

FROM RESEARCH TO INDUSTRY



ELEMENTTYPES
CONFIGURATION
FRAMEWORK



Papyrus Team

Florian NOYRIT – florian.noyrit@cea.fr

list

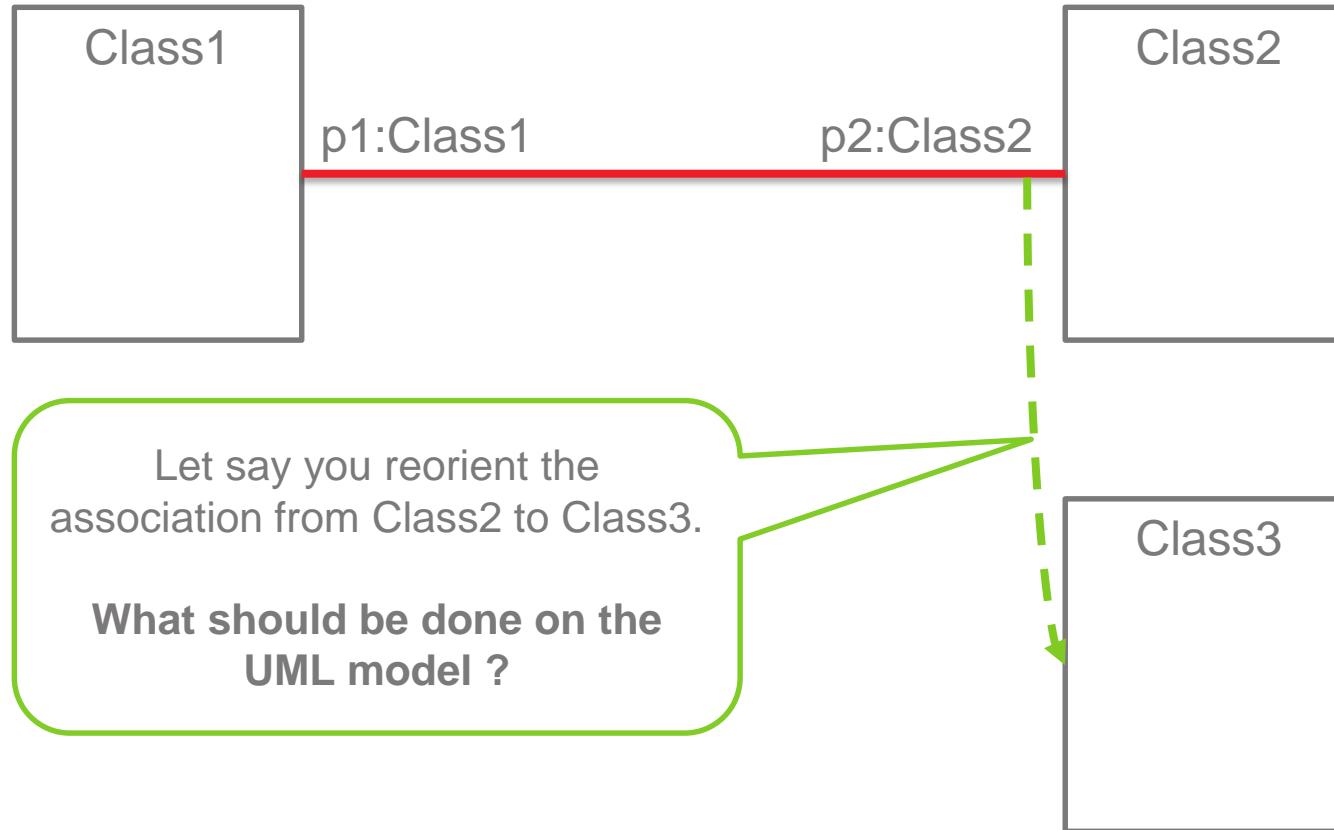
AGENDA

- Why such a framework?
- The Basics
- How is it used in Papyrus?
 - The Association Example
- ElementTypeConfiguration for DSML designers
- ElementTypes and Diagrams
- Advanced Elementtype configuration
- Further developments

