

## 3. Exercises on Class Modeling

### Objectives

- To install the modeling environment (Eclipse Modeling Distribution)
- To learn basic modeling tools of EMF
- To operationally recall modeling using class diagrams

I estimate that you have about 2 hours in class to complete this task + about 4 hours of self study time at home. One possible reference on class modeling is [9], but many other sources are available, also online.

**Task 1.** Install Eclipse modeling distribution following the tutorial on the course website. The tools should be available in the lab, but we strongly recommend using your own laptop for the task. If you already have Eclipse installed, we recommend making a clean install on the side. Several installations of Eclipse easily co-exist on the same PC.

**Task 2.** Read the class modeling tutorial in the bottom of this exercise set. Then proceed to the next task.

**Task 3.\*** Create simple class diagram using the tree editor of Eclipse following this description of a domain:

*A family consists of persons. Each person may be married to another person. The relation of being married is symmetric (if I am married to you, you are married to me). Each person may have a parent, and each parent may have multiple children. ~~Again a symmetric relation.~~ Each person has a name, age and a CPR number. Each person may be enrolled in a university. University may own one or more study programs.*

Create a valid instance of your diagram representing Bob married to Alice, with their son Sam enrolled in the SDT programme of IT University.

The tree editor is much more robust and solid, but also try to initialize a diagram file and create a diagram in UML Class Diagram visual syntax.

Print your tree view, the diagram view, and a screenshot of your instance and hand-in to the teaching assistant as homework.

**Due by:** beginning of exercise session next week. Expected size: 1 page, no text.

### 3.1. A Short Tutorial on EMF

This tutorial is adapted after Vogel's tutorial <sup>1</sup>.

Our objective is to show how to create a simple class model with EMF. In Eclipse, every file belongs to a project. So we need to first create a project. Moreover the project needs to have the right nature, to enable availability of tools (here this has to be an EMG project).

Create a new project called `dk.itu.smdp.tutorial` via `File → New → Other → Eclipse Modeling Framework → Empty EMF Project`. This can be seen in Fig. 3.1

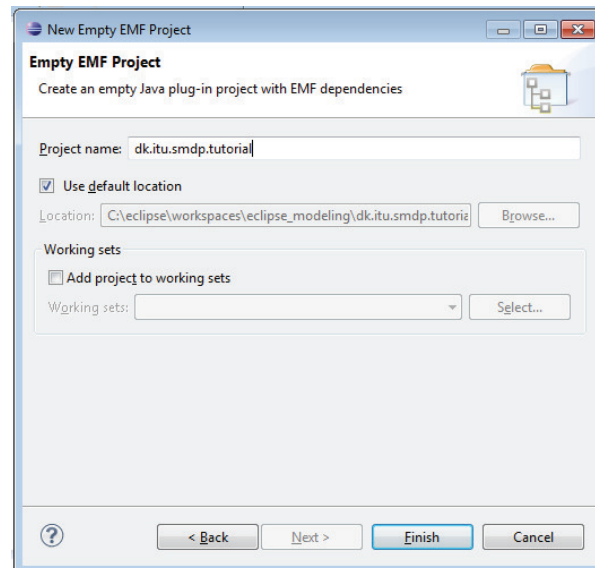


Figure 3.1.: Creating an EMF project

A model needs to be placed in a file, a model file. EMF distinguishes between model files (abstract syntax) and diagram files (concrete syntax). We can create both with one operation in the user interface.

Select the model folder, right-click on it and select `New → Other → Ecore Tools → Ecore Diagram` (c.f. Fig. 3.2).

Choose to create a new model. Enter `WebApp.ecore` as the Domain File Name parameter (Fig. 3.3) (domain model is the wording for "class diagram" here, this file will store the abstract syntax, while the diagram file will store the concrete syntax). This should open a visual editor for creating EMF models. You can see that in the models folder (in the Package Explorer) two files were created `WebApp.ecore` and `WebApp.ecorediag`.

