

Éditeur d'argumentation de sûreté

**Guide de développement**

<b>Objectif de ce document</b>	<b>2</b>
<b>Modification du profil</b>	<b>3</b>
<b>Elementtype sémantique</b>	<b>6</b>
<b>Elementtype graphique</b>	<b>8</b>
<b>Palette</b>	<b>9</b>
<b>New Child menu</b>	<b>11</b>
<b>CSS et customization</b>	<b>12</b>
<b>Génération des propriété</b>	<b>12</b>

## Objectif de ce document

Ce document décrit l'ensemble des procédures à suivre pour modifier ou ajouter un élément au profil et l'intégrer correctement au plug-in.

Du fait de l'existence de dépendances entre les différentes parties du plug-in, il est préférable de réaliser les différentes étapes dans l'ordre.

## Modification du profil

Certaines modifications ou ajouts peuvent nécessiter une modification du profil servant de base au plug-in.

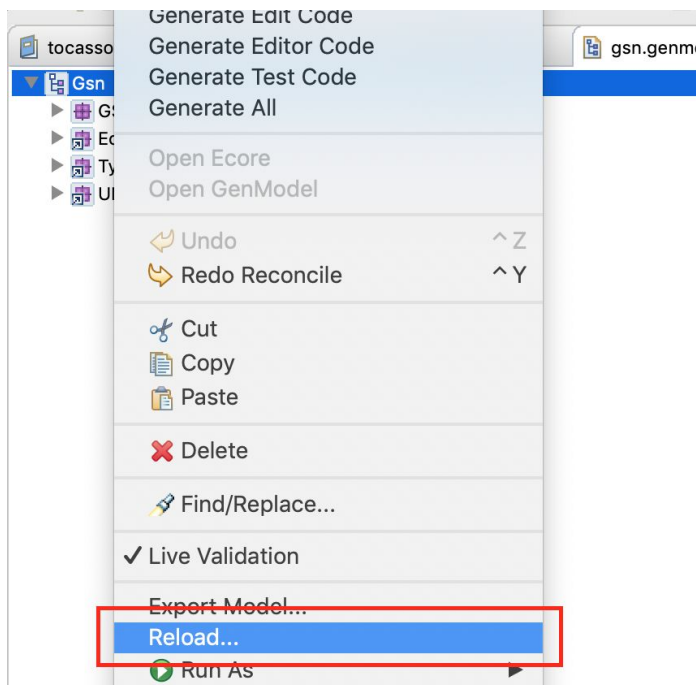
Le profil en lui-même se situe dans le fichier *model/gsn.profile* du plug-in *org.eclipse.papyrus.gsn.profile* et peut être édité graphiquement dans la vue *Papyrus Editor Core*.

## Génération automatique du code du modèle

Après avoir modifié le profil, il est indispensable de re-générer le code source du modèle.

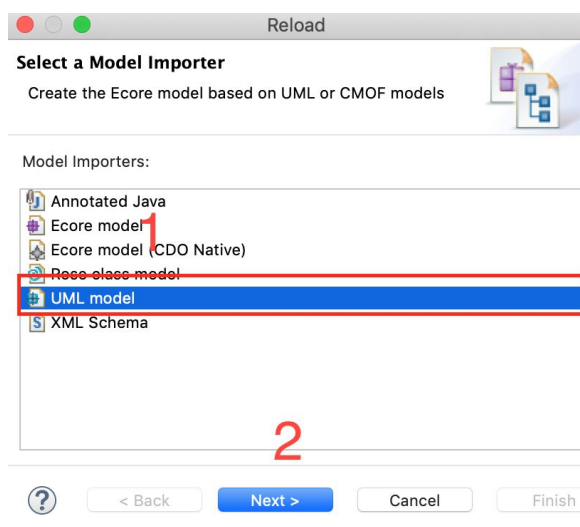
La procédure de génération du code est décrite ci-dessous:

1. Dans le fichier, *model/gsn.genmodel*, faire un clique-droit sur la racine du *genmodel* et sélectionner l'option *Reload*.



