Modif Documentation

-

Co-evolution User Guide

This document explains the steps for using ModifRoundtrip for a Co-evolution (Refactoring and Migration)

June, 2015

by Paola Vallejo, Jean Philippe Babau

Table of contents

1. Create project, folders and ecore metamodel

1.1. Create a project

Create a new Empty EMF project and, for example name it *Test_Vehicles*

File Ÿ New Ÿ Other... Ÿ Eclipse Modeling Framework/Empty EMF Project Ÿ Next Ÿ Specify the project name [Test_Vehicles] Ÿ Finish

1.2. Add Xtext nature

Add the Xtext Nature to the Test project

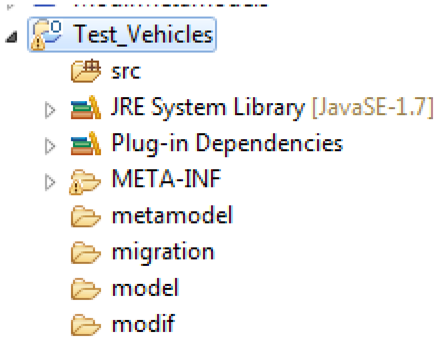
Right click Test_Vehicle project Ÿ Configure Ÿ Add Xpand/Xtext Nature

1.3. Add Folders

Keep the **model** folder and add three additional folders:

Right click *Test_Vehicles*Ÿ New Ÿ Folder Ÿ [Specify the folder_name] Ÿ Finish

1. metamodel
2. migration
3. modif

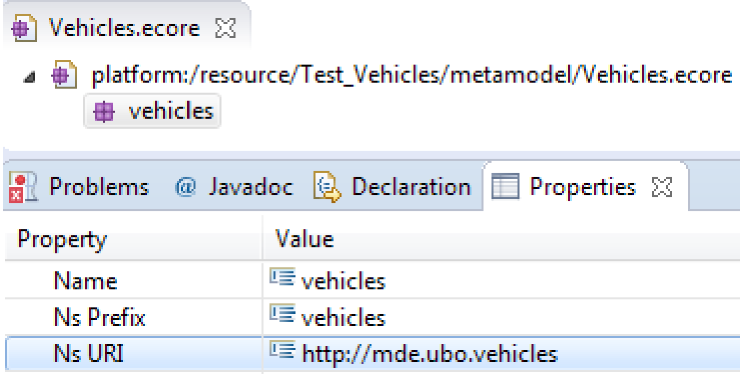


1.4. Create metamodel

In the ***metamodel*** folder, create a new ecore model. And for the example, name it *Vehicles.ecore*

Right click on Test_Vehicles/metamodel Ÿ New Ÿ Other... Ÿ Eclipse Modeling Framework / Ecore Model Ÿ Next Ÿ [give a name for your ecore]Ÿ Finish

Then set the ecore model properties as follows:



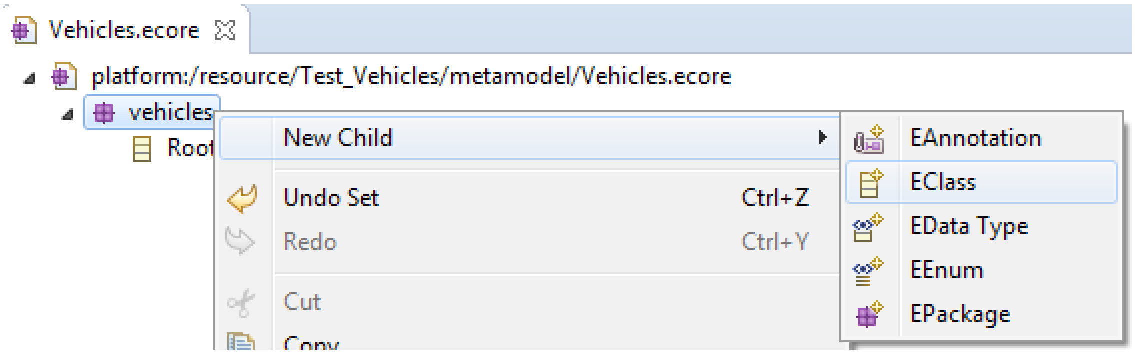
Vehicles.ecore

platform:/resource/Test_Vehicles/metamodel/Vehicles.ecore
  vehicles

Property	Value
Name	vehicles
Ns Prefix	vehicles
Ns URI	http://mde.ubo.vehicles

Add the EClasses to the metamodel.

Right click on *vehicles* package Ÿ New ChildŸ EClass



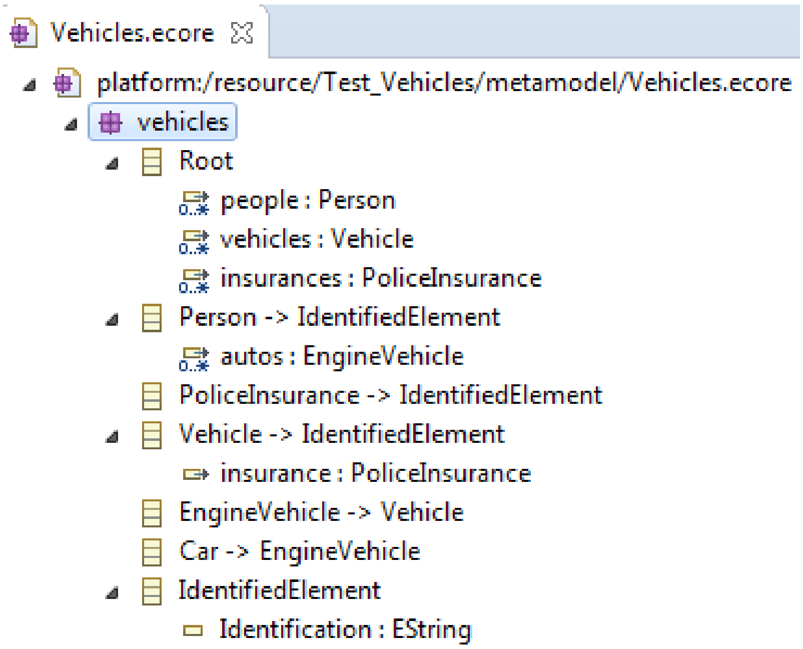
Vehicles.ecore

platform:/resource/Test_Vehicles/metamodel/Vehicles.ecore
  vehicles
    Root

Add the EReferences and EAttributes to the metamodel.

