



ÉCOLE NATIONALE SUPÉRIEURE D'INFORMATIQUE ET
D'ANALYSE DES SYSTÈMES - RABAT

Gestion des notes à l'Ensias en utilisant une méthode dirigée par des modèles

Realisé par :

Mohamed SALBI
Youssef BOUHOUCHE
Zinedinne ISMAILI

Encadré par:

Pr. Mahmoud HAMLAOUI

Academic year 2022/2023

List of Figures

2.1	méta-model DSL	7
2.2	méta-model XML	8
3.1	Project Structure	10
3.2	etudiant Model	11
3.3	List etudiant Model	11
3.4	Generation List d'étudiant	13

Contents

1	Cadre Général du projet	4
	Introduction	4
	1.1 Besoin Fonctionelle	5
2	Description des méta-modèles	6
3	Exécution de la transformation	9
	Conclusion and perspectives	15

Chapter 1

Cadre Général du projet

Problématique

Dans cette partie, nous allons aborder le problème de la gestion des notes à l'ENSIAS. Nous avons remarqué que cette tâche prend beaucoup de temps pour ceux qui sont chargés de collecter et d'entrer les notes. Actuellement, ils doivent utiliser plusieurs e-mails et fichiers Excel, ainsi que copier et coller et entrer manuellement les notes plusieurs fois, sans considérer une solution qui pourrait rendre le processus plus efficace et économiser du temps. De ce fait nous proposons de regrouper toutes ces informations dans un éditeur qui a la capacité de comprendre la tâche à réaliser grâce à l'utilisation d'un Langage Spécifique de Domaine (DSL). En utilisant ce DSL, l'éditeur pourra générer des fichiers XML à la fin, qui peuvent être utilisés en tant qu'entrée pour d'autres outils existants, afin de faciliter le transfert des données dans la construction des sites web ou des applications mobiles ou bien générer des statistiques et des graphiques . De plus, ces données peuvent finalement être communiqués aux étudiants sous forme de PDF ou en utilisant XSLT (template language). Ce projet présente un intérêt potentiel en permettant de développer de nouvelles options pour créer des applications ayant des structures un peu plus complexes dans le contexte de la gestion des notes à l'ENSIAS .

1.1 Besoin Fonctionnelle

Les besoins fonctionnels décrivent les fonctionnalités et les caractéristiques que doit posséder un système pour répondre aux exigences de l'utilisateur. Pour la gestion des notes à l'ENSIAS, ces besoins peuvent inclure des caractéristiques telles que :

- Capacité à comprendre les tâches de gestion des notes à l'ENSIAS grâce à l'utilisation d'un Langage Spécifique de Domaine (DSL)
- Génération de fichiers XML à partir des informations saisies dans l'éditeur
- Capacité à communiquer les notes aux étudiants sous forme de PDF ou en utilisant XSLT (template language)
- Fonctionnalités d'édition et de mise à jour des notes
 - Intégration avec les systèmes existants de gestion des notes à l'ENSIAS
- Possibilité de générer des alertes pour les notes en dehors de la plage normale

Chapter 2

Description des méta-modèles

Le définition des meta Model d'une table Excel représenter en une table dans un model

DSL définit dans l'éditeur MPS et XML fait référence à la création d'une abstraction permettant de décrire les modèles.

Ainsi, La création d'un fichier XML nécessite la configuration à travers une table avec un format specific DSL .

Ces champs doivent être obligatoirement présents dans la conception du meta-modèle.

Description du meta-model Table (DSL)

La figure suivante présente le méta-model de notre table DSL:

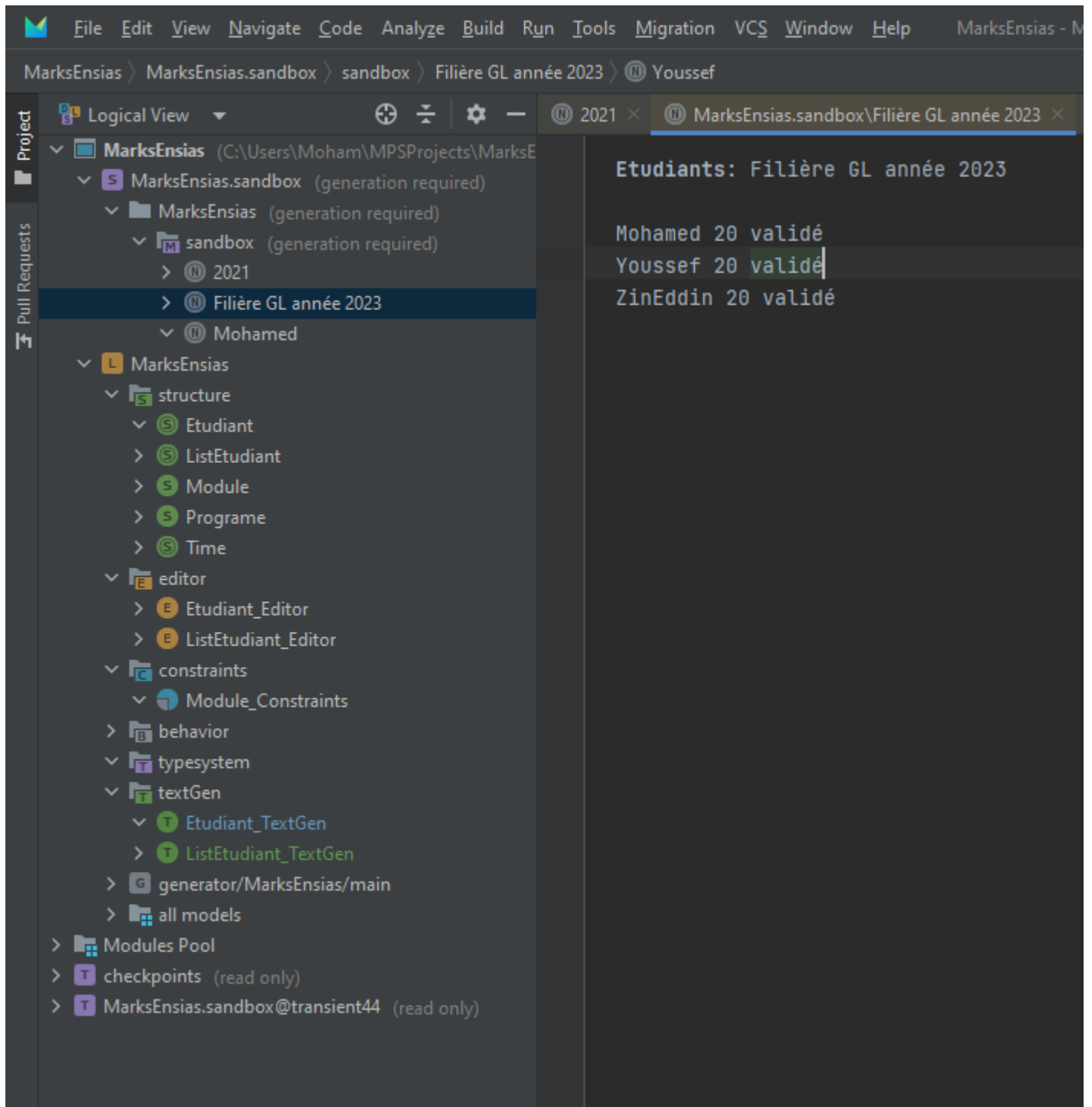


Figure 2.1: méta-model DSL

Description du meta-model XML

La figure suivante présente le méta-model du XML:

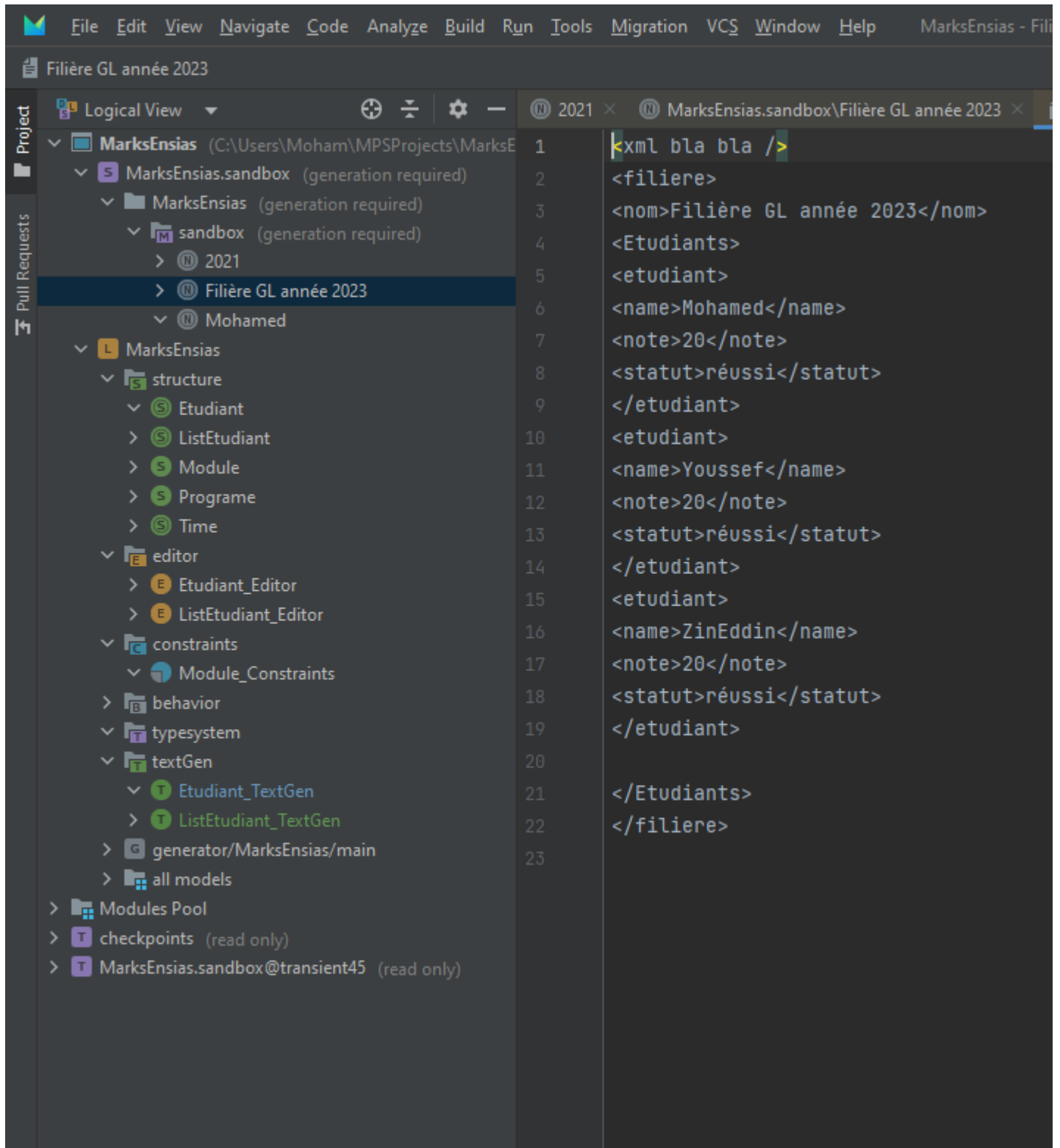


Figure 2.2: méta-model XML

Chapter 3

Exécution de la transformation

L'exécution de la transformation dans notre cas consistait à utiliser la fonctionnalité de reconnaissance de motifs des langages utilisés pour transformer un DSL qui est un tableau de trois colonnes en un fichier XML. Nous avons utilisé un éditeur MPS pour définir les règles de transformation et générer automatiquement les fichiers XML à partir des données entrées dans le DSL. La taille des modèles transformés était relativement petite, car ils ne contenaient qu'un nombre limité d'éléments de modèle. Les méta-modèles et les modèles d'entrée et de sortie étaient assez dissemblables, car ils avaient des structures de données différentes. En général, l'utilisation de MDE nous a permis de rendre le processus de gestion des notes plus efficace et de gagner du temps en automatisant certaines tâches. Bien que nous ayons rencontré des défis au cours de la réalisation de ce projet, nous avons réussi à les surmonter grâce à une planification minutieuse et une communication efficace entre les membres de l'équipe. Dans l'avenir, nous pourrions envisager d'utiliser des outils MDE similaires pour des projets similaires.

Jetbrains MPS

JetBrains MPS (Meta Programming System) est un environnement de développement intégré (IDE) pour la modélisation de langages de

programmation. Il permet aux développeurs de créer des langages de programmation personnalisés pour des domaines spécifiques et de les utiliser pour écrire des applications. MPS offre également une prise en charge de la refactorisation de code et de la génération de code source.