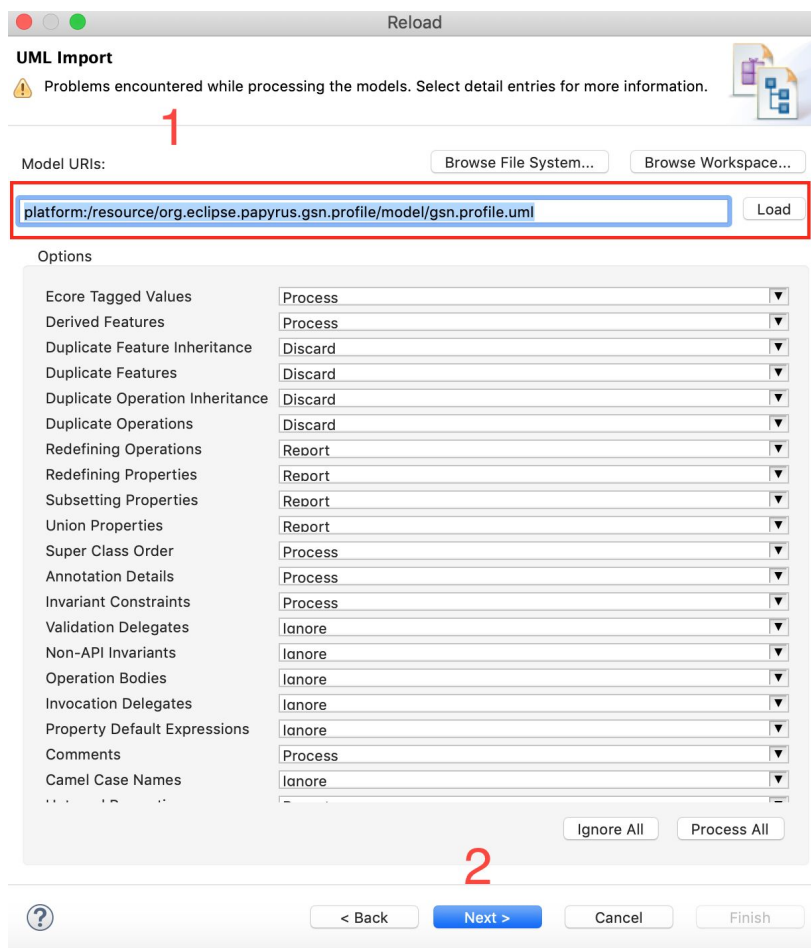
**Figure 1 : Reload du Genmodel**

2. Dans la fenêtre de dialogue qui apparaît, sélectionner l'option *UML Model*.



**Figure 2 : Choix d'UML Model**

3. Dans la boîte de dialogue suivante, sélectionner le fichier UML correspondant au profil, et cliquer sur next sans modifier les options par défaut.



**Figure 3 : Importer le Modèle UML**

4. Dans la boîte de dialogue suivante, cliquer sur *Finish* sans modifier les options par défaut.

## Elementtype sémantique

### Ajout d'un nouvel *ElementType*

Une fois le profil modifié, il faut mettre en place sa partie sémantique dans le fichier *ressources/extLibrary.elementtypesconfigurations* du plugin *org.eclipse.papyrus.gsn.types*.

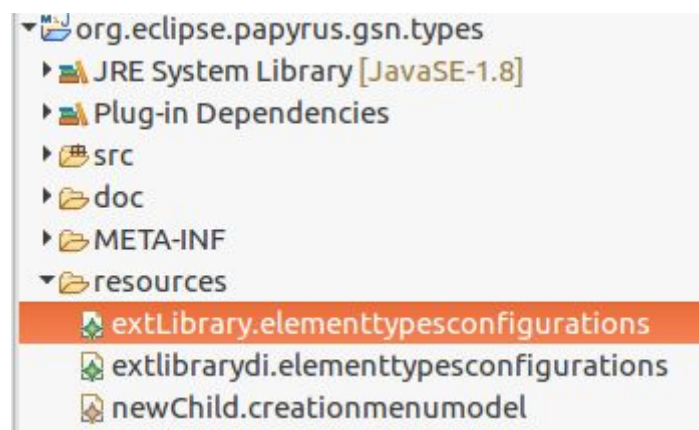


Figure 4 : Le plugin “types”

Afin de faire le binding entre le stéréotype et l'*ElementType*, il faut créer une *Specialization Type Configuration* en indiquant les informations.

