

# Ecore Tools

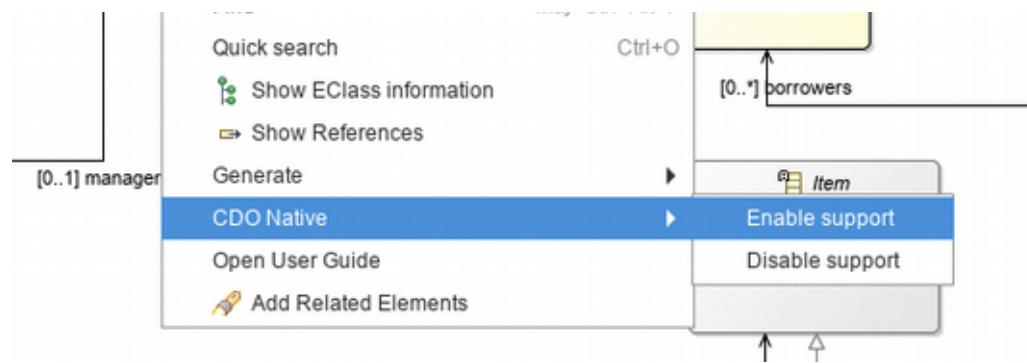
*Your utility belt for Ecore Modeling*

# Productivity tools

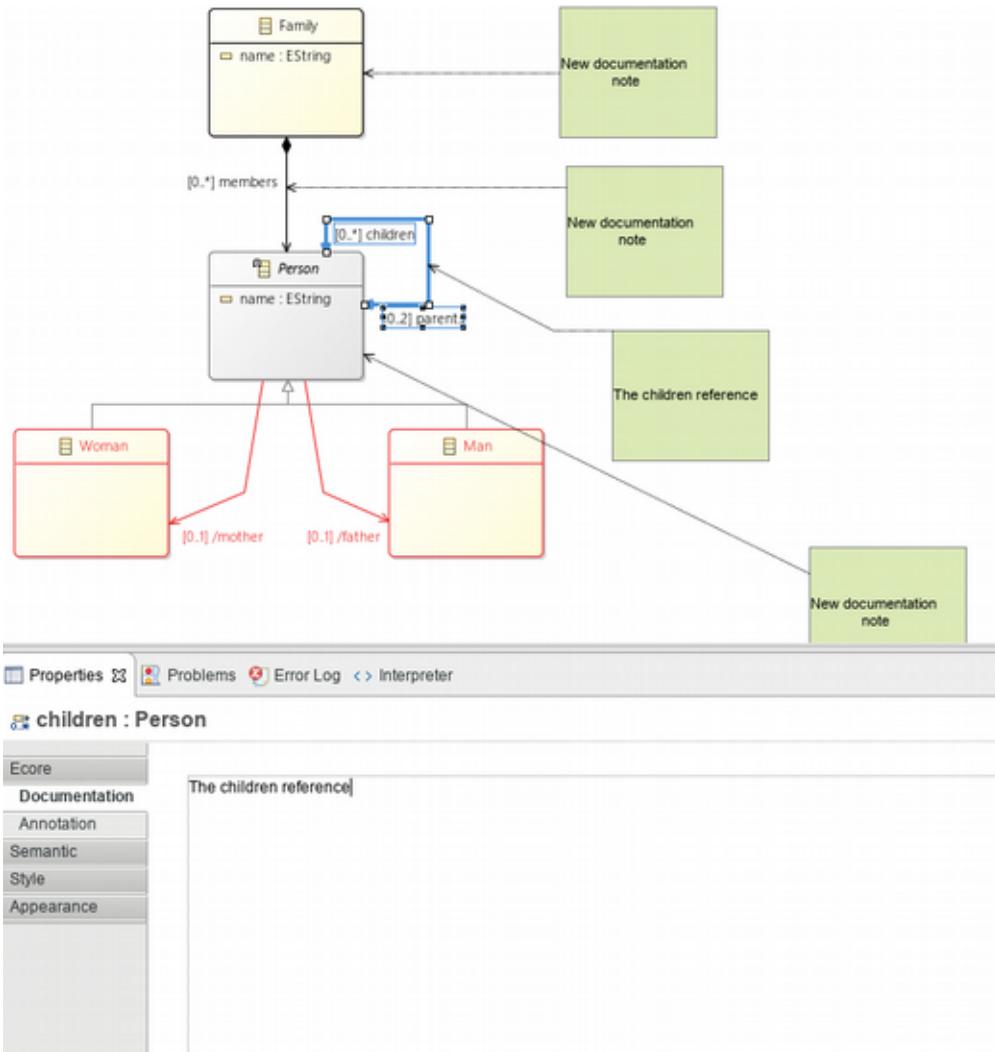
- Ecore Design Checklist :  
<http://cedric.brun.io/eclipse.ecore-design-checklist-part1/>
- Automatic reload of genmodel
- Better default in .genmodel generated by the wizard
- Package Dependencies



