Sample Use-case: University Management System

version 1.1

Student: Hoang Nguyen Phuoc Bao

January 12, 2021

1 Requirements

Before going through this sample use-case, please make sure that you have Java 8 or above and the Eclipse installation with all required plugins on your running computer.

Eclipse IDE installation

As a suggestion, you can have the required installation of Eclipse via the following options:[1]

- For one who does not have Eclipse installed, please download the following complete installation, with all plugins used in the use case via this link:
  `http://miso.es/teaching/mde1718/eclipse.rar`
  In addition, please install the `Epsilon` plugins for ease of registering the meta-models.

  1. On the toolbar of Eclipse IDE, click `Help / Install New Software...`
  2. In the `Install` pop-up window, in the `Work with:` section, entered:
     `http://download.eclipse.org/epsilon/updates/2.2/`
  3. Click `Select All` then complete the installation.

- For one who does have Eclipse installed, please install the following plugins:

  – Work with: 2020-09 - http://download.eclipse.org/releases/2020-09
    - ∗ General Purpose Tool
      - · Eclipse Plug-in Development Environment
    - ∗ Modeling
      - · Acceleo
      - · ATL SDK - ATL Transformation Language SDK
      - · Ecore Diagram Editor (SDK)

[1] Taken from the content of the Formal Model Driven Engineering class.

        · EMF - Eclipse Modeling Framework SDK
        · OCL Examples and Editors SDK
        · Xtext Complete SDK
    – Work with: http://download.eclipse.org/modeling/gmp/gmf-tooling/-updates/releases-3.2.1/
    – Work with: http://download.eclipse.org/modeling/emft/henshin/up-dates/release

2 Execution Instructions

2.1 Import the project to Eclipse IDE Workspace

Download the project

- You will find the complete project in the directory `MDS-SQLSI`.
- In case you clone the project from GitHub, please follow the follwing instructions:
    1. Clone the project using URL
       `git clone https://github.com/npbhoang/MDS-SQLSI.git`
    2. Navigate to the project root.
       `cd MDS-SQLSI`
    3. Checkout the correct branch for this sample use-case.
       `git checkout university-use-case`

Import the project to Eclipse IDE Workspce

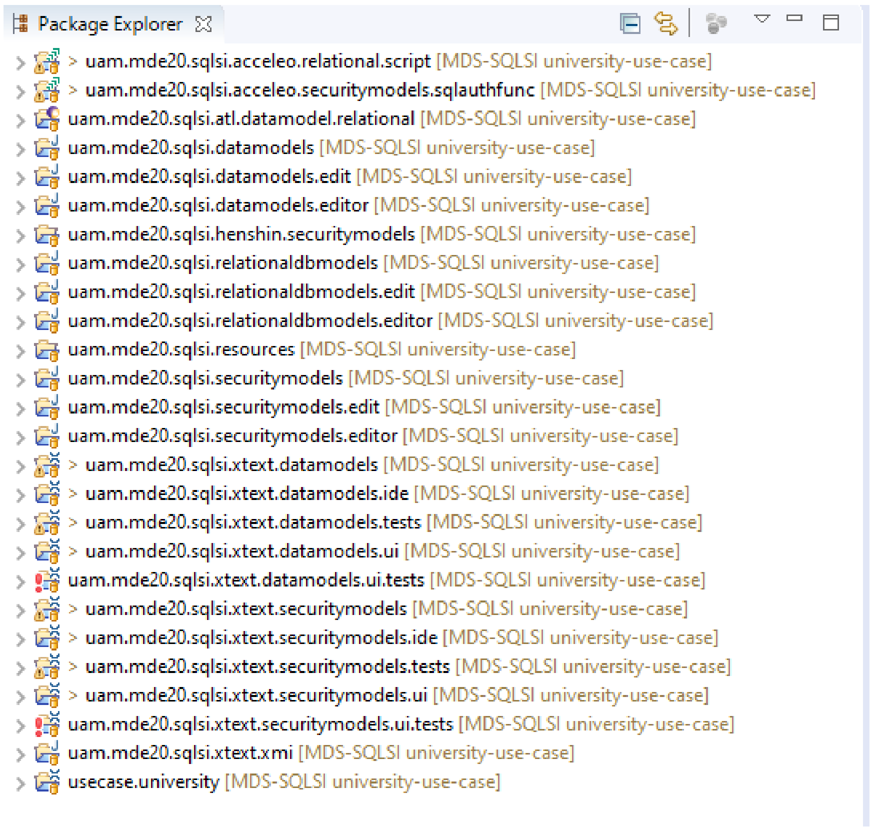
1. In Eclipse, on the tool bar, click `File / Import`
2. In the `Import` pop-up window, click `General / Existing Projects into Workspace`
3. Navigate correctly the project source directory.
4. Add all (sub-)projects.

