

APPLICATION NOTE

HOW TO USE PYTHON FUNCTIONS IN MATELO?

<u>Author</u>	René Christian TUYISHIME
<u>Abstract</u>	This Application Note explains how to use Python functions in MaTeLo.
<u>Keywords</u>	MaTeLo, Transfer function, input, output, Python function
<u>Version</u>	1.0 – July 2013

Table of contents

1. Introduction.....	3
1.1 MaTeLo.....	3
1.2 Why to use Python functions in MaTeLo?	3
2. Python function.....	4
2.1 Transfer function in MaTeLo (Reminder)	4
2.2 How to use Python function in MaTeLo?	5
2.3 Test generation by MaTeLo Testor.....	10
3. Conclusion	11
4. References.....	12

Table of figures

Figure 1: Transfer Function definition.....	4
Figure 2: Function definition	5
Figure 3: Python Function sample.....	6
Figure 4: Transfer Function Menu.....	6
Figure 5: Python function definition	7
Figure 6: MaTeLo input and ERA declaration.....	8
Figure 7: Transfer function window	9
Figure 8: Transfer function window with fields filled.....	9
Figure 9: Link window.....	10
Figure 10: Simulation option	10
Figure 11: sample of Testor generation report	11

1. Introduction

This Application Note explains how to use Python functions in MaTeLo.

1.1 MaTeLo

MaTeLo is a tool chain developed by ALL4TEC (www.all4tec.net) that implements the Model Based Testing approach. In user friendly environment, MaTeLo offers an integrated set of components and features to enable test engineers in designing their tests based on usage models. Then, the test cases are automatically generated from the model depending on the user selected test strategy.

In general, MaTeLo manages successfully test projects in the fields of automotive, railway or energy, IT by providing professional and valuable features to obtain a better test coverage and a higher engineering productivity.

1.2 Why to use Python functions in MaTeLo?

Some System outputs are obtained by simple or complex calculations. Those calculations are performed through functions.

The systems outputs are in MaTeLo the Expected Results Automatic (ERA) associated on transitions of the model. To be able to have the ERA calculated in function of the inputs set on transition, MaTeLo offers the possibility to call function which calculates it.

Before the 4.7.8 MaTeLo version, it was possible to call only functions developed in Matlab and Scilab.

In order to take in consideration users who don't use Matlab and Scilab to perform complex calculations, an evolution was made. This evolution is to give user the possibility to call in MaTeLo his functions developed in Python language.

This new evolution offers also to execute Matlab functions contained in a dll. This is helpful for users who don't have Matlab license. Matlab function will be executed through python function.

In this application note, the use of Python function in MaTeLo will be presented through a simple example.

2. Python function

2.1 Transfer function in MaTeLo (Reminder)

As stated in the introduction of this application note, MaTeLo offers the possibility to call functions developed in Matlab or Scilab tools. When there is a need to call in MaTeLo model, functions (called Transfer functions in MaTeLo); the tool used to develop those functions has to be defined.

The Transfer function definition is done by Go on the Menu Project > Configuration:

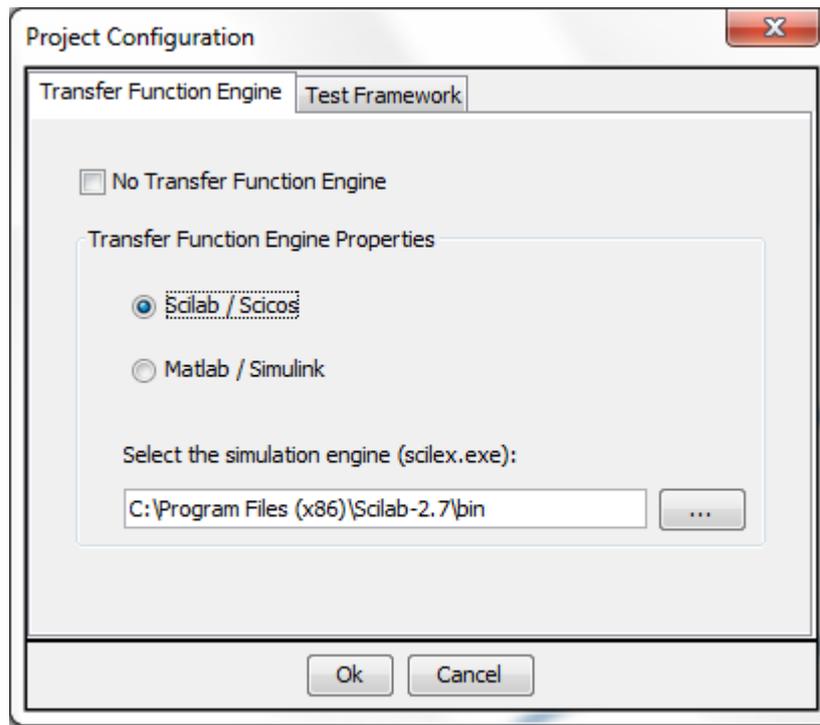


Figure 1: Transfer Function definition

First of all, the tool has to be selected and then the location of the simulation engine. After, the user can begin defining his transfer functions. To do this, user has to right click on “Transfer Functions” menu and click on New for Scilab and Matlab functions and New Python Function for Python function. For each transfer function, the following elements have to be defined:

1. The name of the transfer function;
2. A short description of the transfer function can be given;
3. The path on what file containing the transfer function is saved;
4. The definition of the transfer: name and type of the input and output, eventually their description.