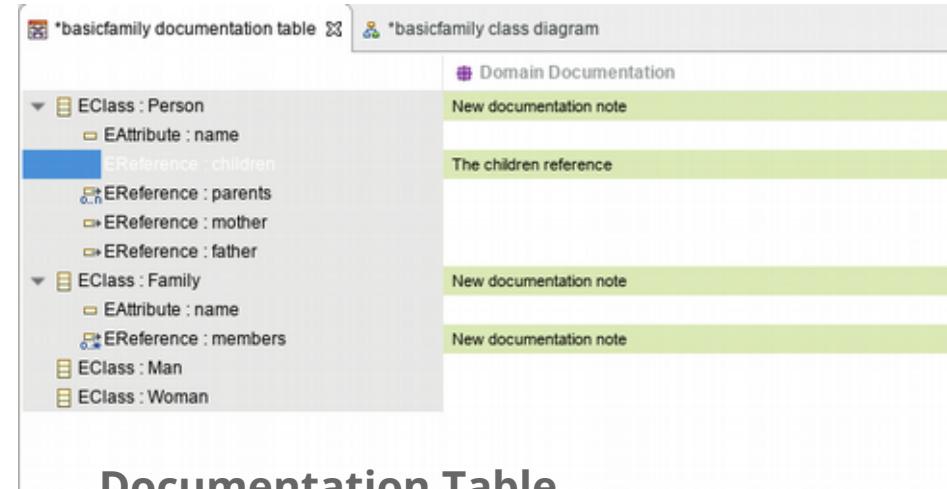
- CDO Configuration



# Documentation annotations



Documentation layer in Entities Diagram



Documentation Table

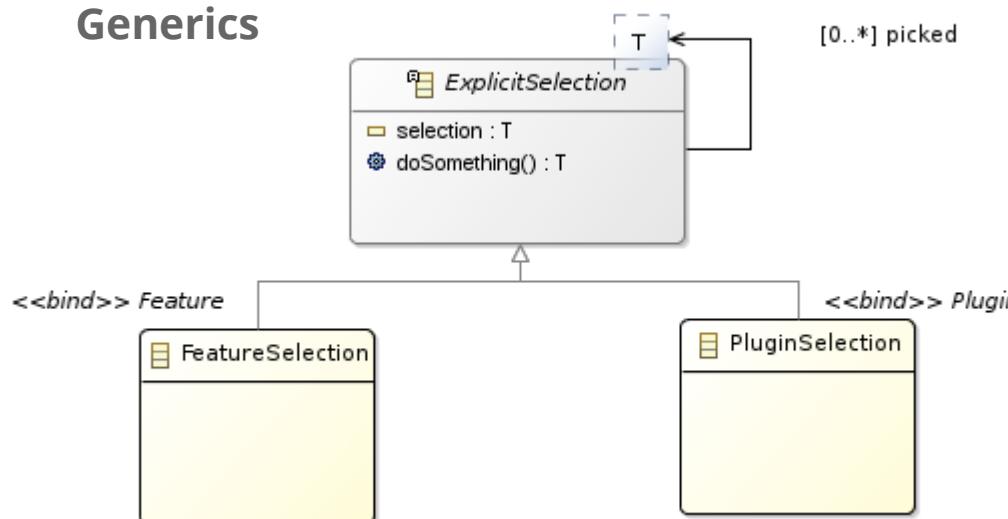
# a few Shortcuts

- **Direct-Edit Syntax**
  - « Something » => change name of feature
  - «:SomeType » => only change the eType
  - «1» => only set cardinality to 1..x
  - « \* » => only set cardinality to x..\*
  - « /Something » => make the feature derived
  - « = something » => set the default value literal
  - [...]
- **CTRL+H** to **Hide** a shape
- **CTRL+O** to navigate within the diagram
- **MAJ+DEL** delete from diagram (vs **DEL** delete from model)

# Modeling

EAnnotation

## Generics



Person

EAnnotation http://www.eclipse.org/emf/2002/GenModel

Source: <http://www.eclipse.org/emf/2002/GenModel>

References: [?](#)

Add Entry

Key: documentation Value: New documentation note [?](#) [X](#)

