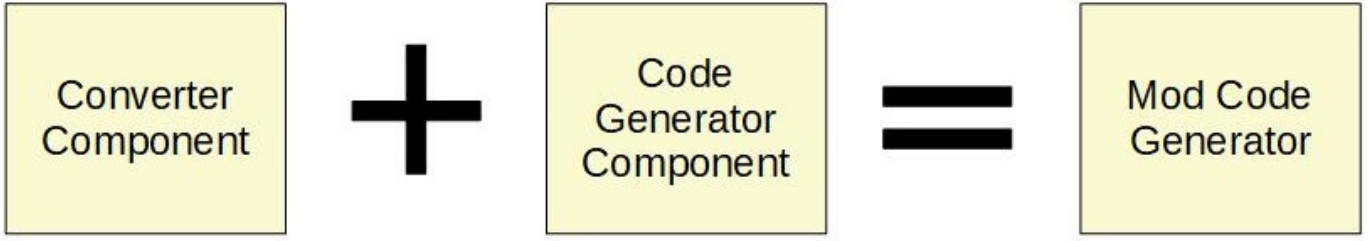
Mod Code Generator Introduction

What is the Mod Code Generator (MCG)?

- The MCG is a code generator tool
- It is written in Python https://www.python.org/
- The MCG works in conjunction with Modelio environment https://www.modelio.org/
- The MCG repository is available under https://github.com/deckamil/mod-code-generator/

The MCG is divided into two components:

- Converter Component (CC), responsible for conversion of a model created within Modelio environment into a configuration file
- Code Generator Component (CGC), responsible for C code generation from the configuration file

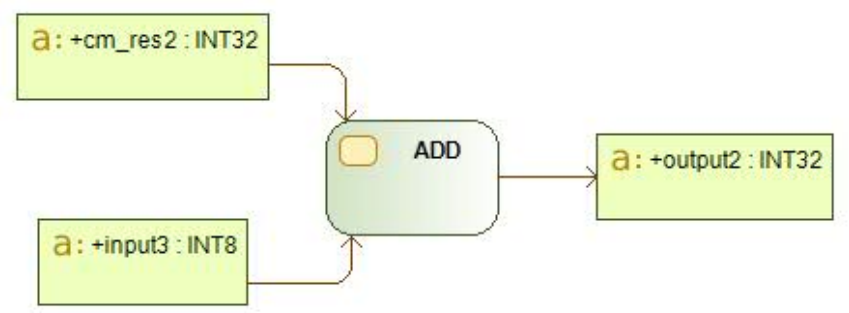


What role does the Modelio environment play?

- The MCG CC and MCG CGC are command line tools
- The Modelio provides modeling environment and graphical user interface for the MCG
- It is standalone program and it is not a part of the MCG
- See list of features under https://www.modelio.org/about-modelio/features.html/

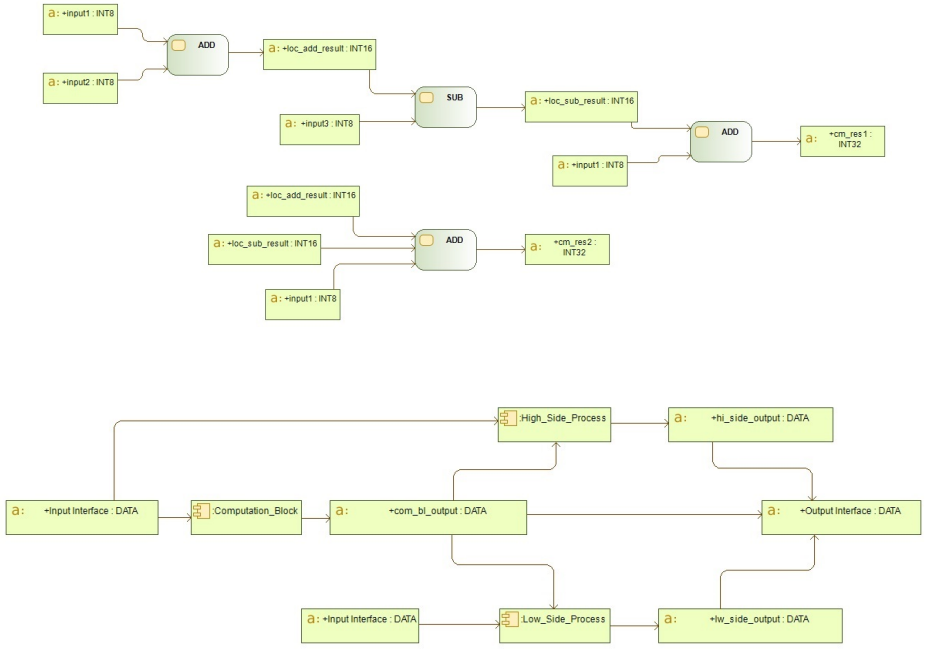
How the Modelio environment is used?

