

Ingénierie des Modèles III

Frédéric Boulanger

frederic.boulanger@centralesupelec.fr



<https://wdi.centralesupelec.fr/mde>



CentraleSupélec

3^e année mention Science du logiciel

Techniques de l'ingénierie dirigée par les modèles

Langages spécifiques à un domaine (DSL)

- définis par opposition aux GPL (General Purpose Languages)
- nombre de concepts restreint
- moins expressifs
- ciblés sur un domaine technique particulier

Techniques de l'ingénierie dirigée par les modèles

Langages spécifiques à un domaine (DSL)

- définis par opposition aux GPL (General Purpose Languages)
- nombre de concepts restreint
- moins expressifs
- ciblés sur un domaine technique particulier

Exemple

- conception d'un automate
 - en C ou en Java : nombreux concepts parasites
structure de l'automate peu visible dans le code
 - avec un DSL : notion d'état, de transition avec garde et actions
 - risque d'erreur réduit

Techniques de l'IDM

Compromis Expressivité / Décidabilité

- avec un DSL, on renonce à de l'expressivité
- ... mais on gagne en décidabilité

Techniques de l'IDM

Compromis Expressivité / Décidabilité

- avec un DSL, on renonce à de l'expressivité
- ... mais on gagne en décidabilité

Exemple

- vérification d'un automate
 - en C ou en Java : langages Turing-complets
pas de procédure de décision
 - avec un DSL : sémantique plus simple, logique moins puissante
 - procédure de décision pour l'atteignabilité d'un état

Techniques de l'IDM

Métamodélisation

- un métamodèle définit la syntaxe d'un DSL
- on le définit à l'aide d'un langage de métamodélisation
 - MOF (Meta Object Facility)
 - eCore (dans Eclipse EMF)

Techniques de l'IDM

Métamodélisation

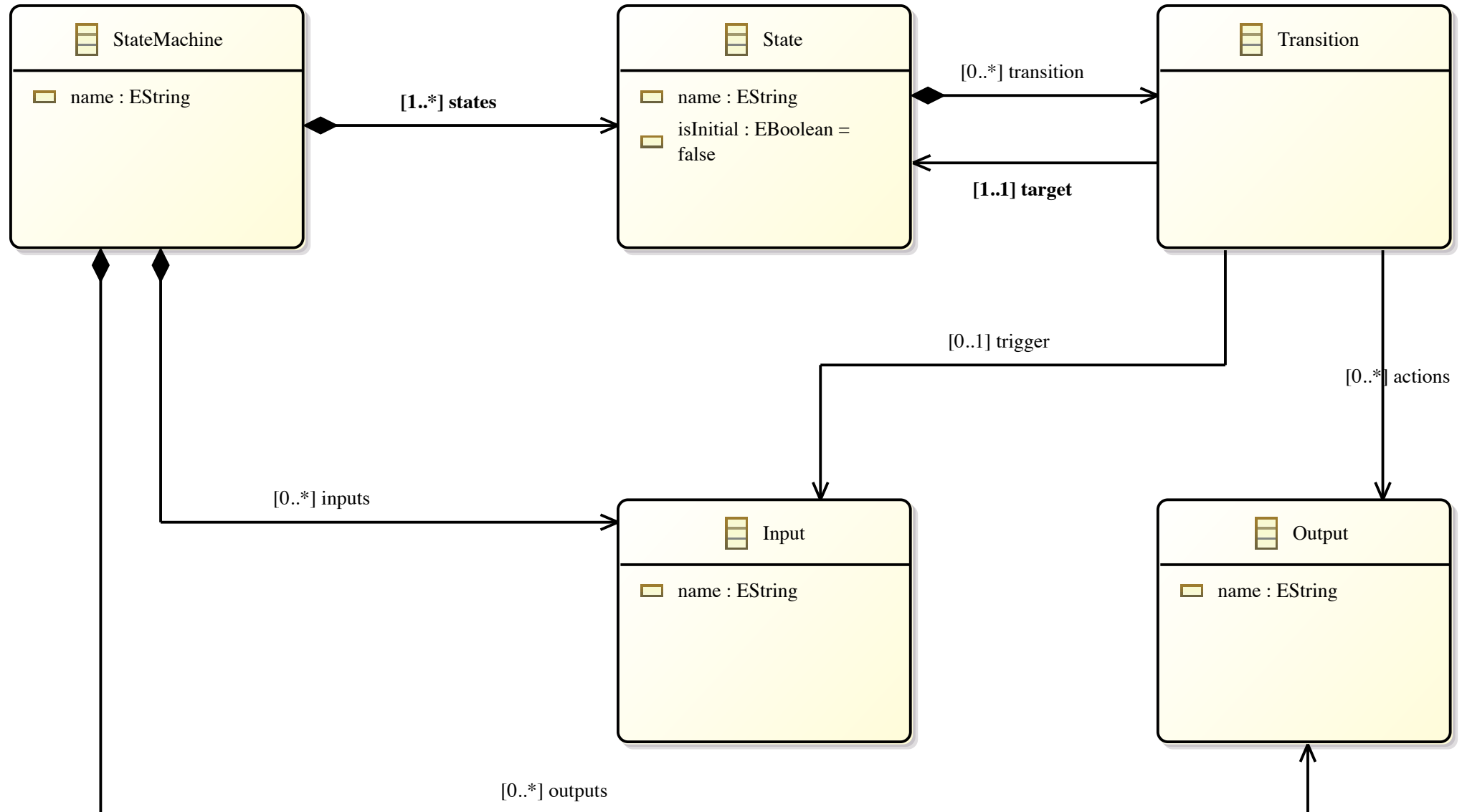
- un métamodèle définit la syntaxe d'un DSL
- on le définit à l'aide d'un langage de métamodélisation
 - MOF (Meta Object Facility)
 - eCore (dans Eclipse EMF)

