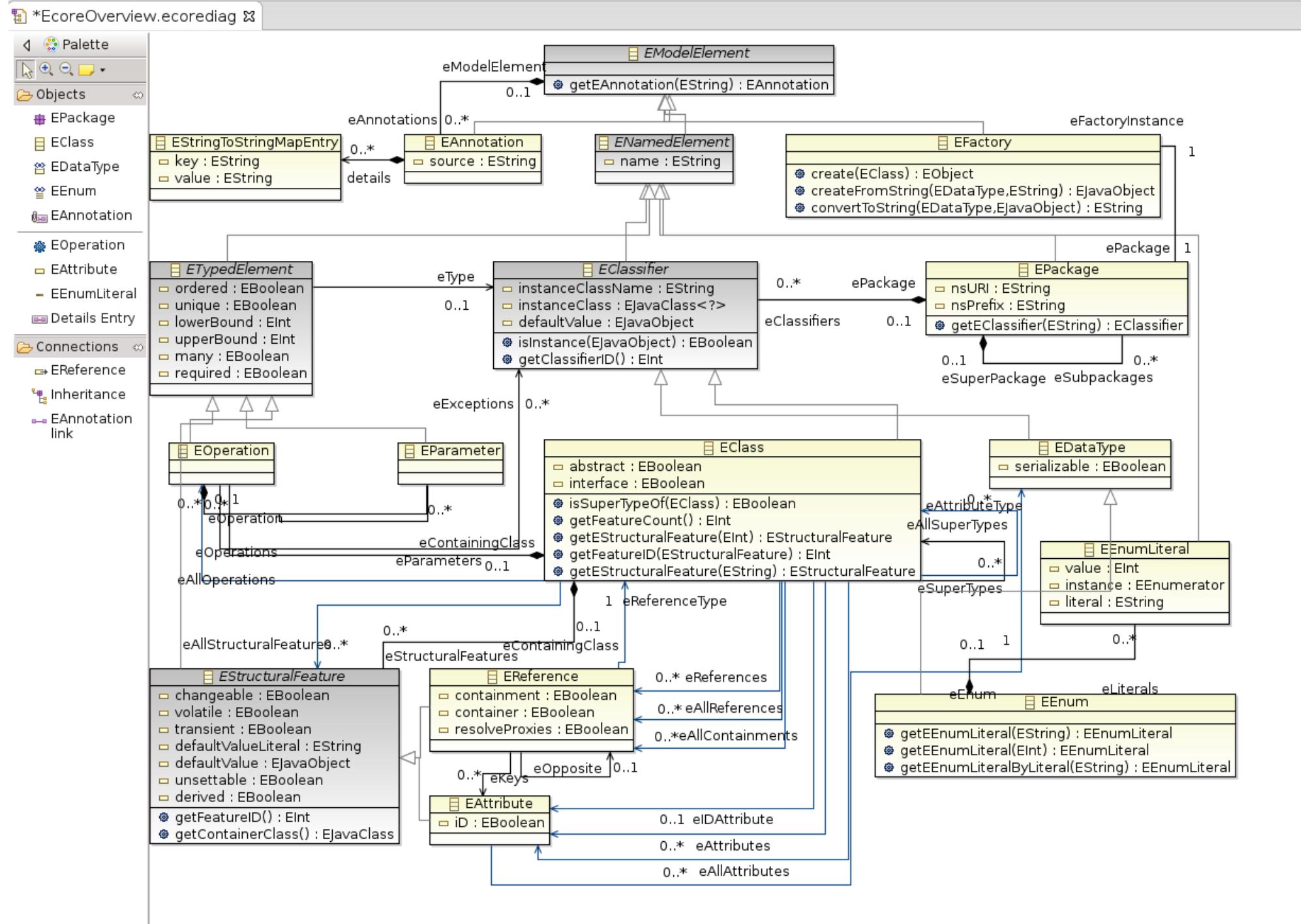


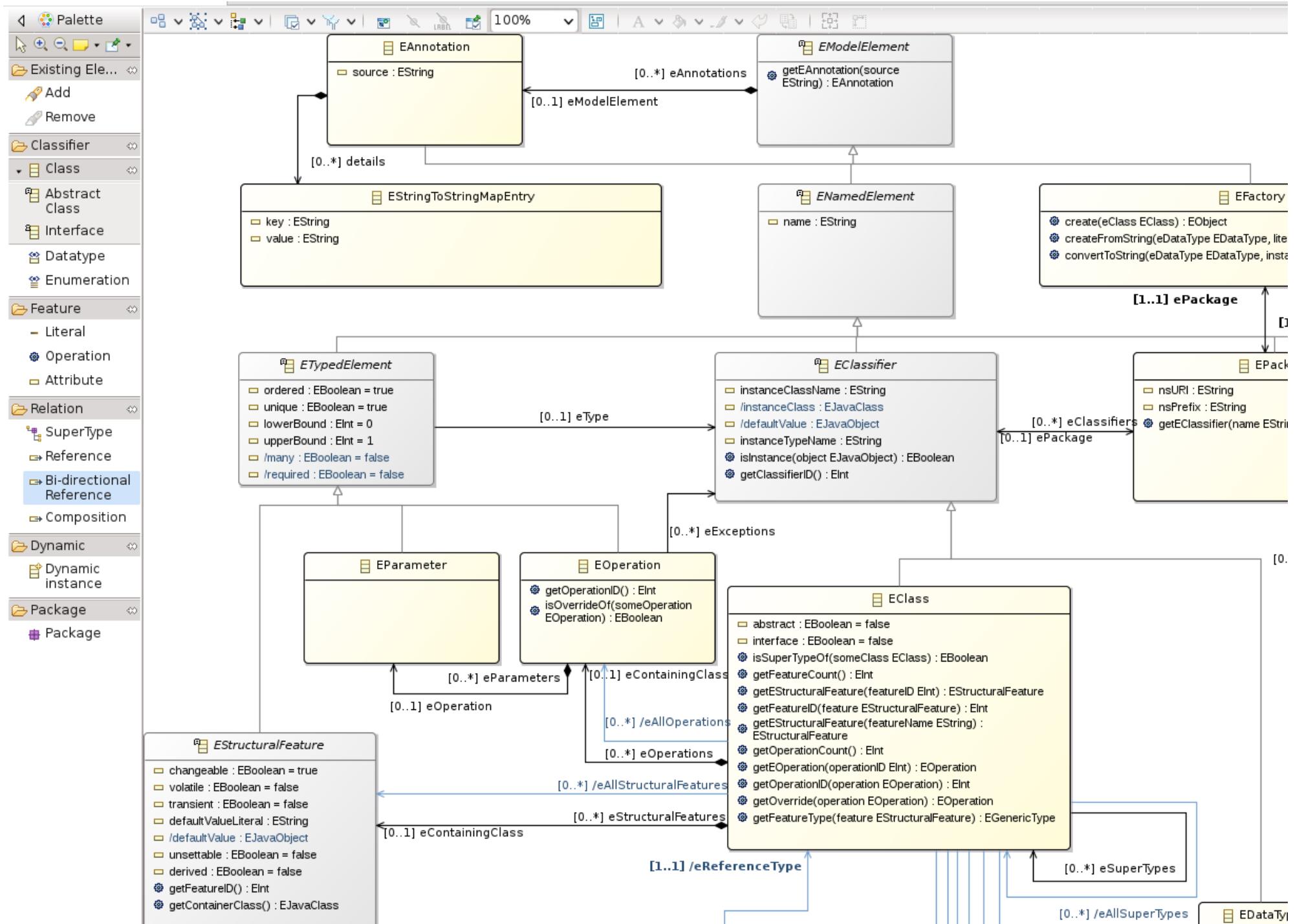


# EcoreTools 2.0

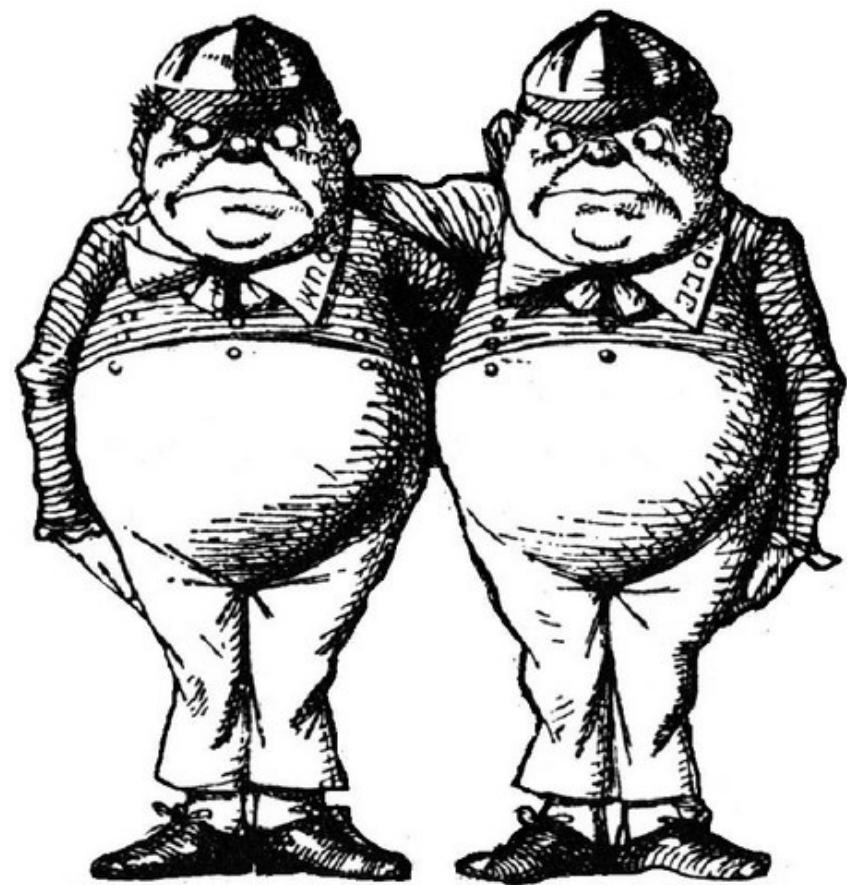
## The Making Of

Cédric Brun <cedric.brun@obeo.fr> / @bruncedric





# Personnas



# The Life of Alice

- Prototyping w. Customer
- Focused Diagrams
- Doc and Notes
- Reviewing Design



# Demo Wrap-up

Semi synchronized class diagram

Layers : documentation, validation

Import elements in the diagram one by one

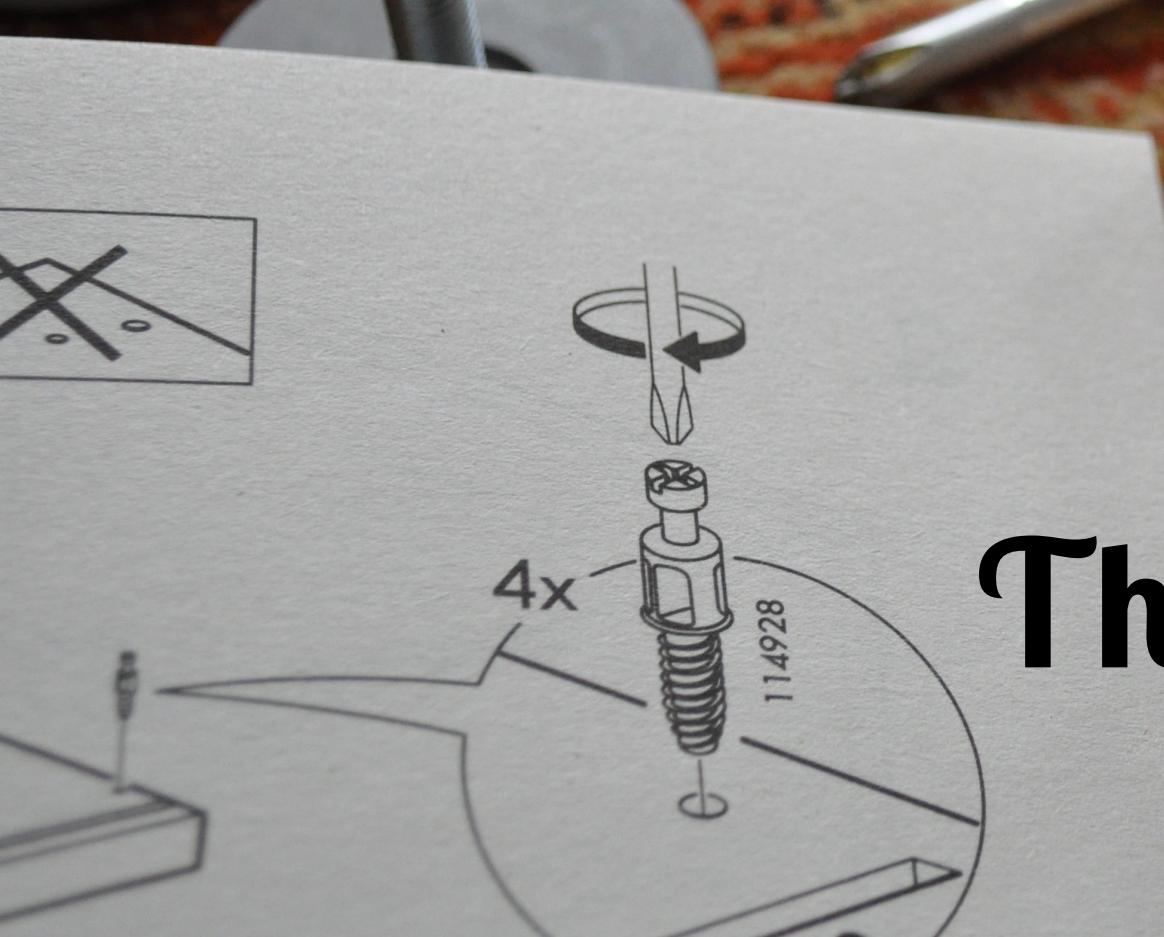
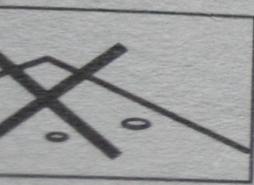
Slightly redesigned graphical style

A lot of shortcuts, smarter palette, smarter edit

A documentation Table

And many more...

