



Ingénierie Dirigée par les Modèles

Projet IDM : chaiseMinute

Groupe L34-2

Élèves :

LEBOBE Timothée

LECUYER Simon

SABLAYROLLES Guillaume

THEVENET Louis

Novembre 2024 - Decembre 2024

Table des matières

1. Introduction	3
2. Méta-Modèles	3
2.1. ChaiseMinute	3
2.2. Algorithm	4
2.3. Function	4
3. Transforamtions Texte à Modèle de ChaiseMinute	5
4. Transformations Modèle à Texte	5
4.1. ChaiseMinute vers Python	5
5. Transformations Modèle à Modèle	5
6. Edition graphique	5
7. Exemples	5
8. Conclusion	5
9. Annexes	5

1. Introduction

Ce projet consiste en la réalisation d'un environnement de calcul Domaine-Scientifique nommé : ChaiseMinute.

ChaiseMinute permet à tous ses utilisateurs de développer des schémas de tables et des librairies pour traiter automatiquement des données respectant ces schémas. Ainsi l'utilisateur pourra produire des outils pour d'autres utilisateurs finaux souhaitant manipuler des données sans avoir à créer leur propres outils.

L'utilisateur de ChaiseMinute dispose d'une syntaxe textuelle ainsi qu'un outil graphique permettant de définir ses schémas de tables. Les schémas de tables pourront être transformés dans des langages plus spécifiques au calcul pour pouvoir transformer et vérifier des données.

2. Méta-Modèles

Nous avons décidé de séparer le modèle en trois sous-méta-modèles :

1. ChaiseMinute.Ecore
 - Point d'entrée, il permet de définir les schéma de table
2. Algorithm.Ecore
 - Définir les algorithmes utilisés pour les ComputedColumns et les Constraints
3. Function.Ecore
 - Pour spécifier les Functions utilisées par les Algorithm

2.1. ChaiseMinute

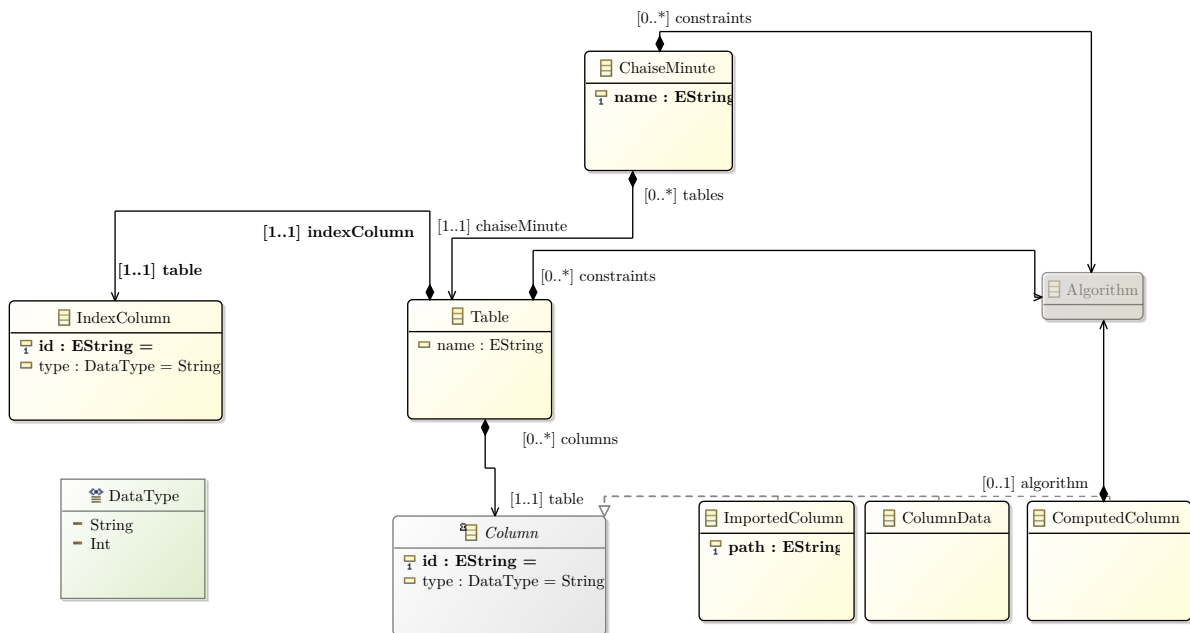


Fig. 1. – ChaiseMinute Ecore Diagram.

