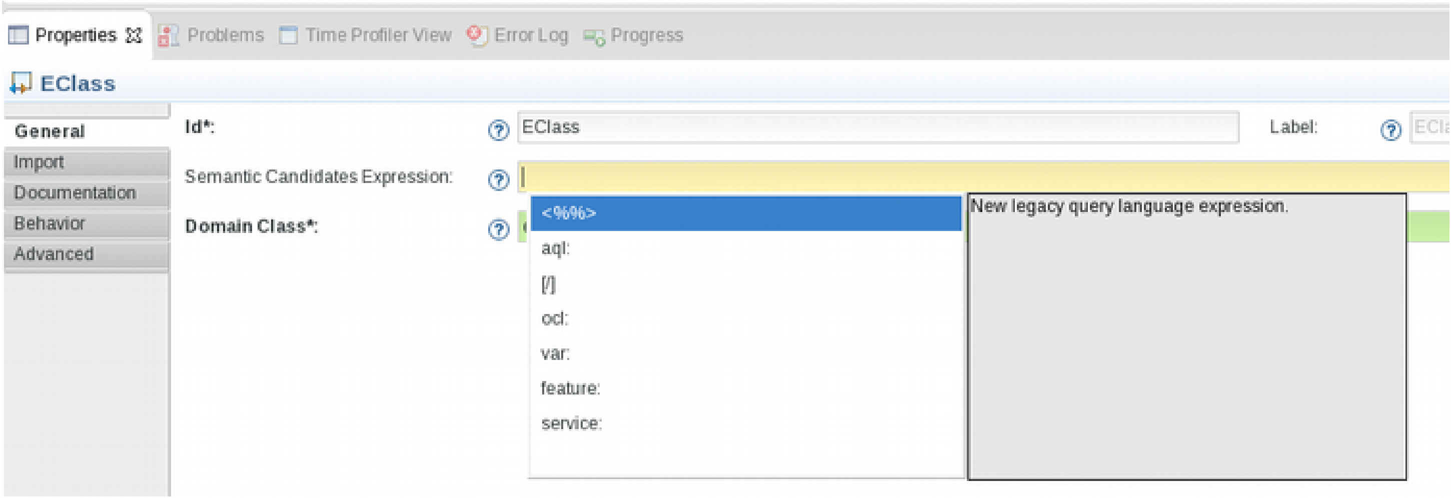
ACCELEO QUERY LANGUAGE

The small, fast & strong sidekick for your tooling

A Guided Tour of Eclipse Modeling, Monday, October 23, 2017

our Motivation



Id*: EClass

Label: ECla

Semantic Candidates Expression:

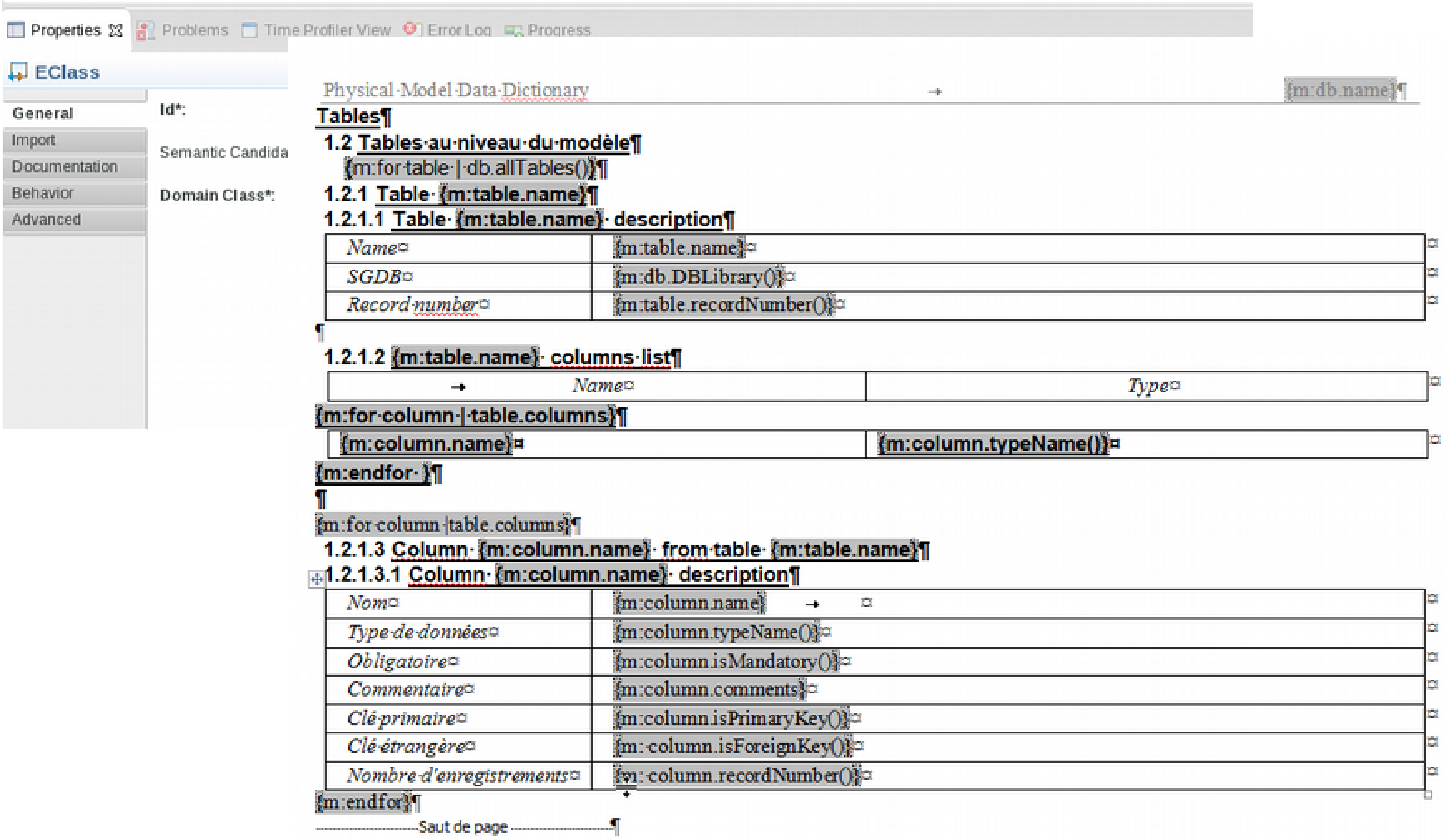
Domain Class*:

```
<%%>
aql:
[/]
ocl:
var:
feature:
service:
```

New legacy query language expression.