- The Modelio allows to build graphical representation of a model, its data flow and data interactions, mainly through UML activity diagrams.
- A model is usually composed from package and component elements.
- Activity diagrams (as most of other Modelio elements) are saved in form of .exml files.
- exml. files provide input information about diagram elements and their connections to the MCG CC.



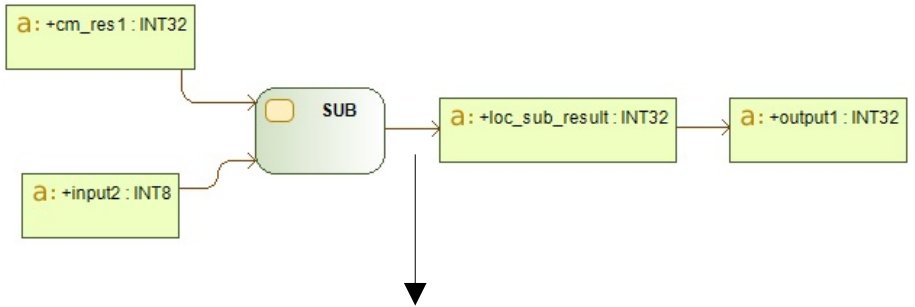

```
<REFOBJ>
  <ID name="cm_res2" mc="Standard.Attribute" uid="8e26c36c-cf8e-4f00-9819-a470ba01f5cd"/>
</REFOBJ>
</LINK>
<COMP relation="Outgoing">
  <OBJECT>
    <ID name="ObjectFlow" mc="Standard.ObjectFlow" uid="92d78caa-d13c-493e-893b-8859f93d899a"/>
    <ATTRIBUTES>
      <ATT name="TransformationBehavior"></ATT>
      <ATT name="SelectionBehavior"></ATT>
      <ATT name="IsMultiCast">false</ATT>
      <ATT name="IsMultiReceive">false</ATT>
      <ATT name="Effect">ReadFlow</ATT>
      <ATT name="Guard"></ATT>
      <ATT name="Weight"><![CDATA[1]]></ATT>
      <ATT name="Name"><![CDATA[ObjectFlow]]></ATT>
      <ATT name="status">1970354901745664</ATT>
    </ATTRIBUTES>
    <DEPENDENCIES>
      <LINK relation="Target">
        <REFOBJ>
          <ID name="ADD" mc="Standard.OpaqueAction" uid="fd5be3ed-0d38-42d0-ab56-d1058657eee8"/>
```

Difference between component and package



- Both component and package should have interface definition along with activity diagram, which will be converter later from .exml format into configuration file.
- Component activity diagram is used to define data interactions (like arithmetic operations) and is treated as a separate C module.
- Package activity diagram is used to define data flow between components. If connections between components are not defined under package activity diagram, then components integration at C level has to be provided manually after code generation.

Model connections and nodes



- The MCG CC finds information about connections between diagram elements in .exml files and base on that creates a list of connections, which further is turned into list of nodes.
- A node represents one, specific instance of data interaction (like SUBtraction operation) from activity diagram, along with interaction inputs and output.

