

INRIA & LINA - Obasco group	AMW Use case Migrating Person model	Kelly Garcés kelly.garces@emn.fr
	User Guide	

Introduction

This guide illustrates how to generate migration transformations by using matching transformations and HOT's provided by AMW. The process is illustrated with a very simple example; it migrates a model, conforming to metamodel Persons, to another model, conforming to metamodel PersonsNew.

Requirements

This use case requires a set of the matching transformations and HOT's provided by AMW. Additionally, it is necessary plugging other matching transformations. In order to enable these transformations, we require Eclipse 3.2.2 + EMF 2.2.2. We should download ATL, AMW and ATL, AMW and AM3 from the sources. The following links provide the installation instructions:

ATL: http://wiki.eclipse.org/index.php/ATL/How_Install_ATL_From_CVS/

AM3: http://wiki.eclipse.org/AM3/How_Install_AM3_From_CVS

AMW: <http://www.eclipse.org/gmt/amw/download/>

When we finish the installation, check if all the projects required by ATL, AM3 and AMW are in the workspace.

Now, we need to modify org.eclipse.gmt.weaver.amw4atl project:

- Download the simmetrics API (from UK Sheffield University) available at <http://sourceforge.net/projects/simmetrics/>. The jar file must be placed in the lib folder of the project and renamed to simmetrics.jar.
- Add simmetrics.jar to the java build path. Right click on the projet > Properties > Java Build Path, select the tab > Libraries > Add JARs, later select simmetrics.jar
- Edit plugin.xml. Select the tab Runtime > add uk.ac.shef.wit.simmetrics.similaritymetrics to Exported Packages > add lib/simmetrics.jar to Classpath. Later, select the tab Build > select the folder lib in the Binary Build.
- Edit ASMAMWModel.java. Uncomment the line

```
addVMOperation(ASMString.myType, toVMOperation(ASMAMWModelElement.class, "getStringSim"));
```

- Edit ASMAMWModelElement.java. Uncomment the method

```
public static ASMReal getStringSim(StackFrame frame, ASMString left, ASMString right)
```

Next, we have to plug 3 matching transformation to AMW. These transformations are already provided by the library matching transformations. We have modified them to obtain the expected results.

- Go to `org.eclipse.gmt.weaver.transformation.extension` project.
- Copy `AMWtoATL_Match.asm`, `Assign_NameSimbyType.asm`, `CreatePropagationbyType.asm` (zipped into the use case example) into transformations folder.
- Edit `plugin.xml`. Select the tab `plugin.xml`, delete the following lines:

```
<!--  
<matching transformation="transformations/Assign_NameSimbyType.asm" description="Name similarity" file_suffix="name">  
  <binding weavingReference="leftM" header="left"/>  
  <binding weavingReference="rightM" header="right"/>  
</matching-->
```

Remember adding these lines:

```
<matching transformation="transformations/Assign_NameSimbyType.asm" description="Name similarity" file_suffix="name"  
category="Metamodel match">  
  <binding weavingReference="leftM" header="left" metamodelHeader="MOF"/>  
  <binding weavingReference="rightM" header="right" metamodelHeader="MOF"/>  
</matching>
```

Finally, we can launch the Run-time workbench. Click on `Run > Run as > Run-time workbench`. Go to the new eclipse.

We import the `Persons2PersonsNew` project (zipped into the use case example) into the current workspace. The ATL project structure should look like this:

Persons2PersonsNew/

- **match:** contains the weaving models created during the matching process and the generated transformation.
- **metamodels:** contains `Persons` and `PersonsNew` metamodels
- **models:** contains the models to verify the generated transformation

Execute the Use Case

Now we explain the process to generate the transformation. The following steps are necessary:

1. Execute the matching transformations.
2. Execute the HOT.

1. Execute the matching transformations

The first step is to create a weaving model. This weaving model contains references to the metamodels.

We use the Weaving Model Wizard to create the weaving model. First, we select the weaving metamodels (mmw_match.km3, mmw_propagation.km3, mmw_compare.km3 and mw_base_extension.km3). See figure 1.