## newOperation1(EString, EInt)

Ecore

Parameters

Documentation

Annotation

Generation

Semantic

Style

Appearance

Parameters

Name: param1 EType: EString

Name: param2 EType: EInt

## Parameters

# Constraints

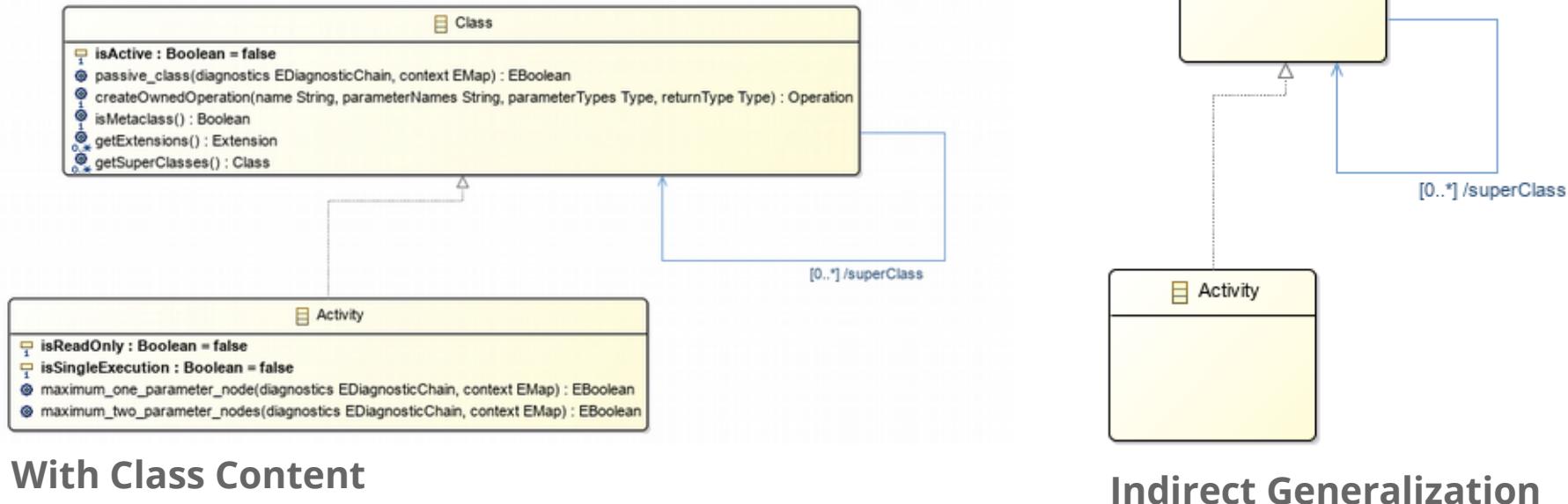
The screenshot shows the Obeo Development Environment interface. On the left, the 'ConstraintsValidator.java' editor displays Java code for validating constraints. In the center, the 'constraints class di' editor shows a UML diagram with a 'Purchase' class having an attribute 'name : EString'. A constraint named 'namelsConform purchaseisValid someOtherConstraint' is associated with this attribute. On the right, the 'constraints.genmodel' palette lists various model elements and constraints.

```
/*
 * Validates the someOtherConstraint constraint of '<em>Purchase</em>'.
 * <!-- begin-user-doc -->
 * <!-- end-user-doc -->
 * @generated
 */
public boolean validatePurchase_someOtherConstraint(Purchase purchase) {
    // TODO implement the constraint
    // -> specify the condition that violates the constraint
    // -> verify the diagnostic details, including severity,
    // Ensure that you remove @generated or mark it @generated if (false) {
        if (diagnostics != null) {
            diagnostics.add
                (createDiagnostic
                    (Diagnostic.ERROR,
                     DIAGNOSTIC_SOURCE,
                     0,
                     " UI_GenericConstraint_diagnostic",
                     new Object[] { "someOtherConstraint", g,
                     new Object[] { purchase },
                     context)));
        }
        return false;
    }
    return true;
}

/**
 * Validates the purchaseInvariant constraint of '<em>Purchase</em>'.
 * <!-- begin-user-doc -->
 * <!-- end-user-doc -->
 * @not-generated
 */
```

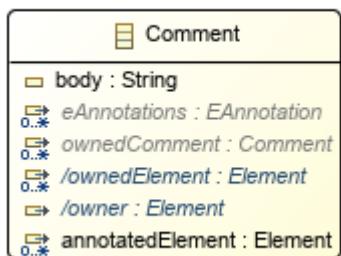
(\*) would be nice to have tooled support for AQL constraints, it just did not happen yet

# Focused diagrams

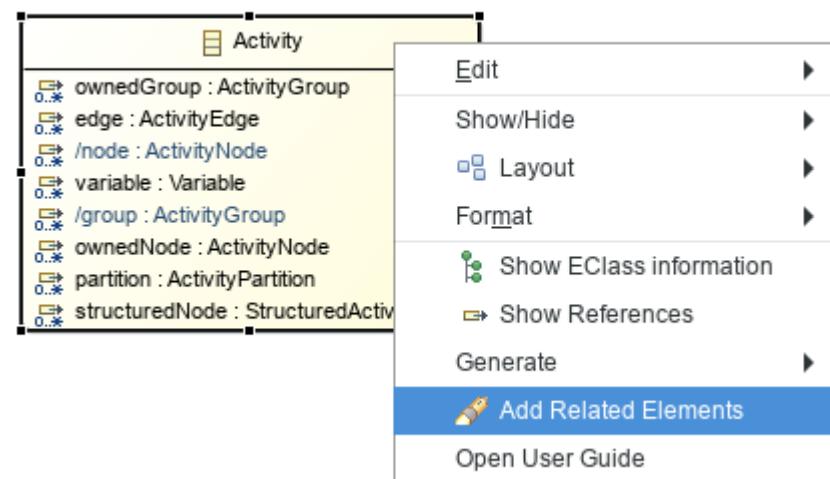


With Class Content

Indirect Generalization



With inherited features



Add Related Elements in Diagram

# Generation settings & launch

UML Class Diagram illustrating inheritance:

```
classDiagram
    class Woman
    class Man
    Woman <|-- Man
```