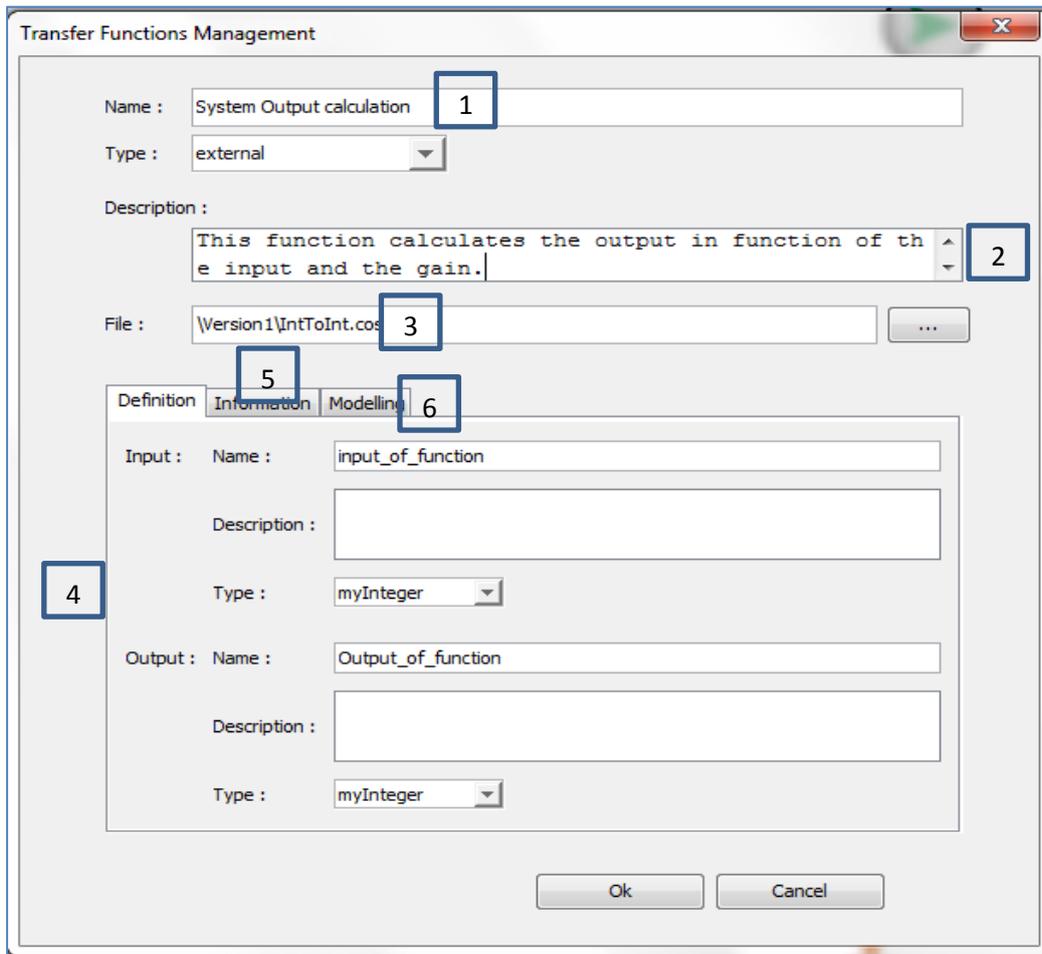


Figure 2: Function definition

The tab Information (5) displays on which transition(s), the transfer function has been associated. The Tab Modelling(6) allows the user viewing the transfer function in the correspondent tool.

2.2 How to use Python function in MaTeLo?

To illustrate the usage of Python function in MaTeLo, a simple Python function will be used. This function aims to calculate the max of two values. Here is a screenshot of the function:

```

7% max.py - C:\Users\rctuyishime\Documents\Application Note\Python_function\Version1\max.py
File Edit Format Run Options Windows Help

# Do not change: Typing of the data passed from Testor
self.Input_List = args[0]
self.Input_List_Size = args[1]
# End Fix Code

def calculateOutputs(self):
    """Function to calculate the output values.

    Keyword arguments:
    self -- reference to the class itself
    """

    # User Code: Precise the Inputs Parameters
    if(self.Input_List[0] > self.Input_List[1]):
        self.outputData = self.Input_List[0]
    else:
        self.outputData = self.Input_List[1]
    # End User Code

    # Do not change: Init of the data passed to Testor, this is a list
    result = list()
    # Do not change: End of Init

    # Update: Append output data to the Output list that will pass to MaTeLo
    result.append(self.outputData)

    print "Outputs = ", result
    return result
    
```

Figure 3: Python Function sample

2.2.1 Python Function declaration in MaTeLo

The declaration of the Python function is performed in this way:

1. Right click on **Transfer Function** Menu, then click on **New Python Function**

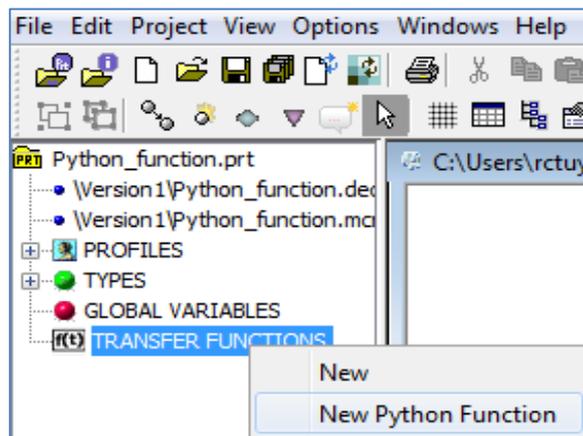


Figure 4: Transfer Function Menu

2. Function definition (for procedure, see chapter 2.1)

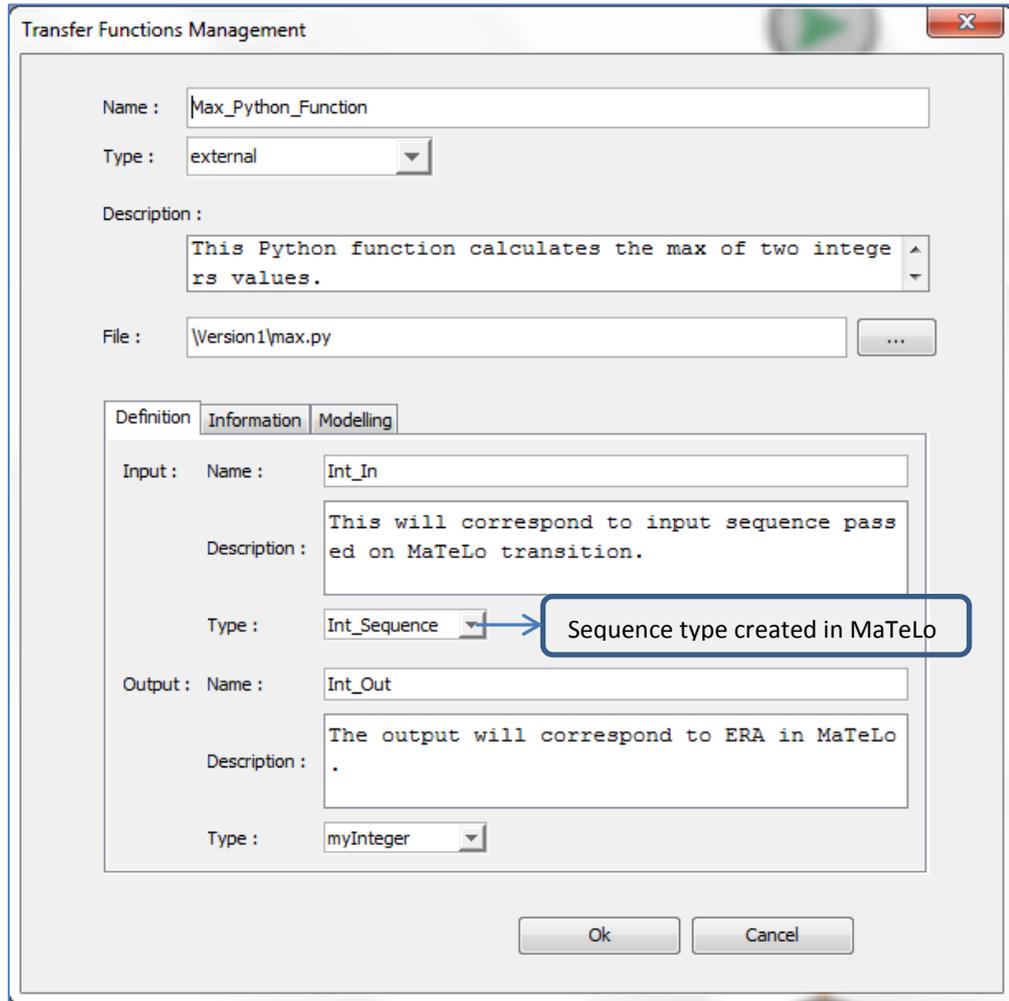


Figure 5: Python function definition

2.2.2 Python function association to MaTeLo model

Before associating the Python function on the MaTeLo transition, input and Expected Result must be defined. The input of the transition will be mapped to the Python function input and the ERA to the Python function output.

