued)

## Advantages

- Easy and intuitive configuration based on rules concepts ;
- Generic parametrizable rules for all specific design ;
- Readable reports of errors automatically generated ;
- For each generic rule : 9 specific rules
  - 3 métadata (name, visibility, stereotype)
  - 2 rule's parameters -> 9 combinations
- For each generic rule there are an infinity of configured specific rules

# Operating results



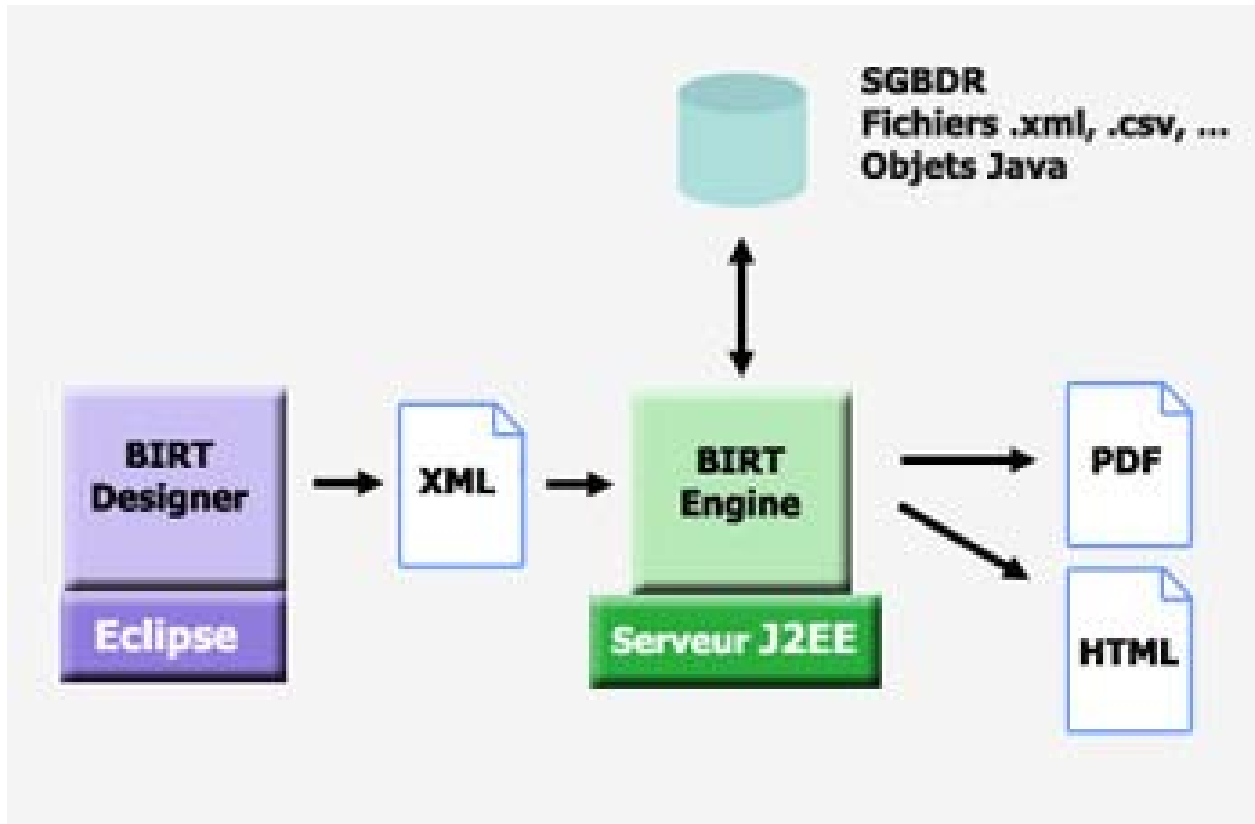
M3
  M2
  M1
 c2 : "conform to"

# Contents

- UML models measurements
- Objectives
- General process
- Sub processes
  - UML representation to inner representation
  - Generic measurements
  - Control design rules
  - **Operating results**
- Example
- Conclusion

# Operating results (continued)

## BIRT API





# Contents

- UML models measurements
- Goals
- General process
- Sub processes
- Example
- Conclusion

# Example



## UMLQA report

### General measurements :

Total number of packages	<b>220</b>
Total number of classes	<b>362</b>
Total number of attributes (inherited included)	<b>791</b>
Total number of attributes inherited	<b>385</b>
Total number of operations (inherited included)	<b>164</b>
Total number of operations inherited	<b>2</b>
Total number of associations (inherited included)	<b>828</b>
Total number of association inherited	<b>303</b>

# Example (continued)

## 2- Chidamber and Kemerer measurements

### 2.1- Depth Inheritance Tree (DIT)

This measurement permits to know the depth of the inheritance tree for a package, for a class and the average DIT for a package.