**WHY SUCH A
FRAMEWORK ?**

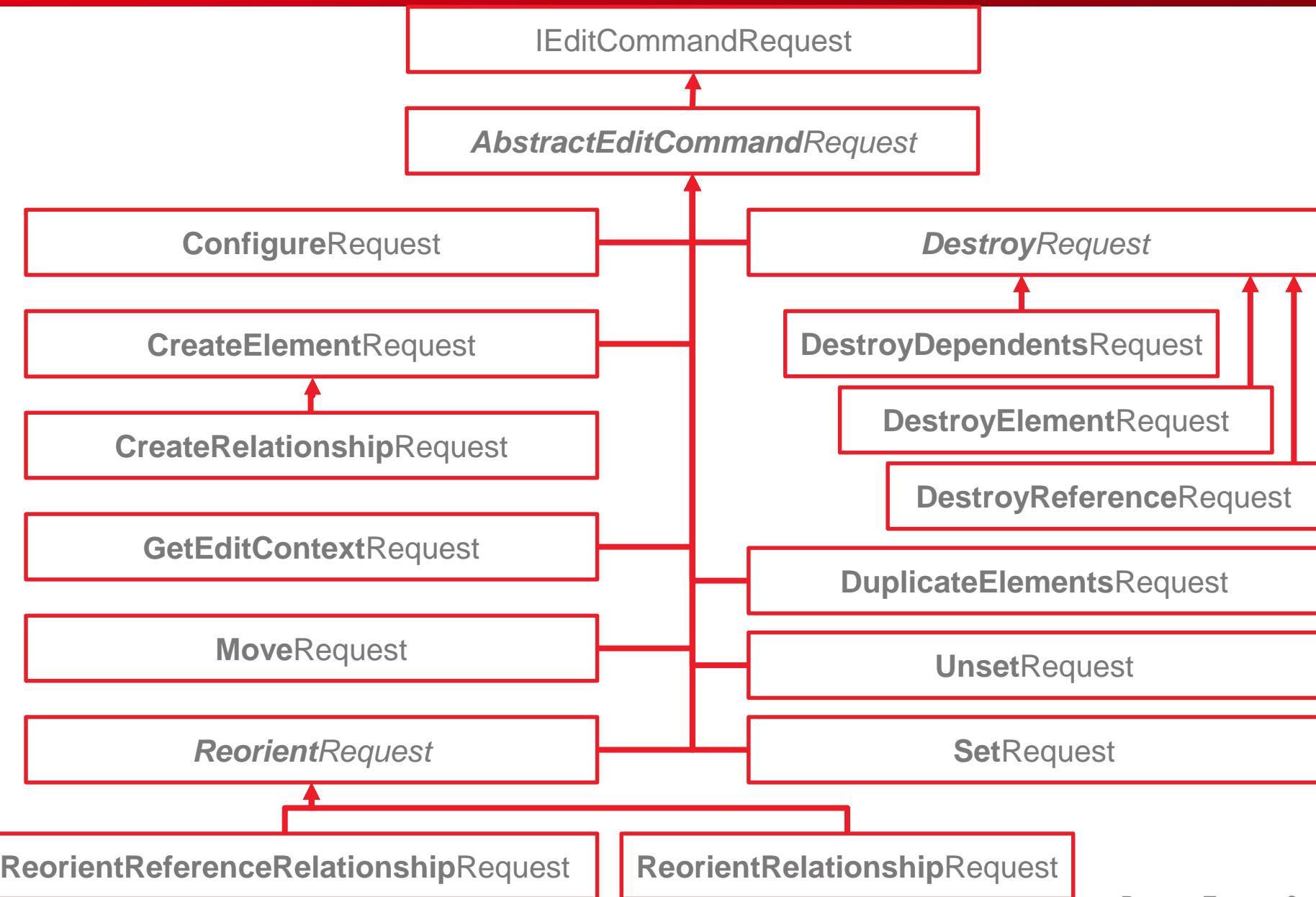
MOTIVATING EXAMPLE

- **UML2 API provides low level metamodel edition facilities :**
- **Example of the creation of a Association :**
 - Create an Association :
 - `association = UMLFactory.eINSTANCE.createAssociation();`
 - Name it “association1”
 - `association.setName("association1");`
 - Create the member ends
 - `p1 = UMLFactory.eINSTANCE.createProperty();`
 - `p2 = UMLFactory.eINSTANCE.createProperty();`
 - Set the member ends
 - `assoc.getMemberEnds().add(...)`
 - Type the member ends, set the multiplicities, ...