Figure 1 shows what you should obtain in your `Package Explorer` afterwards.

2.2 Register meta-models and start a new Eclipse instance

Register meta-models

1. Navigate to `uam.mde20.sqlsi.datamodels / metamodels / datamodels.ecore`
2. Right click on `datamodels.ecore` and choose `Register EPackages`


Package Explorer

```
> uam.mde20.sqlsi.acceleo.relational.script [MDS-SQLSI university-use-case]
> uam.mde20.sqlsi.acceleo.securitymodels.sqlauthfunc [MDS-SQLSI university-use-case]
uam.mde20.sqlsi.atl.datamodel.relational [MDS-SQLSI university-use-case]
uam.mde20.sqlsi.datamodels [MDS-SQLSI university-use-case]
uam.mde20.sqlsi.datamodels.edit [MDS-SQLSI university-use-case]
uam.mde20.sqlsi.datamodels.editor [MDS-SQLSI university-use-case]
uam.mde20.sqlsi.henshin.securitymodels [MDS-SQLSI university-use-case]
uam.mde20.sqlsi.relationaldbmodels [MDS-SQLSI university-use-case]
uam.mde20.sqlsi.relationaldbmodels.edit [MDS-SQLSI university-use-case]
uam.mde20.sqlsi.relationaldbmodels.editor [MDS-SQLSI university-use-case]
uam.mde20.sqlsi.resources [MDS-SQLSI university-use-case]
uam.mde20.sqlsi.securitymodels [MDS-SQLSI university-use-case]
uam.mde20.sqlsi.securitymodels.edit [MDS-SQLSI university-use-case]
uam.mde20.sqlsi.securitymodels.editor [MDS-SQLSI university-use-case]
> uam.mde20.sqlsi.xtext.datamodels [MDS-SQLSI university-use-case]
> uam.mde20.sqlsi.xtext.datamodels.ide [MDS-SQLSI university-use-case]
> uam.mde20.sqlsi.xtext.datamodels.tests [MDS-SQLSI university-use-case]
> uam.mde20.sqlsi.xtext.datamodels.ui [MDS-SQLSI university-use-case]
uam.mde20.sqlsi.xtext.datamodels.ui.tests [MDS-SQLSI university-use-case]
> uam.mde20.sqlsi.xtext.securitymodels [MDS-SQLSI university-use-case]
> uam.mde20.sqlsi.xtext.securitymodels.ide [MDS-SQLSI university-use-case]
> uam.mde20.sqlsi.xtext.securitymodels.tests [MDS-SQLSI university-use-case]
> uam.mde20.sqlsi.xtext.securitymodels.ui [MDS-SQLSI university-use-case]
uam.mde20.sqlsi.xtext.securitymodels.ui.tests [MDS-SQLSI university-use-case]
uam.mde20.sqlsi.xtext.xmi [MDS-SQLSI university-use-case]
usecase.university [MDS-SQLSI university-use-case]
```

Figure 1: Package Explorer after Step 1

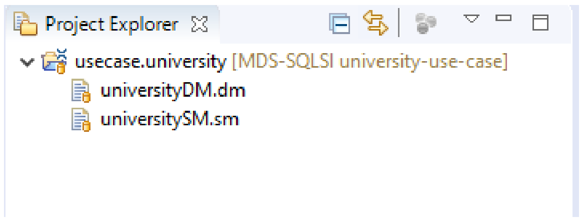
3. Navigate to `uam.mde20.sqlsi.relationaldbmodels / metamodels / relationaldb.ecore`
4. Right click on `relationaldb.ecore` and choose `Register EPackages`
5. Navigate to `uam.mde20.sqlsi.securitymodels / metamodels / securitymodels.ecore`
6. Right click on `securitymodels.ecore` and choose `Register EPackages`

Start Eclipse runtime instance

1. Navigate to the project `uam.mde20.sqlsi.xtext.securitymodels`.
2. Right click the top of the project and choose `Run as / Eclipse Application / Launch Runtime Eclipse` (Choose the second one instance). Please ignore the error warning, if any.