# The Making Of



EcoreTools 2.0

EcoreTools 1.0

Sirius Runtime

EEF

GMF runtime

Acceleo

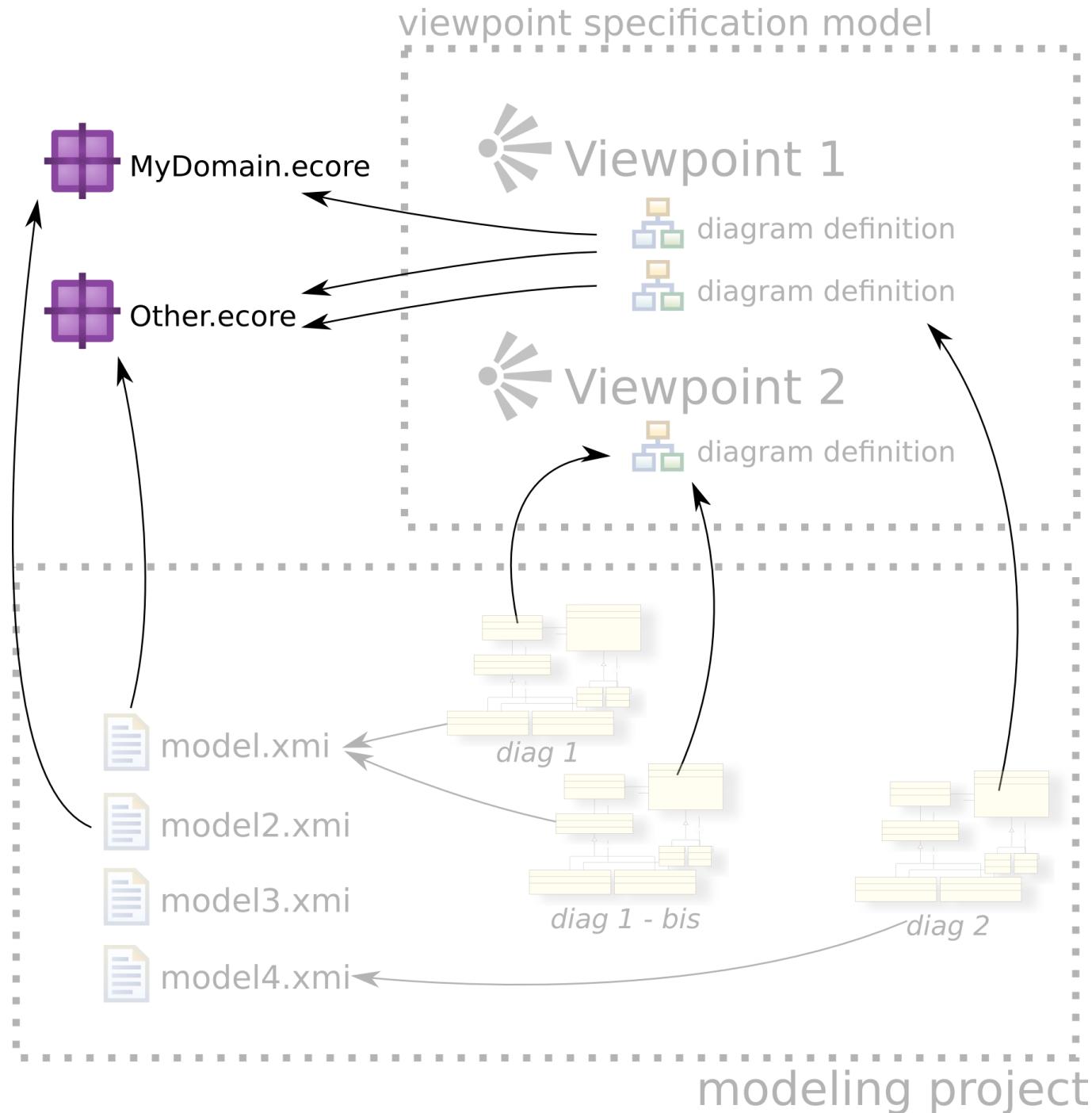
GEF

EMF

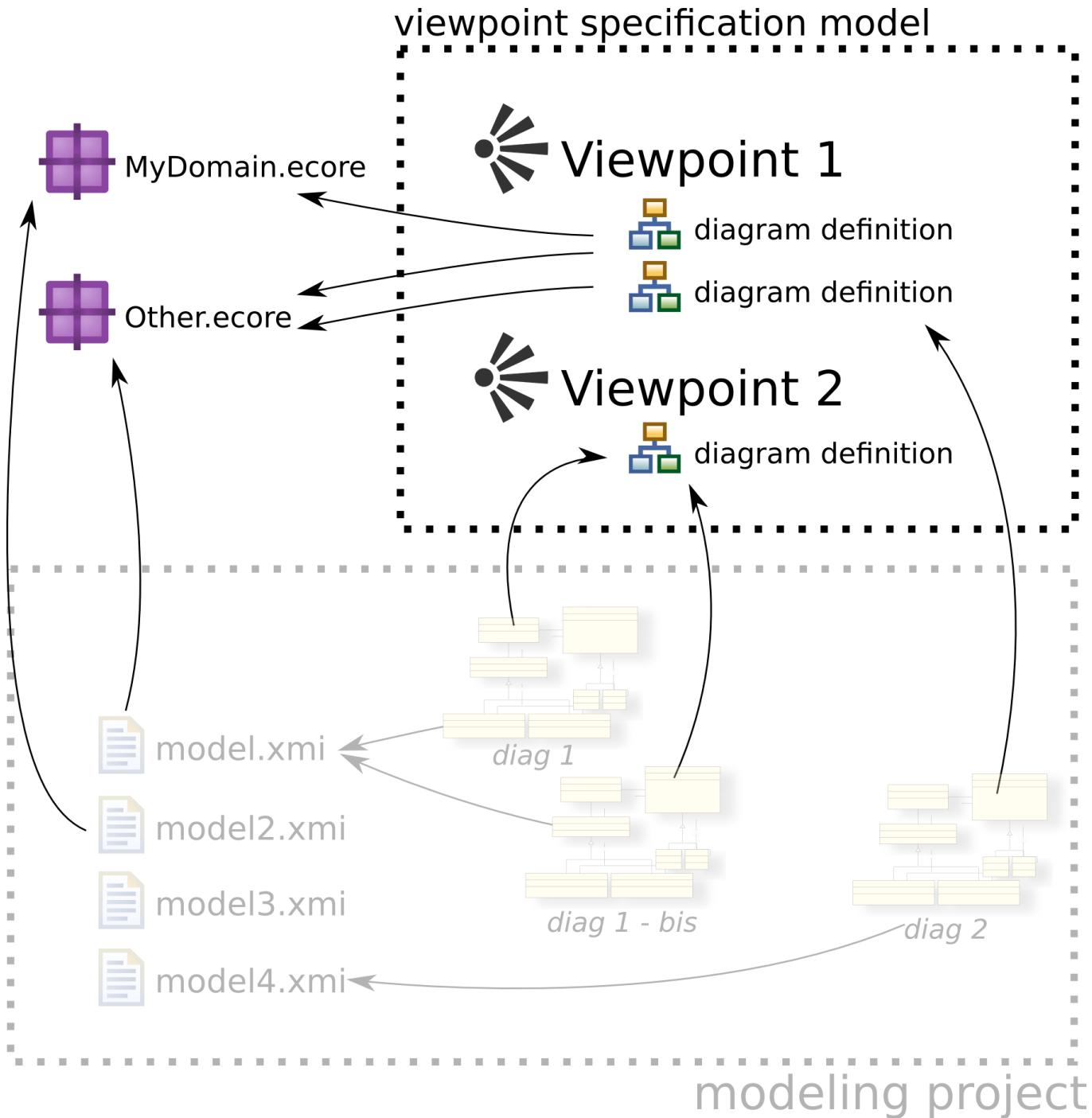
Eclipse IDE



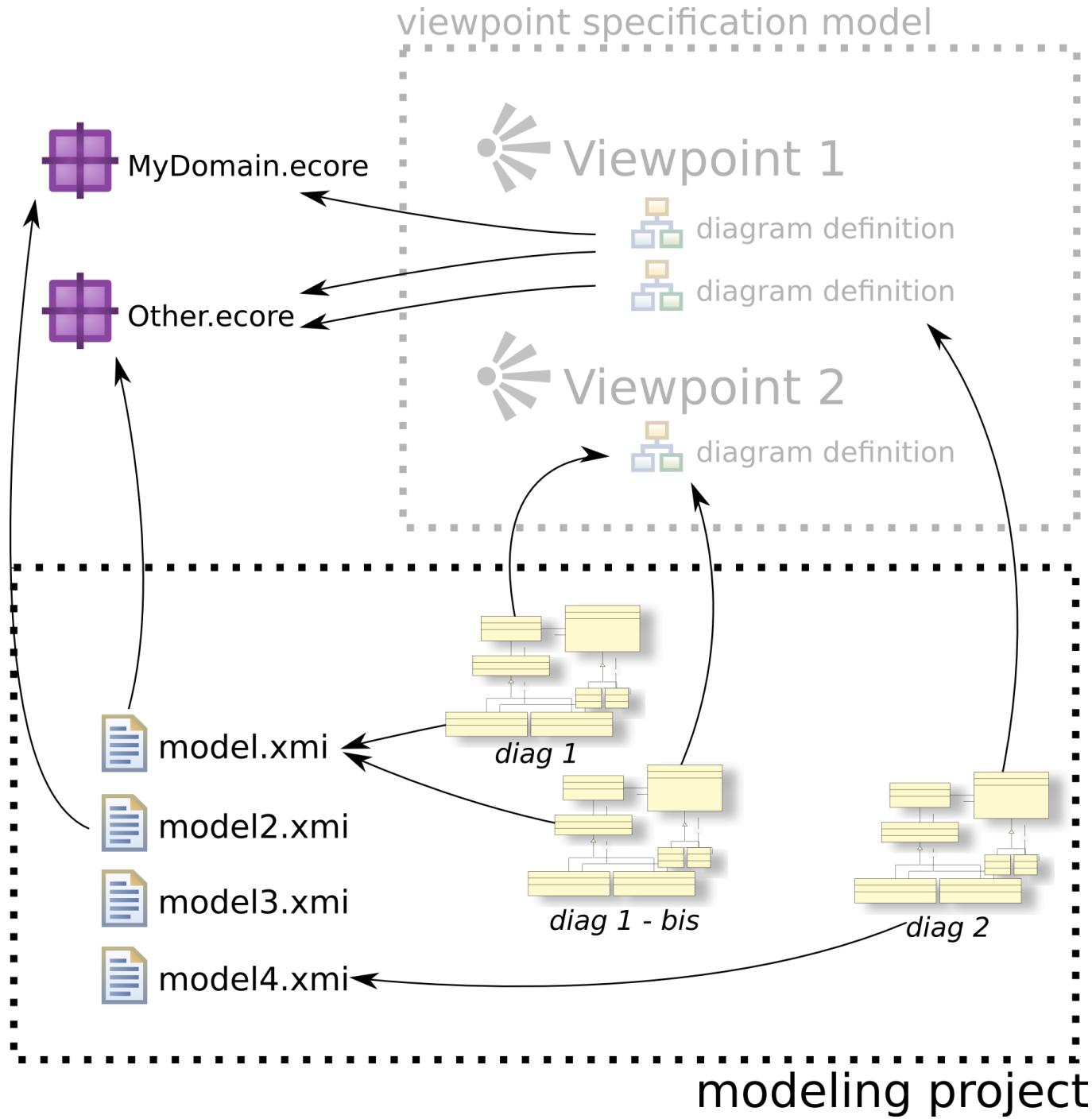