Open the Properties view via the menu `Window → Show View → Other... → General → Properties`. This view allows you to modify the attributes of your model elements (Fig. 3.4). At this point it is also useful to mention the quick access feature of Eclipse. Almost any deeply

<sup>1</sup><http://www.vogella.com/tutorials/EclipseEMF/article.html>

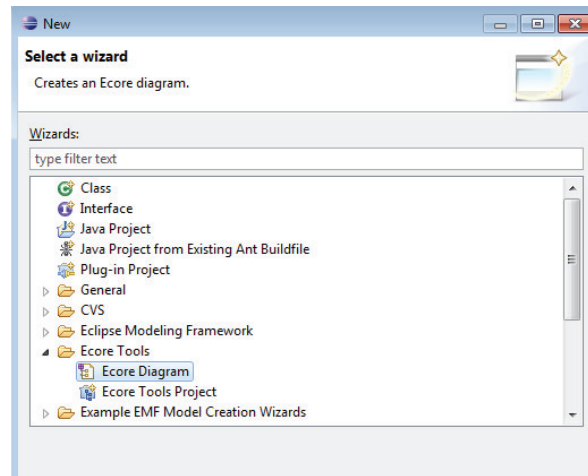


Figure 3.2.: Create Ecore diagram

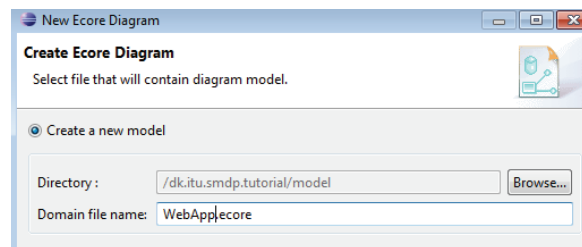


Figure 3.3.: Domain File Name

nested property or menu item, can be accessed quickly, if you remember its name. Press Ctrl+3 and type "properties", then select from the menu. You will open the same view as the previous long menu selection. This way a lot of complex operations can be speeded up.

**Classes.** In order to create a new class, select Eclass from the tool palette on the left of the editor, and click anywhere on the canvas to create a class. Create classes for Website, Webpage, Category and Article (see Fig. 3.5).

**Class Attributes.** Use the EAttribute tool from the palette to assign to each class an attribute *name* (see Fig. 3.6).

The name attribute should be of type String. Go to EType in the properties window and press in the white space or the icon on the right. The Object selection should open (see Fig. 3.7). Select EString and press OK. When you do this again for the other classes, EString will be in the top of the list. Add *title*, *description*, *keywords* to Website and Webpage, and set them as EString as well.