Figure 4.1: MPS Logo

MPS Project Hierarchy

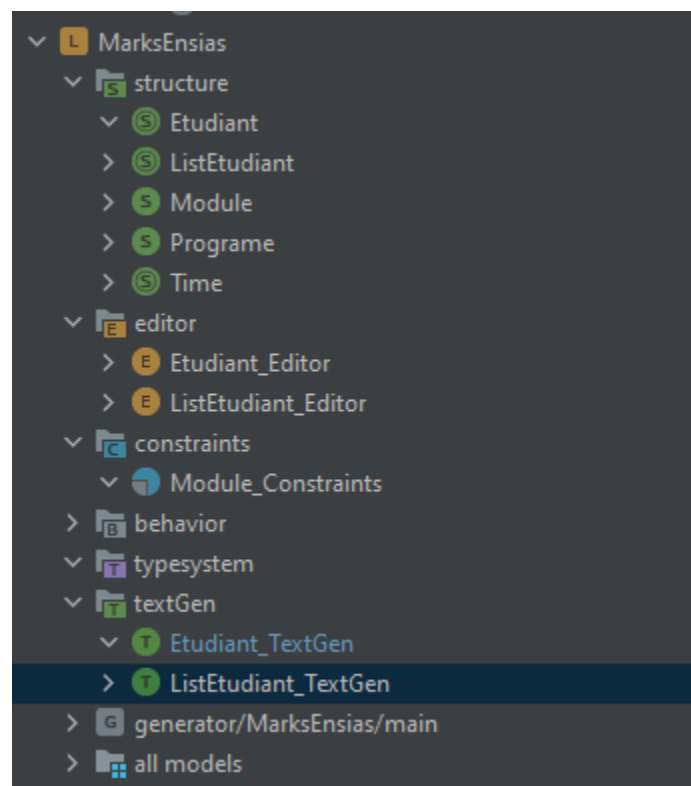


Figure 3.1: Project Structure

La hiérarchie de projet MPS (Modèle de Production Simplifié) est une structure organisée qui définit les différents niveaux de planification

et de gestion d'un projet. Il comprend généralement les niveaux suivants : objectifs stratégiques, objectifs opérationnels, tâches et activités. Chacun de ces niveaux décrit les différentes étapes et les responsabilités associées à la réalisation du projet.