# a dedicated Tooling ?



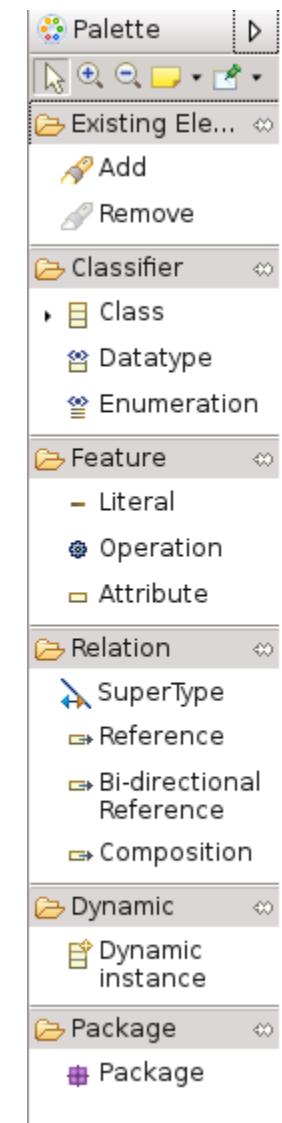
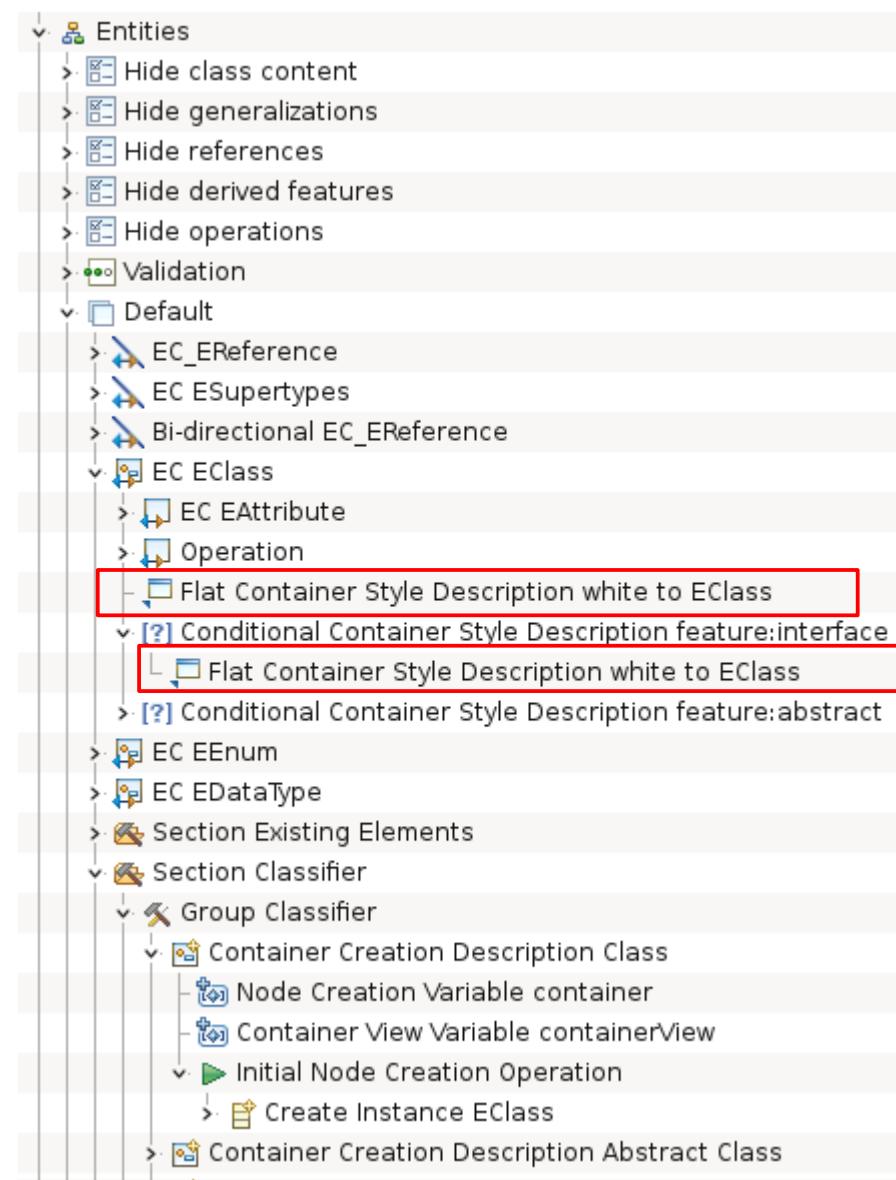
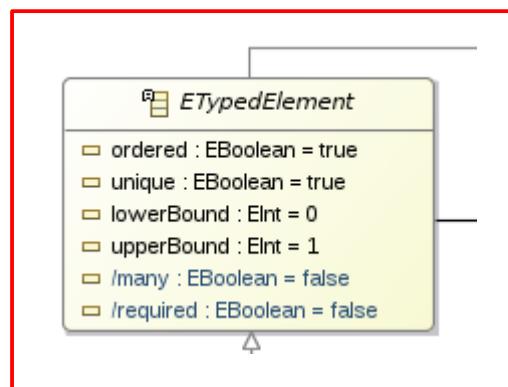
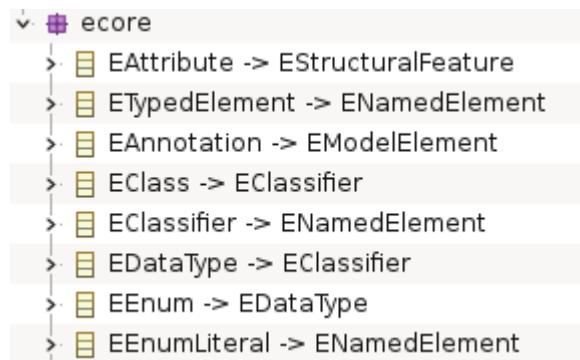
# a dedicated Tooling ?