- $SOURCE$: $FIRST$ cm_res1 $TARGET$: SUB 6d076cfa-6ce0-4bd4-bde4-e327457e8691
- $SOURCE$: input2 $TARGET$: SUB 6d076cfa-6ce0-4bd4-bde4-e327457e8691
- $SOURCE$: output1 $TARGET$: $EMPTY$
- $SOURCE$: loc_sub_result $TARGET$: output1
- $SOURCE$: SUB 6d076cfa-6ce0-4bd4-bde4-e327457e8691 $TARGET$: loc_sub_result

- $INPUTS$: cm_res1 input2 $INTERACTION$: SUB 6d076cfa-6ce0-4bd4-bde4-e327457e8691 $OUTPUT$: loc_sub_result
- $INPUTS$: loc_sub_result $INTERACTION$: ASSIGNMENT $OUTPUT$: output1

How activity diagram nodes are sorted?

- $INPUTS$: input1 input2 $INTERACTION$: ADD 7ea36f0f-22c9-404f-adbb-1b770e2c188c $OUTPUT$: loc_add_result
- $INPUTS$: loc_sub_result input1 $INTERACTION$: ADD 9b1cef49-19f9-47a5-9247-e94b09d8b9a3 $OUTPUT$: cm_res1
- $INPUTS$: loc_add_result loc_sub_result input1 $INTERACTION$: ADD 8abfc438-1b8d-4771-a445-53855a80ea10 $OUTPUT$: cm_res2
- $INPUTS$: loc_add_result input3 $INTERACTION$: SUB 56cfd233-ec84-405e-8de8-22caf60c369d $OUTPUT$: loc_sub_result

- $INPUTS$: input1 input2 $INTERACTION$: ADD 7ea36f0f-22c9-404f-adbb-1b770e2c188c $OUTPUT$: loc_add_result
- $INPUTS$: loc_add_result input3 $INTERACTION$: SUB 56cfd233-ec84-405e-8de8-22caf60c369d $OUTPUT$: loc_sub_result
- $INPUTS$: loc_sub_result input1 $INTERACTION$: ADD 9b1cef49-19f9-47a5-9247-e94b09d8b9a3 $OUTPUT$: cm_res1
- $INPUTS$: loc_add_result loc_sub_result input1 $INTERACTION$: ADD 8abfc438-1b8d-4771-a445-53855a80ea10 $OUTPUT$: cm_res2

Once nodes are defined, the MCG CC will sort them:

1. At beginning the MCG CC looks for nodes, which use only input interface data (no dependency from local data) and then add such nodes at beginning of sorted nodes list.

2. After that the MCG CC looks for nodes, which use either input interface data or local data outputted from previously sorted nodes, then append them to the list.

3. The MCG CC repeats the process from (2) for the rest of unsorted nodes till all are sorted.

Conversion into configuration file

```
COMPONENT START
COMPONENT SOURCE 48bfdb13-00ff-4337-a0e1-c9c4a7ec221a.exml
COMPONENT NAME Computation_Block
INPUT INTERFACE START
type INT8 name input1
type INT8 name input2
type INT8 name input3
INPUT INTERFACE END
OUTPUT INTERFACE START
 type INT32 name cm_res1
 type INT32 name cm_res2
 OUTPUT INTERFACE END
 LOCAL DATA START
 type INT16 name loc_add_result
 type INT16 name loc_sub_result
LOCAL DATA END
BODY START
COM Action Interaction ADD 7ea36f0f-22c9-404f-adbb-1b770e2c188c
INS loc_add_result = input1 + input2
COM Action Interaction SUB 56cfd233-ec84-405e-8de8-22caf60c369d
INS loc_sub_result = loc_add_result - input3
COM Action Interaction ADD 9b1cef49-19f9-47a5-9247-e94b09d8b9a3
INS cm_res1 = loc_sub_result + input1
COM Action Interaction ADD 8abfc438-1b8d-4771-a445-53855a80ea10
INS cm_res2 = loc_add_result + loc_sub_result + input1
BODY END
COMPONENT END
```

Sorted nodes are latter converted into configuration file format by the MCG CC.

The configuration file is treated as input data to the MCG CGC process and it contains information about interfaces of model element and set of instructions which represent data interactions from activity diagram.