Right click on EClass Ÿ New ChildŸ [Select EReference or EAttribute]

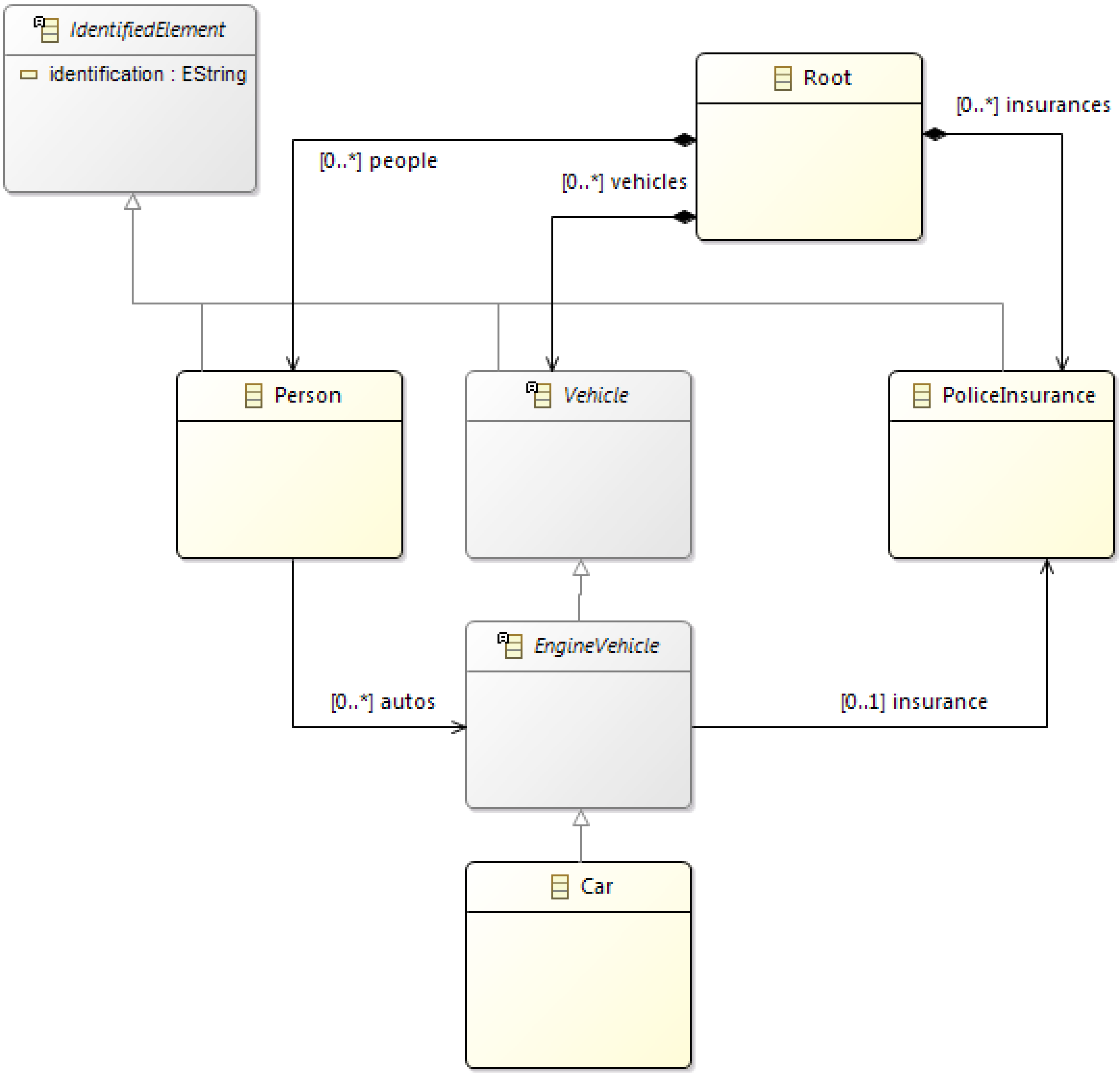
It must look as follows:



```
Vehicles.ecore

platform:/resource/Test_Vehicles/metamodel/Vehicles.ecore
  vehicles
    Root
      people : Person
      vehicles : Vehicle
      insurances : PoliceInsurance
    Person -> IdentifiedElement
      autos : EngineVehicle
    PoliceInsurance -> IdentifiedElement
    Vehicle -> IdentifiedElement
      insurance : PoliceInsurance
    EngineVehicle -> Vehicle
    Car -> EngineVehicle
    IdentifiedElement
      Identification : EString
```

In this metamodel there is a **Root** EClass composed of **Person**s (*person* EReference), **Vehicle**s (*vehicles* EReference) and **PoliceInsirance**s (*insurance* EReference). Person, Vehicle and PoliceInsurance have an identification (*identification* EAttribute). A Person can have some Vehicles. Each Vehicle has an asociated PoliceInsurance.

Graphically, it looks as follows:



Please note: as usual with Ecore, an *EClass* plays a root role in the graph of *EClasses*; the name *Root* is not mandatory but, what is important is that, this *EClass* contains directly or indirectly all the other concrete *EClasses.*

