

# Assignment Sheet 9

Due: Tuesday, January 10, 2012, 23:59

For help, contact [atsc-lecture@lists.iai.uni-bonn.de](mailto:atsc-lecture@lists.iai.uni-bonn.de) (staff and participants).

Submit your solution into your group's SVN under the folder "assignment09".  
The solutions (if not otherwise mentioned) must be submitted as **PDF files**.

## Task 1. *UML Profiles and MOF based Meta Models* (1 Points)

Discuss and present the main advantages and disadvantages of UML Profiles in comparison to MOF-based Meta-models.

## Task 2. *Create a UML Profile for a Component Model* (3 Points)

For this task the following description is provided:

*Components have ports, which have required and provided interfaces. An interface has a name and contains at least one method. An interface may be used in several components.*

Your task is to define a meta-model with stereotypes for the elements. For the definition of the profile you can use a UML2 tool of your choice or use a simple class diagram where you use inheritance instead of the *extension* arrows (see slide 15 of the first lecture).

Create an exemplary class diagram and annotate the model elements accordingly.

*In this assignment sheet we want to create an Ecore meta-model and domain specific language (DSL) editor with the help of the Xtext framework. In the end it should be possible to define a software architecture based on components.*

### **Task 3.** *Prerequisite: Installing EMF (0 Points)*

Installation of Eclipse Modeling Framework (EMF) can be done in Eclipse via the Eclipse Update Manager. In this dialog select “Modeling” and install the “EMF – Eclipse Modeling Framework SDK” as well as the “Ecore Tools”.

A good tutorial on using the EMF framework for modeling is written by Lars Vogel:

<http://www.vogella.de/articles/EclipseEMF/article.html>

To get acquainted with Xtext, read the following tutorial:

[http://www.eclipse.org/Xtext/documentation/2\\_1\\_0/020-domainmodel-step-by-step.php](http://www.eclipse.org/Xtext/documentation/2_1_0/020-domainmodel-step-by-step.php)

### **Task 4.** *Define an Ecore Meta-model with EMF (3 Points)*

Your task is to build a meta-model for the component model (described in Task 2) based on Ecore.

As a starting point we provided an Xtext project; it is available on the Assignments web site and is named “Assignment9Task4.zip”.

Import the two project into your Eclipse workspace.

All relevant files are stored in the directory:

**/org.cs3.atsc.compdsl/src/org/cs3/atsc/compdsl/**

You have two options to create the Ecore meta-model for the described component model. You can either edit the Ecore **components.ecore** file directly, or use the graphical *Ecore Diagram Editor* to edit the file **components.ecorediag**.

The starting point is the **System** class, which should contain all defined (oplevel) elements.

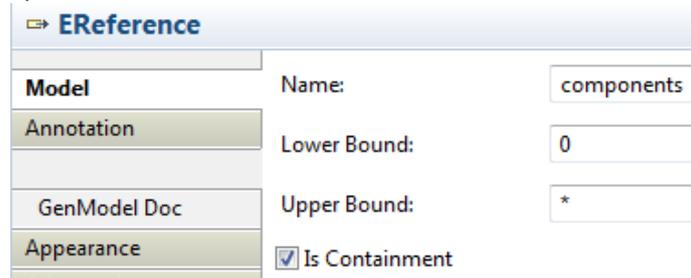
In the case you are using the diagram editor, be aware that deleting a graphic element is not automatically reflected in the Ecore model. To delete an element in the diagram **and** in the model, right-click onto the element and select "Delete from Model".

If you want to gain more insights on what needs to be done “behind the graphical editor” to define an Ecore model, use the Ecore editor instead of the graphical editor. This editor provides a tree-based view of the Ecore model. A short introduction was given in the EclipseCon 2011 presentation by Maximilian Kögel, a blog post of it can be found:

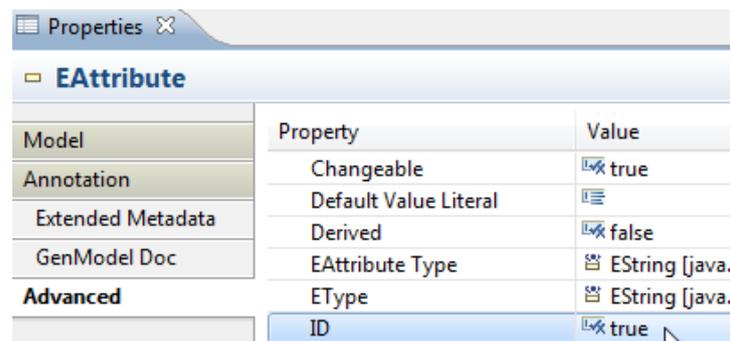
<http://eclipsesource.com/blogs/2011/03/22/what-every-eclipse-developer-should-know-about-emf-part-1/>