You should have a diagram the same as the one in Fig. 3.8

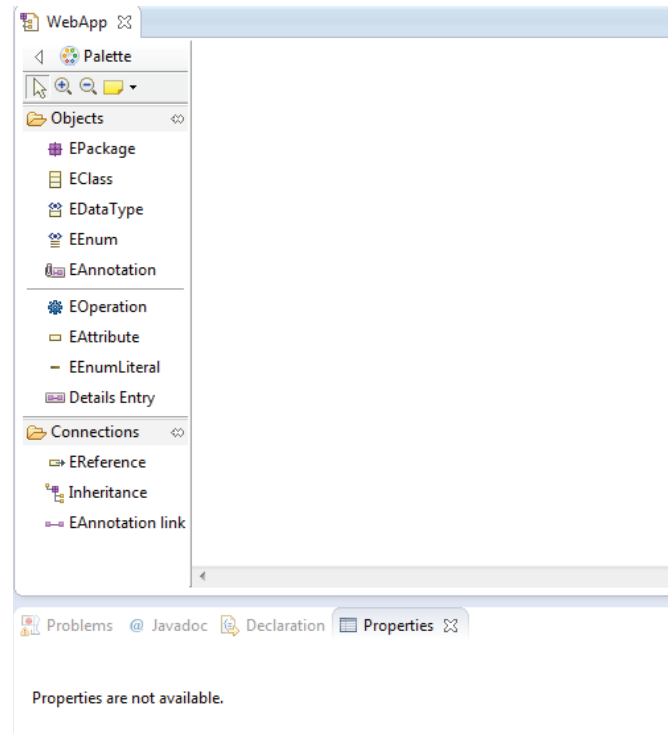


Figure 3.4.: View of the editor, with properties view in the bottom

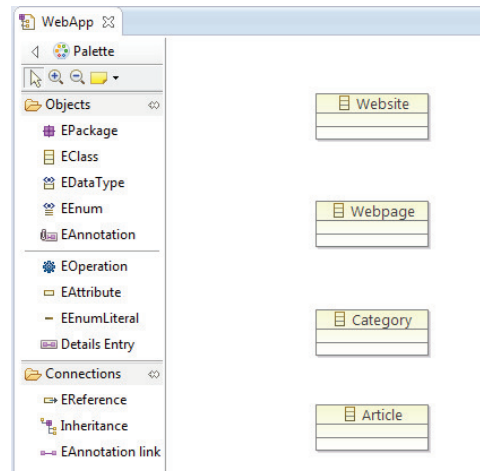


Figure 3.5.: An Ecore Diagram with 4 classes

**References.** Select EReference and create an arrow similar to the following picture. You first press on the Website class and drag it to the Webpage class. Name the reference as *pages*. Make sure the upper bound is set to "\*" and that the "Is Containment" property is enabled (Fig. 3.9).

Create similar references from Webpage to Category and from Category to Article. In the end