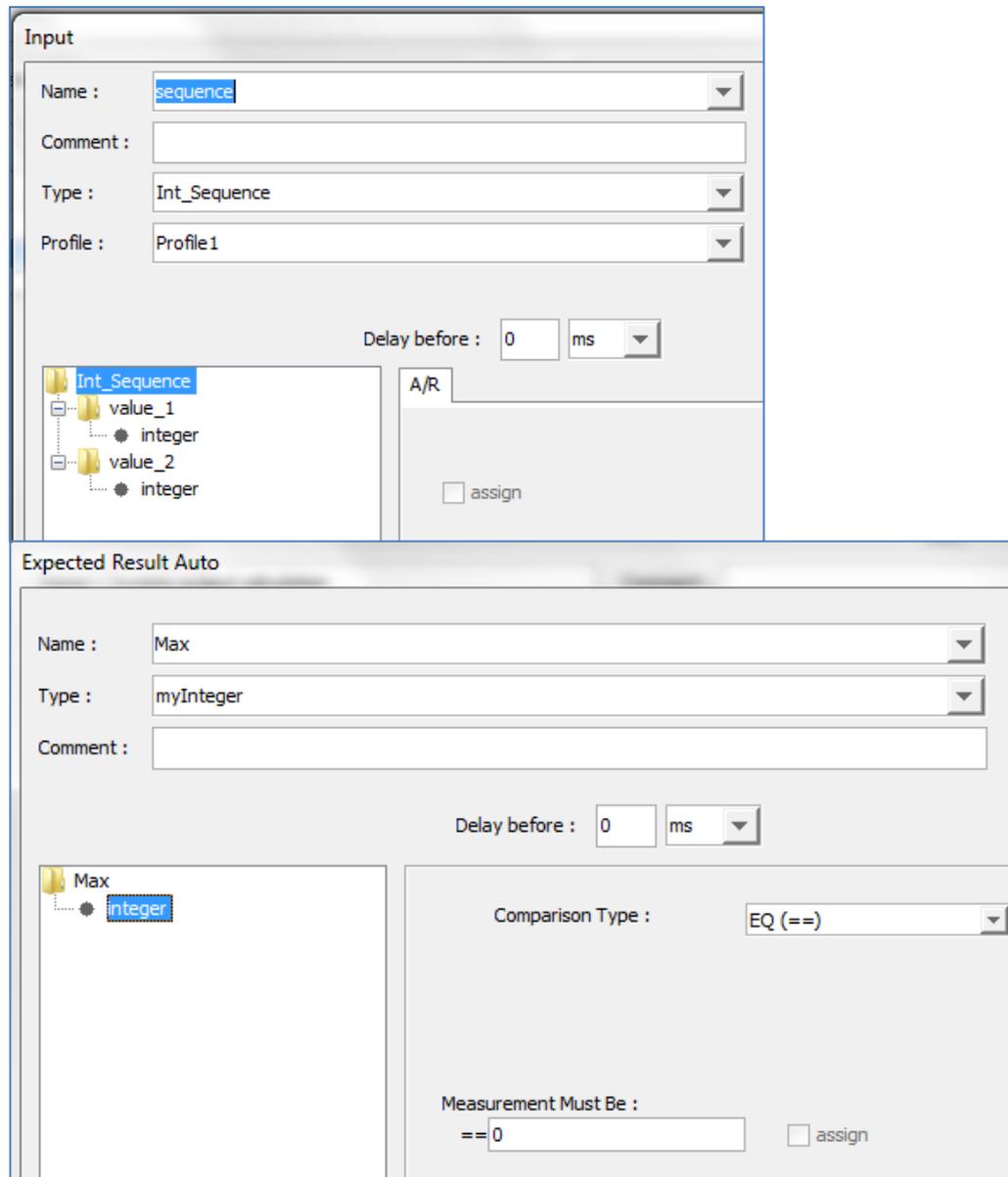


Figure 6: MaTeLo input and ERA declaration

For the Expected Result Automatic, User has to set a value by default. In this example, the default value is 0.

After the definition of the input and the ERA, the Python function can be associated to the transition. The function is added to the transition in the same way as for input and ERA, by double clicking on the transition, then add button, select function item and click Ok.

A new window “Transfer Function “appears.