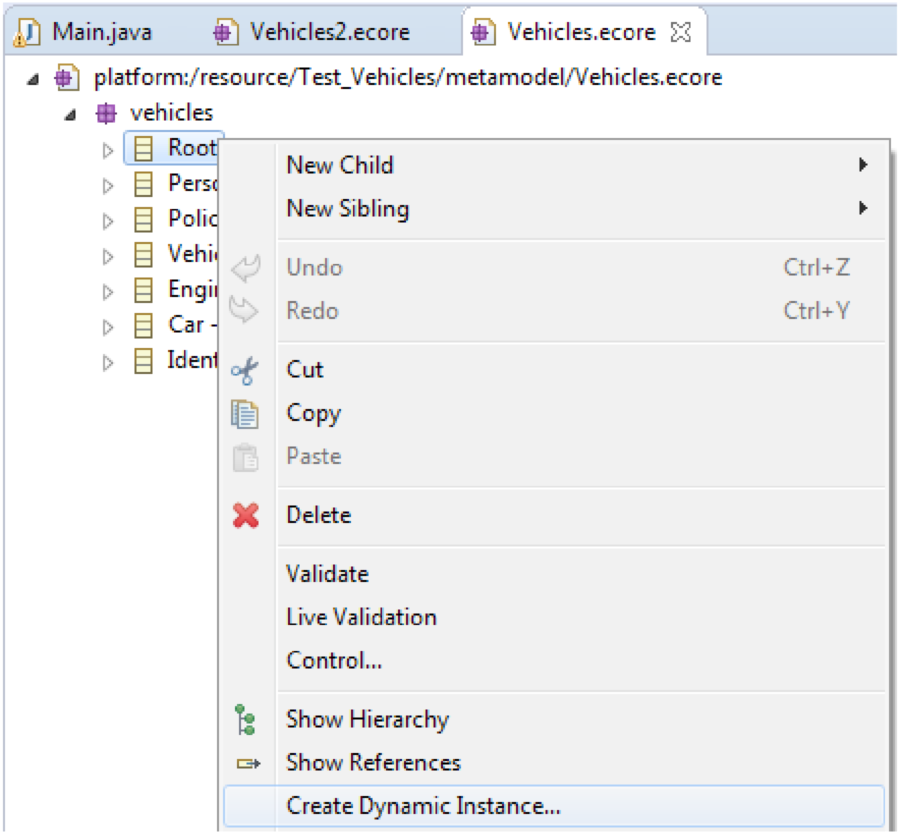
1.5. Create model

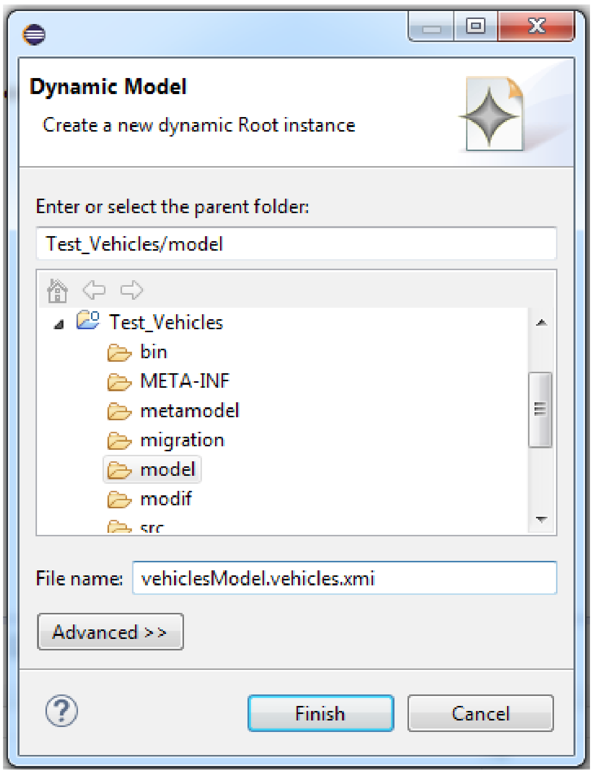
In the ***model*** folder, create a new ecore model. And for the example, name it *vehiclesModel.**vehicles.xmi***.

Please note:

the model name must respect the following rule: modelName.**metamodelName.xmi**

Right click on the root EClass [Root] Ÿ Create Dynamic Instance




Dynamic Model

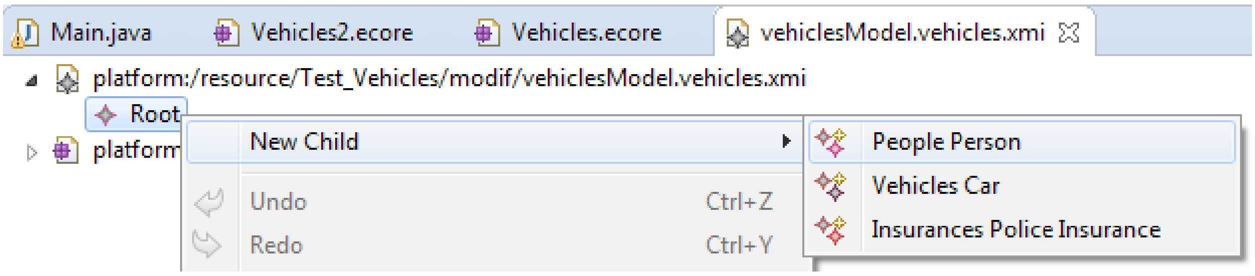
Create a new dynamic Root instance

Enter or select the parent folder: Test_Vehicles/model

File name: vehiclesModel.vehicles.xmi

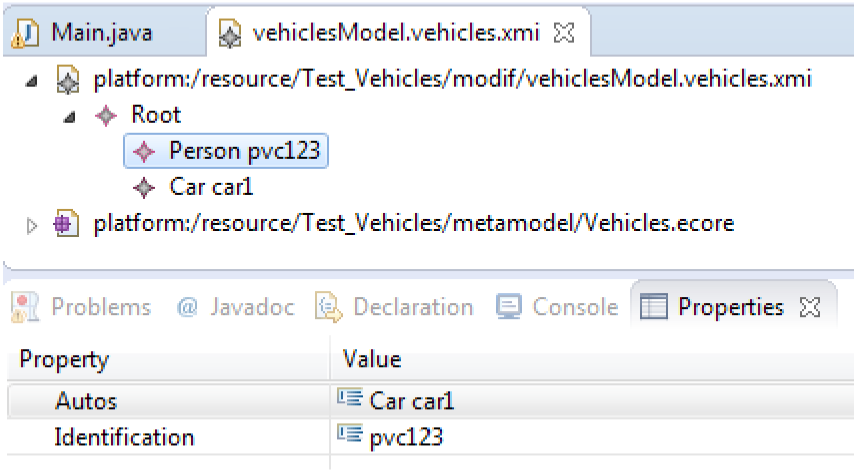
Add the elements to the model

Right click on EClass Ÿ New ChildŸ [Select element to create



platform:/resource/Test_Vehicles/modif/vehiclesModel.vehicles.xmi
Root
platform

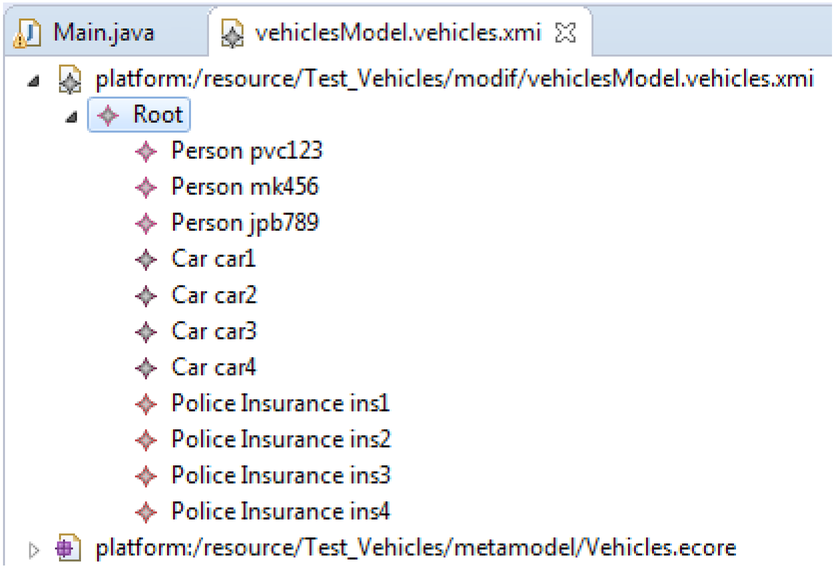
Fill the properties of the element



platform:/resource/Test_Vehicles/modif/vehiclesModel.vehicles.xmi
Root
Person pvc123
Car car1
platform:/resource/Test_Vehicles/metamodel/Vehicles.ecore