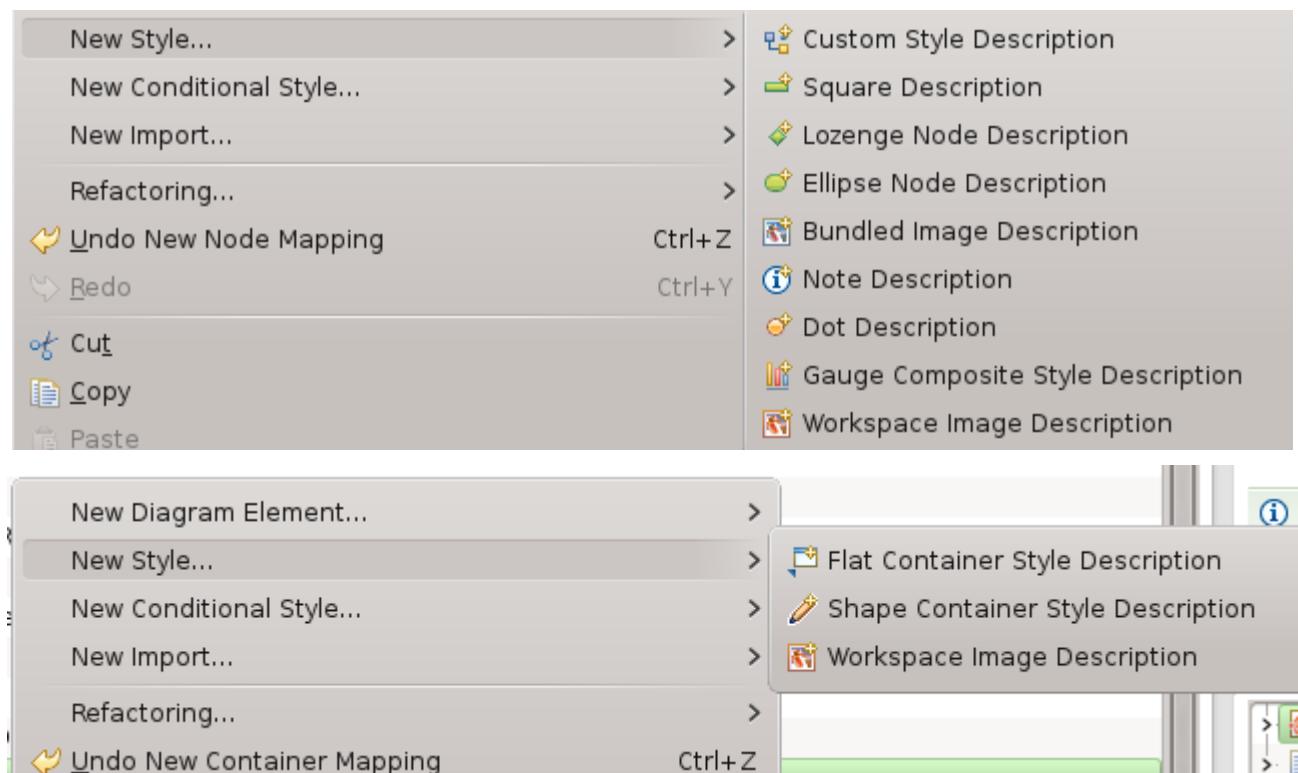
# a dedicated Tooling ?

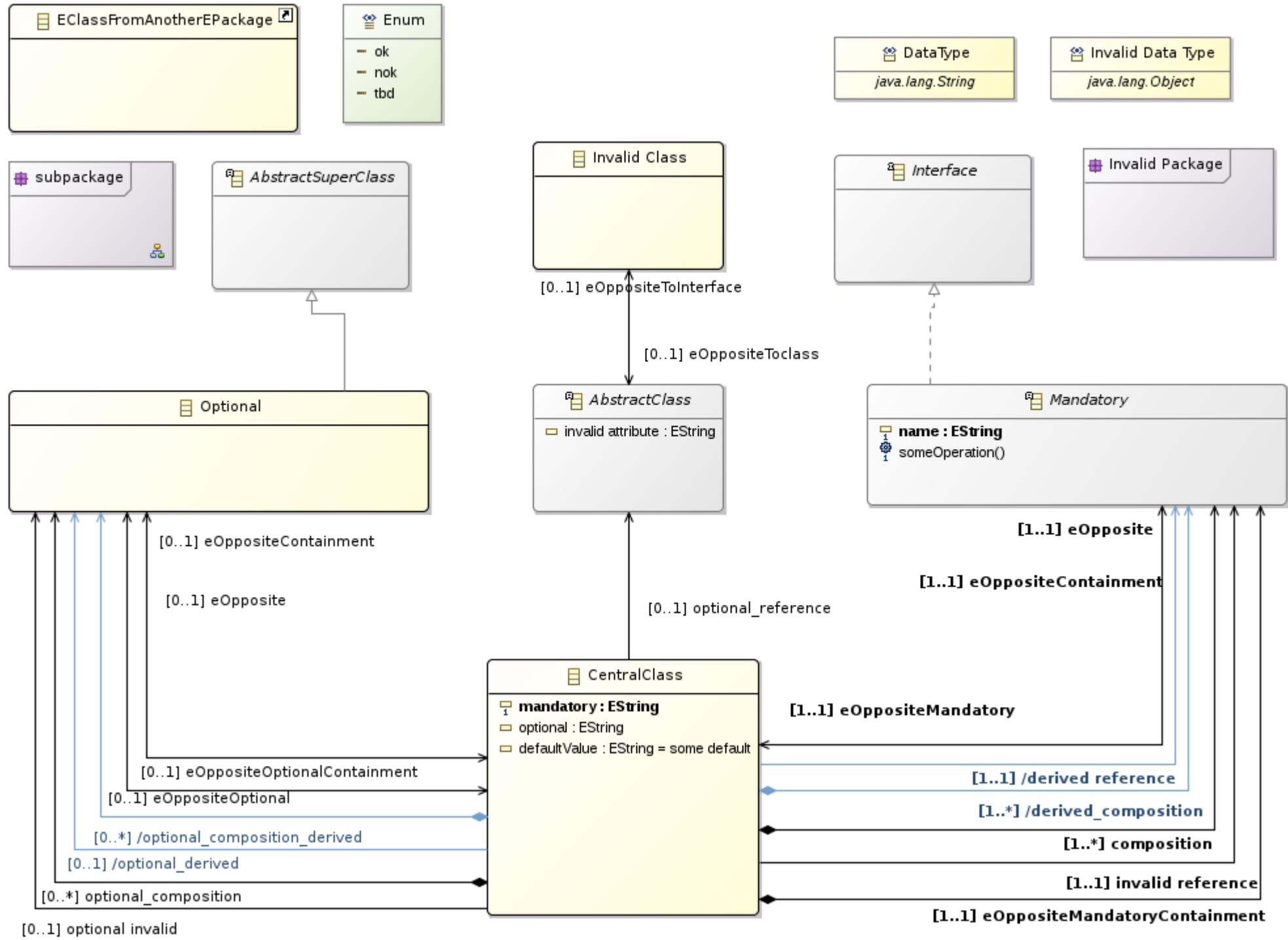


# Diagram definition - styles

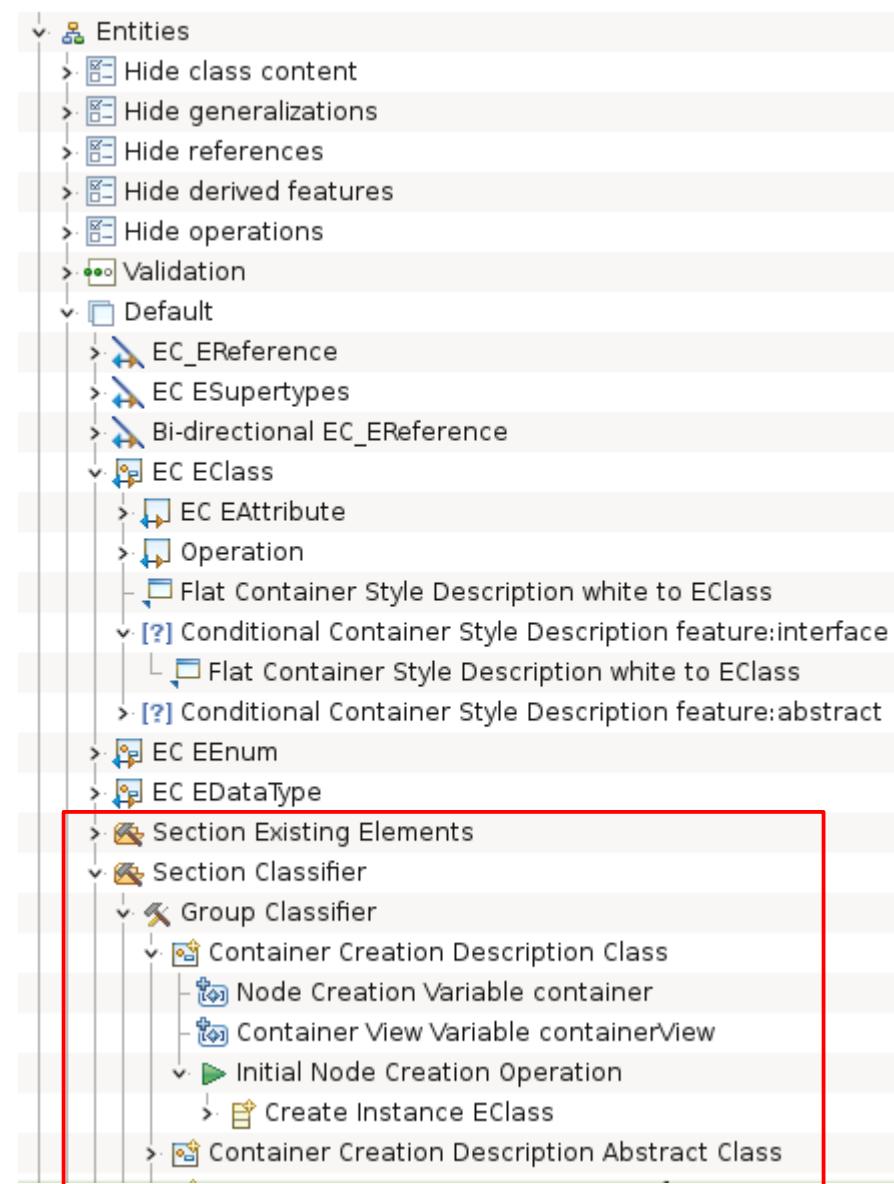
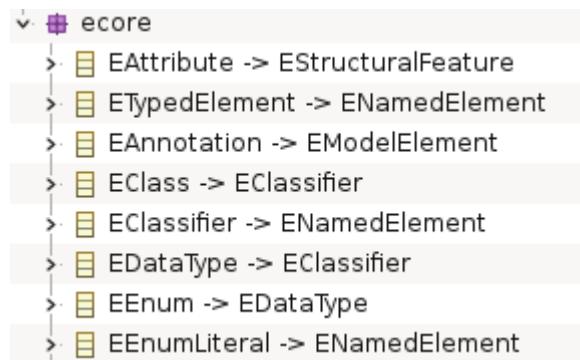