The diagram shows two classes, **Woman** and **Man**. A generalization relationship exists between **Woman** and **Man**, indicated by a line from **Woman** to **Man** with an open diamond symbol. There are two associations: one from **Woman** to **Man** labeled **[0..1] /mother**, and another from **Man** to **Woman** labeled **[0..1] /father**.

Sirius Generation Settings (basicfamily)

**Directories**

- Model: `/org.eclipse.sirius.sample.basicfamily/src`
- Edit: `/org.eclipse.sirius.sample.basicfamily.edit/src`
- Editor: `/org.eclipse.sirius.sample.basicfamily.editor/src`

**GenPackage**

Prefix: `Basicfamily`

Base Package: `org.eclipse.sirius.sample`

Resource Type:  
 None  Basic  XML  XML

Disposable Provider Factory  
 Adapter Factory  
 Initialize by Loading

Interface:

Metadata:

Implementation: `impl`

Utility: `util`

Provider: `provider`

Presentation: `presentation`

Tools:

UML Class Diagram illustrating associations:

```
classDiagram
    class Library
    class Book
    class Employee
    Library "0..1" --> "0..1" Book : parent
    Library "0..*" --> "0..*" Employee : branches
    Book "*" --> "0..1" Employee : stock
```

The diagram shows three classes: **Library**, **Book**, and **Employee**. There are three associations:

- A bidirectional association between **Library** and **Book** labeled **[0..1] parent**.
- A bidirectional association between **Library** and **Employee** labeled **[0..\*] branches**.
- A unidirectional association from **Book** to **Employee** labeled **[0..1] stock**.

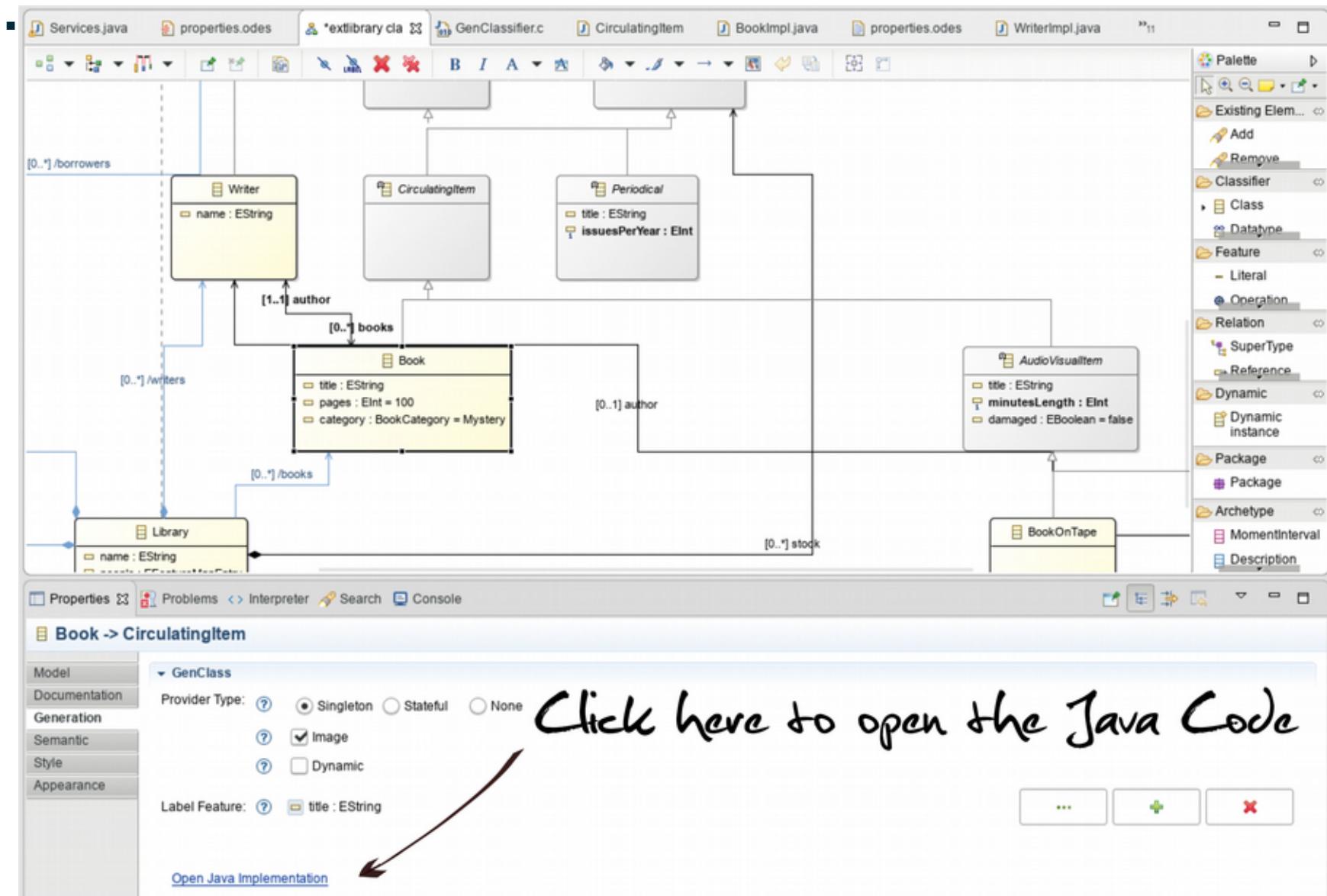
Sirius Context Menu (F5) for **Library**