Property	Value
Autos	Car car1
Identification	pvc123

The model should looks as follows:



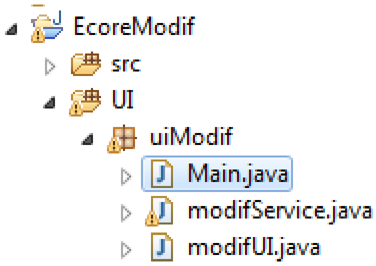
platform:/resource/Test_Vehicles/modif/vehiclesModel.vehicles.xmi
Root
Person pvc123
Person mk456
Person jpb789
Car car1
Car car2
Car car3
Car car4
Police Insurance ins1
Police Insurance ins2
Police Insurance ins3
Police Insurance ins4
platform:/resource/Test_Vehicles/metamodel/Vehicles.ecore

- pvc123 has car1
- mk456 has car2
- jpb789 has car3 and car4
- car1 is associated to i1
- car2 is associated to i2
- car3 is associated to i3
- car4 is associated to i4

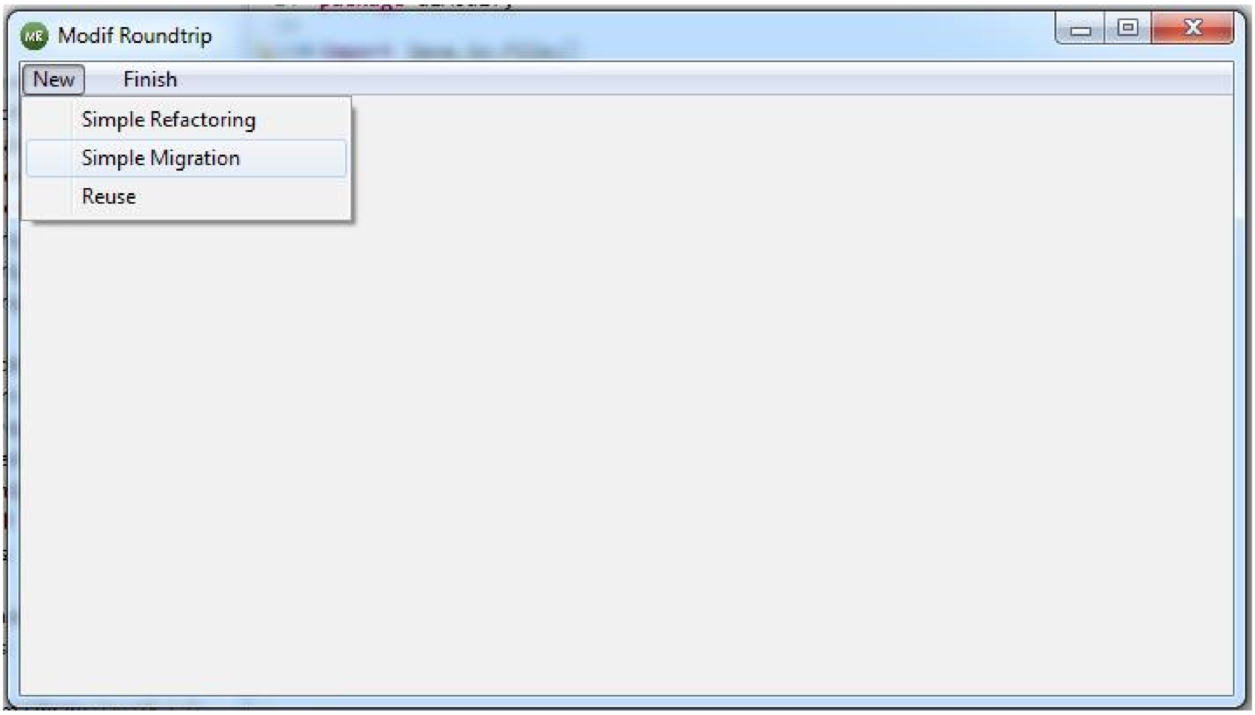
2. Execute Ecore Modif

1. Refactoring

Open the EcoreModif project, navigate to *UI/ uiModif* and execute *Main.java*



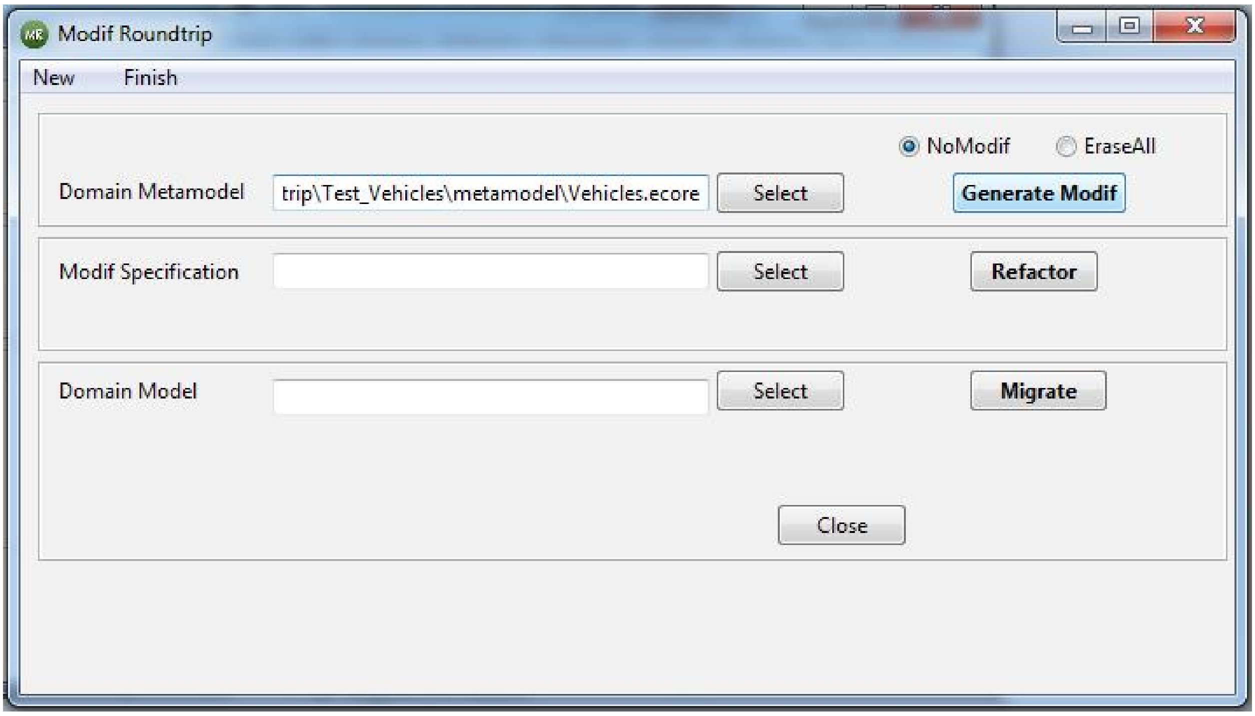
In the oped ModifRoundtrip form, click on *New* and then click on *Simple Migration*.