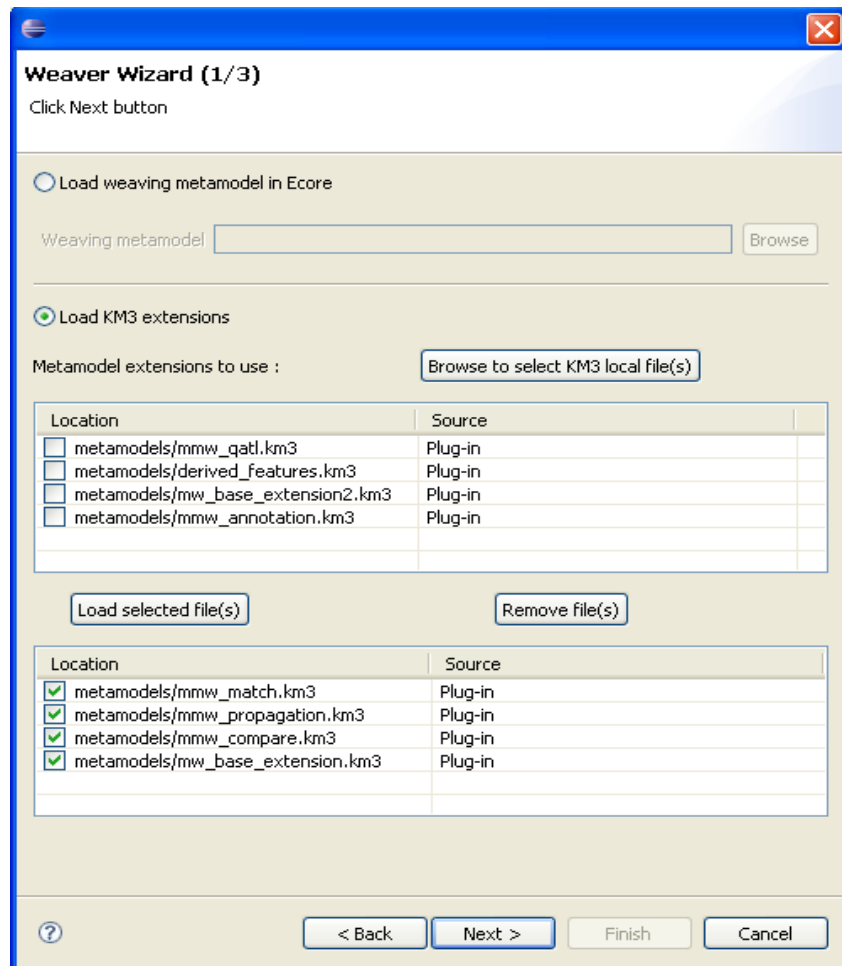


Figure 1. Selecting weaving metamodels

Later, we enter a name for the weaving model, for example Persons_Comp.amw. We select TransformationWeavingPanel and MatchModel (see figure 2).