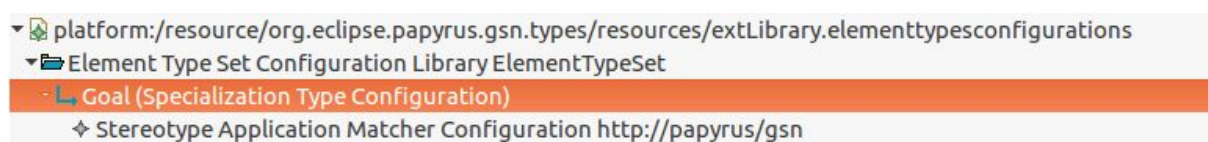


Figure 5 : Création d'un ElementType

Description	UML::Class
Hint	UML::Class
Identifier	org.eclipse.papyrus.gsn.profile.gsn.Goal
Kind	org.eclipse.gmf.runtime.emf.type.core.IHintedType
Name	Goal
Specialized Types	UML::Class (Metamodel Type Configuration)

Figure 6 : Les propriétés d'un ElementType

Il faut également créer un *Stereotype Matcher*



Profile Uri	 http://papyrus/gsn
Stereotypes Qualified Names	 gsn::Core::Goal

Figure 7 : Les propriété d'un ElementType

et un *Apply Stereotype Advice Configuration* pour appliquer le stéréotype au spécialisation type



Figure 8 : Appliquer le stéréotype à l'ElementType

en indiquant le chemin du *specialization type*




Identifier	 org.eclipse.papyrus.gsn.profile.gsn.goal
Inheritance	 none
Target	 Goal (Specialization Type Configuration)

Figure 9 : Appliquer la spécialisation type configuration




Property	Value
Required Profiles	 gsn
Stereotype Qualified Name	 gsn::Core::Goal
Update Name	 true

Figure 10 : Le stéréotype à appliqué

## Elementtype graphique

Pour chaque type de stéréotype étendant une *méta-classe* spécifique, il faut définir un ensemble de *Specialization Type*. Si cette étape n'est pas réalisée, l'élément n'apparaîtra pas dans le diagramme. Pour chaque Elementtype, nous devons avoir l'elementtype sémantique et la spécialization type configuration qui correspond au type (voir exemple)

Description	
Hint	Class_MetaclassShape
Identifiant	org.eclipse.papyrus.gsn.profile.gsn.goal_Class_MetaclassShape
Kind	org.eclipse.gmf.runtime.emf.type.core.IHintedType
Name	Goal(MetaclassShape)
Specialized Types	Class Metaclass Shape (Specialization Type Configuration), Goal (Specialization Type Configuration)

Figure 11 : Elementtype graphique (Metaclass\_Shape)

Pour le type UML::Class :

- Class
- MetaclassShape
- Metaclass\_Shape\_CN
- Class\_Shape
- Class\_Shape\_CN
- Class\_NestedClassifierLabel

platform:/resource/org.eclipse.papyrus.gsn.types/resources/extlibrarydi.elementtypesconfigurations
Element Type Set Configuration org.eclipse.papyrus.training.library.profile.extlibrarydi.elementTypes
Goal(Class) (Specialization Type Configuration)
Goal(MetaclassShape) (Specialization Type Configuration)
Goal (Class_Shape_CN) (Specialization Type Configuration)
Goal (Metaclass_Shape_CN) (Specialization Type Configuration)
Goal (Class_NestedClassifierLabel) (Specialization Type Configuration)
Goal (Class_Shape) (Specialization Type Configuration)

Figure 12 : Liste des ElementTypes à définir

Pour le type UML::Association

- Edge



↳ SupportedBy(Edge) (Specialization Type Configuration)

**Figure 13 : Elementtype graphiques pour les associations**

Pour le type : *UML::Package*

- *Class\_Shape*
- *Class\_Shape\_CN*
- *class\_diagram*
- *Use\_case*
- package sequence diagram

- ▶ ↳ Module(Shape) (Specialization Type Configuration)
- ▶ ↳ Module(Shape\_CN) (Specialization Type Configuration)
- ▶ ↳ Module(class\_diagram) (Specialization Type Configuration)
- ▶ ↳ Module(sequence\_diagram) (Specialization Type Configuration)
- ▶ ↳ Module(Use\_case) (Specialization Type Configuration)