Avec EMF et eCore

- les concepts sont représentés par des classes
- les classes ont des attributs
- les relations entre concepts sont représentées par du sous-typage et des références
- en plus du modèle eCore, un fichier genmodel décrit comment générer du code pour un modèle

Métamodélisation

Exemple de métamodèle



Techniques de l'IDM

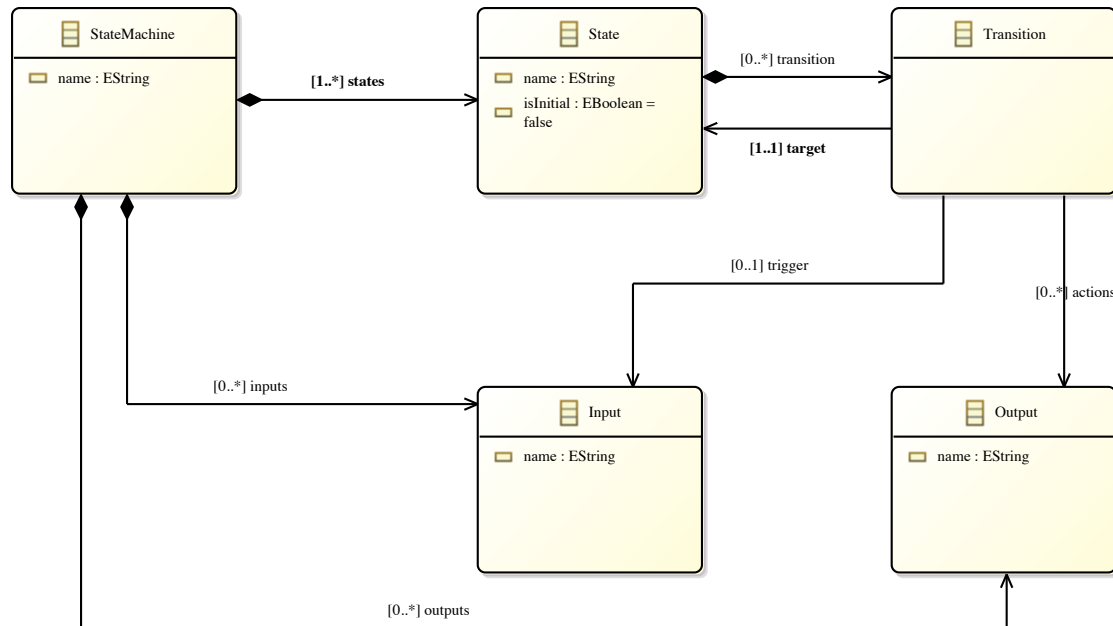
Transformations de modèles M2M

- définies entre deux métamodèles
- transforment un modèle conforme à MM1 en un modèle MM2
- réalisation :
 - codées en Java, Xtend, Python,...
 - décrites dans un langage de description de transformation