Query language for **Sirius**

our Motivation



Physical·Model·Data·Dictionary → {m:db.name}¶

Tables¶

1.2 Tables·au·niveau·du·modèle¶

{m:for·table·|·db.allTables()}¶

1.2.1 Table·{m:table.name}¶

1.2.1.1 Table·{m:table.name}·description¶

Name¤	{m:table.name}¤	¤
SGDB¤	{m:db.DBLibrary()}¤	¤
Record·number¤	{m:table.recordNumber()}¤	¤

¶

1.2.1.2 {m:table.name}·columns·list¶

→ *Name*¤	*Type*¤	¤

{m:for·column·|·table.columns}¶

{m:column.name}¤	{m:column.typeName()}¤	¤

{m:endfor·}¶

¶

{m:for·column·|table.columns}¶

1.2.1.3 Column·{m:column.name}·from·table·{m:table.name}¶

1.2.1.3.1 Column·{m:column.name}·description¶

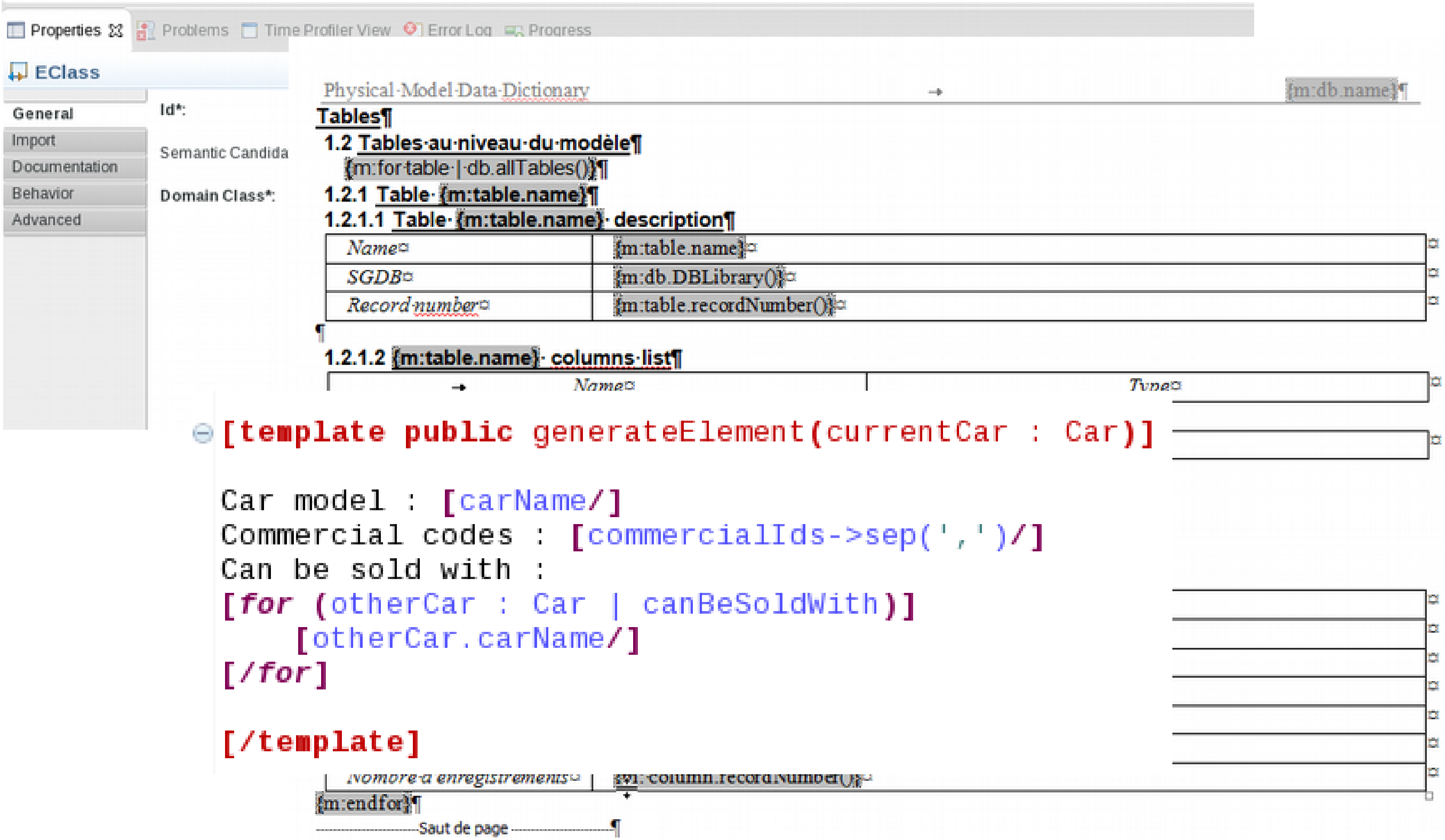
Nom¤	{m:column.name} → ¤	¤
Type·de·données¤	{m:column.typeName()}¤	¤
Obligatoire¤	{m:column.isMandatory()}¤	¤
Commentaire¤	{m:column.comments}¤	¤
Clé·primaire¤	{m:column.isPrimaryKey()}¤	¤
Clé·étrangère¤	{m:·column.isForeignKey()}¤	¤
Nombre·d'enregistrements¤	{m:·column.recordNumber()}¤	¤

