

Ingénierie dirigée par les modèles III

Frédéric Boulanger

17 novembre 2022

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

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 : beaucoup de concepts parasites
structure de l'automate peu visible dans le code

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 : beaucoup de concepts parasites
structure de l'automate peu visible dans le code
 - ▶ avec un DSL : notion d'état, de transition avec garde et actions

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 : beaucoup de 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'ingénierie dirigée par les modèles

Compromis Expressivité/Décidabilité

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

Techniques de l'ingénierie dirigée par les modèles

Compromis Expressivité/Décidabilité

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

Exemple

- ▶ vérification d'un automate

Techniques de l'ingénierie dirigée par les modèles

Compromis Expressivité/Décidabilité

- ▶ dans 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

Techniques de l'ingénierie dirigée par les modèles

Compromis Expressivité/Décidabilité

- ▶ dans 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

Techniques de l'ingénierie dirigée par les modèles

Compromis Expressivité/Décidabilité

- ▶ dans 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'ingénierie dirigée par les modèles

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'ingénierie dirigée par les modèles

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, on a un fichier genmodel pour décrire comment générer du code pour le modèle.

Techniques de l'ingénierie dirigée par les modèles

Transformations de modèles M2M

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

Techniques de l'ingénierie dirigée par les modèles

Transformations de modèles M2M

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

Transformations de modèles M2T

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

Mise en œuvre

Travaux pratiques cet après-midi

- ▶ aperçu des techniques et possibilités du MDE
- ▶ environnement Eclipse Modeling Framework
 - ▶ création de métamodèles
 - ▶ transformation M2M
 - ▶ transformation M2T (génération de code)
- ▶ possibilité de faire l'installation sur votre machine
- ▶ possibilité d'utiliser un conteneur docker
- ▶ lancement facile avec les scripts :
 - ▶ `start-cs3asl.sh` (Linux et MacOS)
 - ▶ `start-cs3asl.ps1` (PowerShell Windows)

disponibles sur :

github.com/Frederic-Boulanger-UPS/docker-cs3asl

Exemple de définition et d'utilisation d'un DSL

Exemple

- ▶ Définition d'un DSL pour représenter des QCM
 - ▶ permet de s'affranchir d'une implémentation particulière
 - ▶ permet de viser différentes plateformes cible :
 - ▶ 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.