Modif Documentation

- Refactoring User Guide

This document explains the steps for using ModifRoundtrip for a Metamodel Refactoring

June, 2015

by Paola Vallejo, Jean Philippe Babau
# Table of contents

1. Create project, folders and ecore metamodel ................................................................. 3  
   1.1. Create a project ............................................................................................................. 3  
   1.2. Add Xtext nature ....................................................................................................... 3  
   1.3. Add Folders .............................................................................................................. 3  
   1.4. Create metamodel .................................................................................................... 3  

2. Execute Ecore Modif ....................................................................................................... 5  
   1. Refactoring .................................................................................................................. 5  
      Specify Domain Metamodel and Generate Modif model ........................................... 6  
      Edit Modif Model ....................................................................................................... 7  
      Refactor ...................................................................................................................... 8  

3. Contact ......................................................................................................................... 10
1. Create project, folders and.ecore metamodel

1.1. Create a project

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

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

1.2. Add Xtext nature

Add the Xtext Nature to the Test project

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

1.3. Add Folders

Keep the model folder and add two additional folders:

Right click ModifTest → New → Folder → Specify the folder_name → Finish

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_Vehicle/metamodel → New → Other... → Eclipse Modeling Framework / Ecore Model → Next → [give a name for your.ecore] → Finish
Then set the ecore model properties as follows:

Add the EClasses to the metamodel.

Right click on `vehicles` package → New Child → EClass

Add the EReferences and EAttributes to the metamodel.

Right click on EClass → New Child → [Select EReference or EAttribute]

It must look as follows:
In this metamodel there is a **Root** EClass composed of **Persons** (*person* EReference), **Vehicles** (*vehicles* EReference) and **PoliceInsurances** (*insurance* EReference). Person, Vehicle and PoliceInsurance have an identification (*identification* EAttribute). A Person can have some Vehicles. Each Vehicle has an associated 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

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 Refactoring.

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_Vehicle/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.
• Refresh the modif folder, the modif model appears.

Edit Modif Model

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

```java
root vehicles to vehicles2
Prefix vehicles to vehicles2
URI "http://mde.ubo.vehicles" to "http://mde.ubo.vehicles"

class {
  Root to Root {
    ref person to person bounds (0,-1) to (0,-1)
    ref vehicles to vehicles bounds (0,-1) to (0,-1)
    ref insurance to insurance bounds (0,-1) to (0,-1)
  }
  Person to Person {
    ref vehicle to vehicle bounds (0,-1) to (0,-1)
  }
  PoliceInsurance to PoliceInsurance;
  Vehicle to Vehicle;
  EngineVehicle to EngineVehicle {
    ref insurance to insurance bounds (0,1) to (0,1)
  }
  Car to Car;
  IdentifiedElement to IdentifiedElement {
    att identification to identification bounds (0,1) to (0,1)
  }
}
```

Edit the modif file in this way:

• For URI, rename it to `http://mde.ubo.vehicles2`
• For class `Root`
  ○ For `person` reference, rename it to `people`
  ○ For `insurance` reference, rename it to `insurances`
• For class `EngineVehicle`, put the key words `hide` and `flatten`
• For class `IdentifiedElement`, put the key words `hide` and `flatten`
Finally, save the model, it will look as shown below

```java
root vehicles to vehicles2
Prefix vehicles to vehicles2
URI "http://mde.ubo.vehicles" to "http://mde.ubo.vehicles2"

class {
  Root to Root {
    ref person to people bounds (0,1) to (0,1)
    ref vehicles to vehicles bounds (0,1) to (0,1)
    ref insurance to insurances bounds (0,1) to (0,1)
  };

  Person to Person {
    ref vehicle to vehicle bounds (0,1) to (0,1)
  };

  PoliceInsurance to PoliceInsurance;
  Vehicle to Vehicle;

  flatten hide EngineVehicle to EngineVehicle {
    ref insurance to insurances bounds (0,1) to (0,1)
  };

  Car to Car;

  flatten hide IdentifiedElement to IdentifiedElement {
    att identification to identification bounds (0,1) to (0,1)
  }
}
```

**Refactor**

- In the ModifRoundtrip form, set the *Modif Specification* Vehicle-to-Vehicles2.modif
- Click on the *Refactor* button
• Refresh the Test_Vehicle/metamodel folder and you will notice that the file vehicle2.ecore is added. It will look as shown below:

Graphically it looks as follows:

Notice the following:

• The reference person has been remaned
• The reference insurance has been renamed
• The class IdentifiedElement disappeared and its attribute identification now exists in Person, Vehicle and PoliceInsurance
• The class VehicleInsurance no longer exists, but there is a new reference from Person to PoliceInsurance, and Car inherits from Vehicle
Congratulations

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

Enjoy Modif

3. Contact

You can report bugs to :

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