{m:endfor}¶

------------Saut de page------------¶

Expressions in **M2Doc**

our Motivation



Physical Model Data Dictionary → {m:db.name}¶

Tables¶

1.2 Tables au niveau du modèle¶

{m:for table | db.allTables()}¶

1.2.1 Table {m:table.name}¶

1.2.1.1 Table {m:table.name} description¶

Name	{m:table.name}
SGDB	{m:db.DBLibrary()}
Record number	{m:table.recordNumber()}

¶

1.2.1.2 {m:table.name} columns list¶

Name	*Type*
Nombre d'enregistrements	{m:column.recordNumber()}

{m:endfor}¶

Saut de page ¶

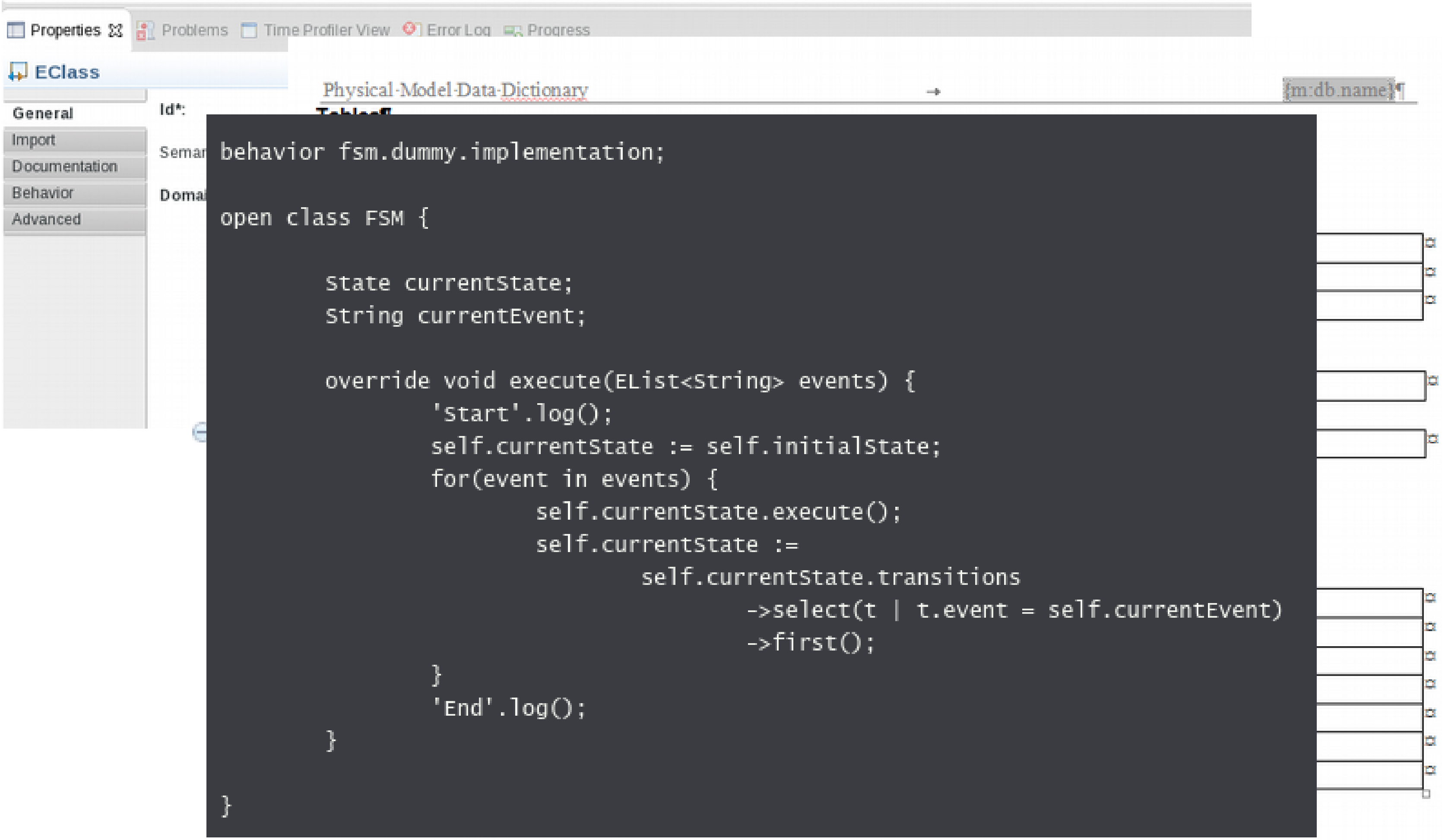
```
[template public generateElement(currentCar : Car)]

Car model : [carName/]
Commercial codes : [commercialIds->sep(',')/]
Can be sold with :
[for (otherCar : Car | canBeSoldWith)]
    [otherCar.carName/]
[/for]

[/template]
```

Expressions in **Acceleo-MTL**

our Motivation



```
behavior fsm.dummy.implementation;

open class FSM {

        State currentState;
        String currentEvent;

        override void execute(EList<String> events) {
                'Start'.log();
                self.currentState := self.initialState;
                for(event in events) {
                        self.currentState.execute();
                        self.currentState :=
                                self.currentState.transitions
                                        ->select(t | t.event = self.currentEvent)
                                        ->first();
                }
                'End'.log();
        }

}
```