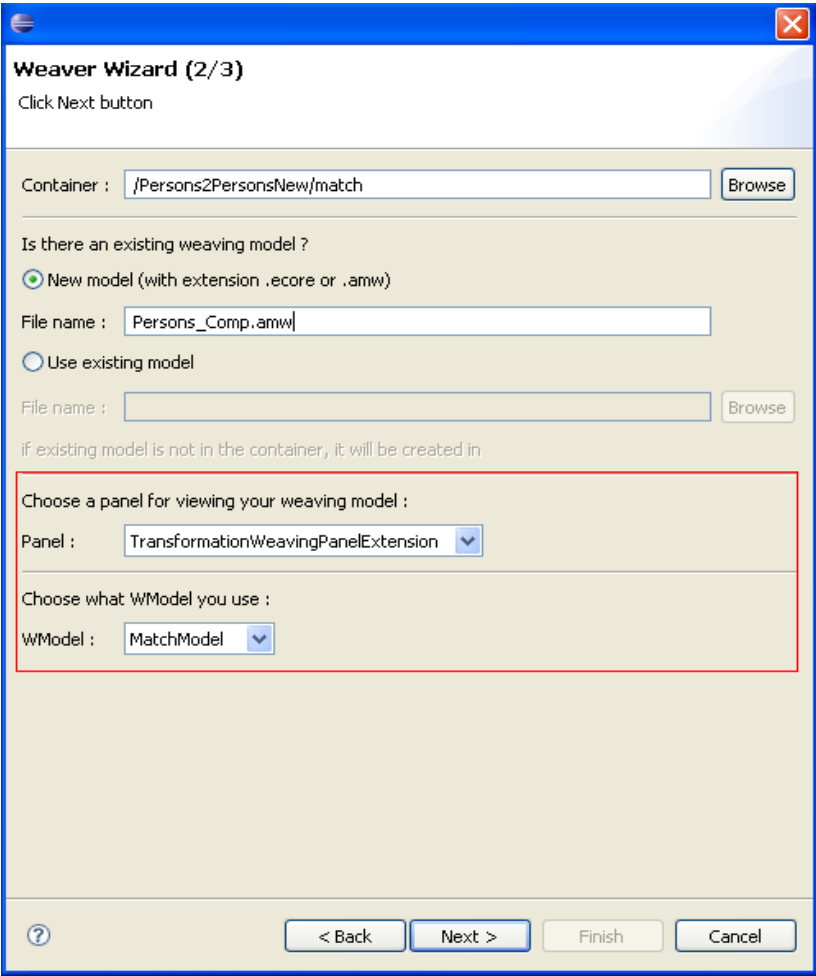


Figure 2. Selecting special parameters

Finally, we select the left (Persons.ecore) and right (PersonsNew.ecore) metamodel.

Subsequently, we execute six matching transformation (Cartesian product, Name similarity, Propagation, Similarity flooding, Threshold and Link rewriting) by following the order showed in the figure 3.

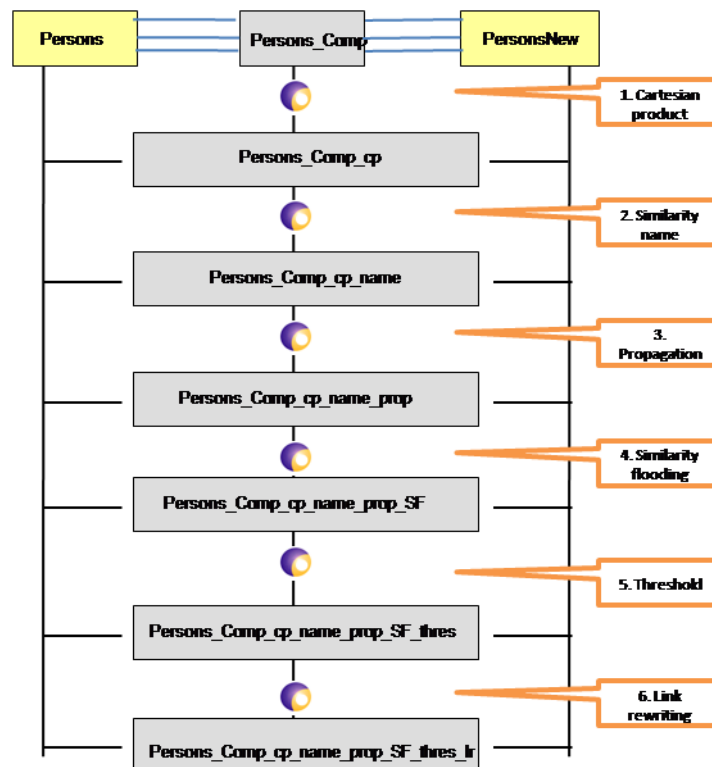


Figure. 3. Matching transformation execution

The transformations can be executed on the weaving panel. For launching a matching transformation, we should:

- Open the previous weaving model
- Display the contextual menu on the middle panel
- Select "Match" item and the required matching transformation

For example, for launching the first matching transformation, we open Persons_Comp.amw, we select Metamodel Match by right clicking on the middle panel. We select Cartesian product item (see figure 4).

