# Disposition: Resolved

## OMG Issue No: 16376

### Title: Ecore is used for the Modelica metamodel in Part III rather than EMOF

**Source:**

Georgia Institute of Technology (Mr. Axel Reichwein,
axel.reichwein(at)me.gatech.edu)

**Summary:**

Ecore is used for the Modelica metamodel in Part III rather than EMOF (EMOF is also supported by the EMF technology). Oddly though I can see no depiction of the metamodel in either ecore or EMOF: I would for example expect to see some UML class diagrams. Instead there is what appears to be Modelica syntax.

**Resolution:**

The abstract syntax of Modelica, in other words its metamodel, needs to be represented in UML class diagrams. Figure 13 containing a representation of some Modelica metaclasses in an Ecore diagram will be replaced by UML class diagrams.

**Revised Text:**

Section 13:

Delete the last paragraph as well as Figure 13 and replace it by:

The main openModelica meta-classes  PROGRAM, CLASS and COMPONENT and their related meta-classes are presented in UML class diagrams in Figures 13, 14 and 15. All meta-classes are subclasses from a higher-level abstract meta-class whose name starts with “u”.

Figure 13 shows the PROGRAM meta-class having a list of class definitions declared at the top level in the source file, combined with a within statement that indicates the hierarchical position of the program.

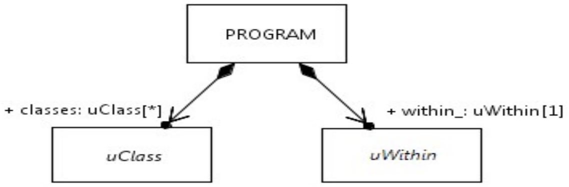


**Figure 13: PROGRAM meta-class and relationships**

Figure 14 shows the CLASS meta-class consisting of a name, a flag to indicate if this class is declared as partial, the declared class restriction, and the body of the declaration. Figure 14 for example shows the different meta-classes representing Modelica class definitions who all inherit from a common abstract meta-class named “uClassDef”.