Expressions in **ALE (Action Language for EMF**

Language = Syntax + Semantic + Tooling + Runtime

- **Syntax :**
  - Familiar for any user of OCL
  - Easily extensible

Language = Syntax + Semantic + Tooling + Runtime

- **Syntax :**
  - Familiar for any user of OCL
  - Easily extensible
- **Semantics**
  - Statically typed
  - Forgiving (null/unsetted values, collections)

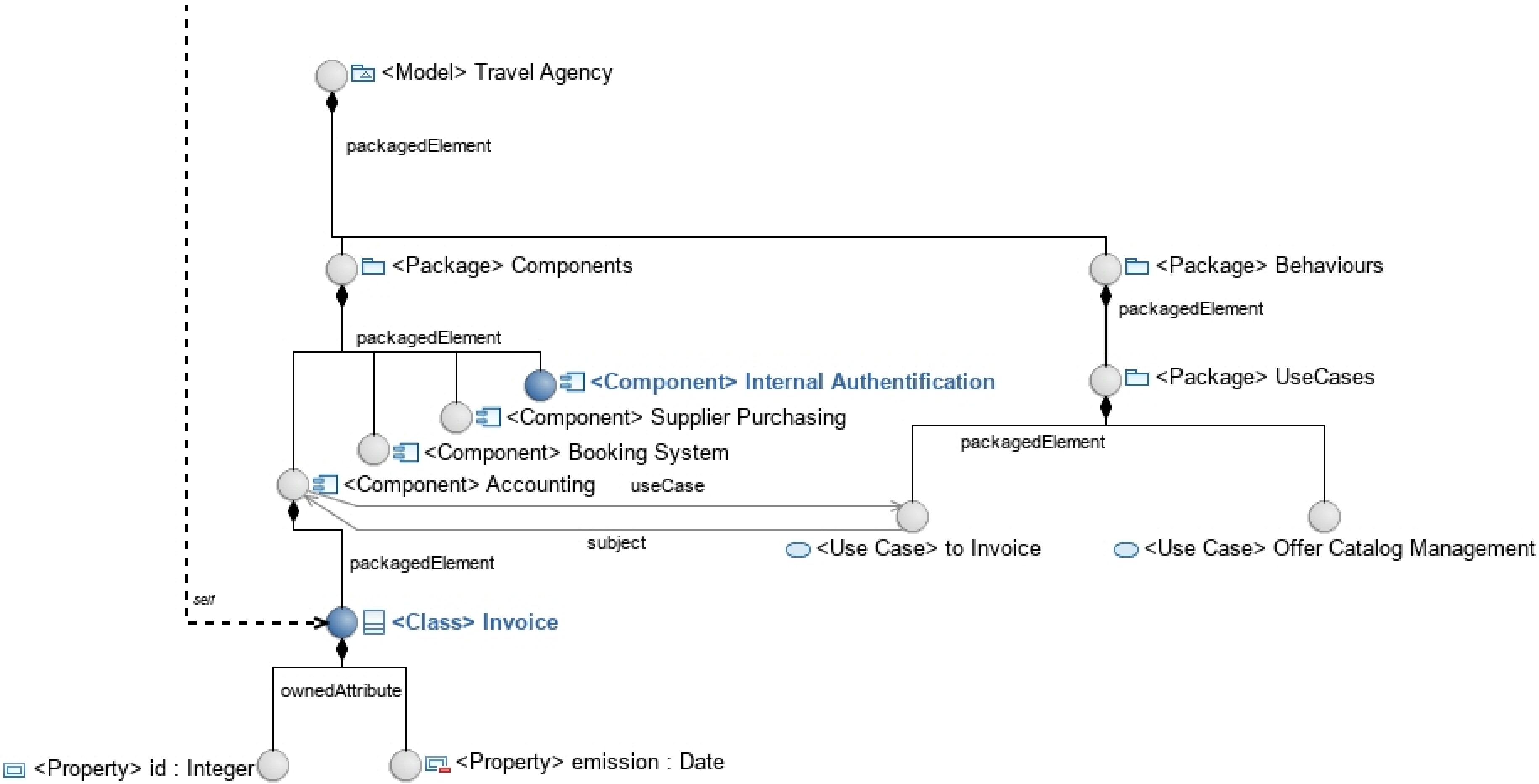
Language = Syntax + Semantic + Tooling + Runtime

- **Syntax :**
  - Familiar for any user of OCL
  - Easily extensible
- **Semantics**
  - Statically typed
  - Forgiving (null/unsetted values, collections)
- **Tooling**
  - Embeddable
  - Statically typed with rich type inference

Language = Syntax + Semantic + Tooling + Runtime

- **Syntax :**
  - Familiar for any user of OCL
  - Easily extensible
- **Semantics**
  - Statically typed
  - Forgiving (null/unsetted values, collections)
- **Tooling**
  - Embeddable
  - Statically typed with rich type inference
- **Runtime**
  - Fast, small, interpreted

```
self.eContainer(uml::Model).eAllContents()->select(a | a.name.startsWith('I'))
```



not **exactly** OCL

- **« . » and « → » notation, select()**

  ✓ collect() and flatten() are implicit in AQL (you won't have a List of Sets)