Now perform the following steps in order:

Specify Domain Metamodel and Generate Modif model

- Click on *Select* for Domain Metamodel and specify the path to the ecore file *Vehicles.ecore* (Test_Vehicles/metamodel folder)
- Select *NoModif* (or *EraseAll*) and then click on the **Generate Modif** button. If the Domain Metamodel path is not correct, an error message appears when executing the modif model generation



Modif Roundtrip

New Finish

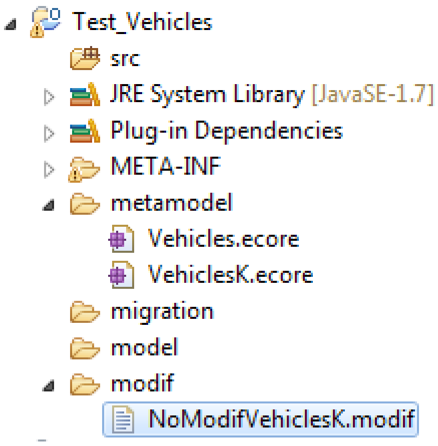
● NoModif ○ EraseAll

Domain Metamodel: trip\Test_Vehicles\metamodel\Vehicles.ecore

Modif Specification:

Domain Model:

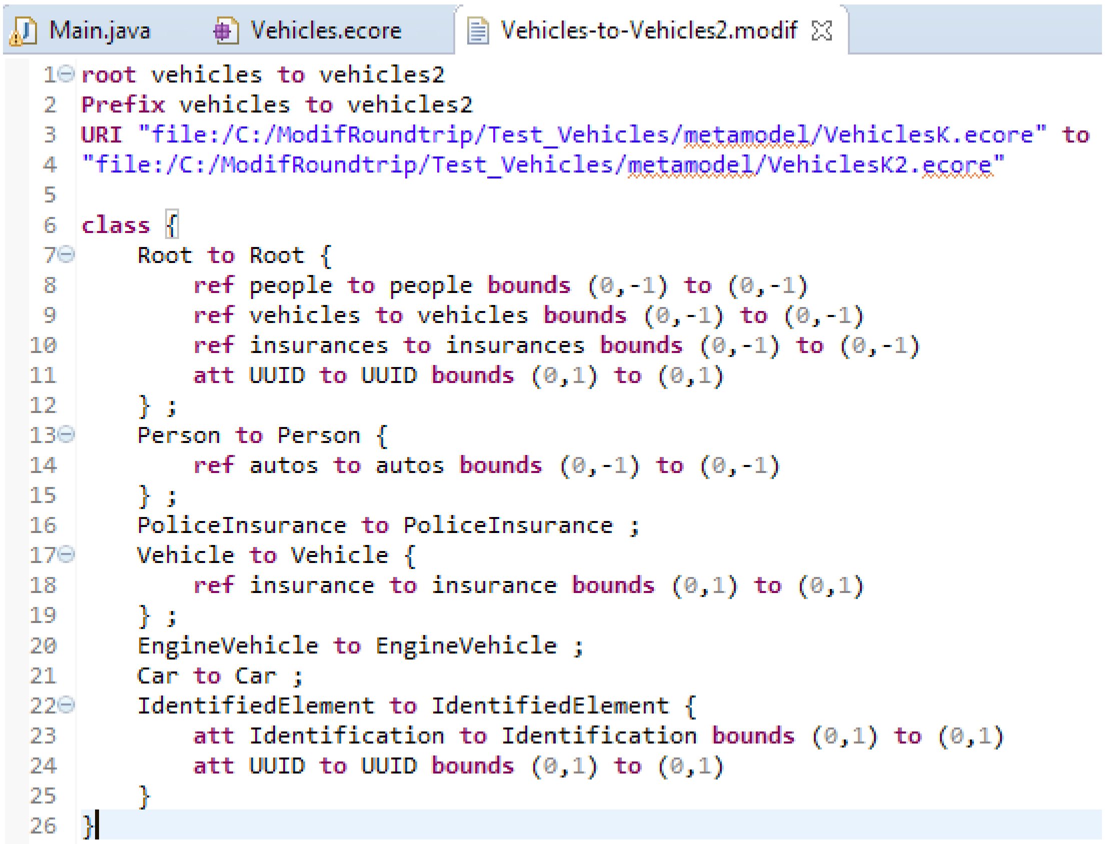
- Refresh the *Test_Vehicles* project, the modif model appears in the *modif* folder



```
Test_Vehicles
  src
  JRE System Library [JavaSE-1.7]
  Plug-in Dependencies
  META-INF
  metamodel
    Vehicles.ecore
    VehiclesK.ecore
  migration
  model
  modif
    NoModifVehiclesK.modif
```

Edit Modif Model

Make a copy of the NoModifVehiclesK.modif (or EraseAllVehiclesK.modif) file, paste it in the ***modif*** folder and name it ñVehicles-to-Vehicles2.modifò Then, open the Vehicles-to-Vehicles2.modif using the text editor or the xText editor as shown below:



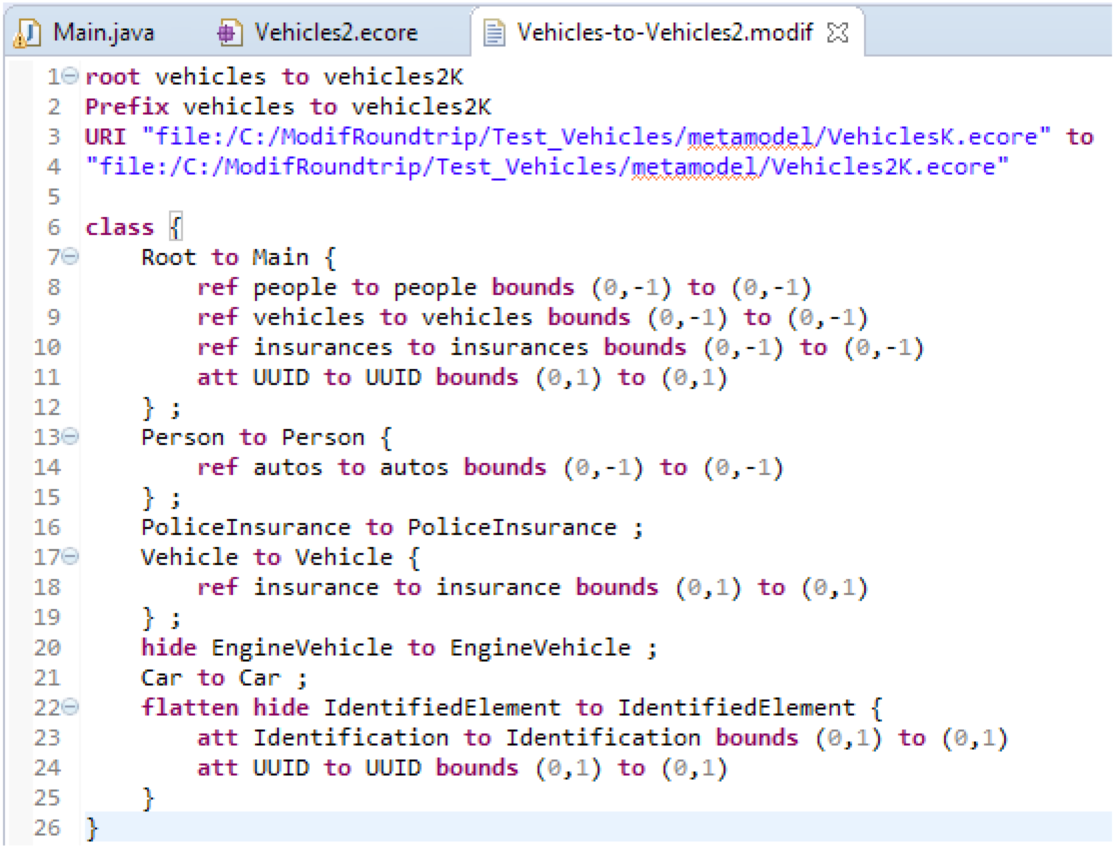
```
root vehicles to vehicles2
Prefix vehicles to vehicles2
URI "file:/C:/ModifRoundtrip/Test_Vehicles/metamodel/VehiclesK.ecore" to
"file:/C:/ModifRoundtrip/Test_Vehicles/metamodel/VehiclesK2.ecore"

class {
    Root to Root {
        ref people to people bounds (0,-1) to (0,-1)
        ref vehicles to vehicles bounds (0,-1) to (0,-1)
        ref insurances to insurances bounds (0,-1) to (0,-1)
        att UUID to UUID bounds (0,1) to (0,1)
    } ;
    Person to Person {
        ref autos to autos bounds (0,-1) to (0,-1)
    } ;
    PoliceInsurance to PoliceInsurance ;
    Vehicle to Vehicle {
        ref insurance to insurance bounds (0,1) to (0,1)
    } ;
    EngineVehicle to EngineVehicle ;
    Car to Car ;
    IdentifiedElement to IdentifiedElement {
        att Identification to Identification bounds (0,1) to (0,1)
        att UUID to UUID bounds (0,1) to (0,1)
    }
}
```

Edit the modif file in this way:

- For **root**, *rename* it **to** vehicles2**K** (do not forget to put the **K** at the end)
- For **Prefix**, *rename* it **to** vehicles2**K** (do not forget to put the **K** at the end)
- For **URI**, change ñVehiclesK2.ecoreò**to** ñVehicles2**K**.ecoreò
- *rename* Root **to** Main
- **hide** EngineVehicle
- **flatten hide** IdentifiedElement

Please note: K at the end of root, Prefix and URI is mandatory

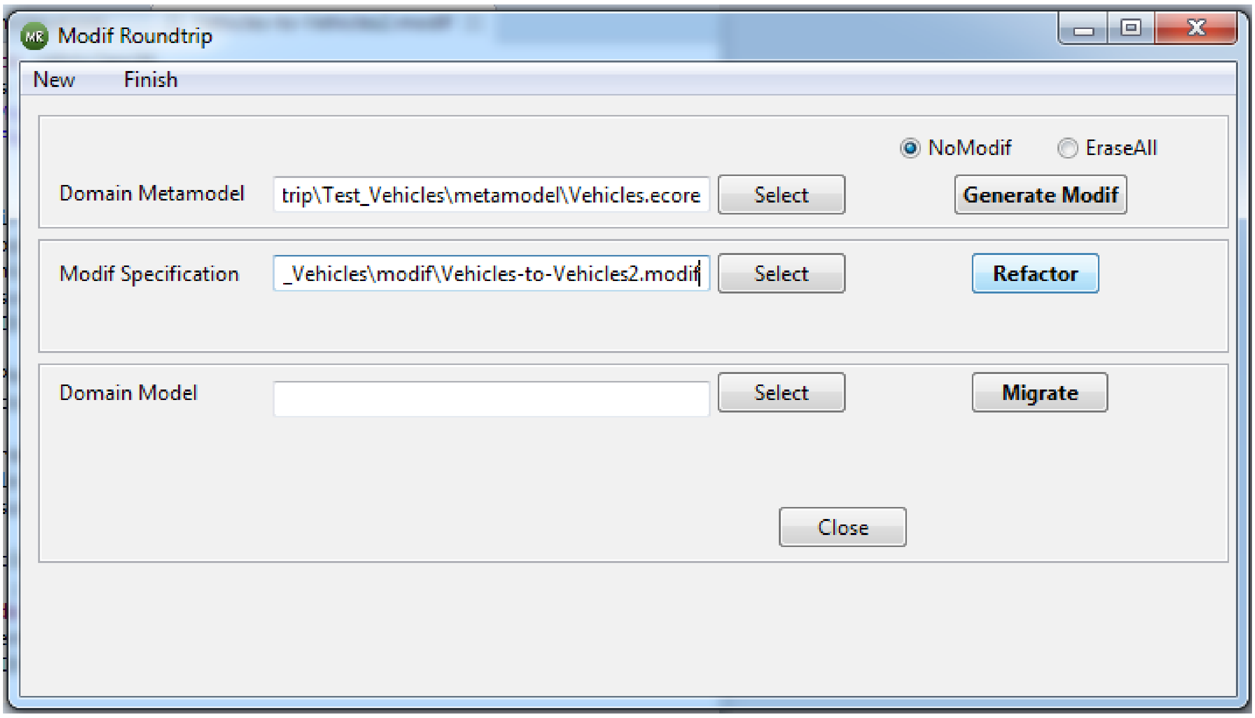
Finally, save the model, it will look as shown below



```
root vehicles to vehicles2K
Prefix vehicles to vehicles2K
URI "file:/C:/ModifRoundtrip/Test_Vehicles/metamodel/VehiclesK.ecore" to
"file:/C:/ModifRoundtrip/Test_Vehicles/metamodel/Vehicles2K.ecore"

class {
    Root to Main {
        ref people to people bounds (0,-1) to (0,-1)
        ref vehicles to vehicles bounds (0,-1) to (0,-1)
        ref insurances to insurances bounds (0,-1) to (0,-1)
        att UUID to UUID bounds (0,1) to (0,1)
    } ;
    Person to Person {
        ref autos to autos bounds (0,-1) to (0,-1)
    } ;
    PoliceInsurance to PoliceInsurance ;
    Vehicle to Vehicle {
        ref insurance to insurance bounds (0,1) to (0,1)
    } ;
    hide EngineVehicle to EngineVehicle ;
    Car to Car ;
    flatten hide IdentifiedElement to IdentifiedElement {
        att Identification to Identification bounds (0,1) to (0,1)
        att UUID to UUID bounds (0,1) to (0,1)
    }
}
```