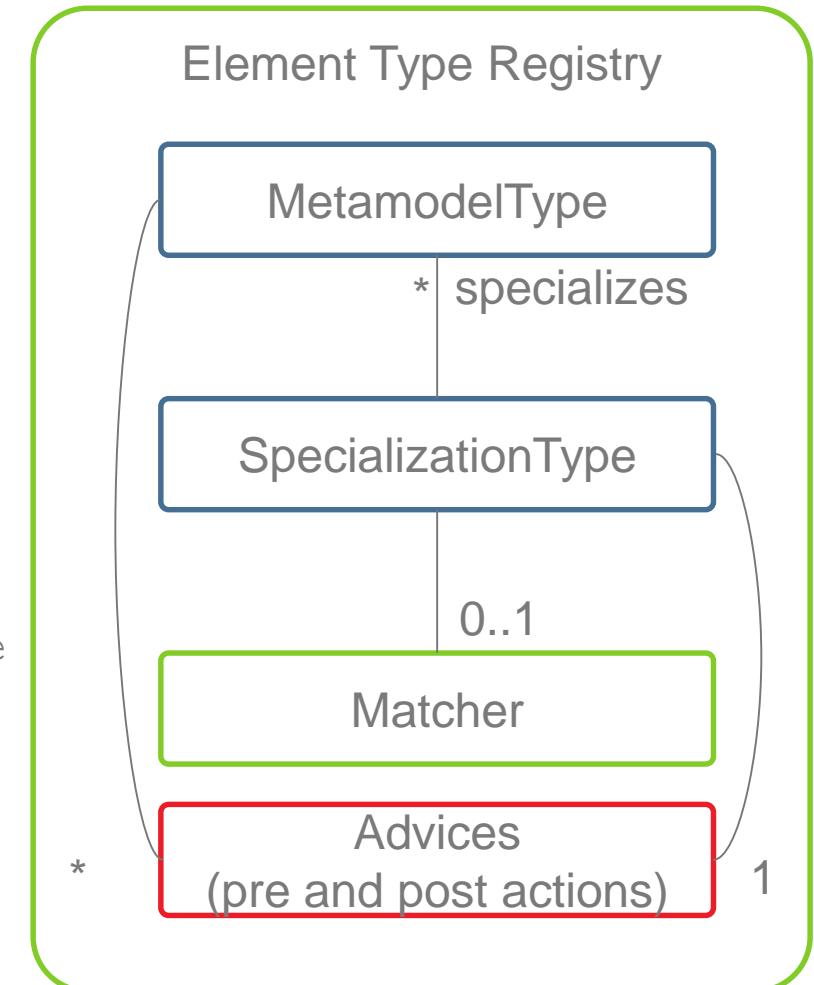
BASICS



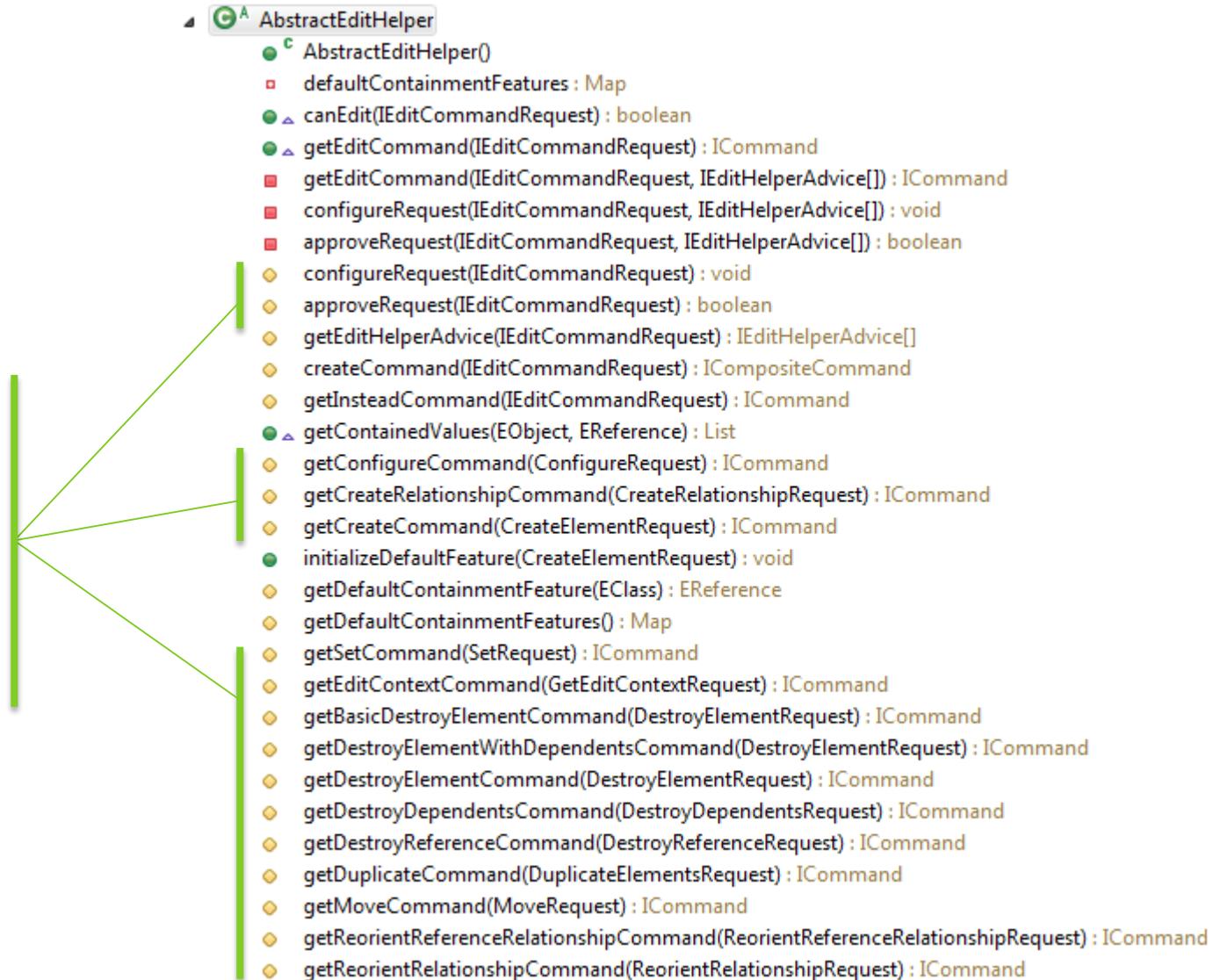


HOW COMMANDS ARE COMPUTED BY THE FRAMEWORK

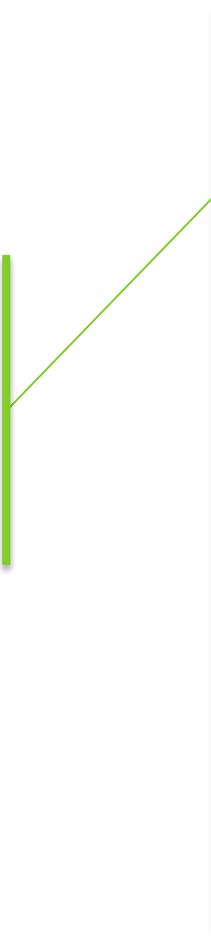
1. Find the **elementType** corresponding to the element being edited
 - Use the **matchers** to find the appropriate one.
2. Get the base command (from the specialized **metamodelType**)
3. Look for **advices** that are bound to the **elementType**.
4. Produce the final command: base command surrounded by the pre and post actions from the **advices** bound to the **elementType**.



Methods to override to provide the appropriate commands



Methods to override to define the pre or post actions



```
class G AbstractEditHelperAdvice
{
    <> C AbstractEditHelperAdvice()
    <> A getBeforeEditCommand(IEditCommandRequest) : ICommand
    <> A getAfterEditCommand(IEditCommandRequest) : ICommand
    <> A getBeforeCreateRelationshipCommand(CreateRelationshipRequest) : ICommand
    <> A getAfterCreateRelationshipCommand(CreateRelationshipRequest) : ICommand
    <> A getBeforeCreateCommand(CreateElementRequest) : ICommand
    <> A getAfterCreateCommand(CreateElementRequest) : ICommand
    <> A getBeforeConfigureCommand(ConfigureRequest) : ICommand
    <> A getAfterConfigureCommand(ConfigureRequest) : ICommand
    <> A getBeforeDestroyElementCommand(DestroyElementRequest) : ICommand
    <> A getAfterDestroyElementCommand(DestroyElementRequest) : ICommand
    <> A getBeforeDestroyDependentsCommand(DestroyDependentsRequest) : ICommand
    <> A getAfterDestroyDependentsCommand(DestroyDependentsRequest) : ICommand
    <> A getBeforeDestroyReferenceCommand(DestroyReferenceRequest) : ICommand
    <> A getAfterDestroyReferenceCommand(DestroyReferenceRequest) : ICommand