Techniques de l'IDM

Transformations de modèles M2M

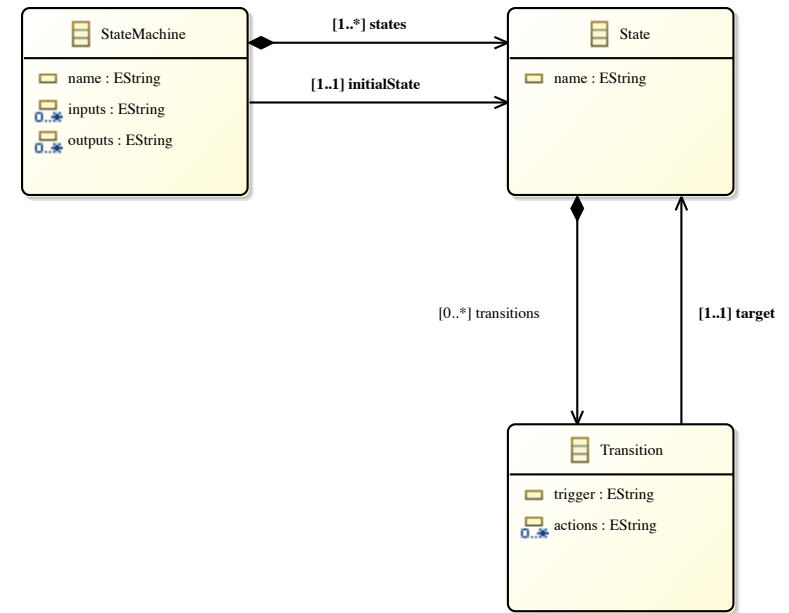
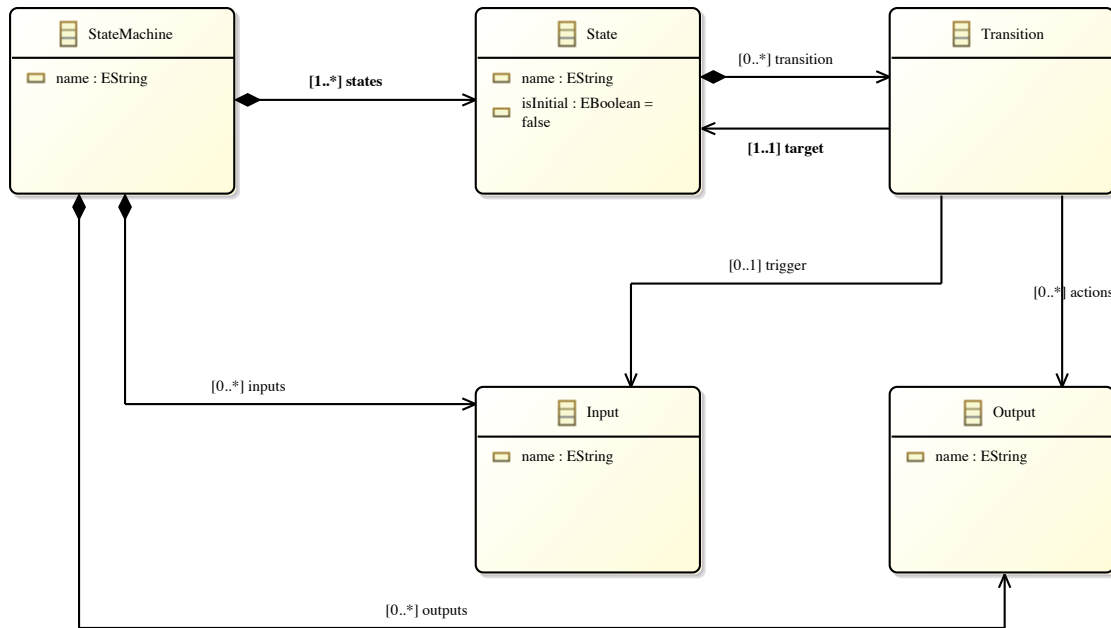
- définies entre deux métamodèles
- transforment un modèle conforme à MM1 en un modèle MM2
- réalisation :
 - codées en Java, Xtend, Python,...
 - décrites dans un langage de description de transformation



Techniques de l'IDM

Transformations de modèles M2M

- définies entre deux métamodèles
- transforment un modèle conforme à MM1 en un modèle MM2
- réalisation :
 - codées en Java, Xtend, Python,...
 - décrites dans un langage de description de transformation



Techniques de l'IDM

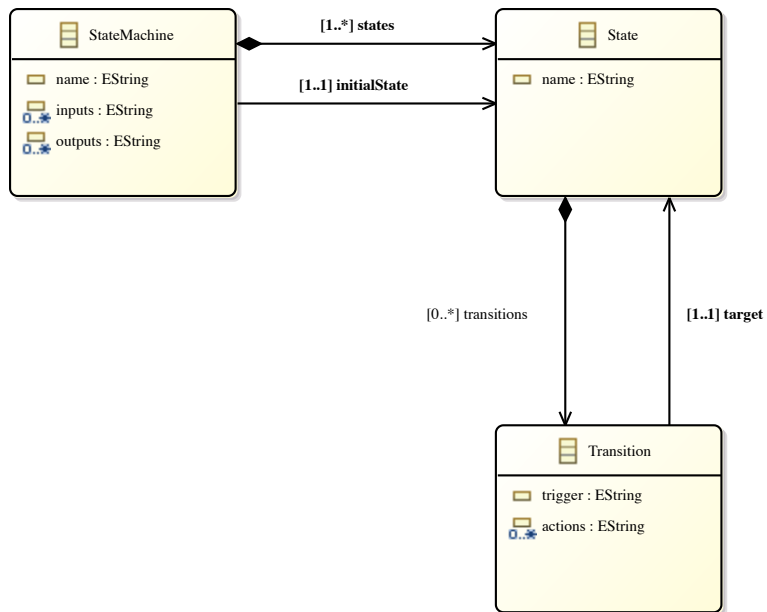
Transformations de modèles M2T

- définies d'un métamodèle vers du texte
- transforment un modèle conforme à un métamodèle en texte
- génération de code (compilation), export vers des outils de vérification

