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 .................................................. 3  
   1.1. Create a project ......................................................................................... 3  
   1.2. Add Xtext nature ....................................................................................... 3  
   1.3. Add Folders ............................................................................................... 3  
   1.4. Create metamodel ..................................................................................... 3  
   1.5. Create model ............................................................................................ 5  

2. Execute Ecore Modif ....................................................................................... 8  
   1. Refactoring .................................................................................................... 8  
      Specify Domain Metamodel and Generate Modif model ................................ 8  
      Edit Modif Model ......................................................................................... 9  
      Refactor ....................................................................................................... 11  
   2. Migration .................................................................................................... 13  
      Specify Domain Model and Migrate ............................................................. 13  

3. Contact ......................................................................................................... 14
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 → New → Other... → Eclipse Modeling Framework/Empty EMF Project → Next → Specify the project name [Test_Vehicles] → 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 three additional folders:

- metamodel
- migration
- modif

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

```
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:
```

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:

```

Modif Documentation- Simple Migration User Guide  4/14
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*.

### 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] → Create Dynamic Instance
Add the elements to the model

Right click on EClass → New Child → [Select element to create]

Fill the properties of the element

The model should looks as follows:

- 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 opened 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
• Refresh the Test_Vehicles project, the modif model appears in the modif folder

![Diagram showing the directory structure including Test_Vehicles, src, JRE System Library, 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 Vehicles-to-Vehicles2.modif. Then, open the Vehicles-to-Vehicles2.modif using the text editor or the xText editor as shown below:
Edit the modif file in this way:

- For **root**, rename it to **vehicles2K** (do not forget to put the **K** at the end)
- For **Prefix**, rename it to **vehicles2K** (do not forget to put the **K** at the end)
- For **URI**, change "VehiclesK2.ecore" to "Vehicles2K.ecore"
- **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
Refactor

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

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

![Modif Roundtrip](image-url)
• Refresh the model folder. The model `vehiclesModel_final.vehicles2.xmi` will appear. It should looks as follows:

![Diagram of the model]

- `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`