**Figure 14 : Elementtype graphiques pour les packages**

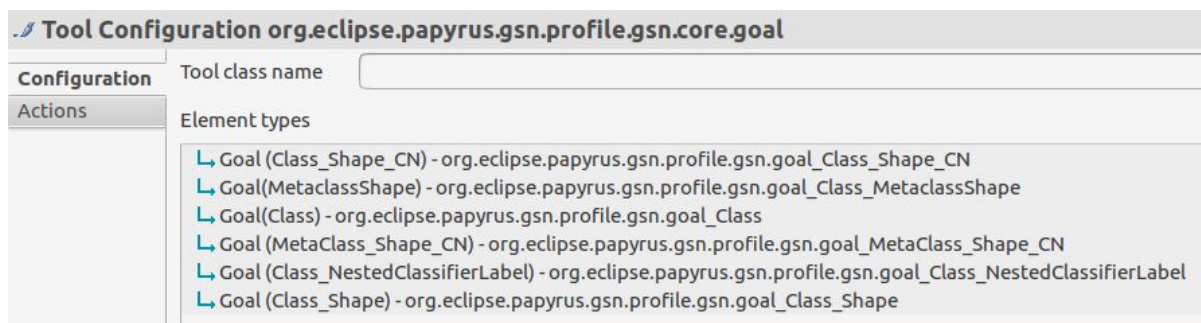
Dans le cas des associations, il faut définir des *Stereotype Advice Configuration* afin que les stéréotypes soient bien appliqués au niveau du diagramme.

⚙️ Apply Stereotype Advice Configuration  
 Stereotype To Apply gsn::Associations::SupportedBy

**Figure 15 : Elementtype graphiques pour les packages**

## Palette

Au niveau de la palette , il faut créer un *New Tool* et remplir les informations. Afin de pointer sur les *Elementtypes* graphiques , il faut modifier directement le fichier XML en ajoutant des références aux identifiants des *Elementtypes* graphiques.



**Figure 16 : Références aux elementTypes graphiques (1)**

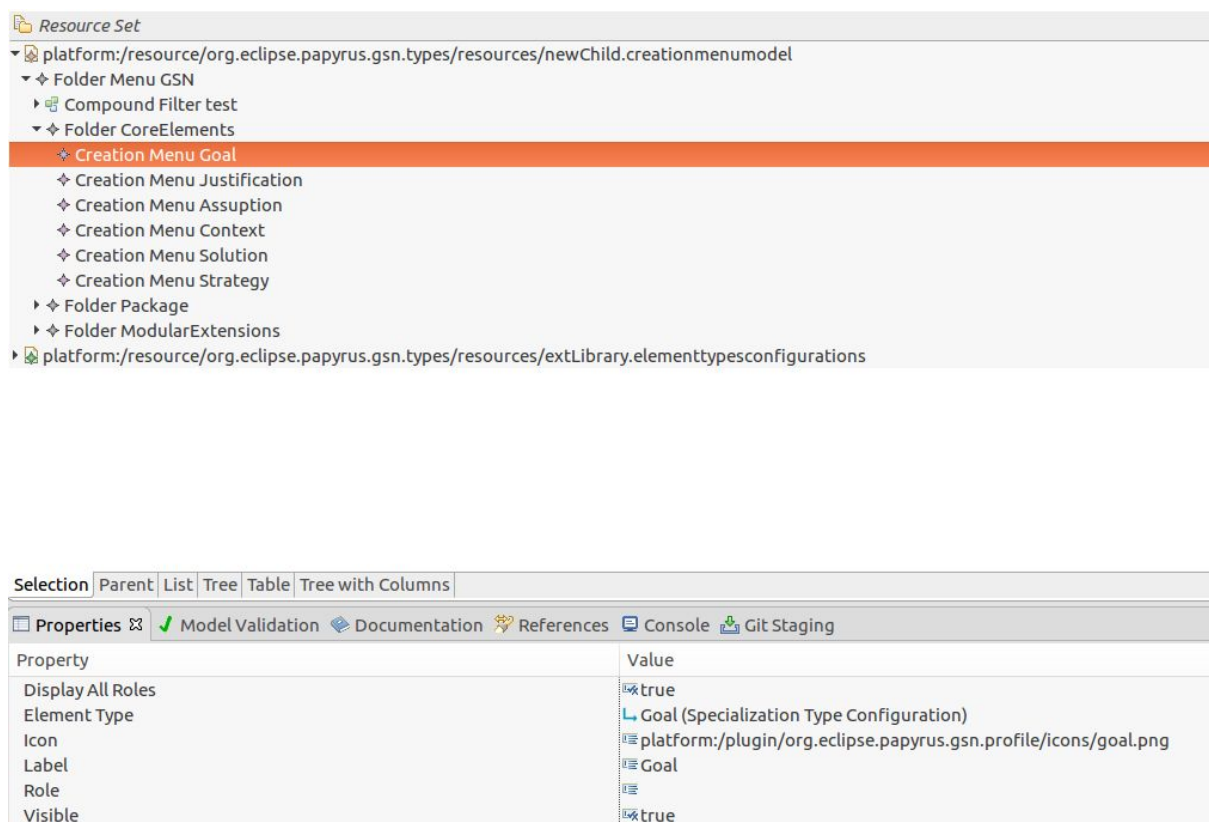
```
href="platform:/plugin/org.eclipse.papyrus.gsn.types/resources/extlibrarydi.elementtypesconfigurations#org.eclipse.papyrus.gsn.profile.gsn.goal_Class_Shape_CN"/>
href="platform:/plugin/org.eclipse.papyrus.gsn.types/resources/extlibrarydi.elementtypesconfigurations#org.eclipse.papyrus.gsn.profile.gsn.goal_Class_MetaclassShape"/>
href="platform:/plugin/org.eclipse.papyrus.gsn.types/resources/extlibrarydi.elementtypesconfigurations#org.eclipse.papyrus.gsn.profile.gsn.goal_Class"/>
href="platform:/plugin/org.eclipse.papyrus.gsn.types/resources/extlibrarydi.elementtypesconfigurations#org.eclipse.papyrus.gsn.profile.gsn.goal_MetaClass_Shape_CN"/>
href="platform:/plugin/org.eclipse.papyrus.gsn.types/resources/extlibrarydi.elementtypesconfigurations#org.eclipse.papyrus.gsn.profile.gsn.goal_Class_NestedClassifierLabel"/>
href="platform:/plugin/org.eclipse.papyrus.gsn.types/resources/extlibrarydi.elementtypesconfigurations#org.eclipse.papyrus.gsn.profile.gsn.goal_Class_Shape"/>
```