3. The new Eclipse runtime instance will pop-up, in this new instance, import only the `usecase.university` project from the local directory.

Figure 2 shows what you should obtain in your `Project Explorer` in the new Eclipse instance afterwards.



Project Explorer

usecase.university [MDS-SQLSI university-use-case]

universityDM.dm

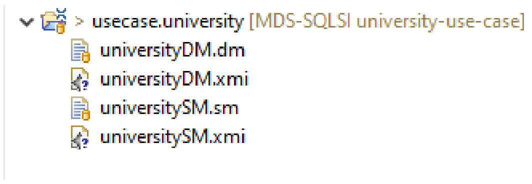
universitySM.sm

Figure 2: Project Explorer of new Eclipse instance after Step 2

2.3 Transform models from specific DSL format to XMI format

1. Switch back to the original Eclipse window.
2. Navigate to `uam.mde20.sqlsi.xtext.xmi / src / xmi / Main.java`.
3. Please change the absolute path of the DSL model of datamodel (`usecase.university/universityDM.dm`) and securitymodel (`usecase.university/universitySM.sm`) accordingly.
4. Right click on `Main.java` and choose `Run as / Java Application`.
5. Refresh the `usecase.university` on both Eclipse windows.

Figure 3 shows what you should obtain in your `Project Explorer` in the new Eclipse instance afterwards. As the result, you can see two new XMI models, one is the university data-model and the other is the university security-model.



> usecase.university [MDS-SQLSI university-use-case]

universityDM.dm

universityDM.xmi

universitySM.sm

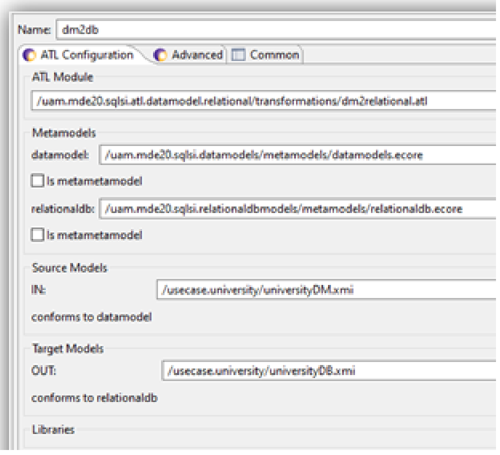
universitySM.xmi

Figure 3: Project Explorer of new Eclipse instance after Step 3

2.4 Transform datamodel to relational database model, from which generate SQL schemata

Transform datamodel to relational database model

1. Switch back to the original Eclipse window.
2. On the toolbar, click `Run / Run Configurations...`
3. Double click on `ATL Transformation`
4. Figure 4 shows a sample configuration of this transformation.
5. Click `Run`
6. Refresh the `usecase.university` on both Eclipse windows. As the result, you can see a new XMI model, which is the university relational database model.



Name: dm2db

ATL Module
/uam.mde20.sqlsi.atl.datamodel.relational/transformations/dm2relational.atl

Metamodels
datamodel: /uam.mde20.sqlsi.datamodels/metamodels/datamodels.ecore
[ ] Is metametamodel
relationaldb: /uam.mde20.sqlsi.relationaldbmodels/metamodels/relationaldb.ecore
[ ] Is metametamodel

Source Models
IN: /usecase.university/universityDM.xmi
conforms to datamodel

Target Models
OUT: /usecase.university/universityDB.xmi
conforms to relationaldb

Libraries

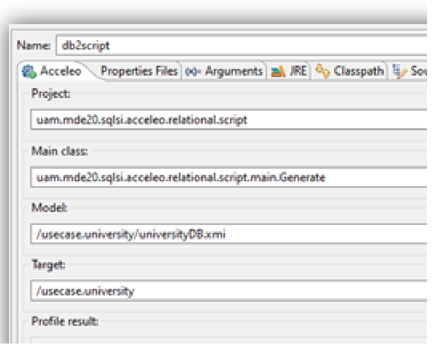
Figure 4: ATL Transformation sample configuration