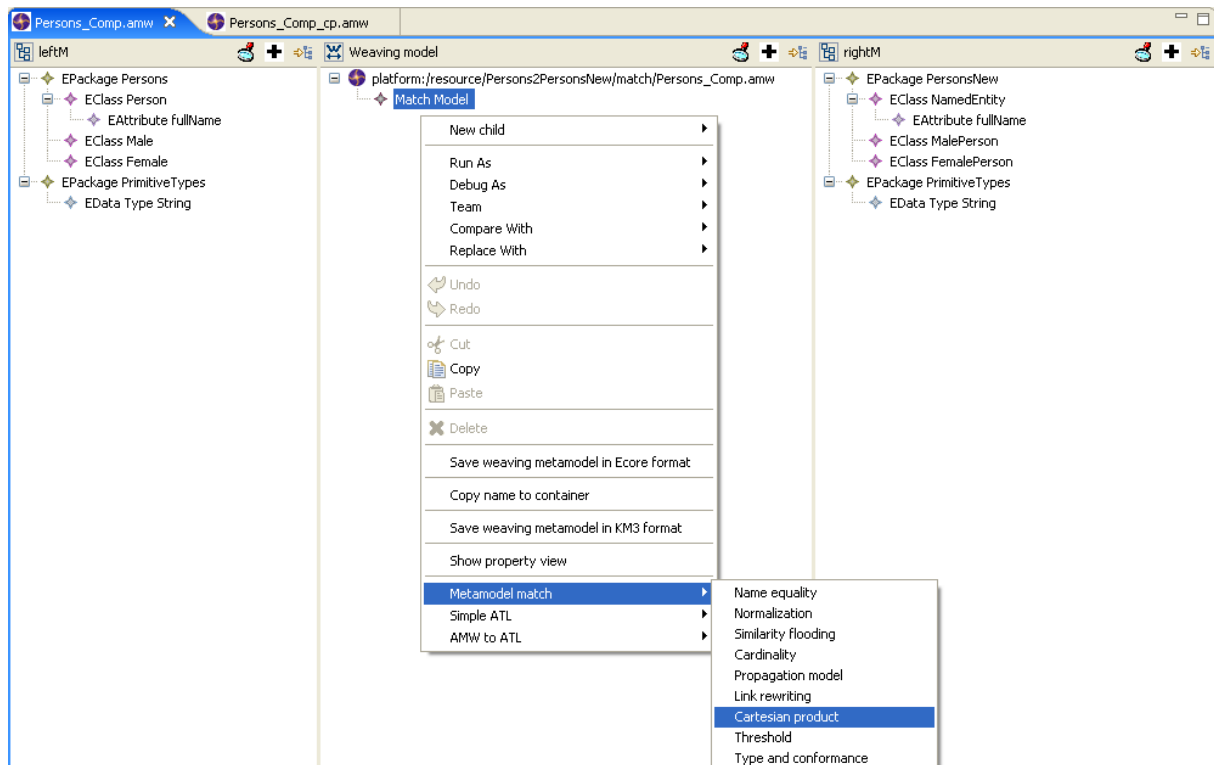


Figure 4. Launching Cartesian product

Note a new weaving model is created (Persons_Comp_cp.amw). Its name consists of the previous weaving model name plus the suffix "cp". Every match transformation adds a predefined suffix to avoid overriding the previous weaving model.

The Threshold transformation has a special execution. We have to assign a threshold before launching. In order that, we open Persons_Comp_cp_name_prop_SF.amw, we select configure transformation item on the middle panel (see figure 5).