**Figure 17 : Références aux elementTypes graphiques (2)**

## New Child menu

Pour le menu *New Child* , il suffit de créer un *Creation Menu* référençant l'Elementtype sémantique pour que le nouvel élément apparaisse dans le menu contextuel.

⚠: Si l'icône du nouvel élément n'est pas renseignée, ce dernier n'apparaîtra pas dans le menu.



**Figure 18 : Création d'un nouvel élément**

## CSS et customization

Il faut créer une image SVG et l'appliquer le stéréotype grâce au CSS .

```
[appliedStereotypes~Goal] {  
    svgFile: 'platform:/plugin/org.eclipse.papyrus.gsn.css/icons/goal.svg';  
}
```

Figure 19 : Appliquer une image SVG au stéréotype “Goal”

## Génération des propriétés

Afin d'ajouter un nouveau stéréotype , il faut régénérer les propriétés à partir du modèle et modifier par la suite le fichier .ctx généré . Il faut bien vérifier le widget type qui doit correspondre au type déjà défini dans

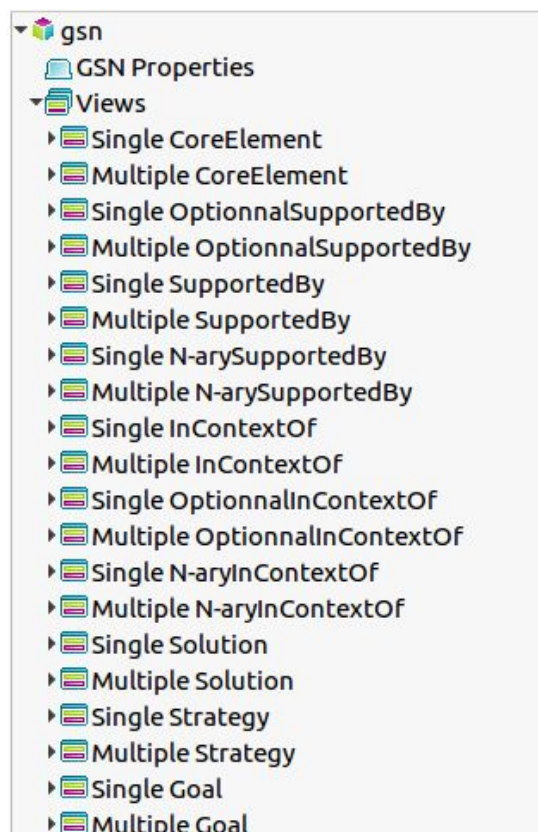


Figure 20 : Génération des propriétés pour les éléments du langage

🔧 Undeveloped

Context

Read only

☐ true ☒ false

Show label

☒ true ☐ false

Property

▢ Undeveloped

Widget type

🔧 Property Editor Type BooleanRadio

Custom label

**Figure 21 : Modification du widget type**