Generate SQL schemata from relational database model

1. Switch back to the original Eclipse window.
2. On the toolbar, click `Run / Run Configurations...`
3. Double click `Acceleo Application`
4. Figure 5 shows a sample configuration of this generation.
5. Click `Run`

6. Refresh the `usecase.university` on both Eclipse windows.

Figure 6 shows what you should obtain in your `Project Explorer` in the new Eclipse instance afterwards. As the result, you can see a new SQL script that is executable in MySQL relational database management system.


Name: db2script

Acceleo | Properties Files | Arguments | JRE | Classpath | Sou

Project:
uam.mde20.sqlsi.acceleo.relational.script

Main class:
uam.mde20.sqlsi.acceleo.relational.script.main.Generate

Model:
/usecase.university/universityDB.xmi

Target:
/usecase.university

Profile result:

Figure 5: Acceleo Code-Generation sample configuration

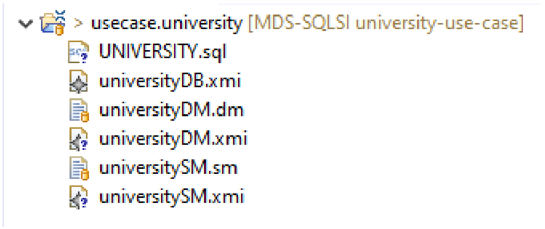

> usecase.university [MDS-SQLSI university-use-case]
UNIVERSITY.sql
universityDB.xmi
universityDM.dm
universityDM.xmi
universitySM.sm
universitySM.xmi

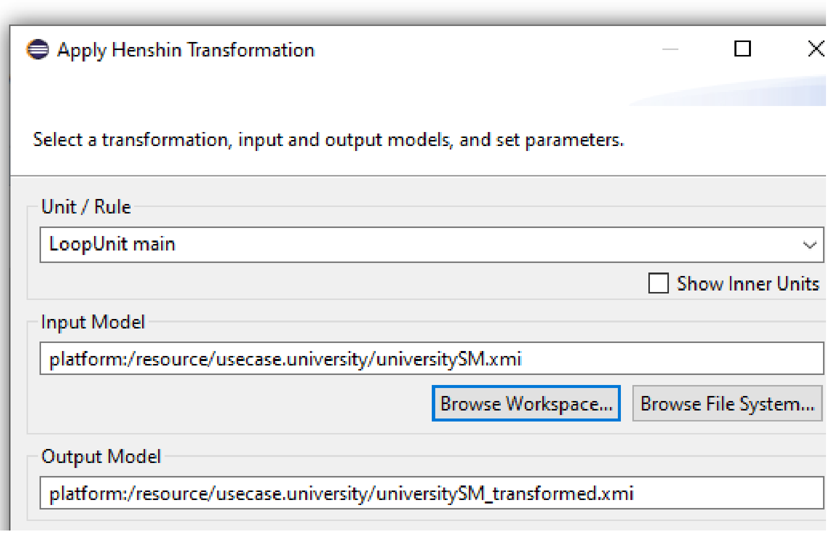
Figure 6: Project Explorer of new Eclipse instance after Step 4

2.5 Manipulate security model then generate SQL authorization functions

Manipulate security model

1. Switch back to the original Eclipse window.
2. Navigate to `uam.mde20.sqlsi.henshin.securitymodels / default.henshin`
3. Right click on it and choose `Henshin / Apply Transformation`
4. Figure 7 shows a sample configuration of this manipulation.
5. Click `Transform`, the following error in Figure 8 window will pop-up. This is indeed a unsolved problem. Please, for the moment, ignore this error by click `OK`.

6. Click `Transform` again, this time it should work!
7. Refresh the `usecase.university` on both Eclipse windows. As the result, you can see a new XMI "transformed" model, which is the university security model in the "normalized" form.



Select a transformation, input and output models, and set parameters.