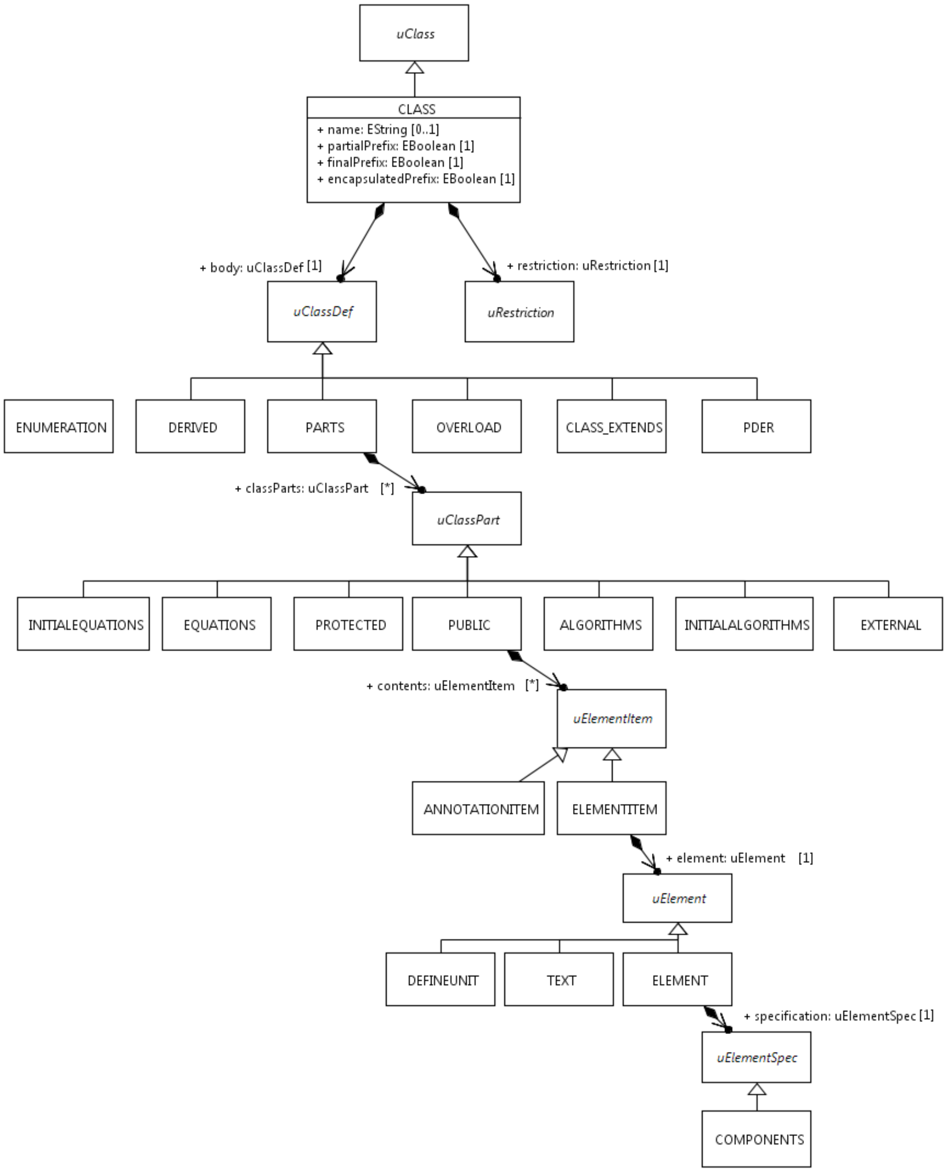


**Figure 14: CLASS meta-class and relationships**

Figure 15 shows the COMPONENTS meta-class and its related meta-classes including the COMPONENT meta-class.

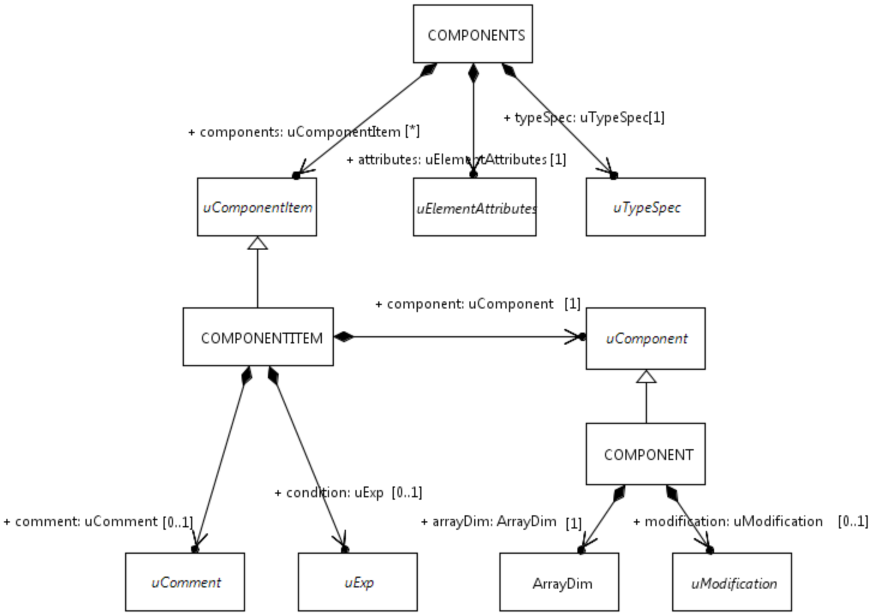


**Figure 15: COMPONENTS meta-class and relationships**

## Disposition: Resolved

# Disposition: Resolved

## OMG Issue No: 16377

### Title: The UML Profile is represented in proprietary Eclipse format

**Source:**

Georgia Institute of Technology (Mr. Axel Reichwein,
axel.reichwein(at)me.gatech.edu)

**Summary:**

The UML Profile is represented in proprietary Eclipse format.

**Resolution:**

The SysML4Modelica has been converted into OMG-grade XMI and has been added to the inventory of files which still includes the profile in Eclipse format.

**Disposition:** **Resolved**

# Disposition: Resolved

# OMG Issue No: 16545

## Title: SysML-Modelica Transformation Spec problem with <reference>

**Source:**

Lockheed Martin (Mr. Michael Jesse Chonoles, michael.j.chonoles(at)lmco.com)

**Summary:**

As found by Lenny Delligatti of Lockheed Martin.

On page 5 of the spec., it shows that the SysML4Modelica profile references the SysML profile (Figure 2 in the screenshot below):

But I believe that’s an error. I believe that a «reference» dependency is only legal from a profile to a metamodel.