- ChaiseMinute représente le container permettant de regrouper les schémas de tables entre eux.
- Table défini un schéma de table comme un ensemble de Column et d'une IndexColumn, des contraintes peuvent également être ajoutées pour vérifier les données d'entrée
- IndexColumn : est une colonne spéciale pour les index des lignes
- Column est une super-classe représentant une colonne, elle contient un type de données et d'éventuelles contraintes sur les données d'entrée ou de sortie
 - ColumnData : une colonne simple pour représenter une donnée,

- ImportedColumn : une colonne provenant d'une autre Table,
- ComputedColumn : représentant une colonne calculée avec un Algorithm

2.2. Algorithm

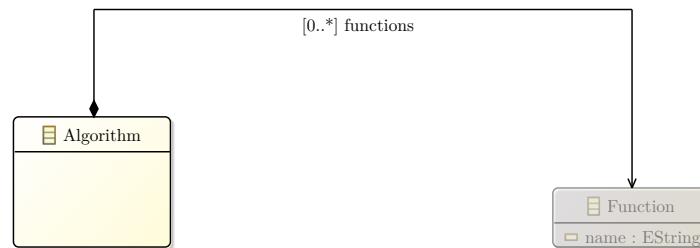


Fig. 2. – Algorithm Diagram.

Algorithm représente le point d'entrée d'un algorithme, il contient les fonctions qui seront appliquées.

L'application des fonctions respecte les règles suivantes :

- La première fonction est appliquée à ses arguments déclarés dans le modèle
- Les fonctions suivantes reçoivent en premier argument la sortie de la fonction précédente, puis leur arguments déclarés

Ces règles nous permettent de chaîner les fonctions dans un algorithme.

2.3. Function

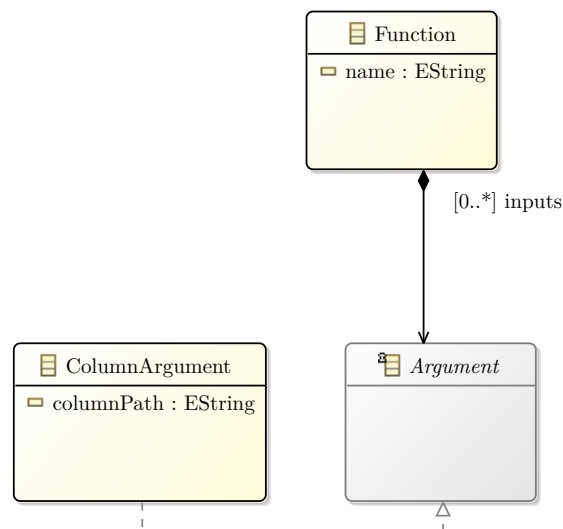


Fig. 3. – Algorithm Diagram.

Une Function est représentée par un identifiant qui référence un programme Python, elle contient des arguments, qui sont des références vers des colonnes. Les colonnes sont représentées sous la forme `nomTable.nomColonne` dans tout le projet (pour les références croisées de Fig. 1, les arguments de fonctions, etc).

2.3.1. Limitations

On aurait pu faire en sorte que les Function soient des arguments, ainsi on aurait pu construire un arbre d'appels et prendre la sortie de plusieurs fonctions à la fois en arguments. On peut quand même obtenir ce résultat avec le système actuel en adaptant les fonctions Python pour

qu'elles renvoient plusieurs colonnes, ainsi en adaptant la fonction suivante pour qu'elle récupère ces deux colonnes, on imite le fonctionnement d'un arbre d'appels.

3. Transformations Texte à Modèle de ChaiseMinute

4. Transformations Modèle à Texte

4.1. ChaiseMinute vers Python

5. Transformations Modèle à Modèle

6. Edition graphique

Nous avons développé un outils graphique permettant de modifier des fichiers .cml (ChaiseMinute) pour modifier les différents schémas de tables et obtenir une visualisation plus pratique pour l'utilisateur.

- Les Tables sont représentées par des container blanc
- Les IndexColumn sont représentées par des container rouge à l'intérieur des Tables

Il est aussi possible pour l'utilisateur de rajouter des fonctions utiles pour une ComputedColumn. Cependant nous avons rencontré des difficultés à choisir des fonctions inutile. En effet, nous ajoutons et enlevons les fonctions en écrivant leur chemin dans une boite de dialogue texte mais pour l'enlever nous n'avons pas réussi à utiliser la valeur renvoyée pour vérifier si elle correspondait à un fichier présent et donc le supprimer en conséquence.

7. Exemples

8. Conclusion

9. Annexes