    <> A getBeforeDuplicateCommand(DuplicateElementsRequest) : ICommand
    <> A getAfterDuplicateCommand(DuplicateElementsRequest) : ICommand
    <> A getBeforeEditContextCommand(GetEditContextRequest) : ICommand
    <> A getAfterEditContextCommand(GetEditContextRequest) : ICommand
    <> A getBeforeMoveCommand(MoveRequest) : ICommand
    <> A getAfterMoveCommand(MoveRequest) : ICommand
    <> A getBeforeReorientReferenceRelationshipCommand(ReorientReferenceRelationshipRequest) : ICommand
    <> A getAfterReorientReferenceRelationshipCommand(ReorientReferenceRelationshipRequest) : ICommand
    <> A getBeforeReorientRelationshipCommand(ReorientRelationshipRequest) : ICommand
    <> A getAfterReorientRelationshipCommand(ReorientRelationshipRequest) : ICommand
    <> A getBeforeSetCommand(SetRequest) : ICommand
    <> A getAfterSetCommand(SetRequest) : ICommand
    <> A createType(EObject, IElementType, IProgressMonitor) : EObject
    <> A createType(EObject, IElementType, Map, IProgressMonitor) : EObject
    <> A configureRequest(IEditCommandRequest) : void
    <> A approveRequest(IEditCommandRequest) : boolean
}
```

**HOW IS IT USED IN
PAPYRUS ?**

Usecase
UML Diagram

Class
UML Diagram

...
UML Diagrams

SysML
Diagrams

UML Types
(org.eclipse.papyrus.uml.service.types)

SysML Types
(org.eclipse.papyrus.sysml.service.types)

ElementTypesConfigurations Framework
(org.eclipse.papyrus.infra.elementtypesconfigurations)

GMF ElementTypes

UML Diagrams

diagram
.elementtype
configuration

specializes

uml
.elementtype
configuration

UML Types (org.eclipse.papyrus.uml.service.types)

EditHelpers
.java

Advices
.java

Conforms to

Elementtypeconfiguration
.ecore

ElementTypesConfigurations Framework
(org.eclipse.papyrus.infra.elementtypesconfiguration)

