

## 4 Transformation de modèles : Patron State

### 4.1 Travail à faire

L'objectif de ce TP est d'implémenter en Java/EMF une transformation de modèle qui, à partir d'un modèle UML  $M$  produit un modèle UML  $M'$  correspondant à  $M$  mais dans lequel on aura appliqué à toutes les classes contexte d'une machine à états le patron de conception State, ainsi que vu en TD. On fera les mêmes suppositions qu'en TD :

- On considère qu'une classe ne peut pas avoir plus d'une machine à états associée.
- On ne s'intéressera pas à l'héritage de machine à états.
- On ne manipule que des machines à états sans états composites ni orthogonaux.
- On ne traitera que la partie statique du patron de conception State. Pour la partie dynamique (changement d'état, délégation vers l'état, ...), on pourra éventuellement attacher des commentaires aux méthodes concernées.

Cette transformation est conceptuellement simple, mais nécessite de réaliser beaucoup de transformations intermédiaires (création de nouvelles classes, de hiérarchies, d'une association, de méthodes, ...). Il est donc fortement déconseillé de tenter d'écrire l'intégralité de cette transformation dans une seule méthode java. Au contraire, il est conseillé de bien décomposer le problème en méthodes atomiques et éventuellement réutilisables dans un autre contexte de transformation impliquant le méta-modèle UML.

### Astuce

Suivez l'enchaînement décrit dans le TD sur les transformations de modèles pour découper le problème en opérations "atomiques".