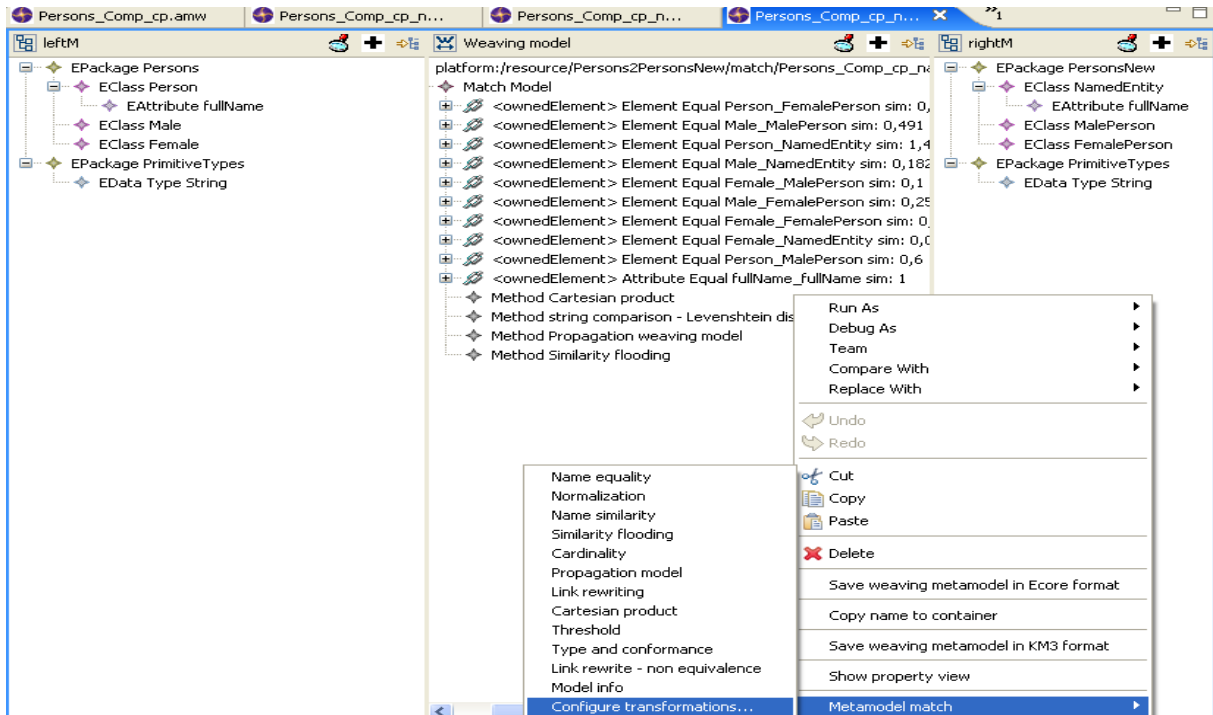


Figure 5. Configuring matching transformation

The matching configuration window is displayed. We assign 0.4 to the threshold and select the checkbox Execute (see figure 6). Finally, we press the button Ok.

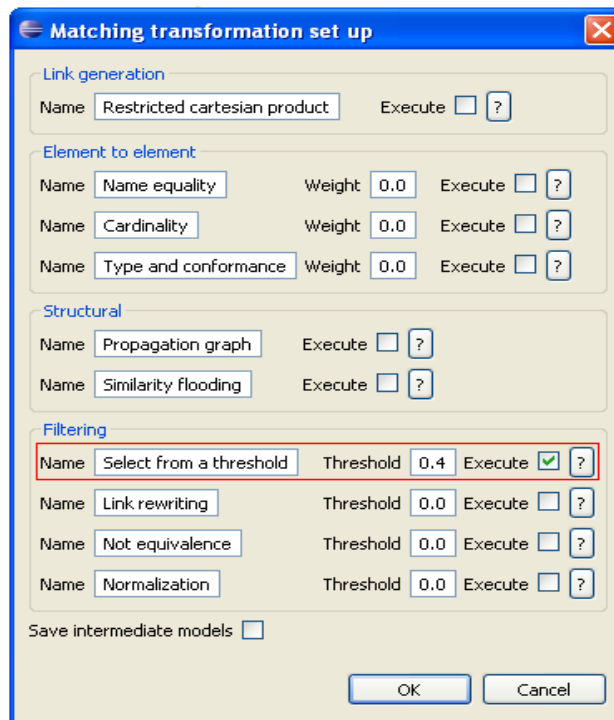


Figure 6. Selecting a threshold

When we finish the matching transformation execution, we get Persons_Comp_cp_name_prop_SF_thres_lr.amw. It contains the more accurate links to generate the transformation. From this weaving model we execute the HOT.

2. Execute the HOT

The HOT can be also executed on the weaving panel. For launching the HOT, we should:

- Open Persons_Comp_cp_name_prop_SF_thres_lr.amw
- Display the contextual menu on the middle panel
- Select "AMW to ATL" item and Generate transformation (HOT), see figure 7.