# Styles

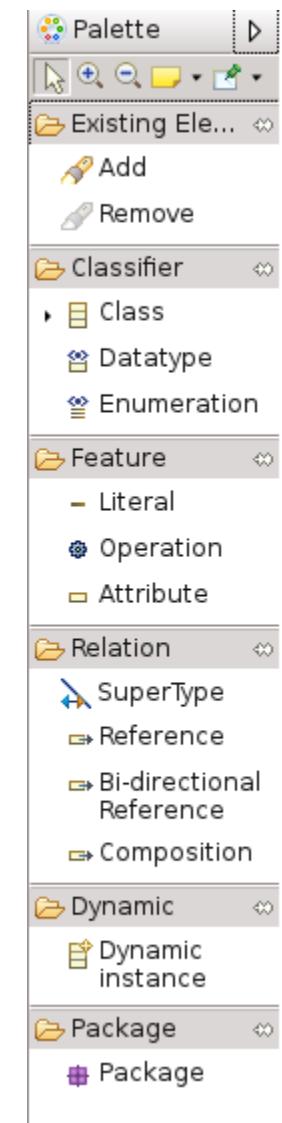
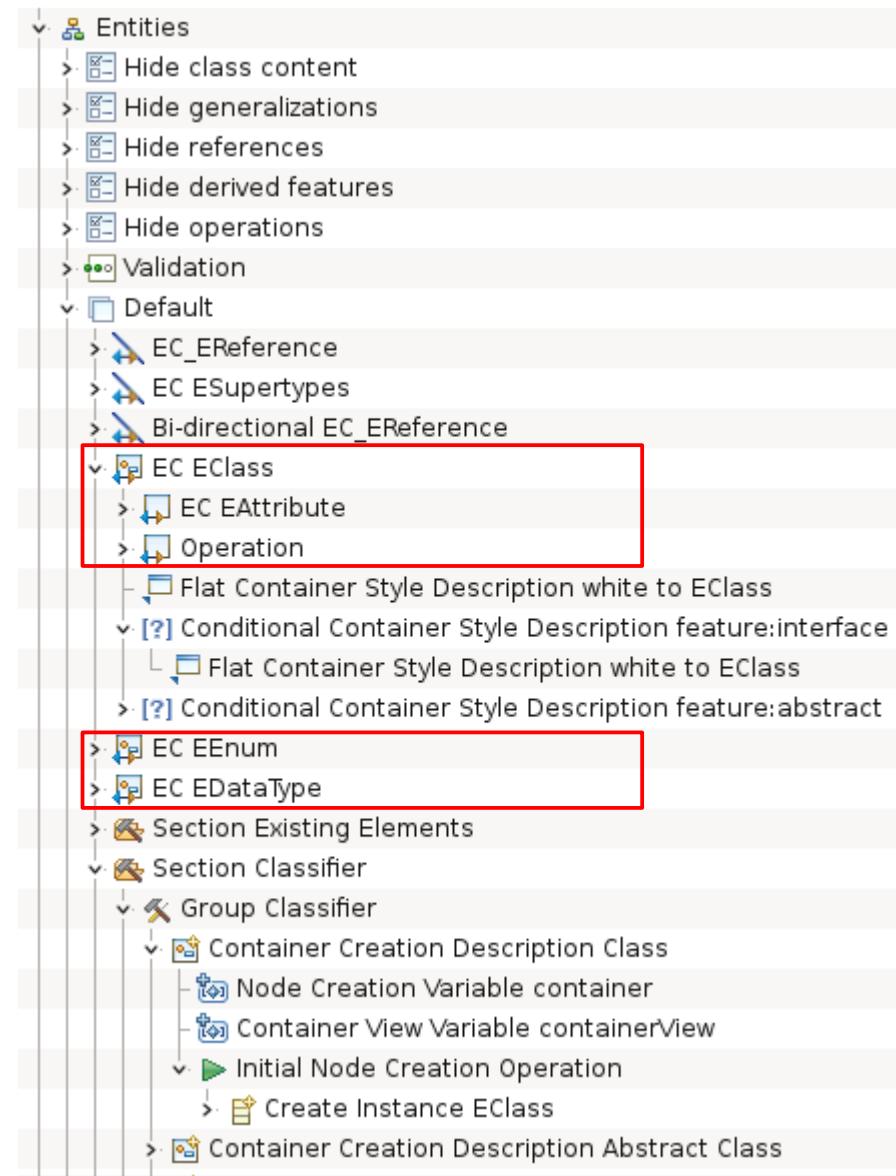
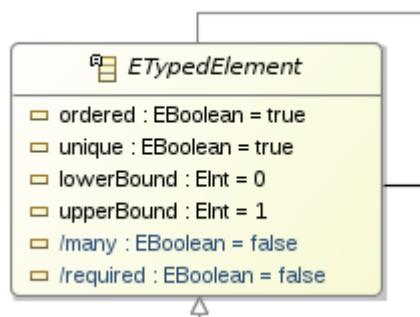
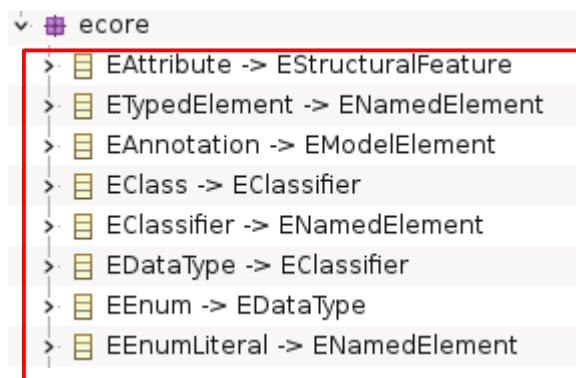