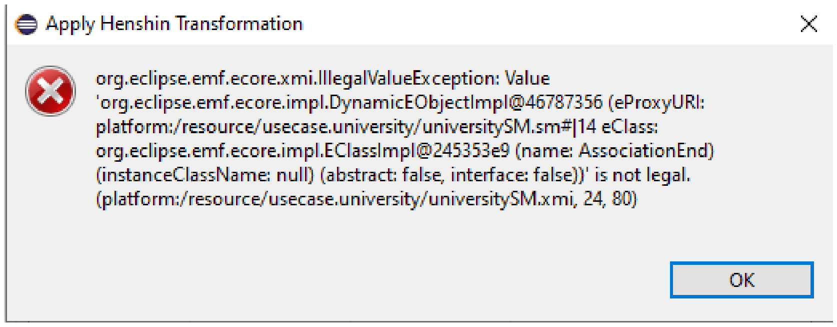
Unit / Rule: LoopUnit main

[ ] Show Inner Units

Input Model: platform:/resource/usecase.university/universitySM.xmi

Output Model: platform:/resource/usecase.university/universitySM_transformed.xmi

Figure 7: Henshin Transformation sample configuration



```
org.eclipse.emf.ecore.xmi.IllegalValueException: Value
'org.eclipse.emf.ecore.impl.DynamicEObjectImpl@46787356 (eProxyURI:
platform:/resource/usecase.university/universitySM.sm#|14 eClass:
org.eclipse.emf.ecore.impl.EClassImpl@245353e9 (name: AssociationEnd)
(instanceClassName: null) (abstract: false, interface: false))' is not legal.
(platform:/resource/usecase.university/universitySM.xmi, 24, 80)
```

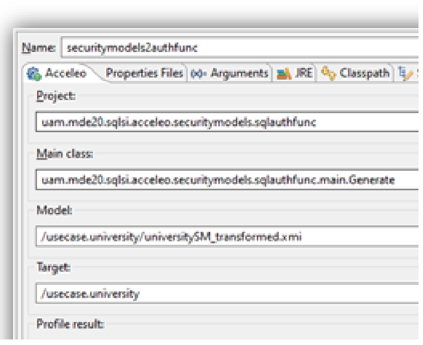
Figure 8: Henshin Transformation Error

Generate SQL authorization functions from security model

1. Switch back to the original Eclipse window.
2. On the toolbar, click `Run / Run Configurations...`

3. Double click `Acceleo Application`
4. Figure 9 shows a sample configuration of this generation.
5. Click `Run`
6. Refresh the `usecase.university` on both Eclipse windows.

Figure 10 shows what you should obtain in your `Project Explorer` in the new Eclipse instance afterwards. As the result, you can see a new SQL script for authorization functions that is executable in MySQL relational database management system.



Name: securitymodels2authfunc

Project:
uam.mde20.sqlsi.acceleo.securitymodels.sqlauthfunc

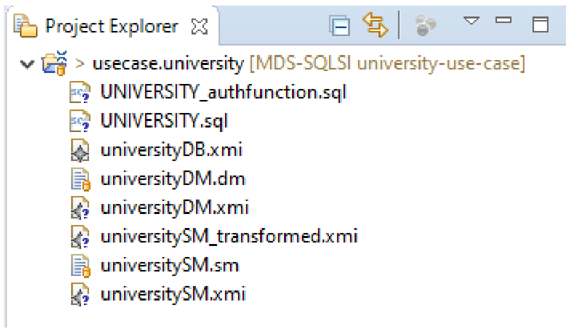
Main class:
uam.mde20.sqlsi.acceleo.securitymodels.sqlauthfunc.main.Generate

Model:
/usecase.university/universitySM_transformed.xmi

Target:
/usecase.university

Profile result:

Figure 9: Acceleo Code-Generation sample configuration



```
> usecase.university [MDS-SQLSI university-use-case]
    UNIVERSITY_authfunction.sql
    UNIVERSITY.sql
    universityDB.xmi
    universityDM.dm
    universityDM.xmi
    universitySM_transformed.xmi
    universitySM.sm
    universitySM.xmi
```

Figure 10: Project Explorer of new Eclipse instance after Step 5