Techniques de l'IDM

Transformations de modèles M2T

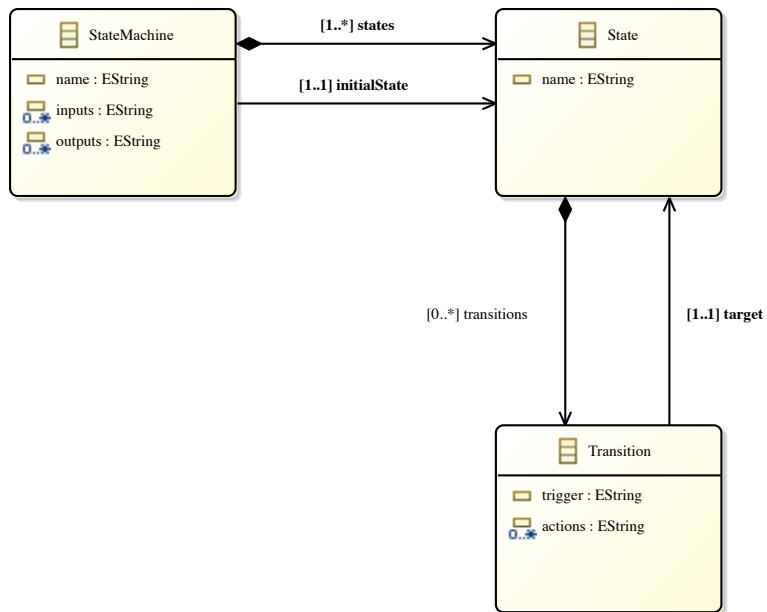
- définies d'un métamodèle vers du texte
- transforment un modèle conforme à un métamodèle en texte
- génération de code (compilation), export vers des outils de vérification



Techniques de l'IDM

Transformations de modèles M2T

- définies d'un métamodèle vers du texte
- transforment un modèle conforme à un métamodèle en texte
- génération de code (compilation), export vers des outils de vérification



```
function react(msg) {
    var out_u = document.getElementById("out_u");
    var out_v = document.getElementById("out_v");
    out_u.checked = false;
    out_v.checked = false;
    var state_A = document.getElementById("state_A");
    var state_B = document.getElementById("state_B");
    if (state_A.checked) {
        if (msg == "a") {
            // Actions
            out_u.checked = true;
            // Target state
            state_B.checked = true;
            return;
        }
    }
    if (state_B.checked) {
        if (msg == "b") {
            // Actions
            out_v.checked = true;
            // Target state
            state_A.checked = true;
            return;
        }
    }
}
```

Mise en œuvre

Cours sur machine et BE

- aperçu des techniques et possibilités du MDE
- environnement Eclipse Modeling Framework
 - création de métamodèles
 - transformations M2M
 - transformations M2T (génération de code)
- possibilité d'installer Eclipse sur votre ordinateur
- possibilité d'utiliser le conteneur Docker `docker-webtop-3asl`
- lancement avec les scripts :
 - `start-3asl.sh` (Linux et MacOS), `start-3asl.ps1` (Windows)
 - <https://github.com/Frederic-Boulanger-UPS/docker-webtop-3asl>

Mise en œuvre

Cours sur machine et BE

- aperçu des techniques et possibilités du MDE
- environnement Eclipse Modeling Framework
 - création de métamodèles
 - transformations M2M
 - transformations M2T (génération de code)
- possibilité d'installer Eclipse sur votre ordinateur
- possibilité d'utiliser le conteneur Docker `docker-webtop-3asl`
- lancement avec les scripts :
 - `start-3asl.sh` (Linux et MacOS), `start-3asl.ps1` (Windows)
 - <https://github.com/Frederic-Boulanger-UPS/docker-webtop-3asl>

Application à un cas d'étude industriel

- Intervention d'un ancien élève de la mention
- Application à la génération de code à partir de modèles Open API

Exemple de définition et d'utilisation d'un DSL

Définition d'un DSL pour représenter des QCM

- permet de s'affranchir d'une implémentation particulière
- permet de viser différentes plateformes cibles :
 - papier
 - formulaire web
 - outils spécifiques
 - génération d'un formulaire HTML pour chaque QCM
 - définition d'une syntaxe textuelle concrète pour ce langage.

Sera plus ou moins développé selon le temps disponible.