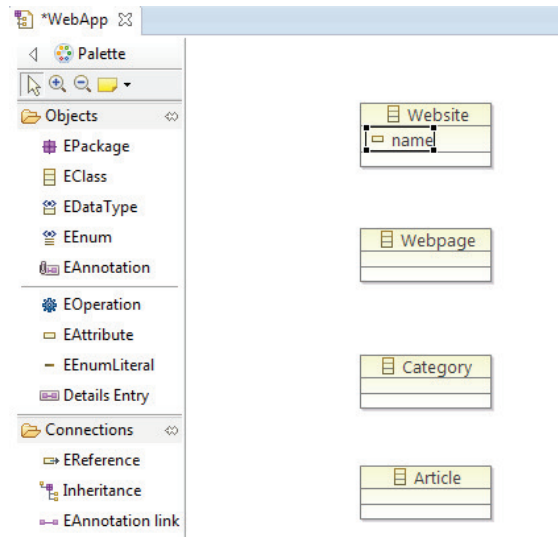


Figure 3.6.: Adding attributes with EAttribute

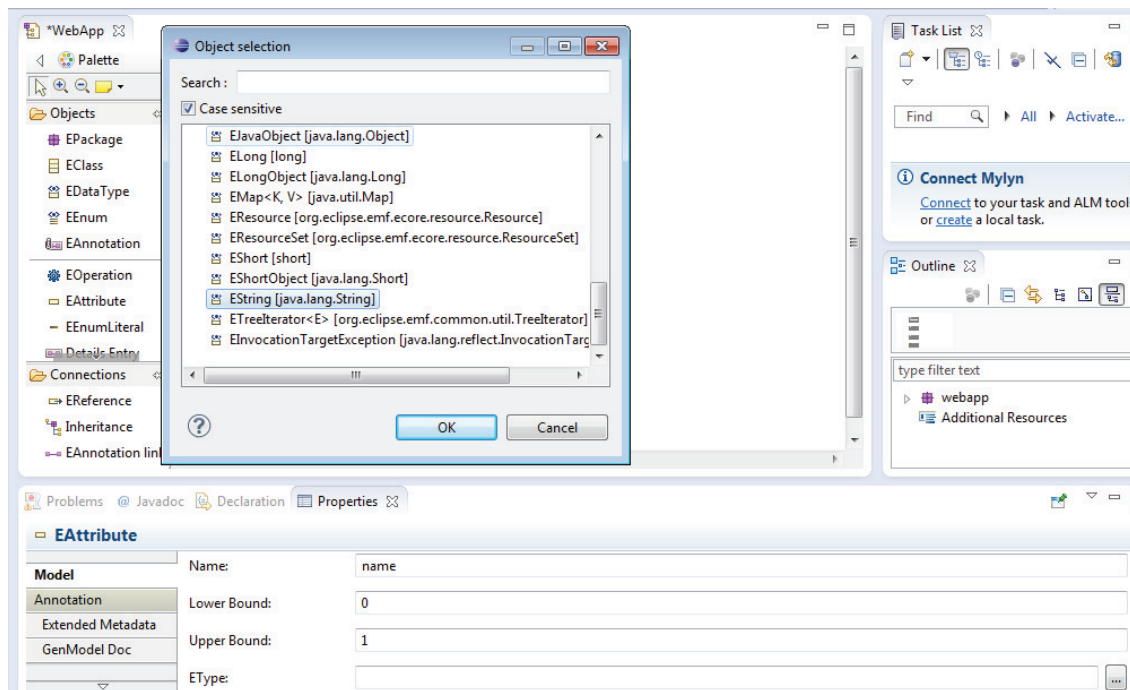


Figure 3.7.: EString type

your diagram should look like the one in Fig. 3.10

**Tree view of the .ecore file.** Close the diagram and open the WebApp.ecore file. The result should look like in Fig. 3.11. This view shows the actual model, so the abstract syntax of the diagram.

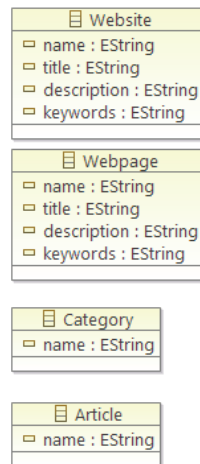


Figure 3.8.: Complete Diagram

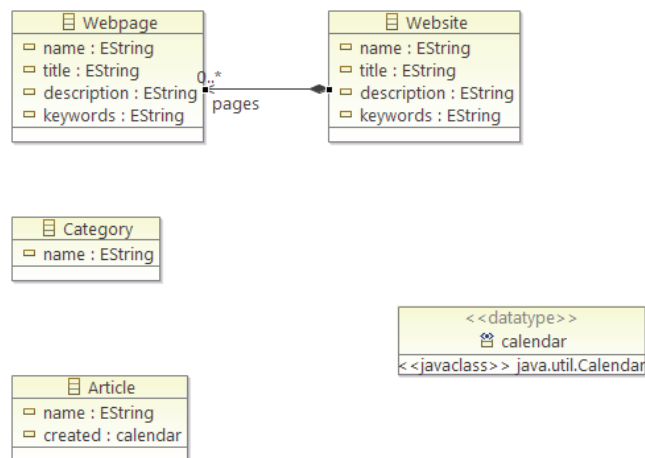


Figure 3.9.: EReferences

Now, in Eclipse, the diagram editor is rather clunky (we use it, since bringing another tool, would further complicate the course). So the recommend way to edit your models is actually to edit the tree view representation of the model in the .ecore file. The .ecorediag file only contains the presentation layer, and it happens once in a while that it gets out of sync with the .ecore file. Then you need to recreate it (loosing all layout information, which can be quite irritating). This is why it is useful to learn to manipulate the tree view editor directly, by passing the class diagram view.

**Create EMF Generator Model.** Right-click your WebApp.ecore file and select New → Other → Eclipse Modeling Framework → EMF Generator model. Create the WebApp.genmodel file based on your Ecore model (see Fig. 3.12). The generator model configures the code generator. We are usually happy with the default settings, so we just create it and never