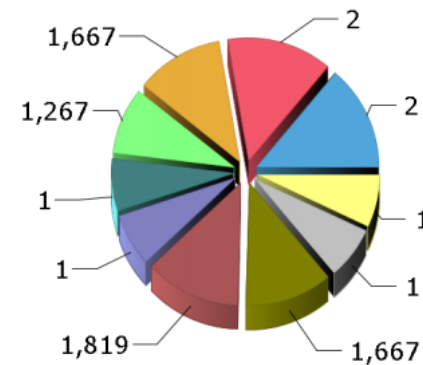
#### 2.1.1- Average DIT per class in package

Package	DIT
MAS.domain.autorisations.autorisation.amep	2
MAS.domain.consolidations.consolidation	2
MAS.domain.referentiel.statique	1,819
MAS.domain.encours.provisions	1,667
MAS.domain.risques.pays	1,667
MAS.domain.limites.limite	1,267
MAS.domain.referentiel.dynamique	1
MAS.domain.referentiel.devises	1
MAS.domain.transverse.langue	1
MAS.domain.transverse.montant	1

Formula for the average DIT in a

package :

$$\frac{\sum_{class} DIT}{\sum_{class}}$$



# Example (continued)

## 4- UMLQA report

### Errors

*Rule LinkedClasses :: Classes de stereotypes bo reliées à des classes de stéréotype bo*

MAS.domain.referentiel.dynamique.AliasBO -> MAS.domain.referentiel.statique.ConsolidationComptableBO

# Contents

- UML models measurements
- Goals
- General process
- Sub processes
- Example
- Conclusion

# Conclusion

- UMLQualityAnalysis computes generic measurements on UML models without concern about implementation ;
- UMLQualityAnalysis checks if a project is consistent with specific design rules ;
- UMLQualityAnalysis automatically generates readable reports.

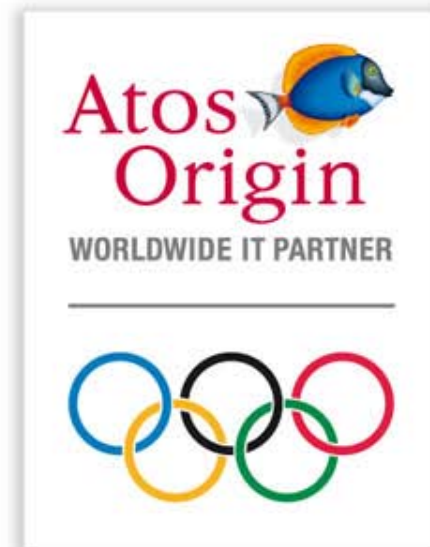
**End of presentation**

**Thanks**  
**Questions, remarks ?**

**AtlanMod / Atos Origin Integration project**  
**Mathieu Vénisse**

[mathieu.venisse@gmail.com](mailto:mathieu.venisse@gmail.com)

- >> AMELIORER LA PERFORMANCE
- >> AUGMENTER LA SOUPLESSE
- >> ASSURER LA TRANSPARENCE
- >> REDUIRE LES COUTS
- >> AMELIORER LA RELATION CLIENT
- >> ACCELERER LA MISE SUR LE MARCHÉ
- >> INNOVER
- >> AMELIORER L'EFFICACITE
- >> ACCROITRE L'ADAPTABILITE
- >> GARANTIR LA CONFORMITE



Mathieu Vénisse

[mathieu.venisse@gmail.com](mailto:mathieu.venisse@gmail.com)

Immeuble Pascal – 6 Impasse Augustin Fresnel  
44812 St Herblain Cedex  
[www.atosorigin.fr](http://www.atosorigin.fr)