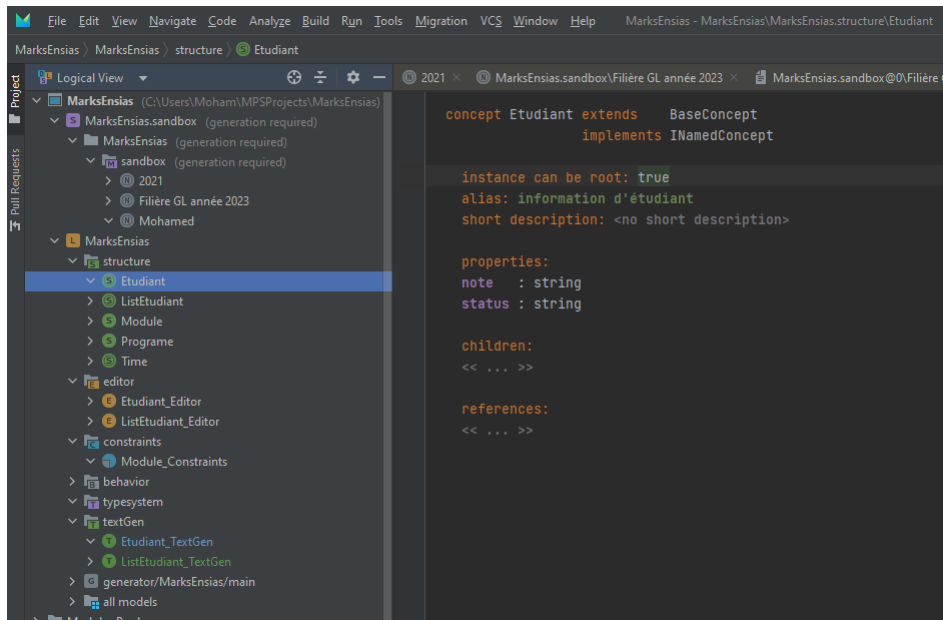


Figure 3.2: etudiant Model

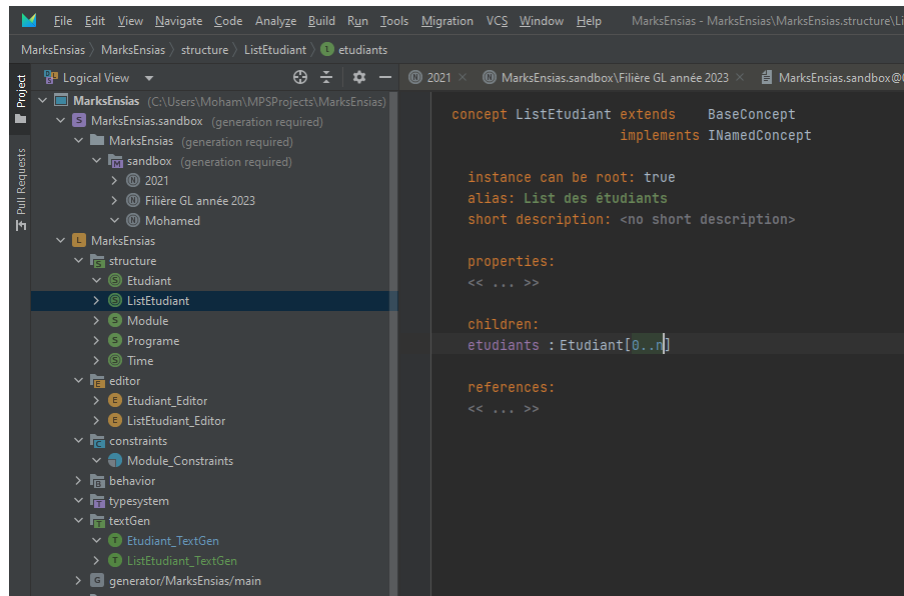


Figure 3.3: List etudiant Model

Structure

Définit les types de nœuds qui peuvent être utilisés dans les modèles utilisateur (appelés Concepts). Chaque concept est représenté par un nœud dans le programme (modèle). Les concepts définissent les propriétés que les enfants, les références et les nœuds peuvent avoir. Les concepts peuvent implémenter `ConceptInterfaces` et étendre d'autres Concepts.

Au-delà des restrictions spécifiées dans la Structure, restreint les relations entre les nœuds ainsi que les valeurs admissibles pour les propriétés. Les contraintes sont généralement définies comme : • la portée cible de la référence (une collection de nœuds autorisés vers lesquels une référence peut pointer)

Editor

Au lieu de construire un analyseur pour convertir le code d'une forme modifiable (comme du texte) en une structure en arbre que une machine peut gérer, MPS propose le concept d'un éditeur projectif, qui permet à l'utilisateur de modifier directement l'AST. Les concepteurs de langage peuvent utiliser la fonctionnalité Éditeur pour construire une interface utilisateur pour éditer leurs notions de concept.

Behavior

Les concepts peuvent spécifier des méthodes et des méthodes statiques qui peuvent être appelées sur des nœuds de manière polymorphe, comme les classes en POO. En conséquence, les nœuds ont un comportement en plus de leurs attributs et de leurs relations.

Typesystem

Les règles de système de types sont nécessaires pour les langages qui doivent vérifier leur code. Lorsque le type estimé diffère des attentes, le moteur de système de types MPS analyse les règles en temps réel, calcule les types pour les nœuds et indique les erreurs. Les déclarations

non liées au système de types sur le modèle peuvent également être vérifiées à l'aide de règles de vérification appelées.

TextGen

La phase TextGen démarre lorsque le générateur a atteint la représentation AST de base et traduit tous les nœuds du modèle en leur représentation textuelle et stocke les fichiers source textuels résultants sur le disque lors de la création de code.