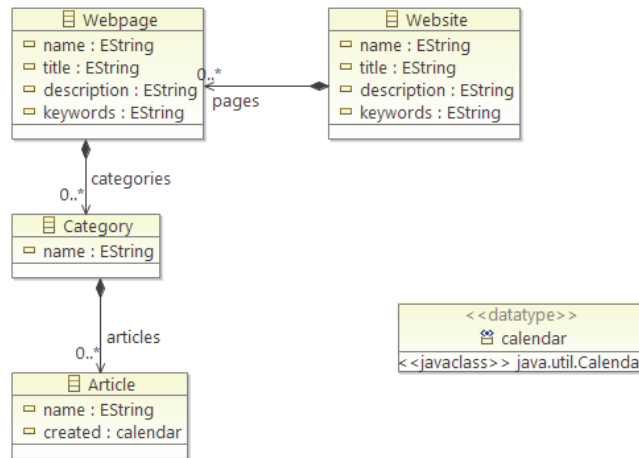


Figure 3.10.: Diagram with references

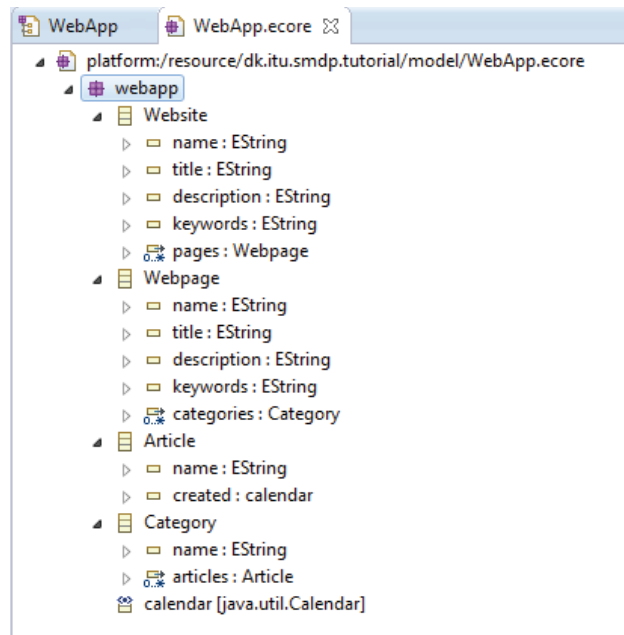


Figure 3.11.: Ecore diagram

modify it.

Select Ecore model.(see Fig. 3.13)

Select your model, press Load, then Next, and Finish.(see Fig. 3.14)

Now it should look like in Fig. 3.15. Set the base package to be `dk.itu.smdp.tutorial.WebApp.model`, and the prefix to be `MyWebApp`.