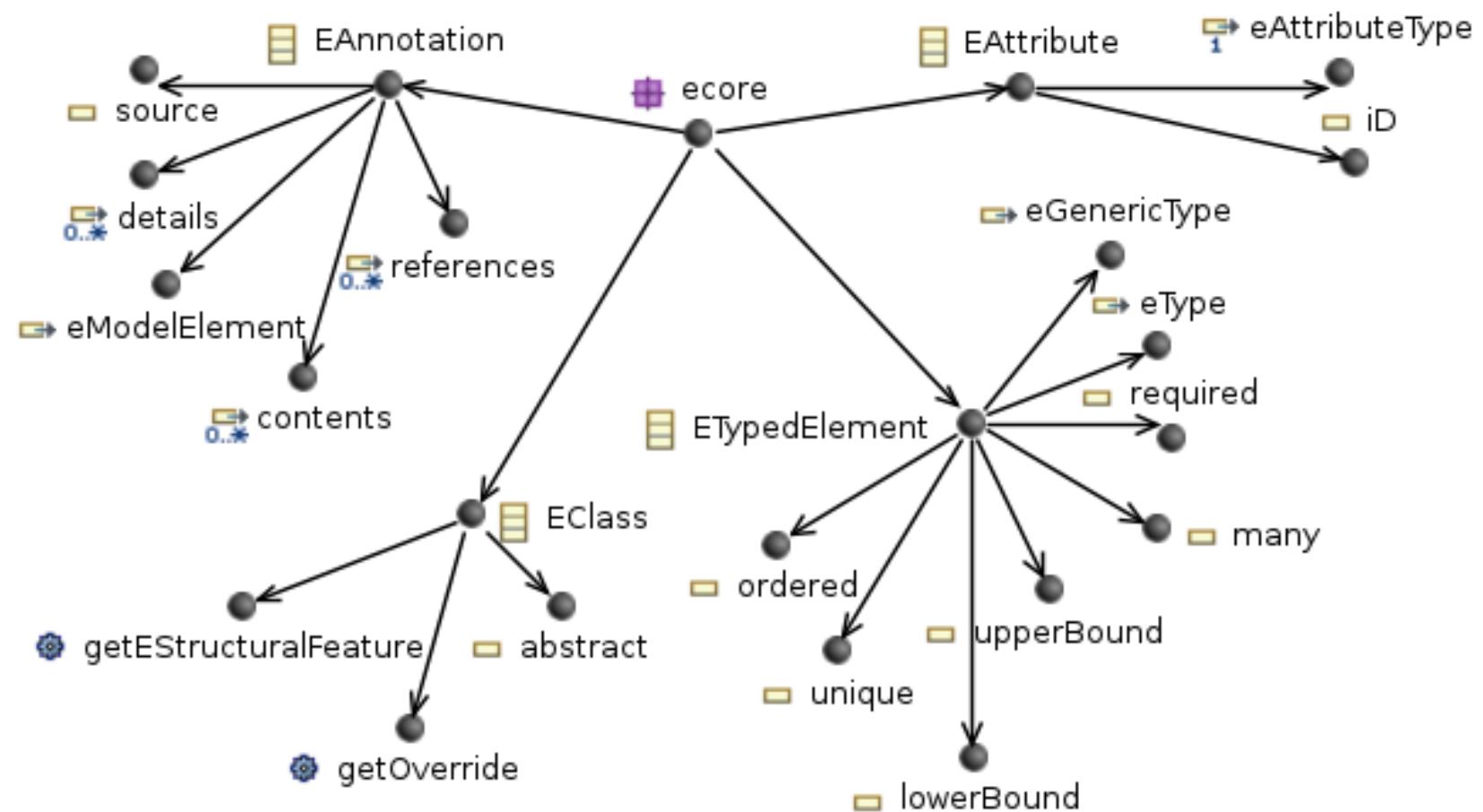
# Diagram definition - tools



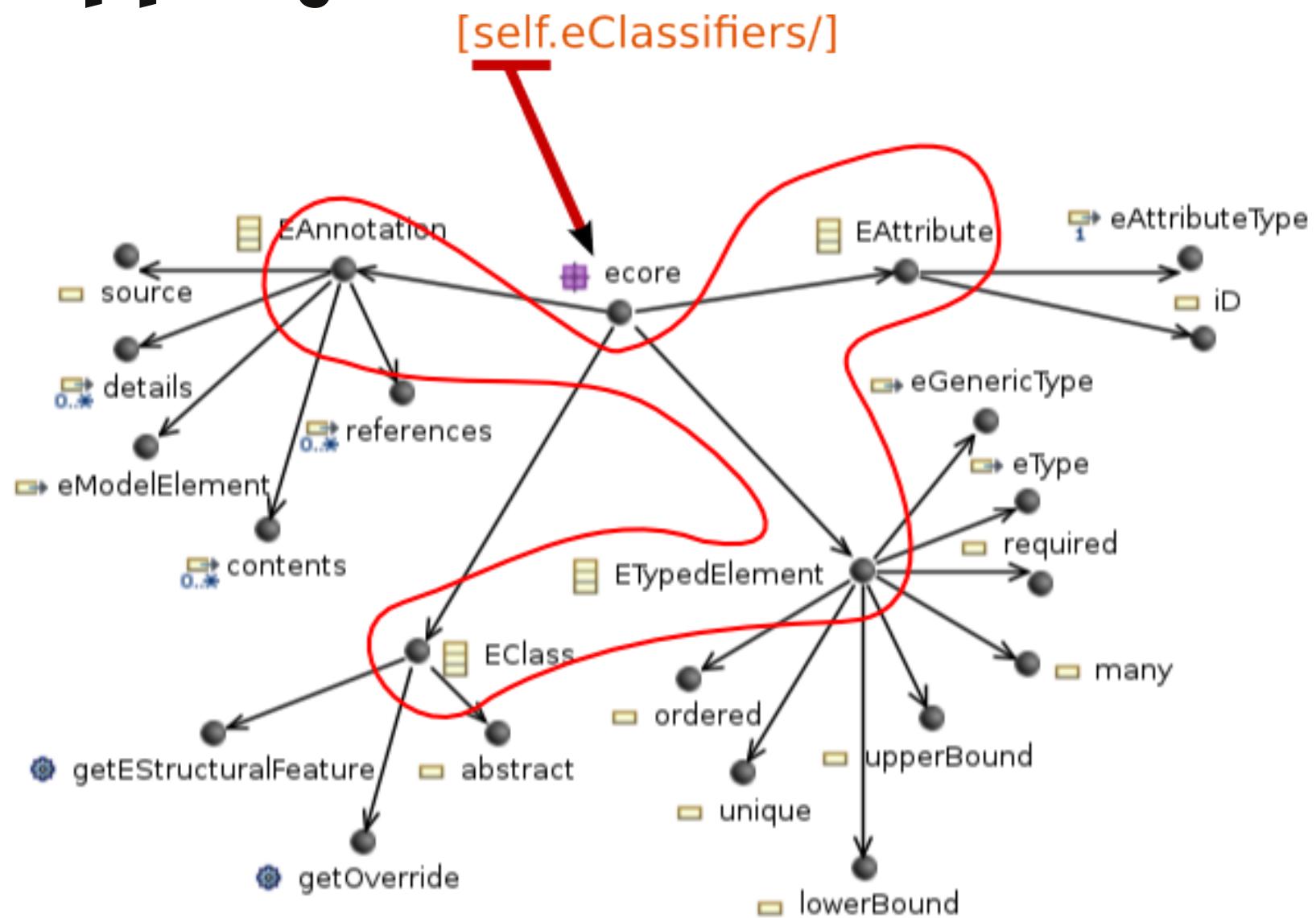
# Diagram definition - mappings



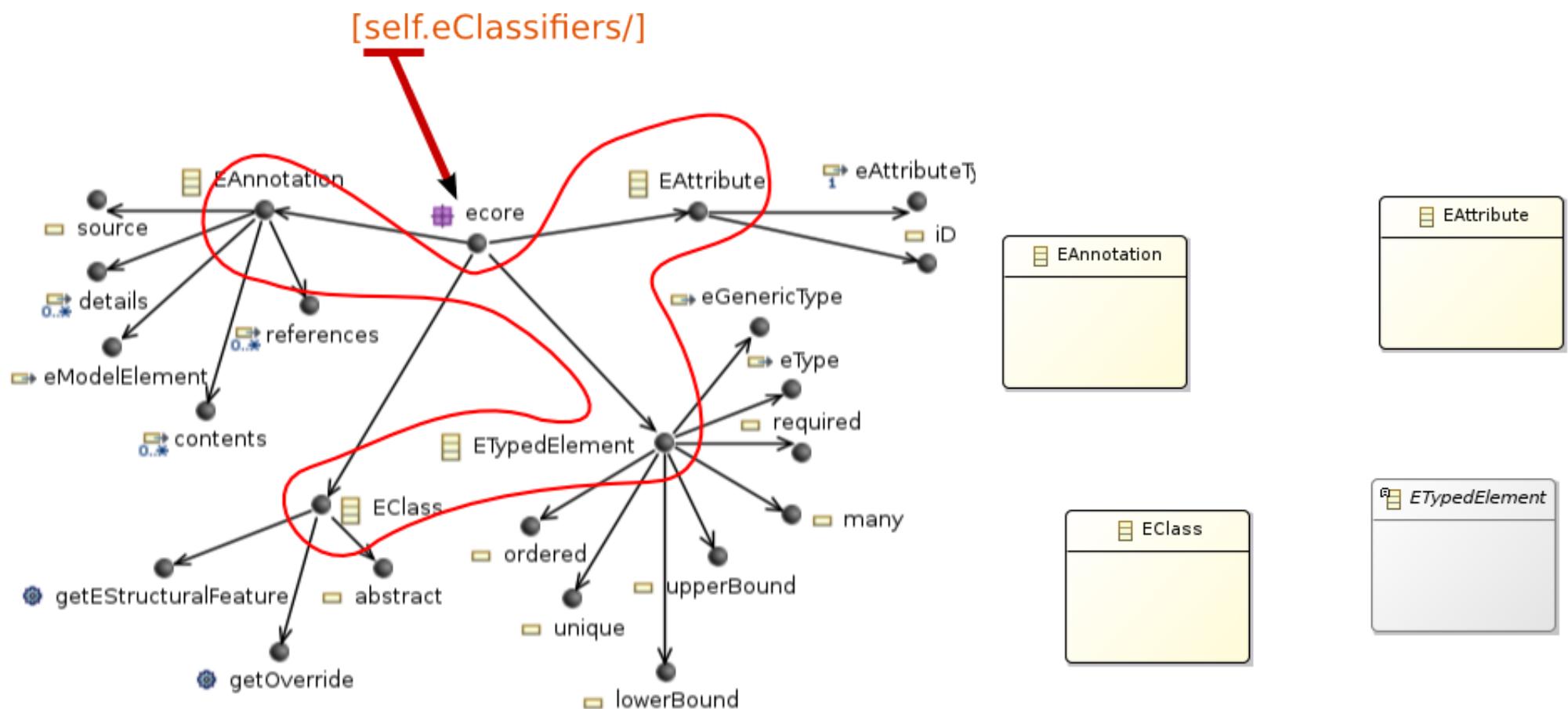
# A Mapping



# A Mapping



# A Mapping



# Synchronization Policy



# Mapping for EClasses

EC EClass

<b>General</b>	<b>Id:</b> <input type="text" value="EC EClass"/>	<b>Label:</b> <input type="text" value="EC EC"/>
<b>Import</b>	<b>Domain Class:</b> <input type="text" value="EClass"/>	
<b>Documentation</b>		
<b>Behavior</b>	<b>Children Presentation:</b> <input type="radio"/> FreeForm <input checked="" type="radio"/> List <input type="radio"/> Horizontal Stack <input type="radio"/> Vertical Stack	
<b>Advanced</b>	<b>Semantic Candidates Expression:</b> <input type="text"/>	
	<b>Associated Elements Expression:</b> <input type="text"/>	

EC EClass

<b>General</b>	<b>Synchronization:</b> <input checked="" type="radio"/> Not synchronized <input type="radio"/> Unsynchronizable <input type="radio"/> Synchronized
<b>Import</b>	
<b>Documentation</b>	
<b>Behavior</b>	
<b>Advanced</b>	<b>Precondition Expression:</b> <input type="text"/>

# Mapping for EReferences

EC\_EReference

<b>General</b>	<b>Id*</b> : <input type="text" value="EC_EReference"/> Label: <input type="text" value="EC_EReference"/>
<b>Path</b>	<b>Source Mapping*</b> : <input type="text" value="EC EClass"/>
<b>Documentation</b>	<b>Domain Class*</b> : <input type="text" value="EReference"/>
<b>Behavior</b>	<b>Source Finder Expression:</b> <input type="text" value="feature:eContainer"/>
<b>Advanced</b>	<b>Target Mapping*</b> : <input type="text" value="EC EClass"/>
	<b>Target Finder Expression*</b> : <input type="text" value="feature:eType"/>
	<b>Semantic Candidates Expression:</b> <input type="text" value="diagram.getDisplayedEClasses().oclAsType.ecore::EClass).eAllReferences-&gt;flatten()"/>
	<b>Associated Elements Expression:</b> <input type="text" value="var:self"/>