Refactor

- In the ModifRoundtrip form, set the *Modif Specification* Vehicle-to-Vehicles2.modif
- Click on the **Refactor** button



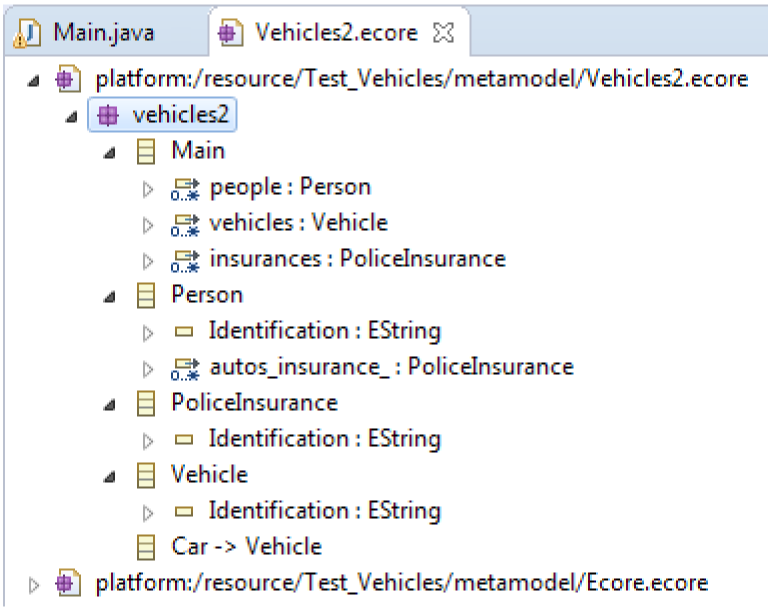
● NoModif ○ EraseAll

Domain Metamodel: trip\Test_Vehicles\metamodel\Vehicles.ecore

Modif Specification: _Vehicles\modif\Vehicles-to-Vehicles2.modif

Domain Model:

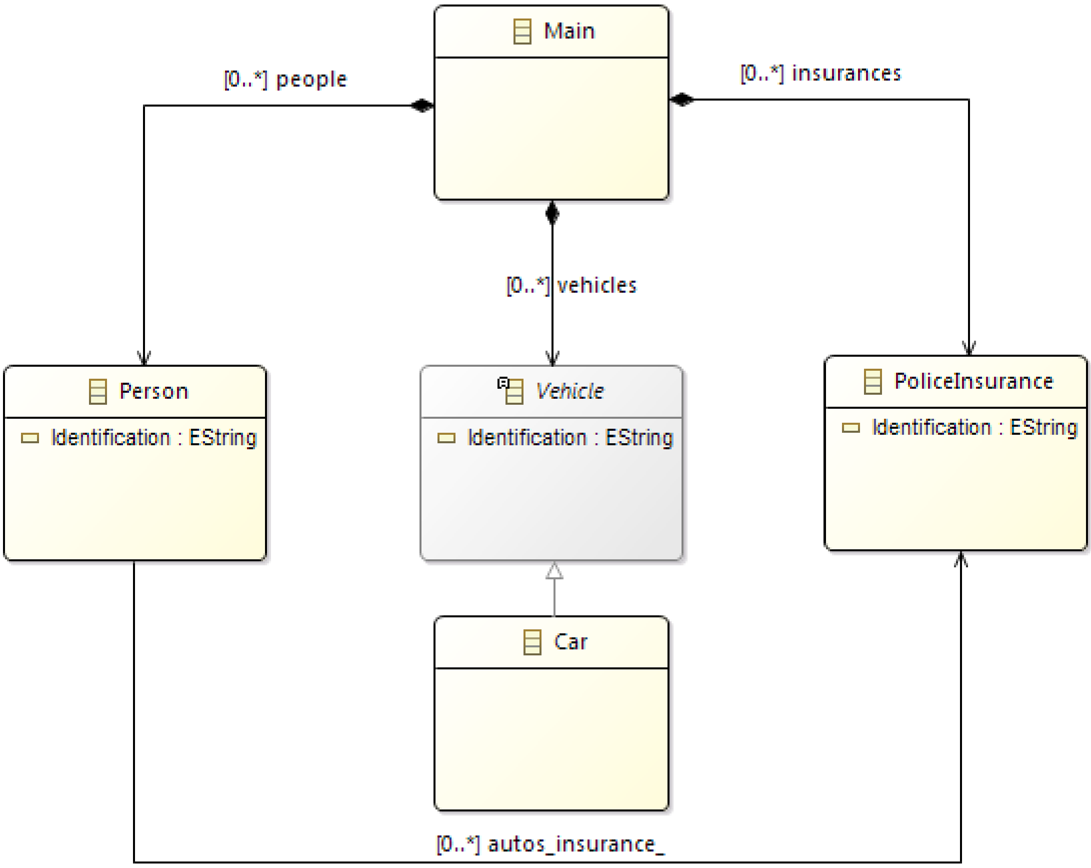
- Refresh the *Test_Vehicles/*metamodel folder and you will notice that the file *Vehicles2.ecore* is added. It will look as shown below:



Main.java | Vehicles2.ecore

- platform:/resource/Test_Vehicles/metamodel/Vehicles2.ecore
  - vehicles2
    - Main
      - people : Person
      - vehicles : Vehicle
      - insurances : PoliceInsurance
    - Person
      - Identification : EString
      - autos_insurance_ : PoliceInsurance
    - PoliceInsurance
      - Identification : EString
    - Vehicle
      - Identification : EString
    - Car -> Vehicle
- platform:/resource/Test_Vehicles/metamodel/Ecore.ecore

Graphically it looks as follows:



Notice the impact of the applied operators:

- *rename* Root **to** Main
  - Root has been renamed to Main
- **hide** EngineVehicle
  - EngineVehicle has been deleted
  - autos has been removed
  - insurance has been removed
  - autos_insurance_ has been added
- **flatten hide** IdentifiedElement
  - IdentifiedElement has been removed
  - Identification has been added to Person, Vehicle and PoliceInsurance