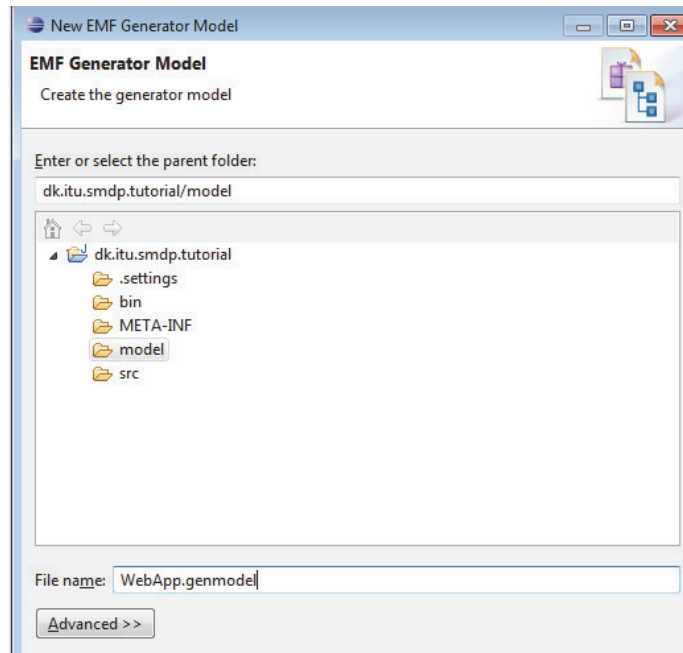


Figure 3.12.: Generator Model

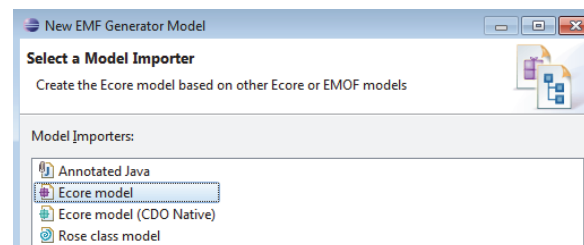


Figure 3.13.: Select Ecore model

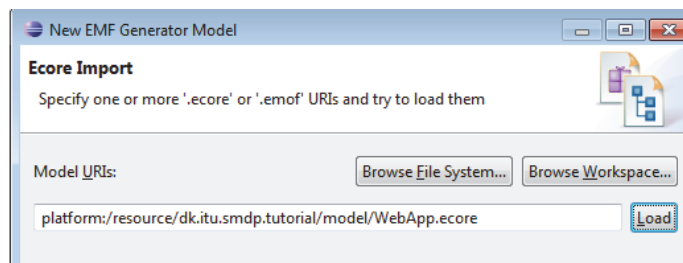


Figure 3.14.: Generator model

**Creating a Dynamic Instance** Up to now we have created a class diagram. Next we will create an instance of this class diagram. Open the WebApp.ecore from the package explorer. In the editor expand the blocks twice. Then right click on the Website and select *Create Dynamic Instance*. Then press Finish (see Fig. 3.16).



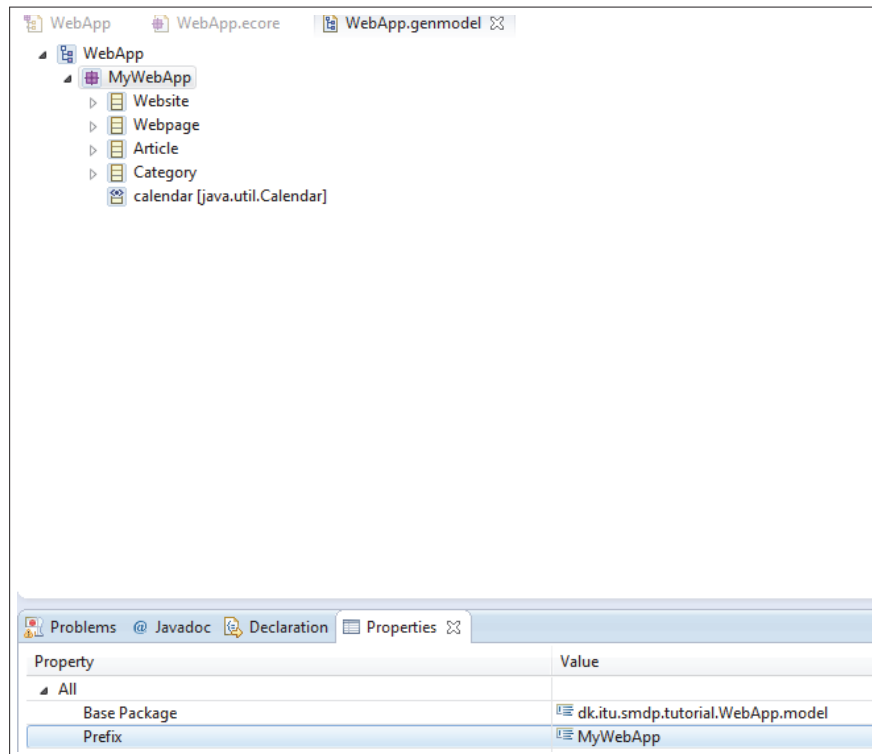


Figure 3.15.: Configuring the genmodel.