- **Optional variable denotation**

  ✗ every expression starts with a var name

- **Sequence, Set, Bag, OrderedSet**

  ✗ Only Lists & Sets, and the order is always stable across executions

- **Types (uml::Class, family::Person…)**

  ✓ are optional in a lambda

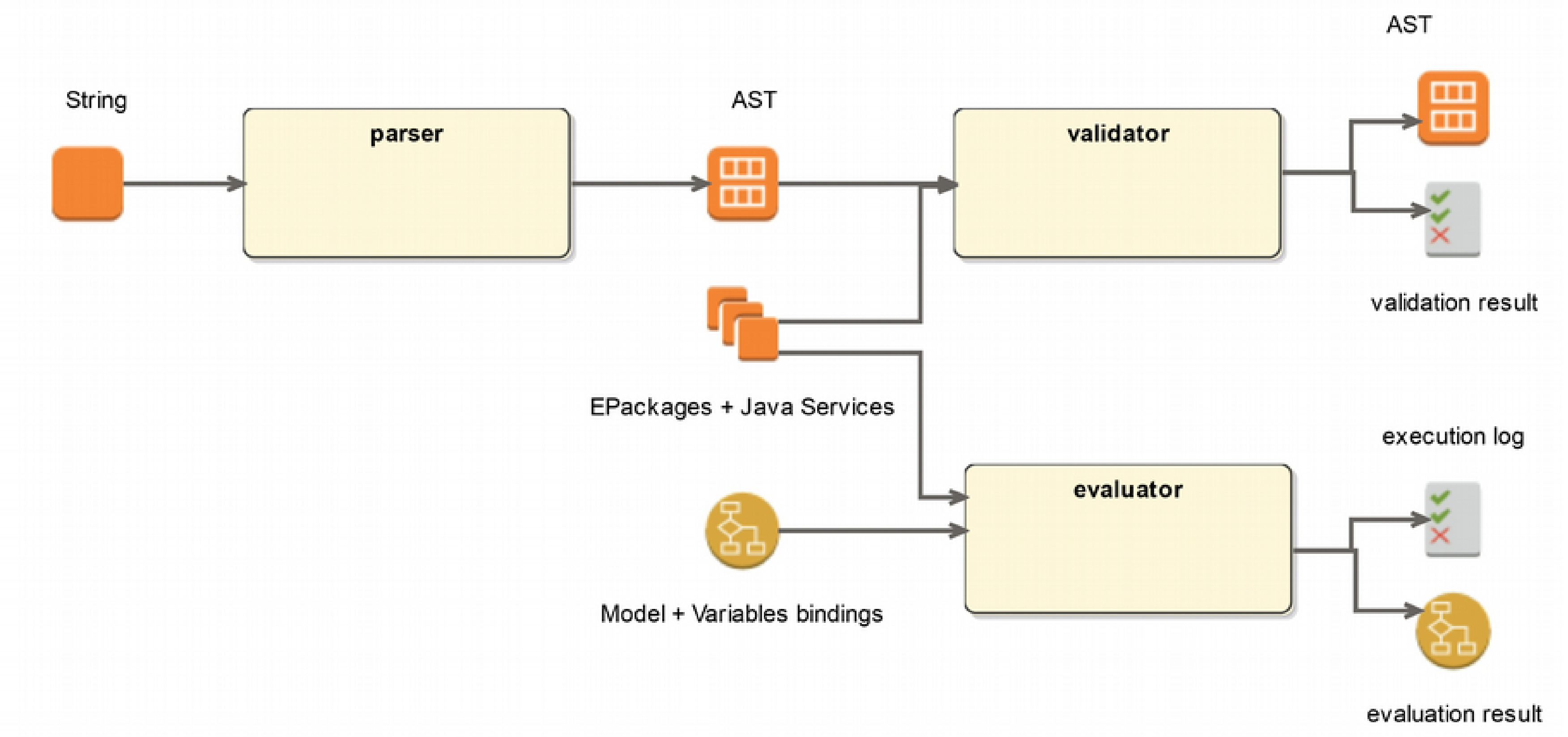
  ```
  ->select(a : family::Person| a.firstName.size > 10)
  ->select(a | a.firstName.size > 10)
  ```