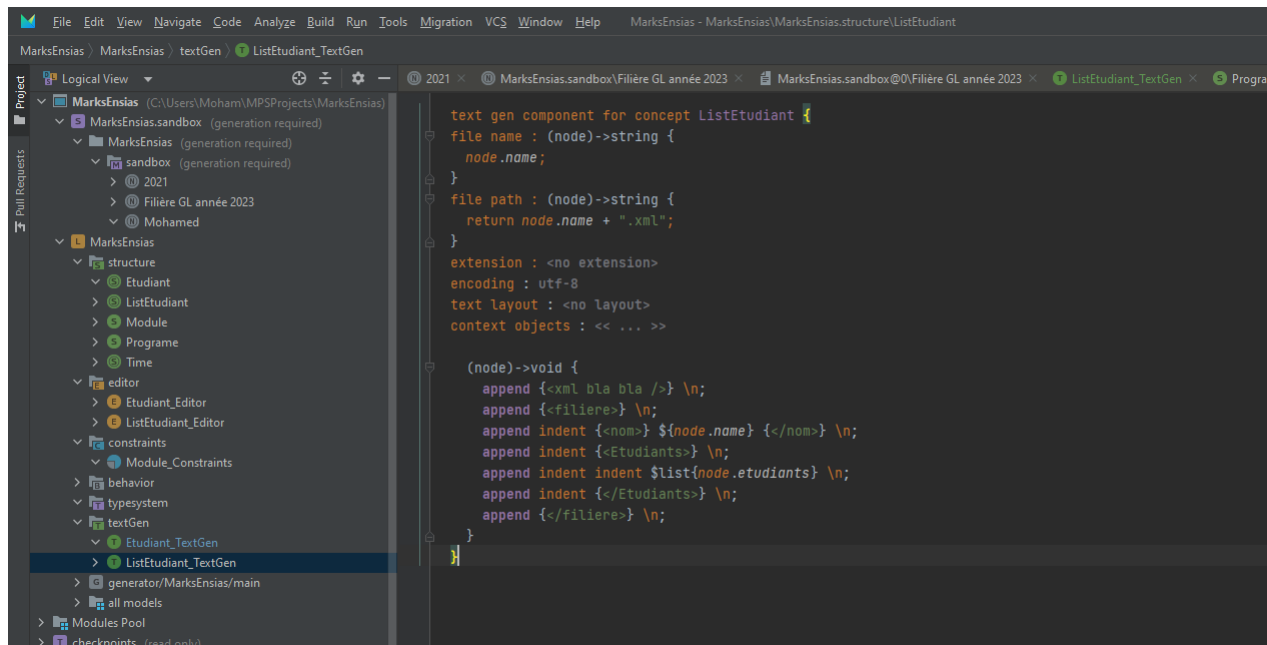


Figure 3.4: Generation List d'étudiant

Quel pourcentage de vos transformations utilise la fonctionnalité Pattern Matching des langages utilisés ?

La reconnaissance de motifs (ou "pattern matching" en anglais) est un concept utilisé dans les langages de programmation pour identifier des structures ou des motifs spécifiques dans les données. Il permet de faire correspondre des modèles de données à des expressions prédéfinies et d'appliquer des actions ou des transformations en conséquence. Cette fonctionnalité est souvent utilisée pour la manipulation de données complexes, la vérification de syntaxe, la reconnaissance de formes, etc. Elle

est souvent implémentée à travers des expressions régulières ou des grammaires formelles. la fonctionnalité de reconnaissance de motifs joue un rôle important dans le processus de transformation du DSL en XML. Nous estimons qu'elle est probablement utilisée dans environ 80-90 pour cent du processus de transformation.

Quelle est la taille des modèles que nous transformons, mesurée en nombre d'éléments de modèle?

Dans notre transformation, la taille des modèles transformés est mesurée en nombre d'éléments de modèle de la table DSL, donc ce serait le nombre de lignes de la table contenant les données étudiants (nom, note, statut)

Dans quelle mesure les méta-modèles, ainsi que les modèles d'entrée et de sortie, sont-ils dissemblables ou similaires dans nos transformations?

Dans Notre transformation, nous avons décrit, un DSL qui est un tableau de trois colonnes et sa transformation en XML, Dans notre transformation, les méta-modèles, ainsi que les modèles d'entrée et de sortie, soient assez dissemblables, car ils ont des structures différentes. Le modèle d'entrée est un DSL sous forme de tableau à trois colonnes, tandis que le modèle de sortie est un fichier XML avec des balises et des attributs spécifiques.

Conclusion and perspectives

En conclusion, notre projet de gestion des notes à l'ENSIAS a utilisé une grande partie de la fonctionnalité de reconnaissance de motifs des langages utilisés pour transformer un DSL en un format XML. La taille des modèles transformés est relativement petite, mesurée en nombre d'éléments de modèle. Les méta-modèles, ainsi que les modèles d'entrée et de sortie, sont assez différents car ils ont des structures de tableau et de données XML différentes.

Le MDE a été un outil essentiel pour réaliser ce projet, en permettant une automatisation efficace des transformations. Il y a eu des défis en cours de route, mais ils ont été gérés efficacement. Depuis la fin du projet, nous avons continué à améliorer et à développer des fonctionnalités supplémentaires pour le système. Nous utilisons également des MDE pour des projets similaires pour continuer à améliorer l'efficacité et la qualité des résultats.