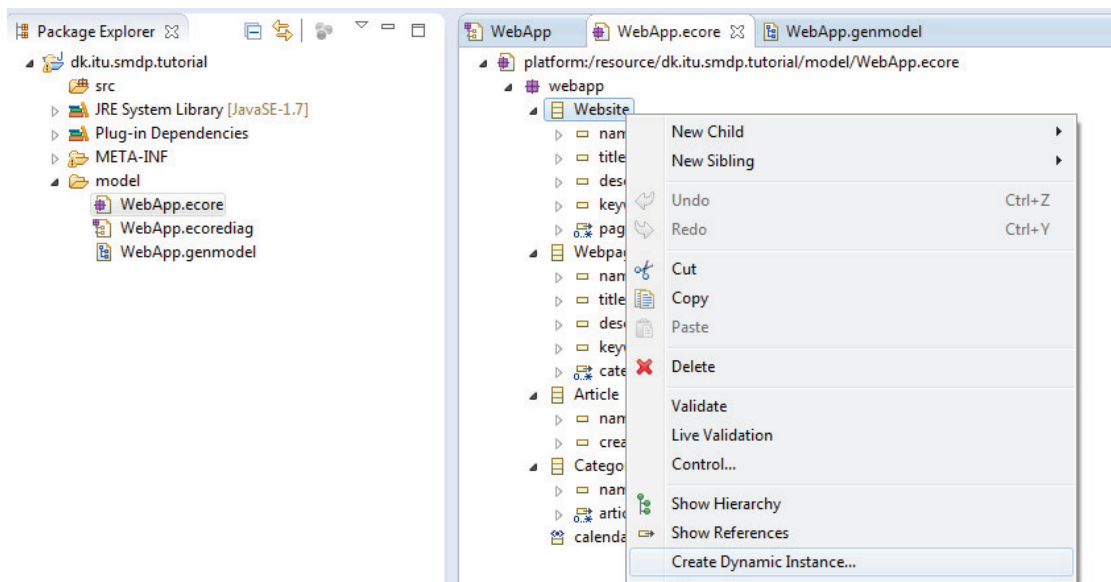


Figure 3.16.: Create Dynamic Instance

Open the Website.xmi file. In the editor expand the platform, select Website and enter values to the attributes of the class in the Properties window.(see Fig. 3.17)

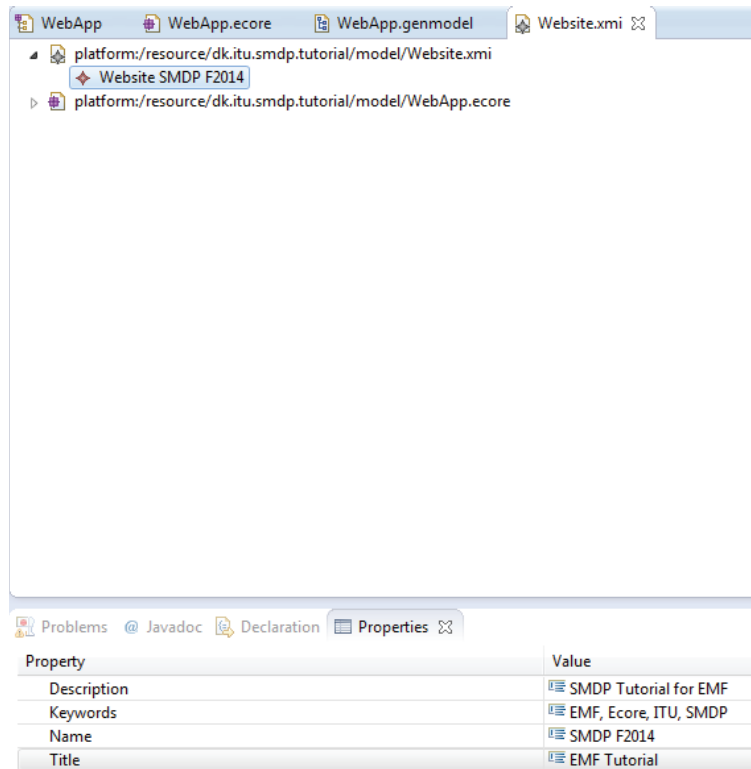


Figure 3.17.: Website.xmi

Right click on the Website class and create a child. Then create a child for Webpage. Then create a child for Category. Add values to the attributes of the classes (see Fig. 3.18).

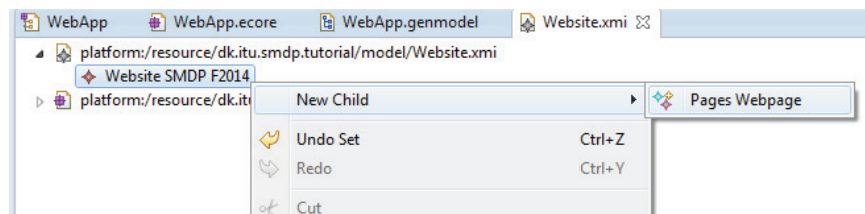


Figure 3.18.: Attribute values