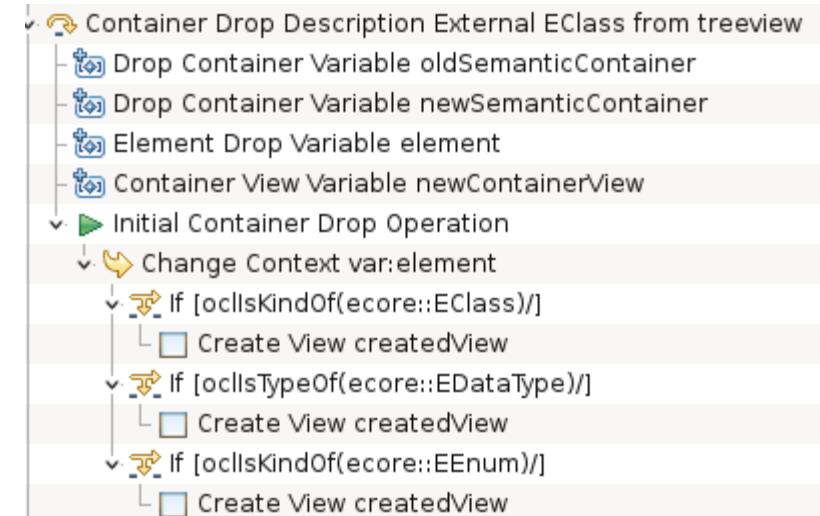
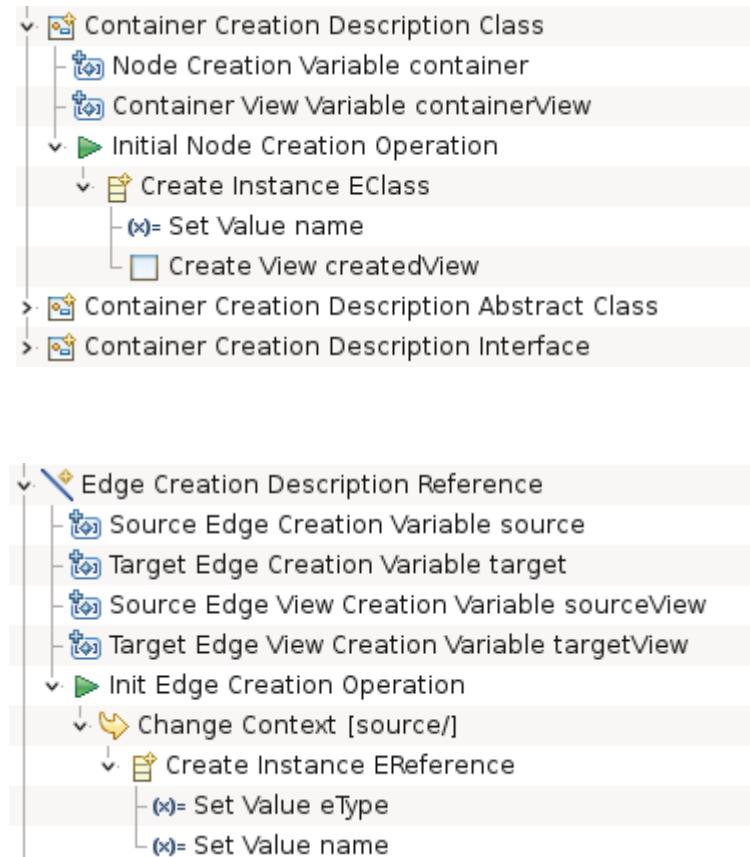
EC\_EReference

<b>General</b>	<b>Synchronization:</b> <input checked="" type="radio"/> Not synchronized <input type="radio"/> Unsyncronizable <input checked="" type="radio"/> Synchronized
<b>Path</b>	<b>Target Expression:</b> <input type="text"/>
<b>Documentation</b>	
<b>Behavior</b>	
<b>Advanced</b>	<b>Precondition Expression:</b> <input type="text" value="service:noEOpposite"/>

# Break 1

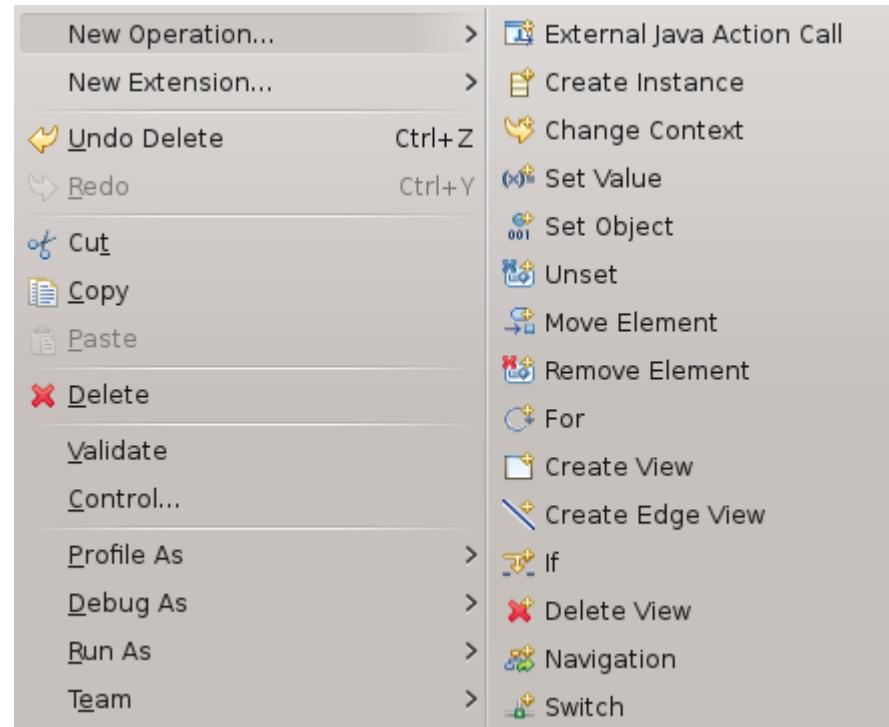
Tooling is specified in a model and interpreted @ runtime  
The coupling to the semantic models is kept low thanks to queries  
Synchronization Policy can be specified on a per mapping basis.  
Queries can be OCL/Acceleo, or even simple Java calls

# the Tools



# Action language

Create  
Reconnect  
drag&drop  
Direct edit  
Double click  
Delete  
Popup actions  
Validation quickfixes



(x) Set Value name

General Feature Name\*: name

Value Expression: ['NewEClass' + eContainer().eContents(ecore::EClass)->size()]

(x) Create Instance EClass

General Reference Name\*: eClassifiers

Type Name: EClass

Variable Name: instance

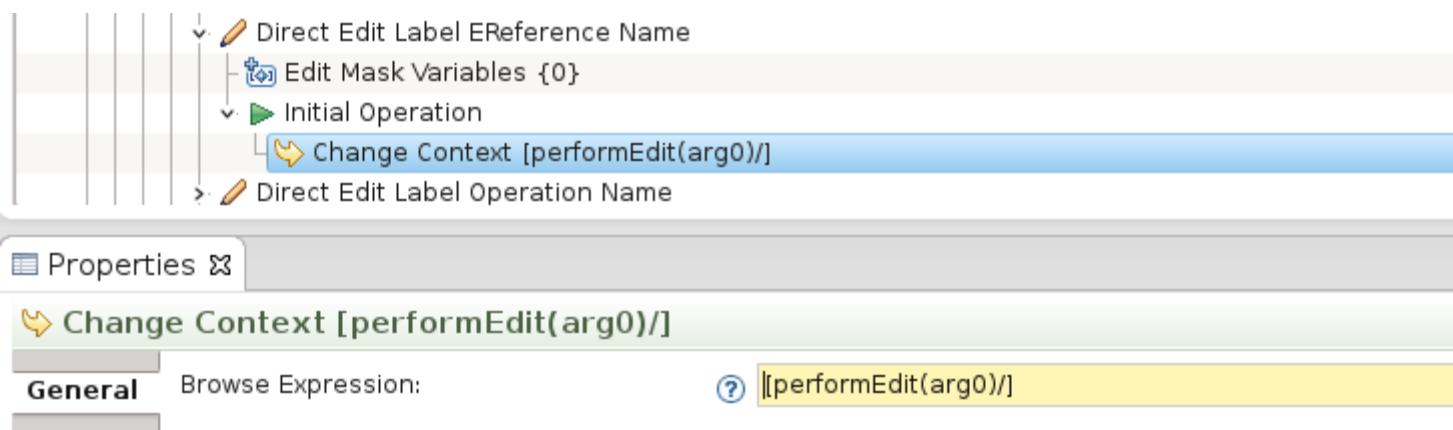
# Label edition

- « Something » => change name of feature
- « :SomeType » => only change the eType
- « l » => only set cardinality to l..x
- « \* » => only set cardinality to x..\*
- « /Something » => make the feature derived
- « = something » => set the default value literal

[...]

# Label edition (bis)

```
public EReference performEdit(EReference ref, String editString) {  
    if ("0".equals(editString.trim())) {  
        ref.setLowerBound(0);  
    } else if ("1".equals(editString.trim())) {  
        ref.setLowerBound(1);  
    } else if (CARDINALITY_UNBOUNDED.equals(editString.trim())) {  
        ref.setUpperBound(-1);  
    } else if (CARDINALITY_UNBOUNDED_ALTERNATIVE.equals(editString.trim())) {  
        ref.setUpperBound(-1);  
    } else {  
        editName(ref, editString);  
        editCardinality(ref, editString);  
    }  
    return ref;  
}
```

