Mod Code Generator Manual

Table of Contents

1 Introduction

2 Code of practice for modeling in Modelio environment

Although Modelio itself is used mainly as UML modeling tool, the modeling environment will be used for Mod Code Generator in different way, i.e. modeling will be focused more on data flow through model elements and interaction of signals (called actions), than on representation of model structure and its elements.

2.1 Model Creation

2.2 Type Creation

2.3 Component Creation

1. Right click on model package and select Create element → Component. Click on new component, press F2 on keyboard and then give desired name to it.
2. Right click on component and select Create element → Interface. Click on new component, press F2 on keyboard and name it Input Interface, then create two other interfaces for the same component and name them Output Interface and Local Parameters.

Table 1: Interface types

Interface Type	Objectives
Input Interface	Defines input interface of the component, i.e. list of signals, which are inputs to the component.
Output Interface	Defines output interface of the component, i.e. list of signals, which are outputs from the component.
Local Parameters	Defines local parameters of the component, i.e. list of signals, which are used internally within the component, but are not in the same time part of input or output interface.

RULE
Always all three interface types **shall** be created under each component, even if you will not use some of them and they will be left empty.

RULE
You **shall not** create signals in any other place of component, apart under three available interfaces as per Table 1: Interface types.

3. Right click on desired interface and select Create element → Attribute. Double-click on new attribute. In Name property type desired signal name and start typing desired signal type in Type property (see Type Creation), then press Ctrl + Space combination on keyboard to see auto-completion list.

   Repeat this step to add other signals to desired interface if needed. If you will not be using some interface type then simply do not add signals under it and leave it empty.

RULE
Please keep in mind that MCG CC recognizes data flow between two different components basing on naming convention, i.e. signals on both sides (output in one component and input in another component) **shall** have same name and type. Therefore please select name of signals and their types carefully.

RULE
All inputs to any component **shall** have unique names, i.e. other component **shall not** have input named in the same way.

RULE
All outputs from any component **shall** have unique names, i.e. other component **shall not** have output named in the same way.

4. Right click on component and select Create diagram, then select Activity diagram from available list of diagram. You should now find Activity element under your component and remove from that Activity two following elements:

   a) this: <component name>

   b) locals

5. Once Activity is created, please find Activity diagram under Activity, click on it and press F2 on keyboard, then enter desired name for Activity diagram.

6. Open Activity diagram and drag and drop interface signals (inputs, outputs and locals, see Table 1: Interface types) into diagram space. If needed, you can drag and drop any signal again if you need to use it more than one time.

   RULE
   Input Interface signal **shall** be connected only as input to another signal or action, i.e. that kind of signal **shall not** have any source in form of another signal or action within scope of diagram.

   RULE
   Output Interface signal **shall** be connected only as output from another signal or action, i.e. that kind of signal **shall not** have any consumer in form of another signal or action within scope of diagram.

   RULE
   Local Parameters signal can be used as input or output of other signals or action, however that kind of signal **shall have** other source and consumer in form of another signal or action within scope of diagram.

   RULE
   Any signal **shall** have only one source, apart from Input Interface signals, which **shall not** have any source within scope of diagram.

7. TBC

2.4 Package Creation

2.5 Naming Convention

Table 2: Allowed characters defines all allowed characters in name of any model element:

Table 2: Allowed characters

Character type	Allowed characters
Upper case letters	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Lower case letters	a b c d e f g h i j k l m n o p q r s t u v w x y z
Digits	1 2 3 4 5 6 7 8 9 0
Special characters	_ -

RULE
Names of model elements **shall** start with upper or lower case letters only.

RULE
Names of model elements **shall not** contain white spaces.

3 Mod Code Generator (MCG)

4 Modeling example with code generation