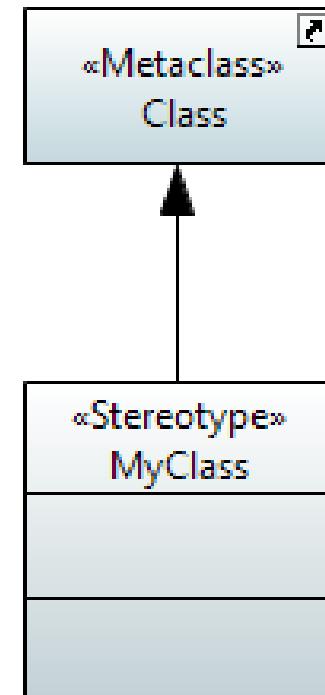
GMF ElementTypes

THE ASSOCIATION EXAMPLE

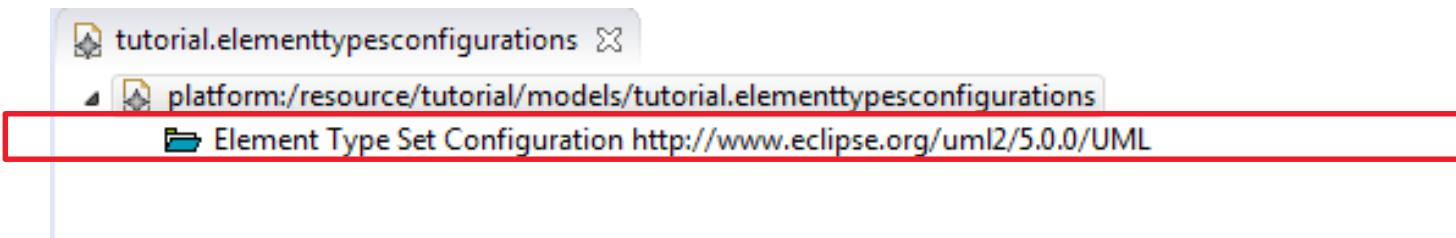
Demo

ELEMENTTYPECONFIGURATION FOR DSML DESIGNERS

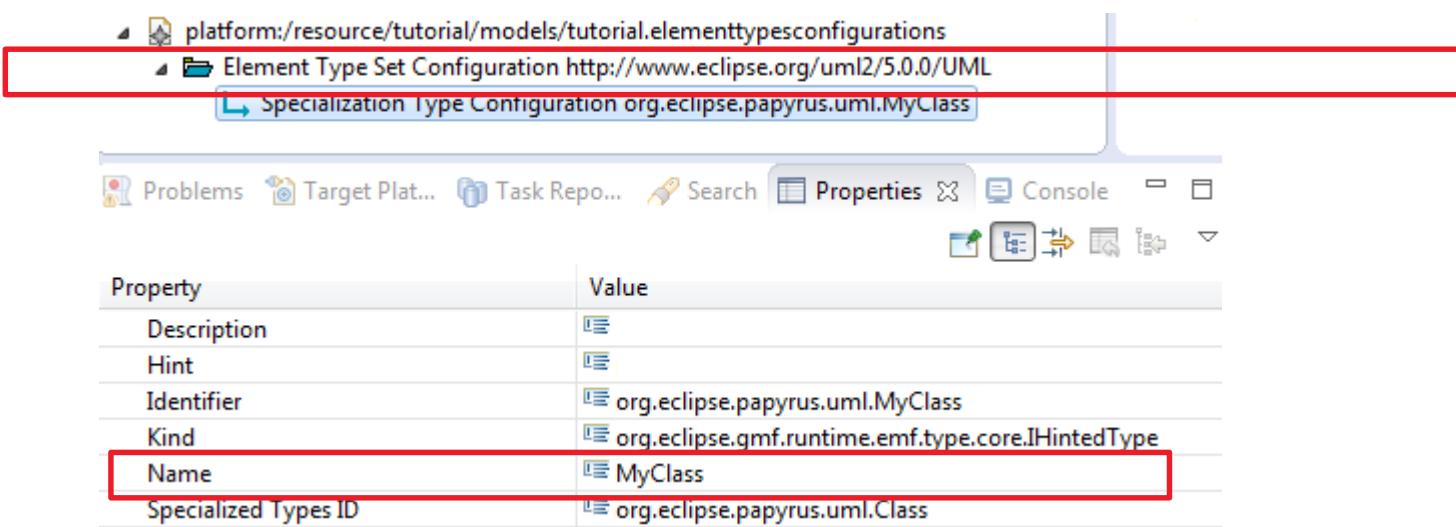
- Create specific element type that will handle a domain specific concept identified by a specific stereotype
 - Let say, there is an OCL constraint that enforces the class stereotyped by "MyClass" to be active