- Edit
- Refresh
- Unsynchronized
- Export diagram as image
- Show/Hide
- Select
- Layout
- Reset Origin
- Validate diagram
- Find Maj+Ctrl+Alt+F
- Quick search Ctrl+O
- Generate**
- Add Related Elements
- Open User Guide

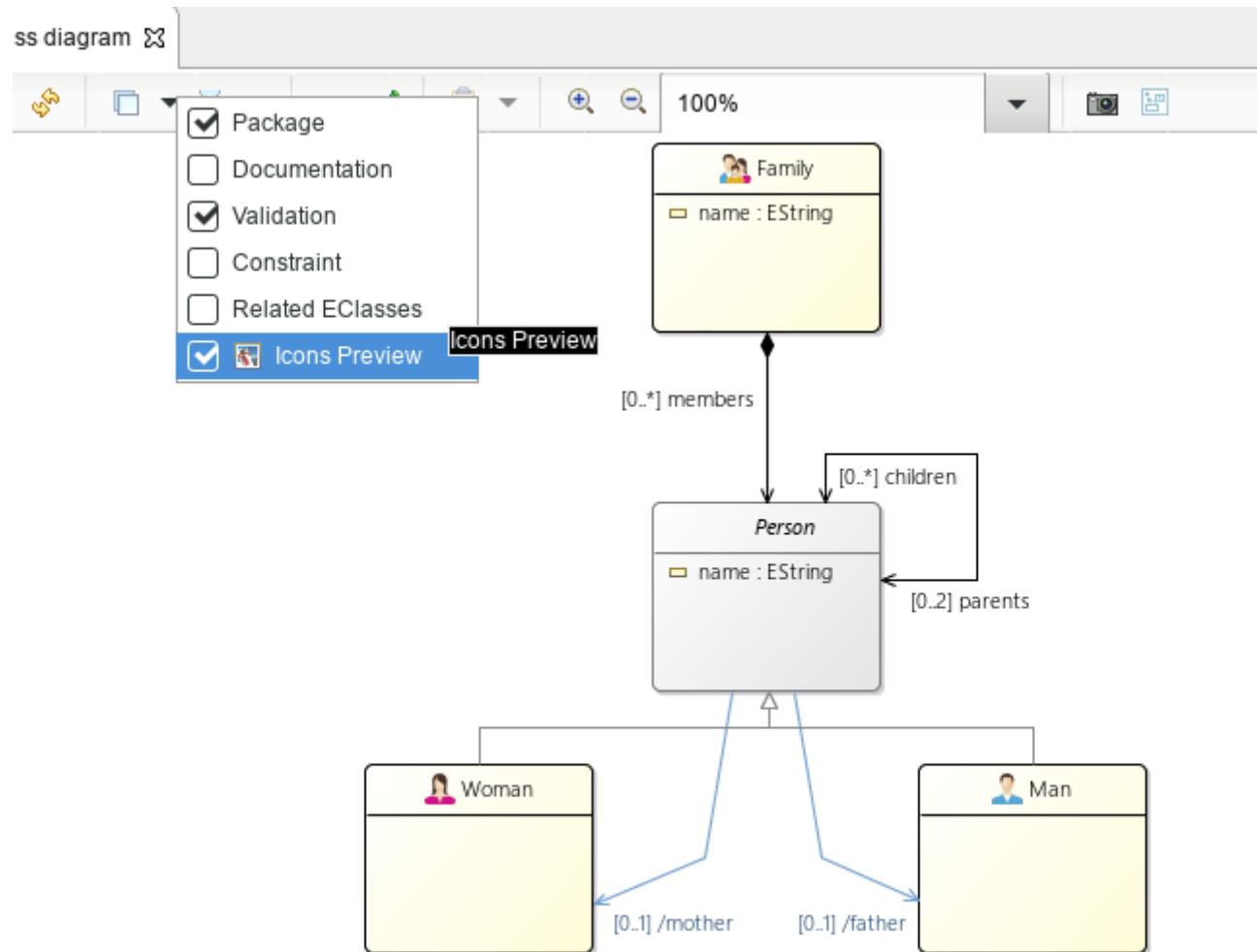
Generate Options:

- Generate Model Code
- Generate Edit Code
- Generate Editor Code
- Generate All**

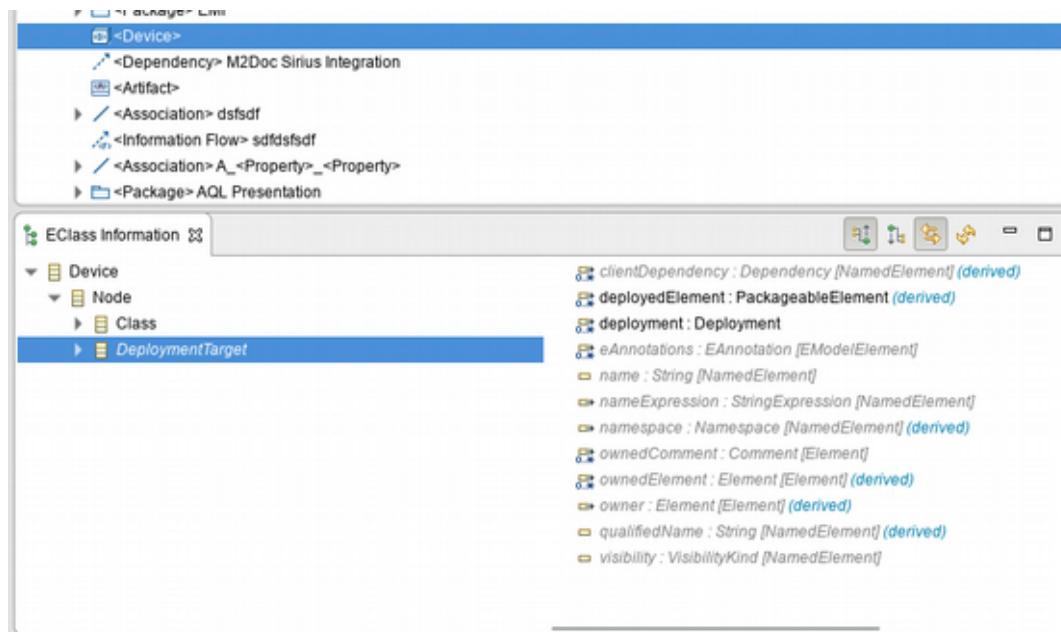
# Navigate to Java



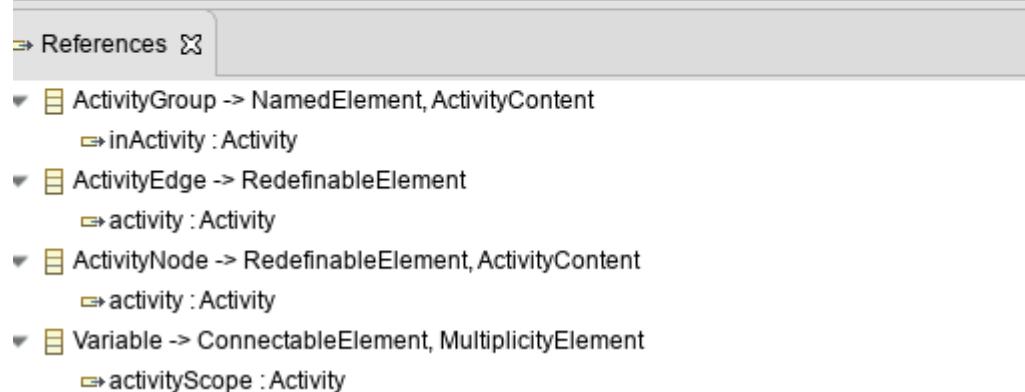
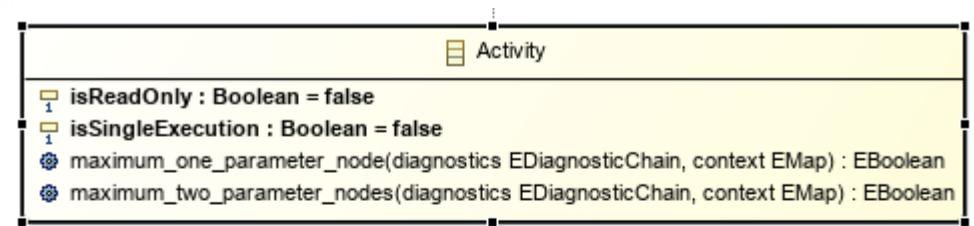
# Icons preview