## Tip for references between classes

Make sure to select the correct bounds (multiplicities) and the "Is Containment" marker (for aggregations) in the Properties view:



Additionally mark every attribute, which uniquely identifies it enclosing class, as an ID:



## Task 5. Prerequisite: Xtext 2.x (0 Points) (Optional for MI)

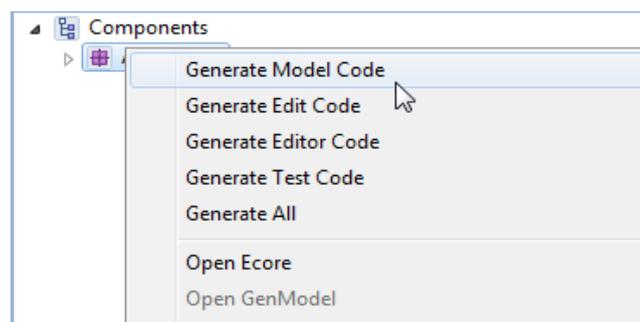
In the following tasks we will use Xtext2.x for Eclipse. You can find more information and screencasts about the tool on their website

<http://www.eclipse.org/Xtext/>

It can be either installed from an Eclipse Update Site, through the Eclipse Marketplace or by Drag & Drop from the "Install Now" button found on the website into your Eclipse.

## Create a Java representation for your Ecore model

Once your model is created open the components.genmodel, right-click onto the package `Atsccomps` to generate the corresponding Java model elements:



Repeat the procedure every time you add a new class. The same operation is carried out by the `GenerateEMFModelClasses.xml` file.

**Task 6.** *Using Xtext to define a DSL for an Ecore meta-model (6 Points) (Optional for MI)*

Working with Xtext, you can either create your Ecore model implicitly with the help of Xtext, or explicitly by creating your Ecore model and then importing it into an Xtext project. In this task you will *create an Ecore meta-model* for the components model.

This task uses the Ecore meta-model that you have defined in Task 4. Based on the Ecore Meta-model create a XText-based DSL. Your grammar should accept examples like the following:

```

component c
  port p provided I I2
  port p2
    required I I2
    provided I I2

interface I
interface I2

```

For simplicity reasons, you can restrict the order in which elements (component, interface, etc.) are defined in the file.

First, read the section about cross-referencing, to get familiar with referencing other model elements by their names:

[http://www.eclipse.org/Xtext/documentation/2\\_0\\_0/070-linking.php](http://www.eclipse.org/Xtext/documentation/2_0_0/070-linking.php)

and

[http://www.eclipse.org/Xtext/documentation/2\\_1\\_0/Xtext%202.1%20Documentation.pdf](http://www.eclipse.org/Xtext/documentation/2_1_0/Xtext%202.1%20Documentation.pdf), Page 59

And here is a simple example for the use of cross-references:

```

ObjectDeclaration : 'object' (name=ObjectName) ';;';
OhThisIs : 'Oh' (argument=[ObjectDeclaration|ObjectName]) 'this' 'is.';
HaveYouSeen: 'Have' 'you' 'seen' (argument=[ObjectDeclaration|ObjectName]) '?';
ObjectName : ID;

```

Your starting point for the DSL is the `SystemRule` in the file

**CompDSL.xtext**

Complete the DSL and run the following workflow file to create editor plugins for files with the extension "comp":

**GenerateCompDSL.mwe2**

Now you can start the runtime workspace by right-clicking on one of the projects and select **"Run As->Eclipse Application"**

A new Eclipse workspace shows up and you can create new file with the extension "comp".

Now, your DSL Editor starts up and your able to create your own component model.

## **Dictionary Task\*** (0 Points)

On each assignment sheet you will find a set of terms used in the lecture. Your (optional) task is to write a short description for each term in your own words. Try to make it as short and precise as possible. We do not accept copied sentences.

The best solution of each week will be added to a combined dictionary on our lecture website:

<http://sewiki.iai.uni-bonn.de/teaching/lectures/atsc/2011/dictionary>

There you can also see which group contributed which description. You can also at any time improve an existing entry.

Creating the dictionary is also an opportunity to check your knowledge on the topics. This also directly gives you feedback on which topics you probably need to improve for the exam.

Terms:

Meta-model

DSL

Ecore