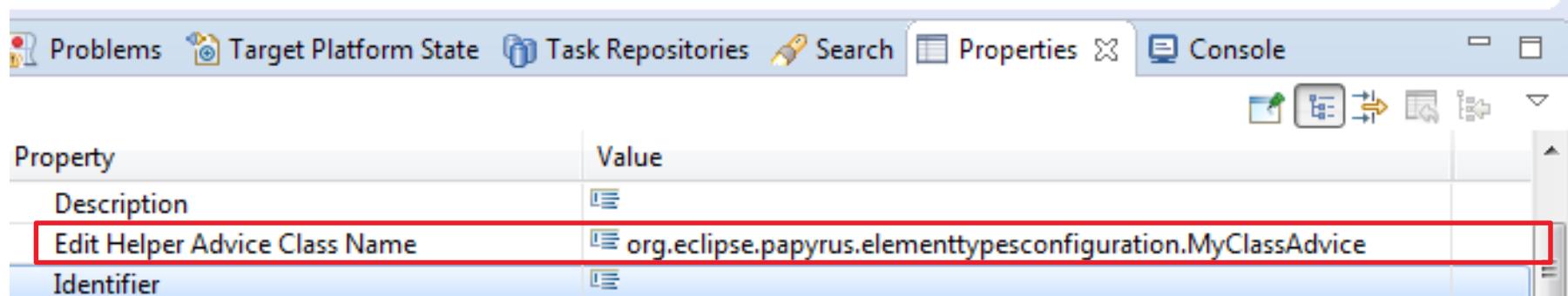
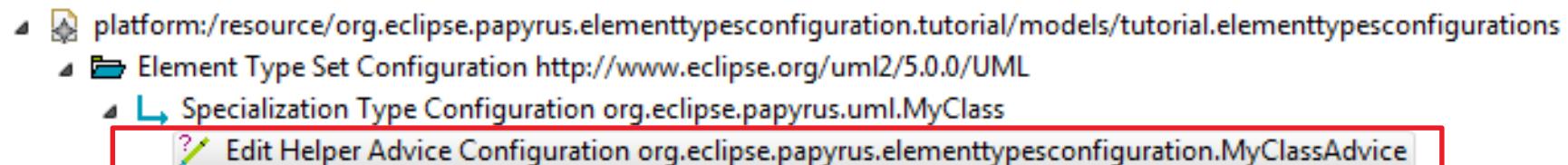
- Create an Elementtypesconfigurations model (e.g. tutorial.elementtypesconfigurations)
 - Create one ElementTypeConfigurationSet



- Add a specialization type



- Create an advice and set the qualifiedName of the class

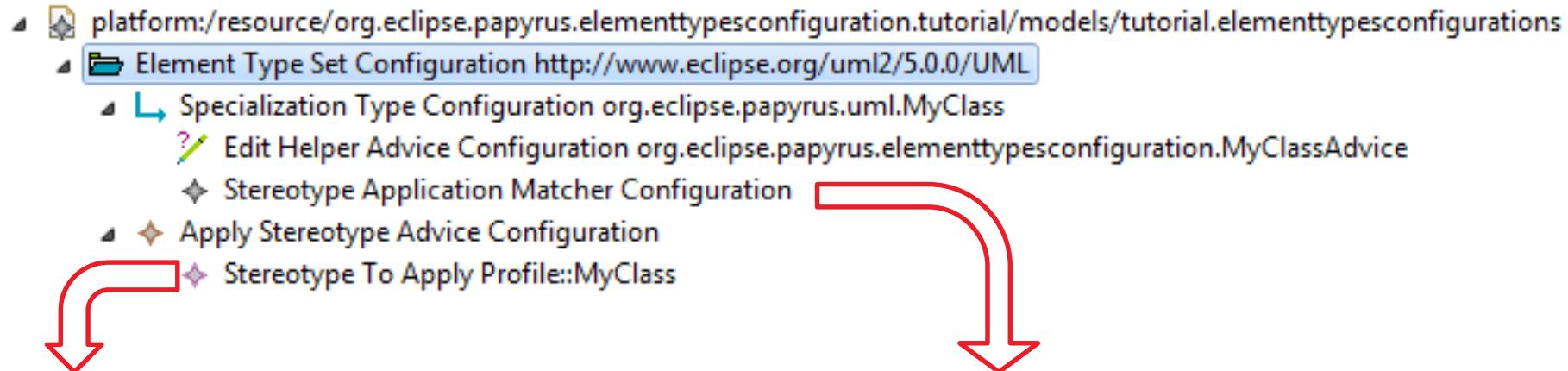


- Develop the advice:

```
@Override  
protected ICommand getAfterConfigureCommand(ConfigureRequest request) {  
  
    ICommand resultCommand = null;  
  
    EObject elementToConfigure = request.getElementToConfigure();  
  
    ...  
  
    if (elementToConfigure instanceof org.eclipse.uml2.uml.Class) {  
        resultCommand = new AbstractTransactionalCommand(editingDomain, "Editing type",  
            Arrays.asList((WorkspaceSynchronizer.getFile(elementToConfigure.eResource())))) {  
  
            @Override  
            protected CommandResult doExecuteWithResult(IProgressMonitor monitor, IAdaptable info)  
                throws ExecutionException {  
                ((org.eclipse.uml2.uml.Class) elementToConfigure).setIsActive(true);  
                return CommandResult.newOKCommandResult(elementToConfigure);  
            }  
        };  
  
        return resultCommand.reduce();  
  
    }  
}
```

- How to use the Stereotype information?

- Add an advice that will apply the stereotype as a post action ?
- Add a matcher to the elementType that will fire when the stereotype is applied to a model element ?



