

# Introduction au méta-modèle UML 2.5

## Spécialisation/généralisation

Module *Ingénierie dirigée par les modèles*

**Documents fournis :** Les diagrammes décrivant la généralisation/spécialisation (**Classifiers**) et les compléments de notation (**Generalization Set**).

## 1 Classifier et Generalization : la vision d'UML 2.5

Un **Classifier** décrit un ensemble d'instances qui ont des caractéristiques communes. C'est donc un élément d'une classification. Les **Classifiers** sont des types et sont organisés par une relation de généralisation.

Dans les termes d'UML, la *conformance de types* entre **classifiers** généralisables est définie de telle sorte qu'un **classifier** se conforme à lui-même et à ses généralisations.

La *substituabilité*, c'est-à-dire le fait que le **classifier** plus spécifique puisse être utilisé partout où le classifier plus général peut l'être, n'est pas automatique en cas de spécialisation : l'attribut **isSubstitutable** permet de préciser si elle s'applique.

Une **Generalization** est une relation taxinomique. Elle apparaît dans le métamodèle comme une relation orientée (**DirectedRelationship**) entre une unité de classification plus générale *G* (rôle **general**) et une unité de classification plus spécifique *S* (rôle **specific**). Toute instance de *S* est une instance de *G* et par conséquent *S* hérite des caractéristiques de *G*. *S* peut ajouter ou redéfinir des **features**.

**Question 1.1** *Proposez un diagramme de classes représentant les triangles et une spécialisation (triangles rectangles) et montrez-le comme instantiation du méta-modèle.*

## 2 GeneralizationSet et powerType

L'une des nouveautés d'UML 2.xx consiste à regrouper des relations de généralisation issues d'un même classifier et ayant une sémantique commune. Un tel regroupement s'appelle un **GeneralizationSet**. Par exemple, pour la figure 1 un **GeneralizationSet** regrouperait les relations de spécialisation entre **Arbre** et ses trois sous-classes **Abricotier**, **Erable** et **Orme**. Un **GeneralizationSet** se décrit par deux attributs permettant de représenter les contraintes **disjoint**, **overlapping**, **complete** et **incomplete**.

On peut également associer à un **GeneralizationSet** un classifier qui joue le rôle de **powerType**. Dans l'exemple, **Espèce D'arbre** est un **powerType** pour les trois sous-classes **Abricotier**, **Erable** et **Orme**. Les sous-classes du classifier spécialisé dans les **GeneralizationSet** (**Arbre**) sont alors considérées comme des instances de ce **powerType** (**Espèce D'arbre**), qui est une sorte de méta-classe pour ces sous-classes (mais il n'y a pas de mécanismes spécifiques associé à cette notation en UML).

**Question 2.1** *Présentez le diagramme de la figure 1 comme une instance du méta-modèle.*