# Eclipse Views



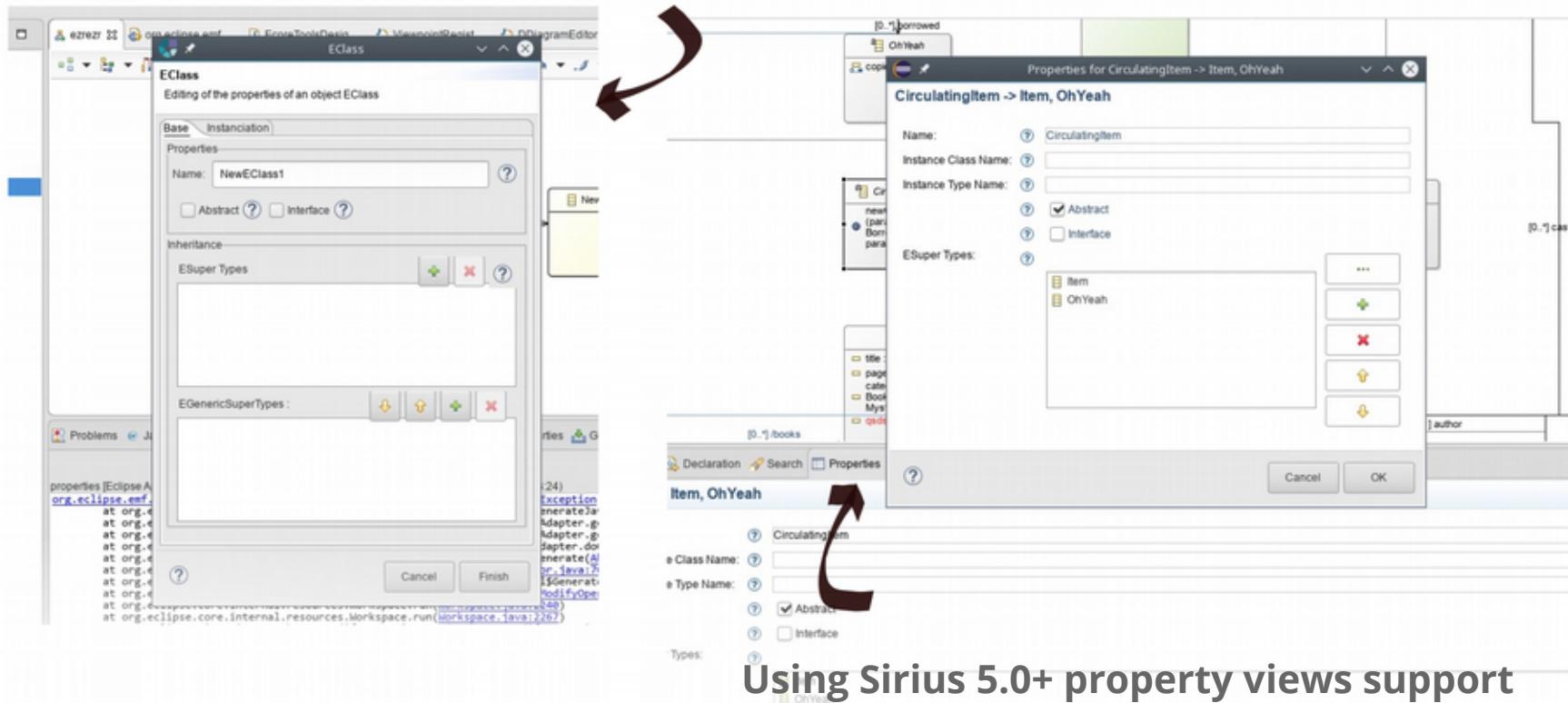
EClass Information to inspect types of selection