Property	Value
Required Profiles	Profile
Stereotype Qualified Name	Profile:: MyClass
Update Name	false

Property	Value
Matcher Class Name	
Profile Uri	
Stereotypes Qualified Names	Profile:: MyClass

- Declare the elementType model using the `org.eclipse.papyrus.infra.elementtypesconfigurations.elementTypeSetConfiguration` extension point

The screenshot shows the Eclipse Papyrus Extensions view. The title bar says "Extensions". Below it, a toolbar has icons for sorting (down arrow), filtering (a-z), and other operations. A "type filter text" input field is present. The main area is titled "All Extensions" and contains a list of extension points. One item is selected: "org.eclipse.papyrus.infra.elementtypesconfigurations.elementTypeSetConfiguration" (with a blue border). To the right of the list are "Add..." and "Remove" buttons. Below the list is a "Extension Element Details" section with a "Required fields are denoted by '*'." note. It contains two input fields: "id*" with value "org.eclipse.papyrus.elementtypesconfiguration.tutorial.elementTypeSet" and "path*" with value "models/tutorial.elementtypesconfigurations" and a "Browse..." button. At the bottom right are four small icons: a green circle, a sun, a gear, and a question mark.

Extensions

All Extensions

Define extensions for this plug-in in the following section.

type filter text

Add... Remove

Extension Element Details

Set the properties of 'elementTypeSet' Required fields are denoted by '*'.

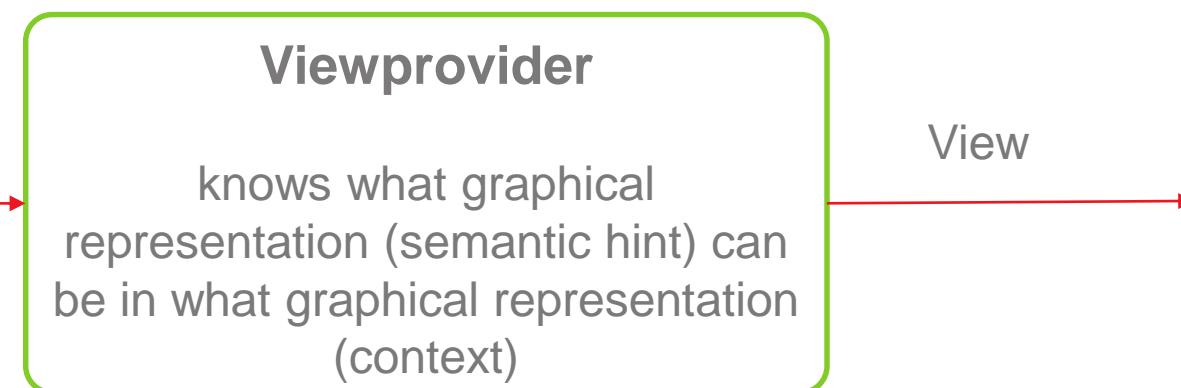
id*: org.eclipse.papyrus.elementtypesconfiguration.tutorial.elementTypeSet

path*: models/tutorial.elementtypesconfigurations Browse...