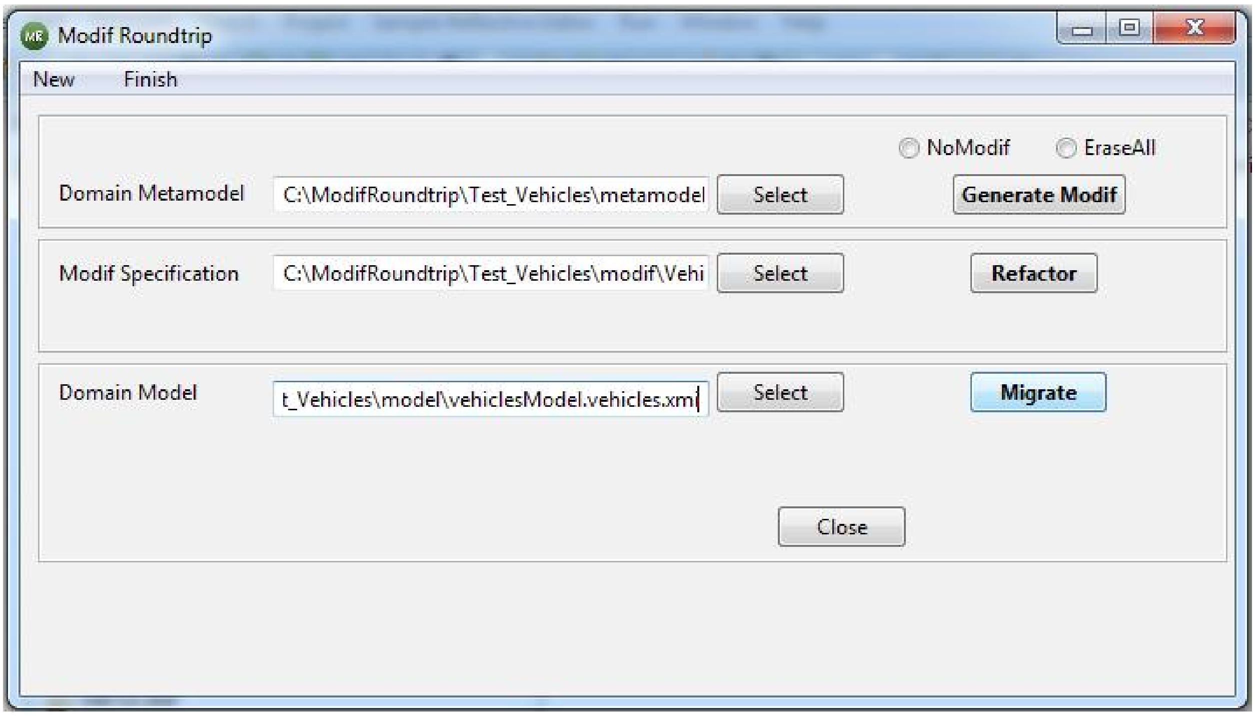
Congratulations

You have used Modif tools to generate a new ecore model by application of Modif operators on initial ecore model

2. Migration

Specify Domain Model and Migrate

- Click on *Select* for Domain Model and specify the path to the model vehiclesModel.vehicles.xmi
- Click on the **Migrate** button



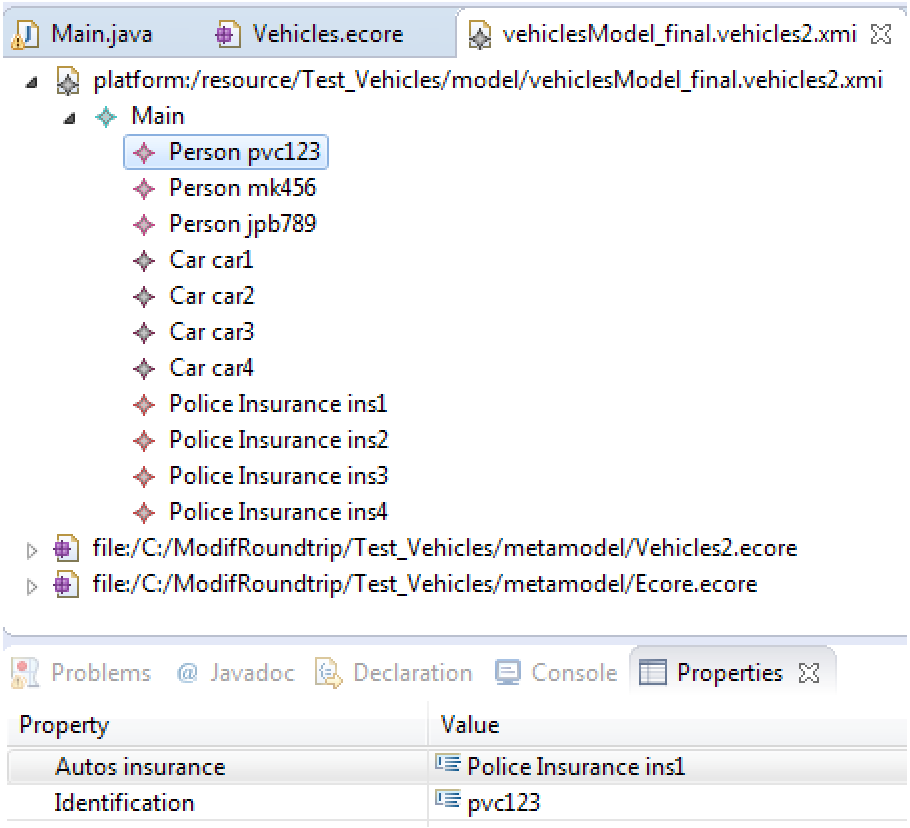
○ NoModif ○ EraseAll

Domain Metamodel C:\ModifRoundtrip\Test_Vehicles\metamodel

Modif Specification C:\ModifRoundtrip\Test_Vehicles\modif\Vehi

Domain Model t_Vehicles\model\vehiclesModel.vehicles.xmi

- Refresh the model folder. The model *vehiclesModel_final.vehicles2.xmi* will appear. It should looks as follows:



```
platform:/resource/Test_Vehicles/model/vehiclesModel_final.vehicles2.xmi
  Main
    Person pvc123
    Person mk456
    Person jpb789
    Car car1
    Car car2
    Car car3
    Car car4
    Police Insurance ins1
    Police Insurance ins2
    Police Insurance ins3
    Police Insurance ins4
file:/C:/ModifRoundtrip/Test_Vehicles/metamodel/Vehicles2.ecore
file:/C:/ModifRoundtrip/Test_Vehicles/metamodel/Ecore.ecore
```

Property	Value
Autos insurance	Police Insurance ins1
Identification	pvc123

- pvc123 is associated to ins1
- mk456 is associated to ins2
- jpb789 is associated to ins3 and inst4

Notice that Person hasn't Cars. Cars has not associated PoliceInsurance. Person has associated PoliceInsurances.

Enjoy Modif

3. Contact

You can report bugs to :

- **babau@univ-brest.fr**
- **vallejoco@univ-brest.fr**