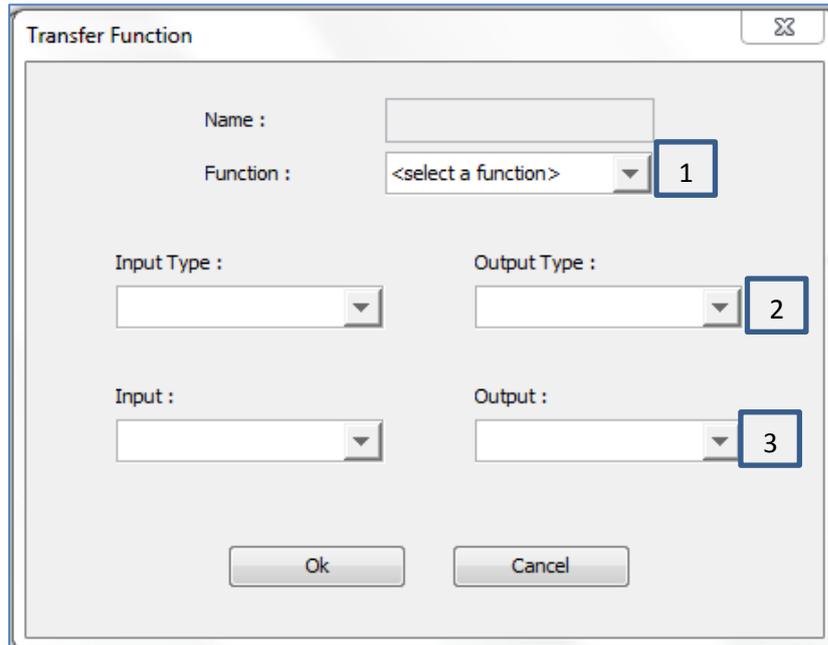


Figure 7: Transfer function window

At this step, the actions presented below are performed:

1. Select the Python function
2. Select input and output type. Here, there is two possibilities for each parameter:
 1. Type “input” for Input type or “ERA” for Output type. This means Function will take the input of MaTeLo as its input and ERA as its output
 2. Type “global” if global variable has been used
3. Select the input and the output

Here is the screenshot of Transfer function window after the execution of the three actions presented above.

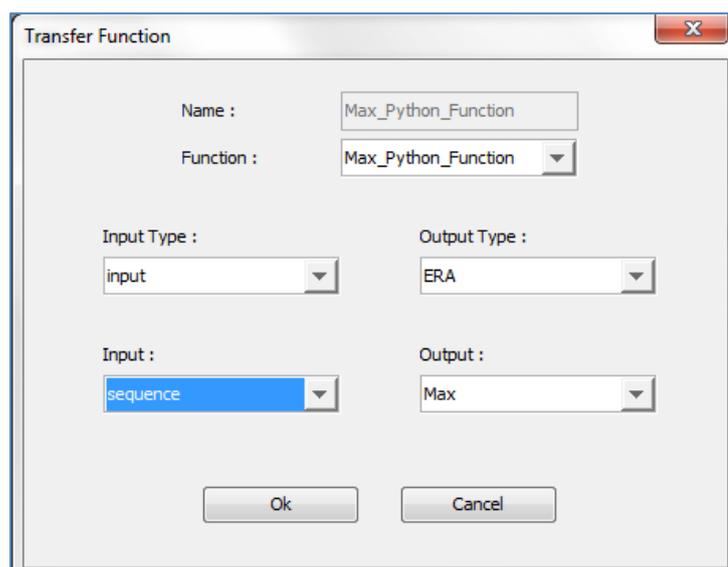


Figure 8: Transfer function window with fields filled

After validating of the association of the function, it appears in the “Link” window, below the input and the Expected Result Auto.