  ✓ are union types

  ✓ are infered at validation time

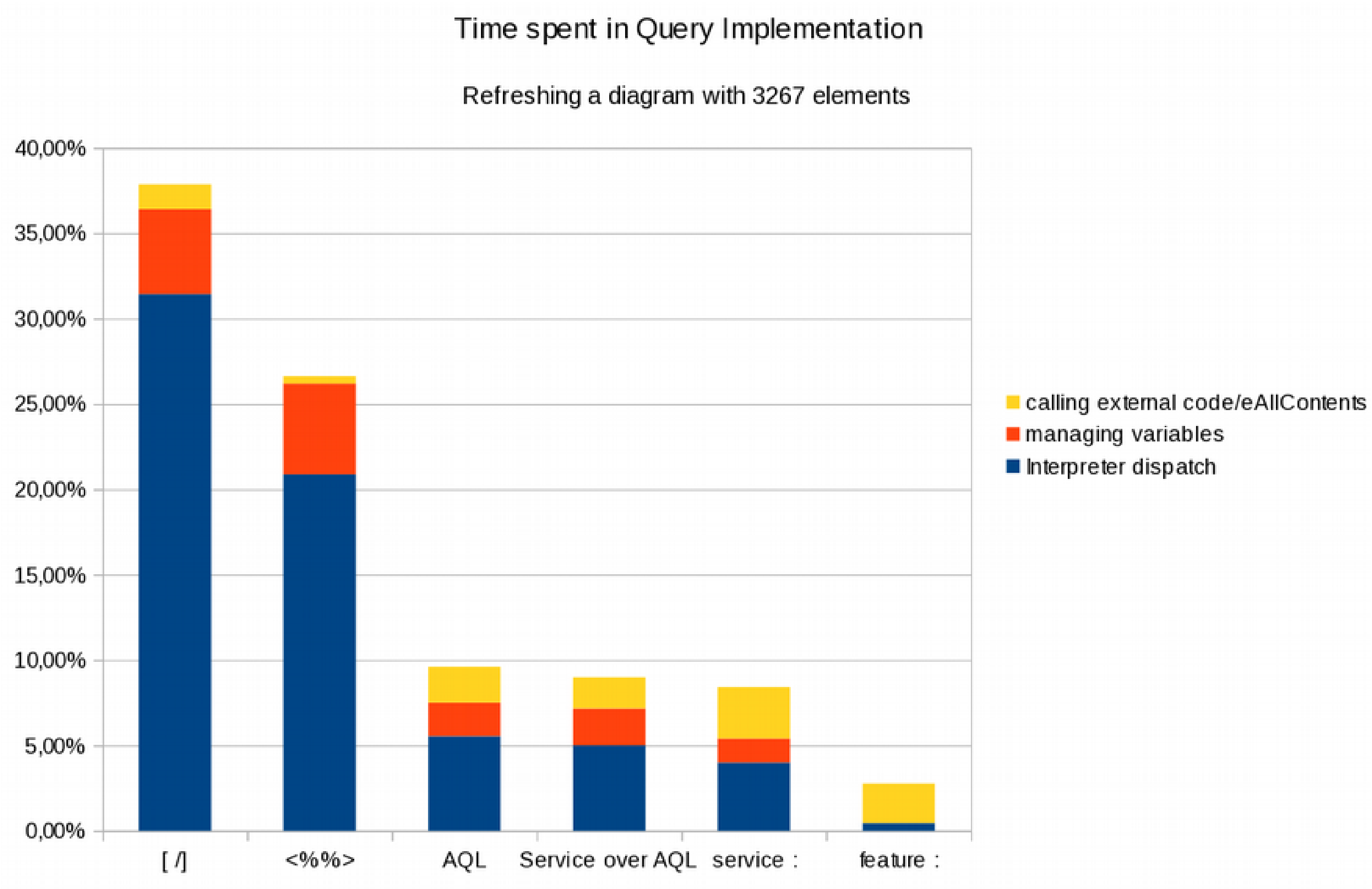
  ✓ no need for cast

Fast during evaluation, **Smart** during validation



Validation is **optional**

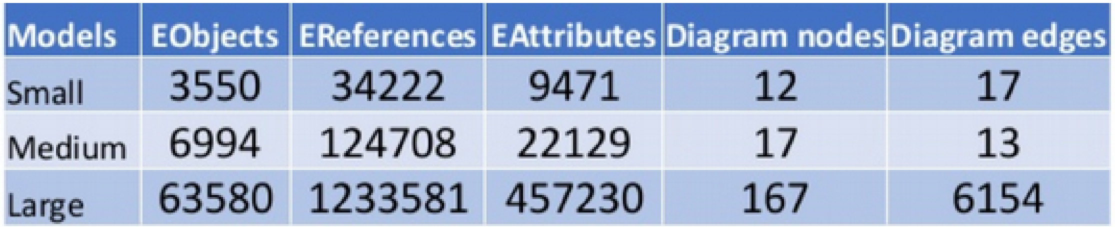
Fast during evaluation, **Smart** during validation



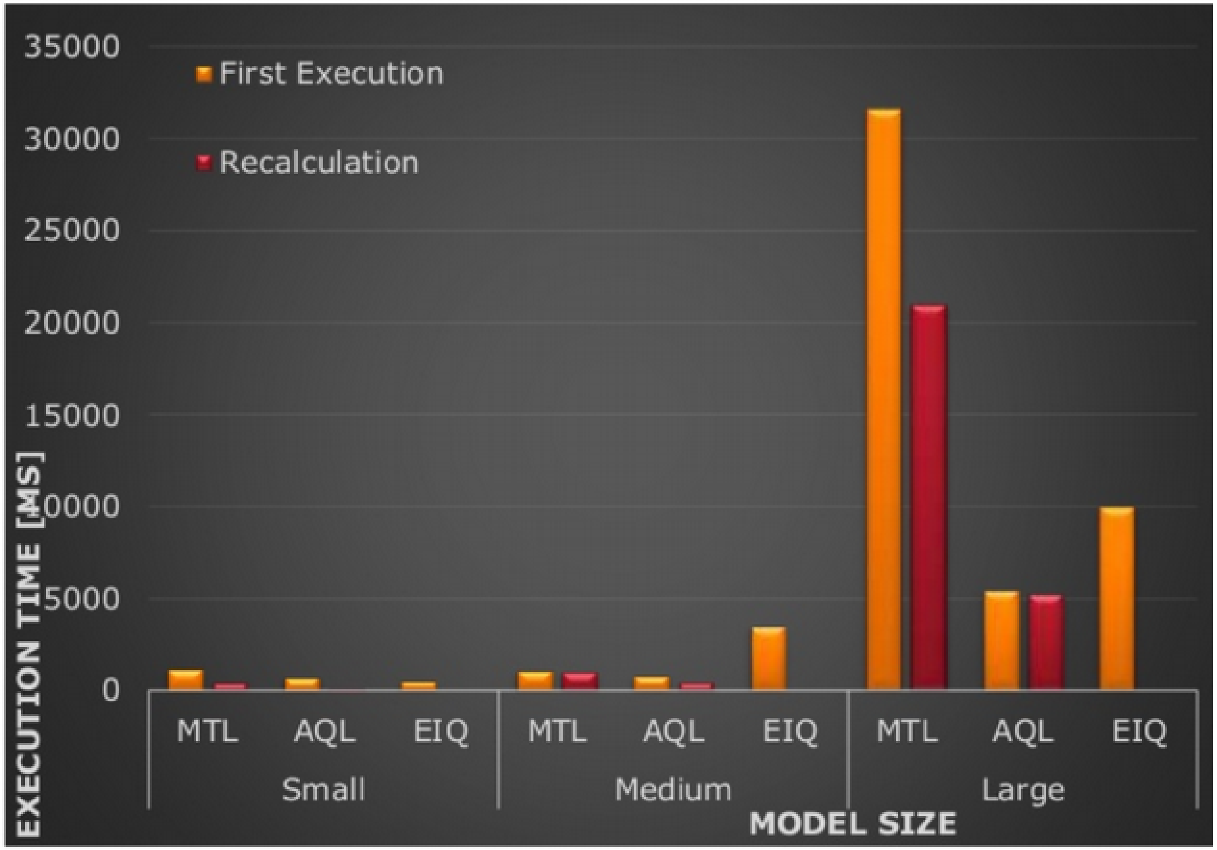
(*) with Sirius 3.1, October 2015, optimizations happened since then