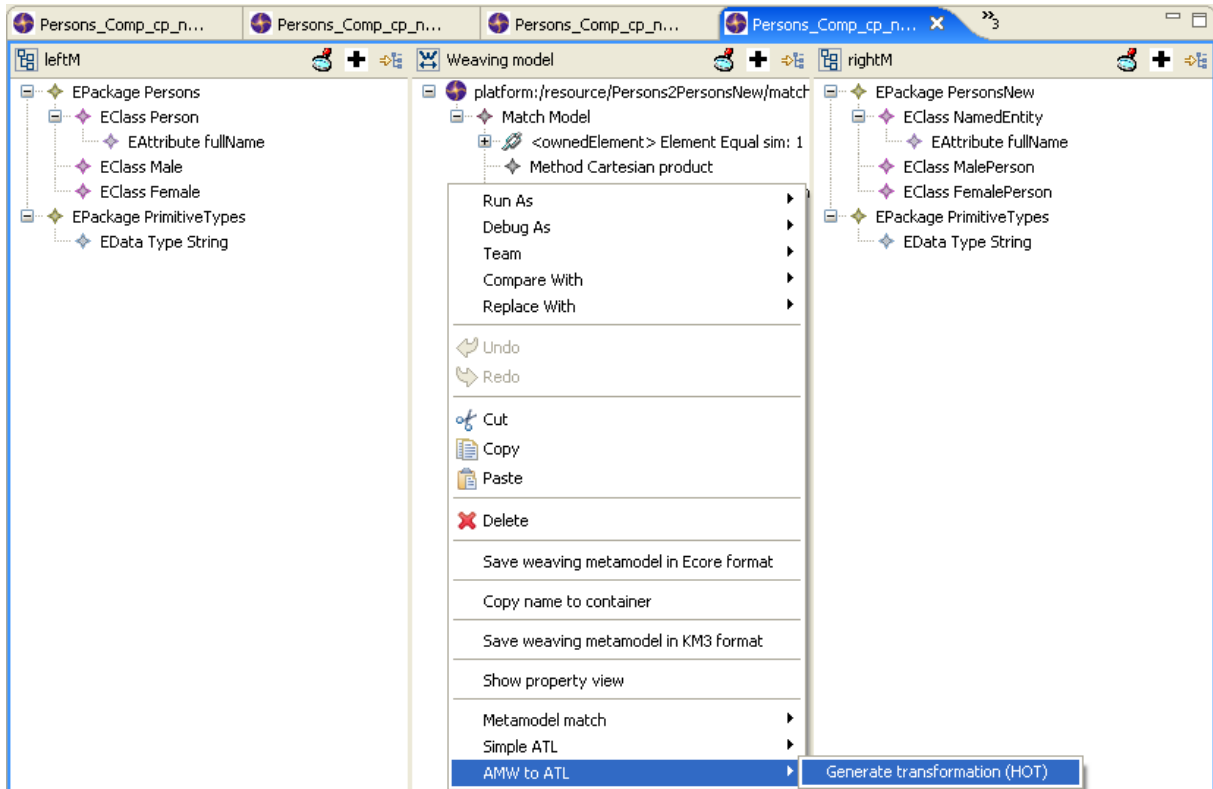


Figure 7. Launching the HOT

We get a model transformation named Persons_Comp_cp_name_prop_SF_thres_lr_hot.ecore. In fact, we can use the AM3 extractor to obtain the ATL transformation (see figure 8).

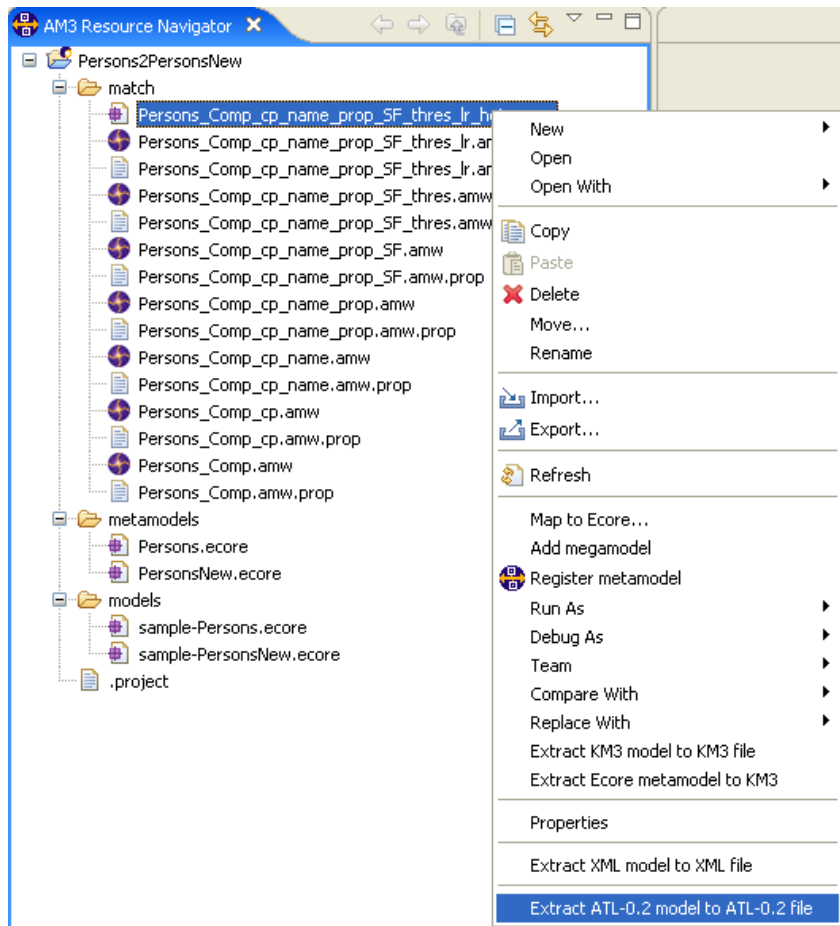


Figure 8. Getting the ATL transformation

Now, we can execute the generated transformation to migrate the model conforming to Persons (sample-Persons.ecore) to the model conforming to Persons New (sample-PersonsNew.ecore). We can use the provided launch file (Persons2PersonsNew).