

# Custom Annotations: Handling Meta-Information in Modelica

Dirk Zimmer<sup>1</sup>, Martin Otter<sup>1</sup>, Hilding Elmqvist<sup>2</sup>, Gerd Kurzbach<sup>3</sup>

<sup>1</sup>German Aerospace Center (DLR), Institute of System Dynamics and Control,  
D-82234 Wessling, Germany,

<sup>2</sup>Dassault Systèmes AB, Ideon Research Park, SE-223 70 Lund, Sweden

<sup>3</sup>ITI GmbH, 01067 Dresden, Germany

[dirk.zimmer@dlr.de](mailto:dirk.zimmer@dlr.de), [martin.otter@dlr.de](mailto:martin.otter@dlr.de), [hilding.elmqvist@3ds.com](mailto:hilding.elmqvist@3ds.com), [kurzbach@itisim.com](mailto:kurzbach@itisim.com)

Annotations and attributes form an important part of the Modelica language. They are used to include various meta-information such as documentation, external C-code, compilation hints, etc. Given the increasingly wide field of potential applications the set of useful annotations becomes too large to be included in the language specification. Hence we present a proposal how a Modelica modeler may define his own annotations and how such custom annotations can be organized within Modelica libraries. In the long term, the goal is to move the definition of standardized annotation, as well as of attributes, from the Modelica specification to a standard library.

For example, a parameter can be marked to be a “Tuner” that shall be optimized. The following statement

```
parameter Real p1 annotation(OptimSetup.Tuner(active=true, min=-2));
```

indicates that for a default optimization setup for this model, parameter `p1` shall be used as tuner and shall have a constraint `p1 >= -2`. A custom annotation is identified by the full path name of the Tuner record class. This defines a new instance of the record, together with a modifier on this record. So, conceptually, this custom annotation is equivalent to the following declaration:

```
OptimSetup.Tuner name(active=true, min=-2);
```

and the name of the instance is not defined, because not needed (the identification of the data is via the class name).

It is proposed to optionally store custom annotations in the `modelDescription.xml` file when exporting the model as Functional Mockup Unit (FMU). Since custom annotation variables are basically standard Modelica variables with all the attributes of Modelica variables, it is proposed to just define them as standard FMI variables and mark the “custom annotation” property in the name. In particular, the name of a custom annotation variable shall be:

```
<ComponentName>.annotation.<CustomAnnotationFullClassName>.<elementName>
```

With a script language the xml-file can be read, the custom annotations retrieved and used to build automatically a default setup for special analysis or synthesis environments.

The proposed concept has been evaluated with a Dymola prototype.