Consistently Fast during evaluation

- Profiler was used to isolate query execution time
- AQL
  - provides good performance
  - Low memory profile
- IncQuery
  - Recalculations < 50 ms
  - Requires up to 2x memory
    - Large ~1.2 Gb

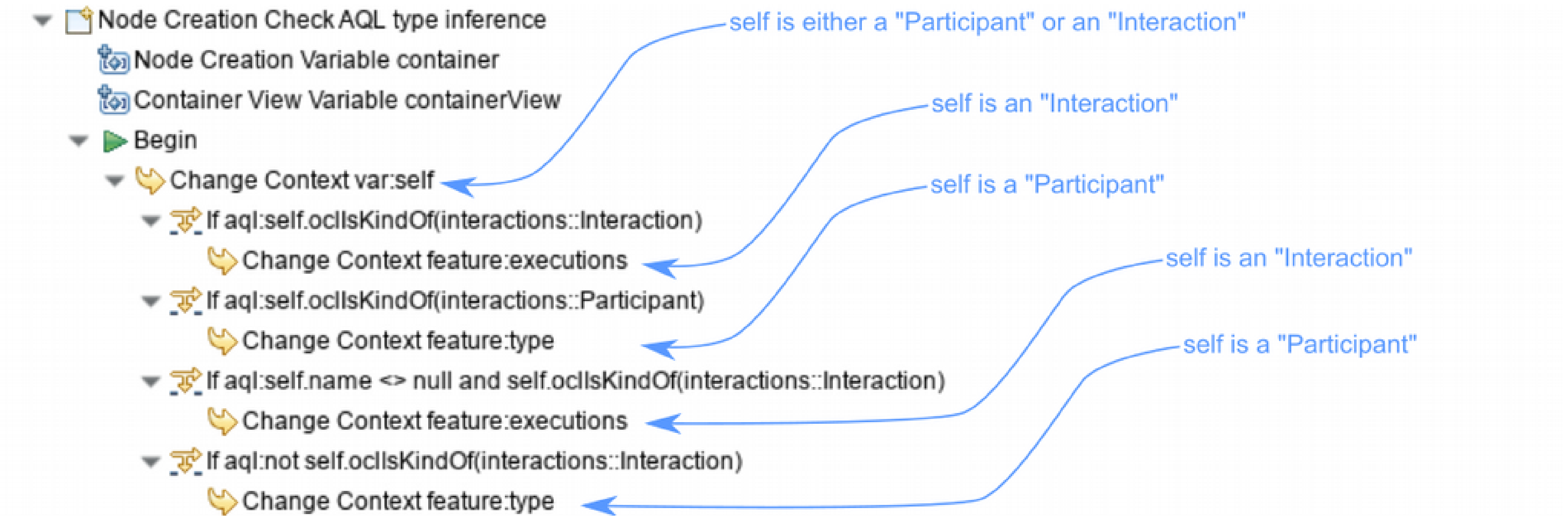


Models	EObjects	EReferences	EAttributes	Diagram nodes	Diagram edges
Small	3550	34222	9471	12	17
Medium	6994	124708	22129	17	13
Large	63580	1233581	457230	167	6154



From „EMF-IncQuery: Blazing-fast reaction time even for very large diagrams (Sirius integration)" by Ákos Horváth, SiriusCon 2015, Paris

Predicates analysis and union types



```
Node Creation Check AQL type inference
    Node Creation Variable container
    Container View Variable containerView
    Begin
        Change Context var:self
            If aql:self.oclIsKindOf(interactions::Interaction)
                Change Context feature:executions
            If aql:self.oclIsKindOf(interactions::Participant)
                Change Context feature:type
            If aql:self.name <> null and self.oclIsKindOf(interactions::Interaction)
                Change Context feature:executions
            If aql:not self.oclIsKindOf(interactions::Interaction)
                Change Context feature:type
```

self is either a "Participant" or an "Interaction"

self is an "Interaction"

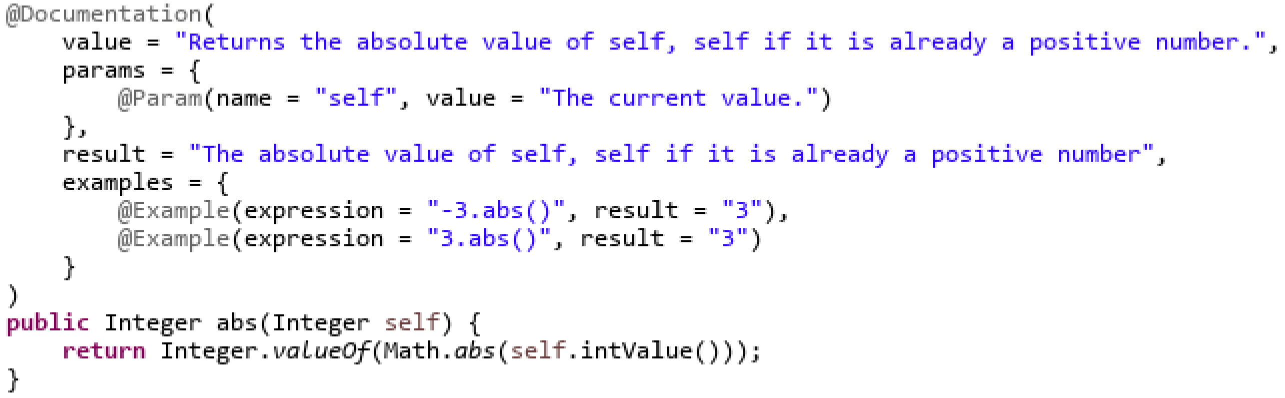
self is a "Participant"

self is an "Interaction"

self is a "Participant"