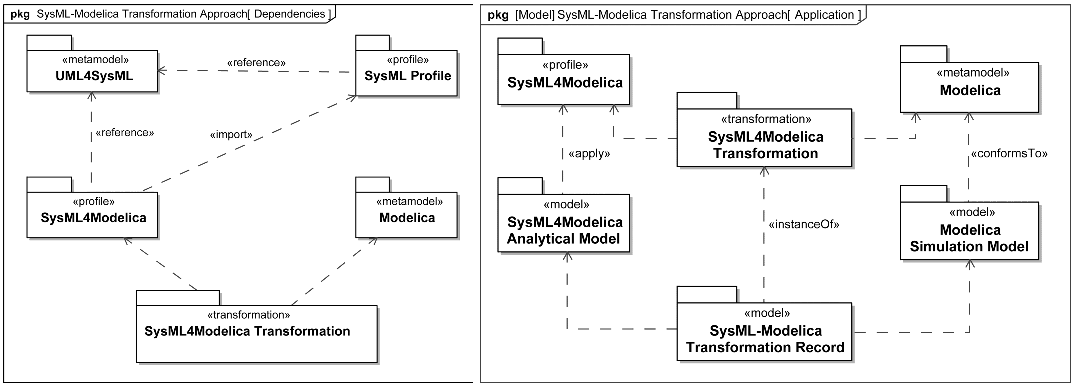
One profile can «import» another profile (and thus transitively reference a metamodel), but not «reference» another profile.

**Resolution:**

A UML dependency can be between any two NamedElements. The dependency can also be tagged with any stereotype and be given any name. So the <<reference>> dependency is by itself legal. However, the <<import>> stereotype seems more suitable between profiles. So the <<reference>> dependencies to the UML4SysML metamodel (as shown in the SysML spec) in Figure 2 are kept and the <<reference>> dependency between the SysML4Modelica and SysML profiles in Figure 2 is replaced by an <<import>> dependency.

**Revised Text:**

Figure 2 will be replaced by this Figure:



The new Figure also renames the Sys4MLModelica <<transformation>> package into “SysML4Modelica Transformation” as there is already another <<profile>> package named “SysML4Modelica”.

**Disposition:** **Resolved**

# Disposition: Duplicate

# OMG Issue No: 16556

## Title: SysML-Modelica Transformation Spec problem with <reference>

### Source:

Lockheed Martin (Mr. Michael Jesse Chonoles, michael.j.chonoles(at)lmco.com)

### Summary:

As found by Lenny Delligatti of Lockheed Martin.

On page 5 of the spec., it shows that the SysML4Modelica profile references the SysML profile (Figure 2 in the screenshot below):

But I believe that's an error. I believe that a «reference» dependency is only legal from a profile to a metamodel.

One profile can «import» another profile (and thus transitively reference a metamodel), but not «reference» another profile.

### Resolution:

**See issue 16545 for disposition**

# Disposition: Resolved

## OMG Issue No: 16593

### Title: clarification needed for the fromLibrary attribute

**Source:**

Georgia Institute of Technology (Prof. Chris Paredis,
chris.paredis(at)me.gatech.edu)

**Summary:**

In the SysML-Modelica Transformation Specification (http://www.omg.org/spec/SyM/1.0/Beta1/PDF/), the attribute fromLibrary for the stereotype «modelicaClassDefinition» is not sufficiently clearly defined (Section 8.2, page 10). The spec mentions that some details (e.g. "value properties and parts") can be omitted when using the fromLibrary tag, but the spec is not sufficiently precise as to which details exactly can/should be omitted and which should still be retained. Since this is a construct that will likely be used extensively, it should be defined more precisely.

**Resolution:**

The identified issue does not relate to the mapping of language constructs between SysML and Modelica but rather to a user- and tool-specific usability aspect of the transformation. The issue therefore does not identify a problem with the specification. Additionally, the “fromLibrary” attribute of the «modelicaClassDefinition» stereotype will be removed since it does not relate to the mapping of language constructs between SysML and Modelica.

**Revised text:**

Chapter 8.2 «modelicaClassDefinition»

Delete the last bullet point describing the “fromLibrary” attribute of the «modelicaClassDefinition» stereotype.

**Disposition:** **Resolved**