A Domain Specific Language for Security Model in Database-centric applications

Student: Hoàng Nguyễn Phước Bảo

Universidad Autónoma de Madrid, Spain

Abstract. To be decided.

1 Introduction

Model-Driven Engineering (MDE) turns the attention on models, instead of code.

Model-Driven Security (MDS) is a specialization on the domain of security.

One of my recent effort in MDS is to propose a model-driven approach in defining security policies for accessing data in database-centric application.

In this report, I am going to realize this proposal into a prototype by using the technology I learnt from the course.

Organization

2 Background and Motivation

Relational Databases and SQL

Access Control in Relational Databases

OCL and Authorization Constraints

Related work on realizing SQLSI

3 The SQLSI Metamodels

3.1 Input Metamodels

Metamodel for data models For SQLSI, a data model contains entities and associations between them. An entity may have properties which are attributes or associations-ends. The multiplicity of an association-end is either 'one' or 'many'.

The data model metamodel for SQLSI is shown in Figure 1. `DataModel` is the root element and contains a set of `Entity`s. Every `Entity` represents an entity in the data model: it has a unique name and is related to a set of `Attribute`s and a set of `AssociationEnd`s. Each `Attribute` represents an attribute of an entity: it has a name and a type. Each `AssociationEnd` represents an association-end: it has a name, a `Multiplicity` value. Each `AssociationEnd` is also linked to its opposite `EAssociationEnd`, and with its `target EEntity`.

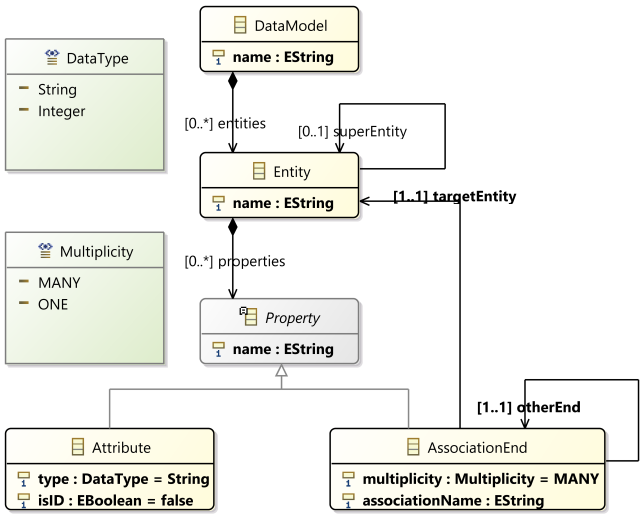


Fig. 1: SQLSI metamodel for data models.

For the sake of consistency, all metamodel invariants (including the ones that can be expressed by the metamodel) will be specified in Xtext. Considering these invariants:

- Within the same datamodel, the name of the entity must be unique.
- Within the same entity, the name of the property must be unique.
- Every entity either has no superclass and has exactly one ID attribute or has a superclass and has no ID attribute.
- An ID attribute must by of type Integer.
- The relation otherEnd is not reflexive.
- The relation superEntity (if exists) is not reflexive nor acyclic.
- For every association end, there exists exactly one another association end such that it be its otherEnd.

Security Model Metamodel [1]

[1] All metamodel invariants will be specified in Xtext, for the sake of consistency.

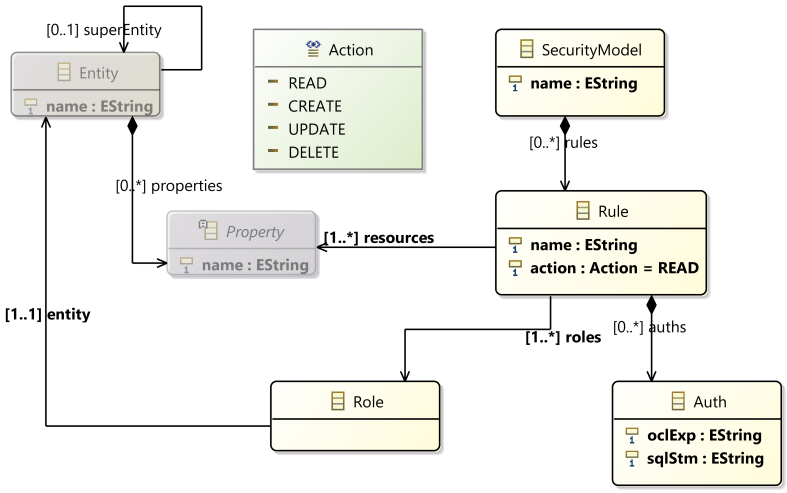


Fig. 2: SQLSI metamodel for security models.

3.2 Output Metamodels

Relational Schema Metamodel [2]

Lightweight version of SELECT-statement Metamodel [3]

4 The SQLSI Language and Design

4.1 A DSL for Security Model

Definition

Building an editor with Xtext

4.2 A M2M Exogenous Transformation from Data Model to Relational Schema

Definition

Database Schema generation with Acceleo

[2] All metamodel invariants will be specified in Xtext, for the sake of consistency.
[3] All metamodel invariants will be specified in Xtext, for the sake of consistency.

4.3 A M2M Endogenous Transformation from a SELECT-statement to a secure SELECT-statement

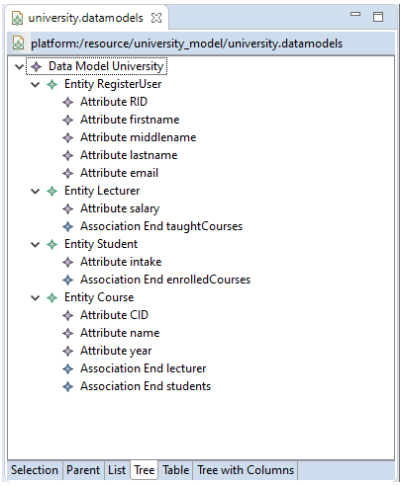
Definition

SQLSI Code generation with Acceleo

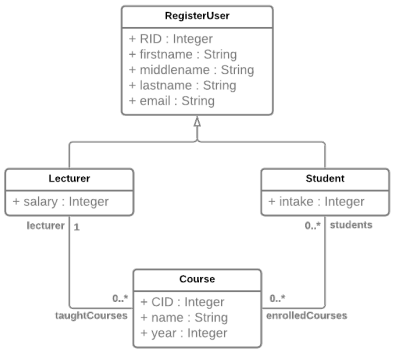
5 Case study: A Simple University Management System

5.1 Models

The input datamodel Consider the simple university management system in Figure 3, it comprises of four entities, namely `RegisterUser`, `Lecturer`, `Student` and `Course`, with their corresponding attributes and associations, whose description shall be omitted for the sake of simplicity.



(a) Abstract Syntax (Tree-like Model)



(b) Concrete Syntax (UML Diagram)

Fig. 3: The University Mangagement System datamodel

6 Conclusions and Future Work