References to inspect type relationships

# Reworked Dialogs

Generated with Extended Editing Framework 1.x



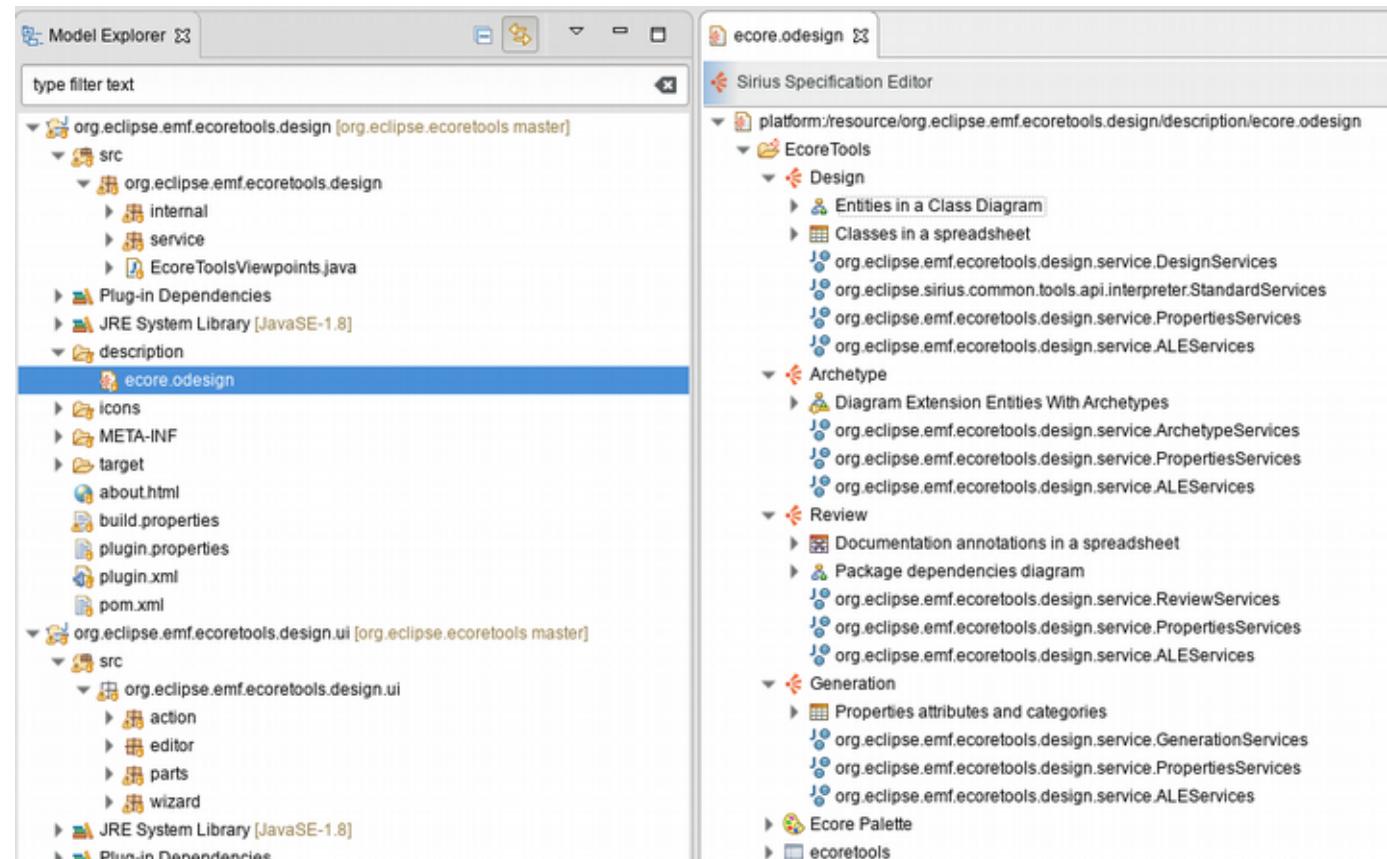
# visualizing Xcore models

The screenshot shows the Sirius IDE interface with three main views:

- Model Explorer**: Shows the project structure under the `basicfamily` package, including `GenModel`, `Person`, `Family`, and their subclasses `Man` and `Woman`.
- Basicfamily.xcore**: Displays the Xcore code defining the model. It includes annotations for Ecore, GenModel, and various class definitions like `Person`, `Family`, `Man`, and `Woman`, along with their associations and multiplicity constraints.
- basicfamily class diagram**: Visualizes the class hierarchy and relationships. It shows `Family` as an abstract class with a `[0..*]` association named `members` pointing to `Person`. `Person` has two subclasses: `Man` and `Woman`. `Man` has a `[0..1]` association named `/father` pointing to `Person`. `Woman` has a `[0..2]` association named `/mother` pointing to `Person`. `Person` also has a `[0..*]` association named `children` pointing back to `Person`.

# powered by Eclipse Sirius

Language	files	blank	comment	code
Java	21	441	534	2659
XML	5	9	22	197
Maven	1	8	4	31
HTML	2	5	0	25
SUM:	29	463	560	2912

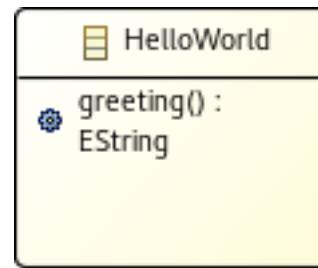


- Sirius **Best Practices**
- Inspectable
- Extensible

# envisionned Roadmap

- Versionning, deprecation annotations
- **ALE (Action Language for EMF)**
  - Contribution in EcoreTools (INRIA)
  - Provides
    - Assignements
    - More control flow
    - Multi-inheritance
    - Runtime data
    - Type inference

Ecore



Open Class

```
open class HelloWorld {  
  
    String msg := "Hi!";  
  
    override EString greeting () {  
        result := self.msg;  
    }  
}
```

# Download

- the Eclipse Modeling Package, through the Marketplace
- Packaged in Obeo Designer Community Edition (EPL)
  - <https://www.obedesigner.com/en/download>



## A Free Eclipse Package

Start using Sirius with **Obeo Designer Community** edition, a ready-to-use package including only Open Source technologies.

Download Obeo Designer Community 10.0

For Linux 64 Bit

Required configuration

Download Obeo Designer for another platform

Note for installation on Mac OS

What's new in Obeo Designer 10.0?

Download previous versions