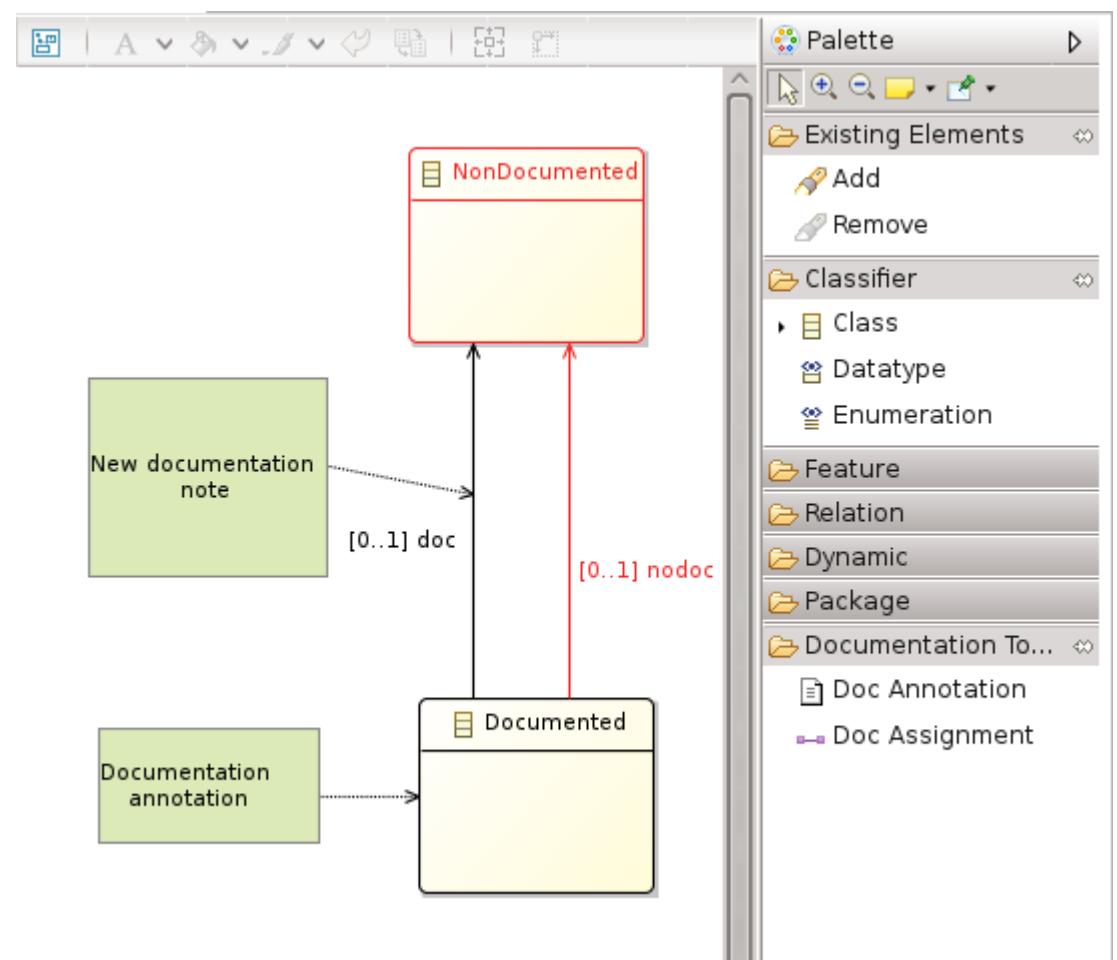
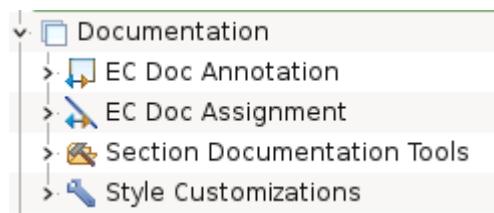


# Documentation Layer

Add new Mappings

Add new Tools

Customize Styles



# Break 2

Because of the low coupling with semantic model, one has to define the effects of the interactions

A Simple Action langage exists for this  
(create/update/delete/ of elements and views)

Calling Java is definitely an option

Expressiveness is pretty rich. Diagram have Layers, filters, tools, decorators

# The Life of Cédric

- Creating EcoreTools
- Consistency
- Performance
- Building
- Testing
- Upgrading



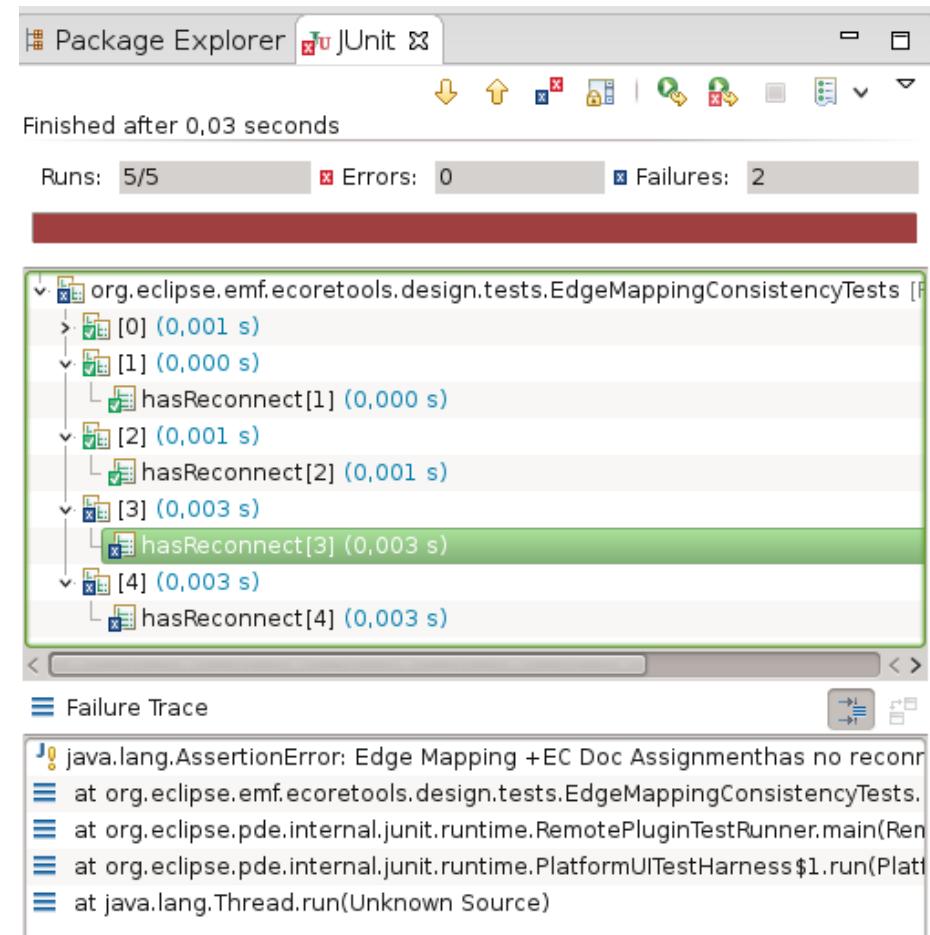
# Checking Consistency

```
@RunWith(value = Parameterized.class)
public class EdgeMappingConsistencyTests {

    private EdgeMapping underTest;

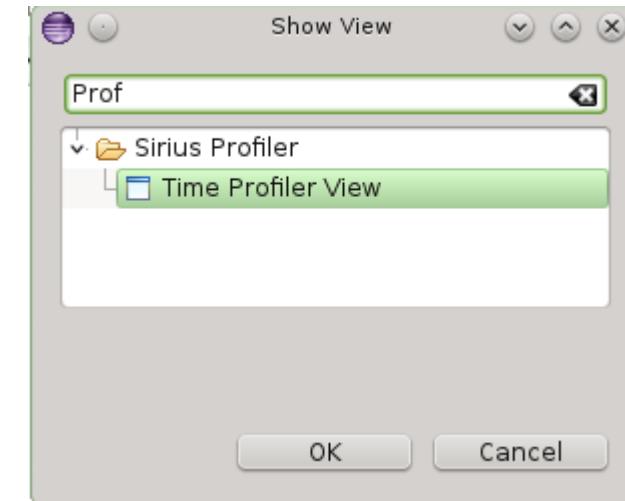
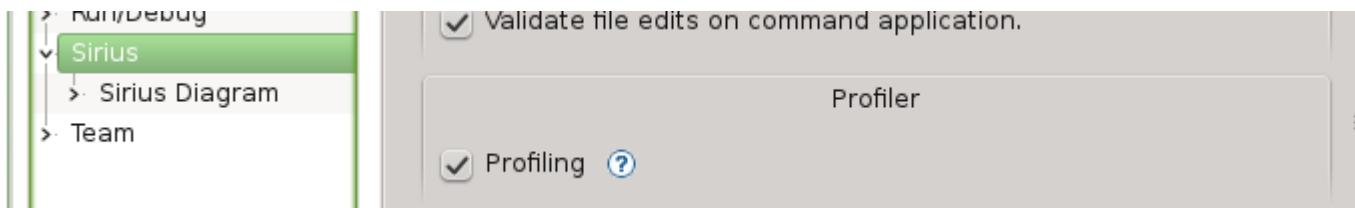
    public EdgeMappingConsistencyTests(EdgeMapping edgemapping) {
        this.underTest = edgemapping;
    }

    @Test
    public void hasReconnect() {
        assertTrue("Edge Mapping " + underTest.getName()
            + " has no reconnect tool",
            underTest.getReconnections().size() > 0);
    }
}
```



Your own checkstyle for Viewpoint Specification model

# Measuring performances



Measure first  
Think about your queries  
Use the cross referencer

Time Profiler View

Task Category	Task Name	Time (ms)	Time (hh:mm:ss,ms)	Occurrences	Minimum
Acceleo	feature:eOperations	0	0:0:0,0	36	0
Acceleo	service:getVisibleAnnotations(diagram)	2	0:0:0,2	12	0
Other	Other	22	0:0:0,22	0	1
DDiagram	Get edge's candidates	18	0:0:0,18	24	1
DDiagram	Compute edge source/target views	5	0:0:0,5	48	0
DDiagram	Get edge's candidates	13	0:0:0,13	24	1
Acceleo	feature:eType	0	0:0:0,0	12	0
Acceleo	service:eContainerEContainer	0	0:0:0,0	12	0
Acceleo	feature:eContainer	0	0:0:0,0	12	0
Acceleo	feature:eSuperTypes	0	0:0:0,0	18	0
Acceleo	Check precondition expressions	1	0:0:0,1	24	0
DDiagram	Get node's candidates	6	0:0:0,6	12	0
Other	Other	6	0:0:0,6	0	1
DDiagram	Updating all edges	37	0:0:0,37	24	0
Other	Other	141	0:0:0,141	0	20

# Just an Eclipse Plugin

Language	files	blank	comment	code
Java	18	353	481	1993
XML	5	9	22	194
Maven	1	8	4	31
HTML	2	5	0	25
SUM:	26	375	507	2243

Tycho build  
JUnit and SWTBot Tests

# Break 3

Model @ runtime means some things are easier.  
You have to care about performances to have good ones  
Modeling Workbenches are just Eclipse Plugins.

# Roadmap

Luna M3 : built and aggregated

Luna M4 : installable and usable in general

Luna M5 : better property views, other stuff

**Luna : Sirius based on GMF based modelers  
shipped with Luna**

# **The Sirius BOF**

## **The Booth**