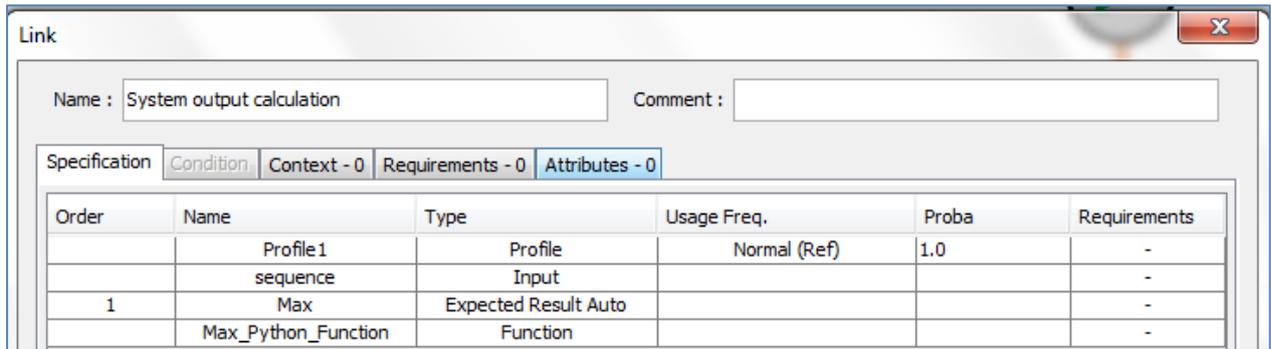


Figure 9: Link window

2.3 Test generation by MaTeLo Testor

To enable the execution of Python function during the test generation in MaTeLo Testor, the user has to check the box “Simulation” in MaTeLo Testor.

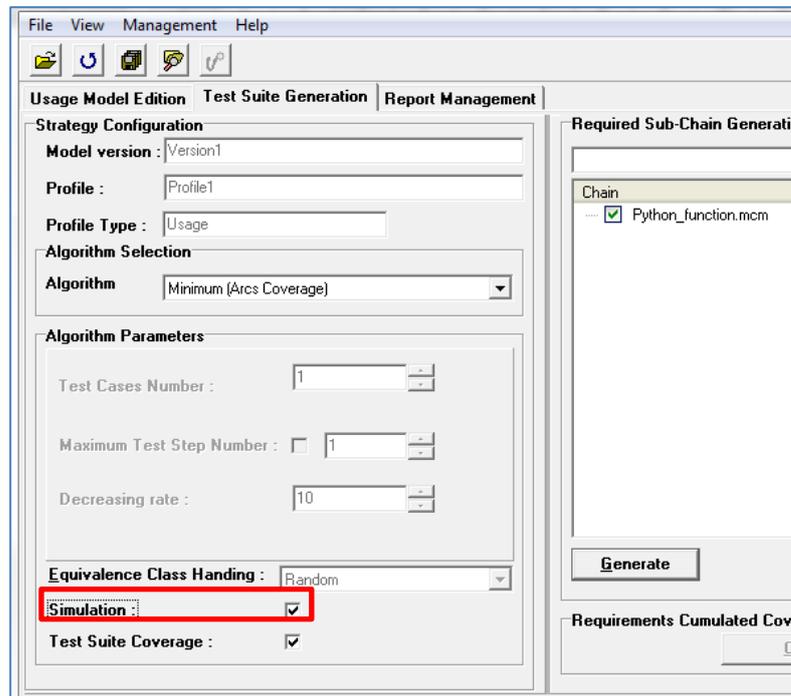


Figure 10: Simulation option

After activating the simulation option in Testor, Test suite can be generated.

Step	Event/Input	Expected Result
1	System output calculation Input : sequence <ul style="list-style-type: none"> • Int_Sequence : <ul style="list-style-type: none"> ◦ value_1 : 5 ◦ value_2 : 9 	ERA : Max <ul style="list-style-type: none"> • myInteger : == 9

Figure 11: sample of Testor generation report

As displayed on the Testor generation report, python function allowed calculating the expected result which was the maximum of two values.

Note: Sequence types including subsequences cannot be processed in Python functions. Data types restrictions can be found on: http://www.all4tec.net/wiki/index.php?title=Transfer_functions#Data_types

3. Conclusion

As presented in this short application note, it is possible to call transfer in order to perform complex calculations.

MaTeLo offers the use of transfer functions designed in Matlab, Scilab and Python.

The recent evolution regarding the use of Python function will allow the user who doesn't have simulation tools like Matlab, to develop their own complex function and call it in MaTeLo.

4. References

Websites:

- MaTeLo: <http://www.all4tec.net/>

Wiki MaTeLo :

<http://www.all4tec.net/wiki>