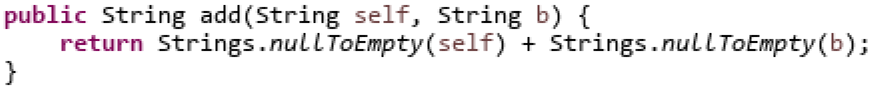
Extensible with Java

- **Included**



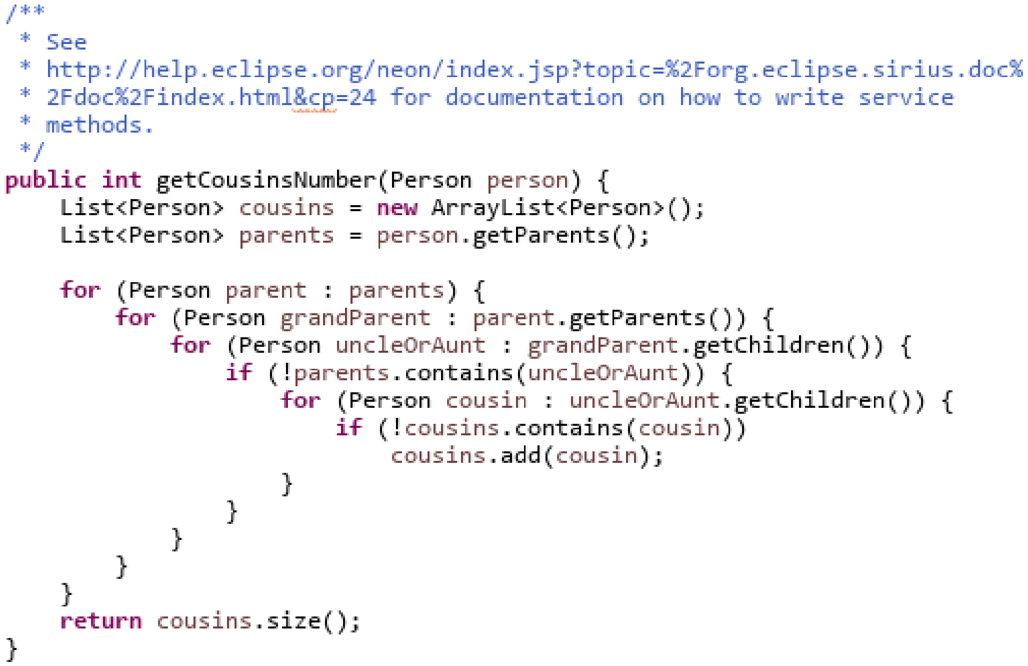
```
@Documentation(
    value = "Returns the absolute value of self, self if it is already a positive number.",
    params = {
        @Param(name = "self", value = "The current value.")
    },
    result = "The absolute value of self, self if it is already a positive number",
    examples = {
        @Example(expression = "-3.abs()", result = "3"),
        @Example(expression = "3.abs()", result = "3")
    }
)
public Integer abs(Integer self) {
    return Integer.valueOf(Math.abs(self.intValue()));
}
```

- **Operators semantics : 'Hello' + self.name**



```
public String add(String self, String b) {
    return Strings.nullToEmpty(self) + Strings.nullToEmpty(b);
}
```

- **Domain specific services**

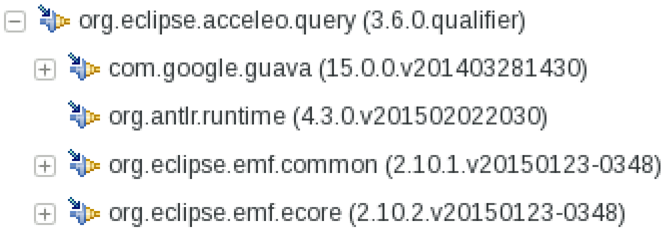


```
/**
 * See
 * http://help.eclipse.org/neon/index.jsp?topic=%2Forg.eclipse.sirius.doc%
 * 2Fdoc%2Findex.html&cp=24 for documentation on how to write service
 * methods.
 */
public int getCousinsNumber(Person person) {
    List<Person> cousins = new ArrayList<Person>();
    List<Person> parents = person.getParents();

    for (Person parent : parents) {
        for (Person grandParent : parent.getParents()) {
            for (Person uncleOrAunt : grandParent.getChildren()) {
                if (!parents.contains(uncleOrAunt)) {
                    for (Person cousin : uncleOrAunt.getChildren()) {
                        if (!cousins.contains(cousin))
                            cousins.add(cousin);
                    }
                }
            }
        }
    }
    return cousins.size();
}
```

The Runtime

- Dependencies



```
⊟ org.eclipse.acceleo.query (3.6.0.qualifier)
    ⊞ com.google.guava (15.0.0.v201403281430)
      org.antlr.runtime (4.3.0.v201502022030)
    ⊞ org.eclipse.emf.common (2.10.1.v20150123-0348)
    ⊞ org.eclipse.emf.ecore (2.10.2.v20150123-0348)
```

- 13K lines of non generated Java code, 21K total (EMF API for AST)
- Is not a singleton

How to get it ?

Part of The Release Train

- **Shipped since 2015 as part of the Acceleo project**
- **Documentation: https://www.eclipse.org/acceleo/documentation/aql.html**
- **Other technologies already using it**
  - Eclipse Sirius
  - M2doc (to generate .docx files from a model)
  - ALE (Action Language for EMF)