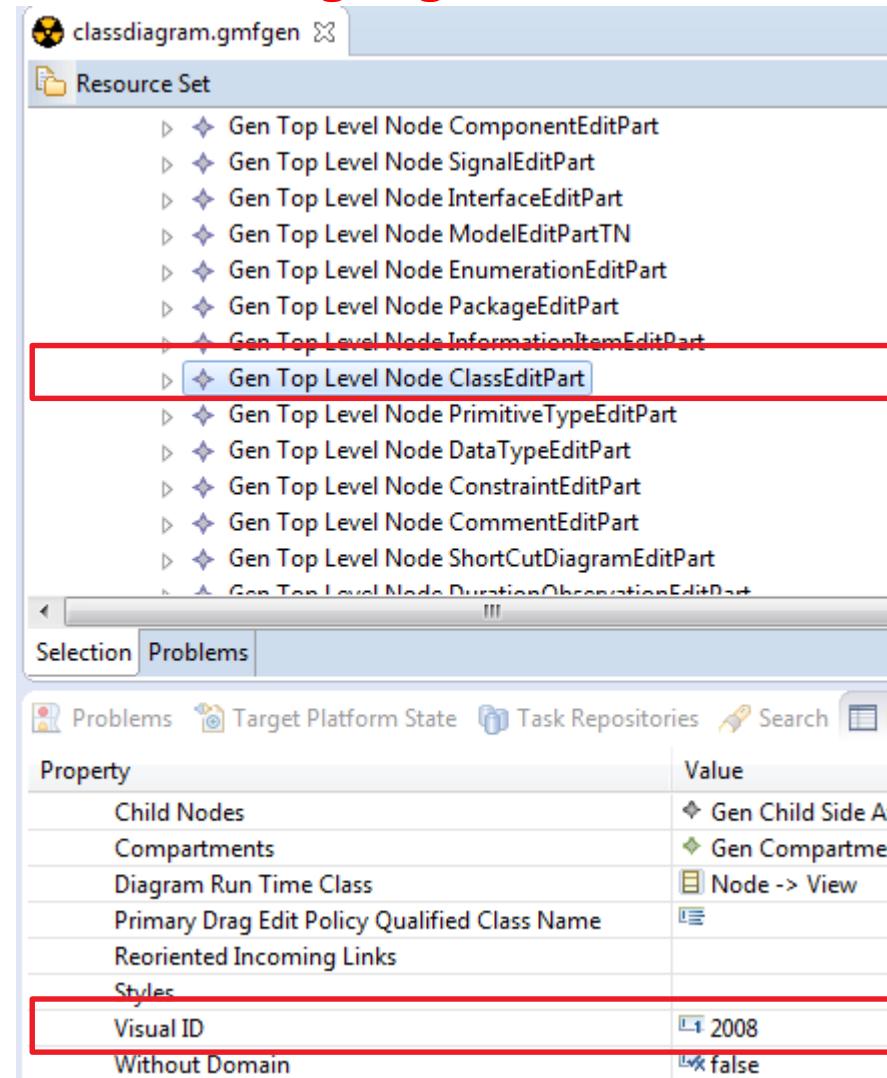
ELEMENTTYPES AND DIAGRAMS

- **ElementTypes are mostly UML model edition information**
- **Nevertheless, diagram editors will “ask” how the elementtype should be represented : the “Semantic Hint”**
 1. A creation tool is selected by user in the palette
 2. The user clicks somewhere in the diagram (the context)
 3. Two commands are prepared
 1. The semantic command (using the elementtype framework as presented before)
 2. The graphical command (using viewproviders)
 4. If both the semantic and graphical command are executable, the element is created in the UML model and the diagrammatic representation is added to the diagram.

- Semantic Hint
- Context



- Where can I find the hints provided by a given diagram in Papyrus ? → in the gmfgen model of this diagram



- Set the hint found in the class diagram to the elementType

The screenshot shows the Eclipse Papyrus Element Types Configuration interface. At the top, there is a tree view of configurations:

- platform:/resource/org.eclipse.papyrus.elementtypesconfiguration.tutorial/models/tutorial.elementtypesconfigurations
- Element Type Set Configuration http://www.eclipse.org/uml2/5.0.0/UML
 - Specialization Type Configuration org.eclipse.papyrus.uml.MyClass
 - Edit Helper Advice Configuration org.eclipse.papyrus.elementtypesconfiguration.MyClassAdvice
 - Stereotype Application Matcher Configuration
 - Apply Stereotype Advice Configuration
 - Stereotype To Apply Profile::MyClass

Below the tree view is a toolbar with icons for Problems, Target Platform State, Task Repositories, Search, Properties, and Console. The Properties tab is selected.

The Properties view displays a table with two columns: Property and Value. The table contains the following entries:

Property	Value
Description	
Hint	2008

The "Hint" row is highlighted with a red border.

ADVANCED ELEMENTTYPE CONFIGURATION

- For example, a **StereotypeElementType** (that will have a predefined advice and matcher):
 1. Create an extension of the elementTypeConfiguration metamodel: an ecore metamodel with an eClass that inherits from SpecializationTypeConfiguration (and generate the implementation with creation extenders)
 - **stereotypeElementType**
 - **StereotypeElementType -> SpecializationTypeConfiguration**
 - (+) **SpecializationTypeConfiguration**
 - **Profile : EString**
 - **StereotypeQN : EString**
 2. Contribute this new type of elementType with the org.eclipse.papyrus.infra.elementtypesconfigurations.elementTypeCo nfigurationType extension point

```
<elementTypeConfigurationType
configurationClass="org.eclipse.papyrus.elementtypesconfiguration.tutorial.stereotypeE
lementType.StereotypeElementType"
factoryClass="org.eclipse.papyrus.elementtypesconfiguration.tutorial.ElementTypeConfig
urationFactory">
</elementTypeConfigurationType>
```

2. Contribute this new type of elementType with the
org.eclipse.papyrus.infra.elementtypesconfigurations.elementTypeCo
nfigurationType extension point
1. Define the elementtype type metaclass that extends the framework (i.e.
configurationClass)
 2. Define the interpreter extension (i.e. the factoryClass)

Extension Element Details

Set the properties of 'elementTypeConfigurationType' Required fields are denoted by '*'.

factoryClass*:

org.eclipse.papyrus.elementtypesconfiguration.tutorial.StereotypeElementTypeFactory

[Browse...](#)

configurationClass*:

org.eclipse.papyrus.elementtypesconfiguration.tutorial.stereotypeElementType.StereotypeElementType

[Browse...](#)

3. Implement the elementTypeConfiguration framework interpreter
 1. Override the factory to predefine an advice that applies the stereotype

```
public class StereotypeElementTypeFactory extends SpecializationTypeFactory {  
  
    @Override  
    protected IEditHelperAdvice getEditHelperAdvice(SpecializationTypeConfiguration elementTypeConfiguration) {  
  
        1 // Override the factory to predefine an advice that applies the stereotype  
  
    }  
  
    @Override  
    protected IElemMatche createElementMatcher(SpecializationTypeConfiguration configuration) {  
  
        2 //Override the factory to predefine a matcher that fires on the stereotype  
  
    }  
  
}
```

Demo

FURTHER DEVELOPMENTS

- **New high-level requests**
 - Profile related requests (e.g. stereotype application)
- **Contextualized elementtypes**
 - Separate elementtypes by specific domain
 - Have different registry depending on the specific domain
- **Enriched the elementtype predefined types, the matched predefined types and the predefined advices types**

THANK YOU FOR YOUR ATTENTION

Commissariat à l'énergie atomique et aux énergies alternatives
Centre de Saclay | 91191 Gif-sur-Yvette Cedex

DRT/LIST
DILS
LISE