Characteristic Map Editor – User Guide

Sofiane Belkhiria, Jan Romberg

V1.0

14/09/2005

1 Introduction

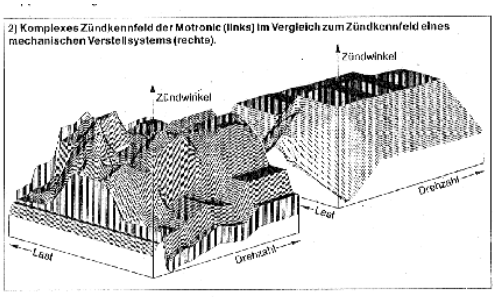
With the increasing adoption of imbedded systems in the industry, the need for powerful and enhanced CASE-Tools is growing.

AutoFOCUS is a CASE-Tool being developed at the Professors Broy Software Engineering Department of the Technical University of Munich.

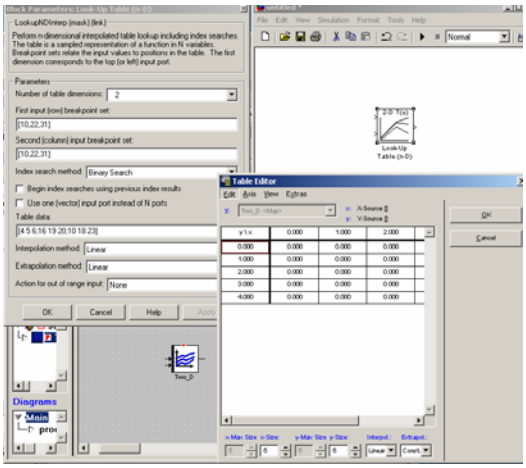
The tool offers the possibility for describing systems and interaction between different system components with a multitude of diagram types such as System Structure Diagrams, Data Flow Diagrams and State Transition Diagrams.

This user guide describes a new feature that enables likewise the description of behavior for software components. The behavior, which relates input and output values, is represented by a table. This form of representation is also called characteristic map.

The characteristic maps are tables that are used to calculate output values based on the input values and an interpolation performed between the appropriate values from the table.



Characteristic map for ignition angle in a gasoline engine controller



Characteristic map editors in commercial tools (ASCET, Matlab Simulink)

In the course of a Systementwicklungsprojekt (SEP) undertaken by Sofiane Belkhiria in 2004, a characteristic map editor was designed, implemented, and integrated into AutoFOCUS.

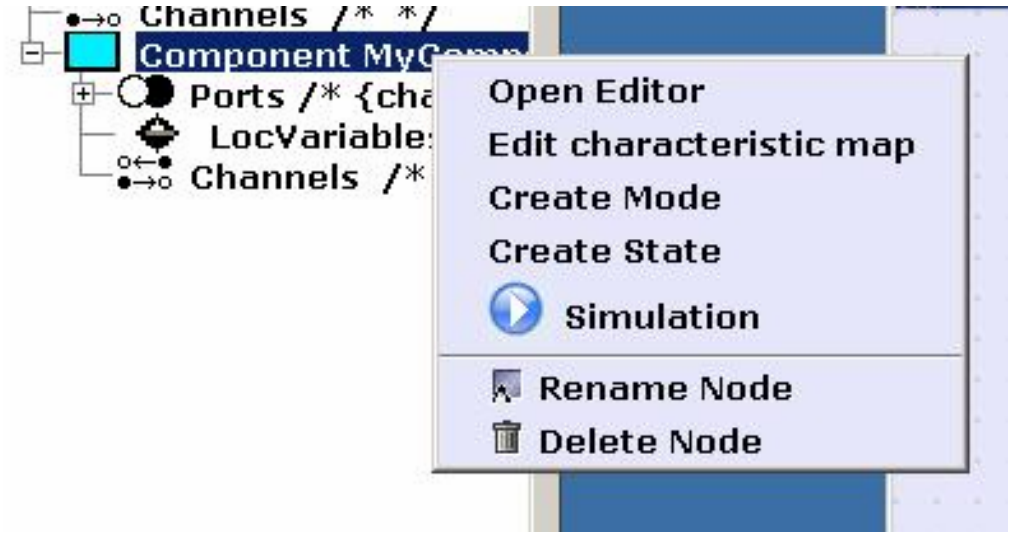
The editor was designed to support the following tasks:

- creating characteristic maps for components.
- support of different data types : integer, double, float and Boolean.
- entering data manually.
- performing data type check for manually entered data.
- editing already existing characteristic maps manually (adding / removing labels).
- saving and loading previously defined characteristic maps automatically.
- preserving consistency of characteristic maps by detecting changes performed on components and ports in other diagram editors such as ports renaming and type changing.
- preventing user from performing illegal operation on maps.
- automatic computing of table values based on input values and mathematical functions. (Not yet implemented at this stage)

2 User Guide

2.1 Opening a Characteristic Map Editor (CME) :

To define a characteristic map for a component, select **Edit characteristic map** in its context menu.



Opening the CME editor

If an ssd editor is already opened for that component, this command will be inactive (gray) so that it will not be possible to open both simultaneously.

Note that it is only possible to define a CM for components that have exactly one outgoing port. Again, if this condition is not fulfilled, the command in the context menu will be inactive.

The CME has its special toolbar, on which figure the buttons corresponding to all its featured functions.

2.2 The CME Toolbar

The CME toolbar

The CME Toolbar has 2 functions groups:

- Add/Delete group
- Precompute group

Add/Delete group:

Consists of 2 buttons:

- Insert Label: enables inserting a label in a dimension. When clicked, a dialogue appears, where the dimension to add label to has to be chosen, and the label itself

has to be entered. The label must be a unique one, so that in a single dimension there are no similar labels.

- Delete Label: enables deleting a label from a dimension.

First select the dimension from of the label to be removed, then choose the label from the list.

Precompute group :

Consists of the **Precompute** button and a field where user can enter an equation. Based on this equation, and the present ports in the CME, the missing values will be calculated and the CM will be automatically filled.

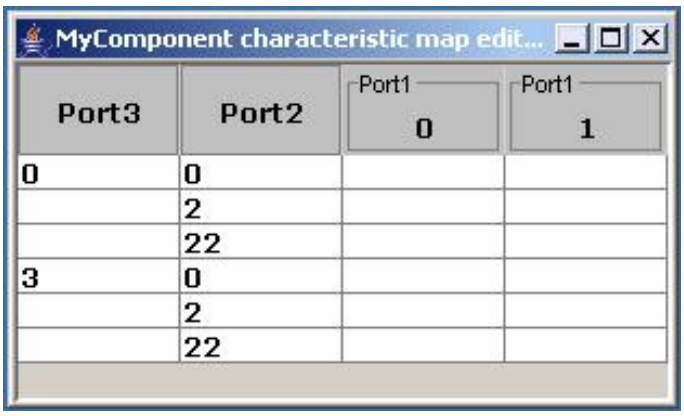
2.3 TheEditor Frame

The characteristic map is a table that enables users to define the output values for all the possible combinations of input values, for every port of any component that has the conditions mentioned above.

It also enables adding and removing labels to every port.

The CM editor is a special table, in which every port is represented by a column(vertically), except the first one, which is represented horizontally.

This means that everyone of its labels occupies a column for itself. These columns have special column headers, that have borders, where the name of the port is indicated.



The editor frame

The order of the columns in the table can not be modified. Therefore it is possible to resize them using the mouse, for example to make the columns larger. This applies also for the whole editor frame. If the frame is resized, the columns will automatically adjust their widths.

The ports are automatically initialized with one default label, which is zero for ports with types int, float or double. Boolean ports are initialized with the default values True and False.

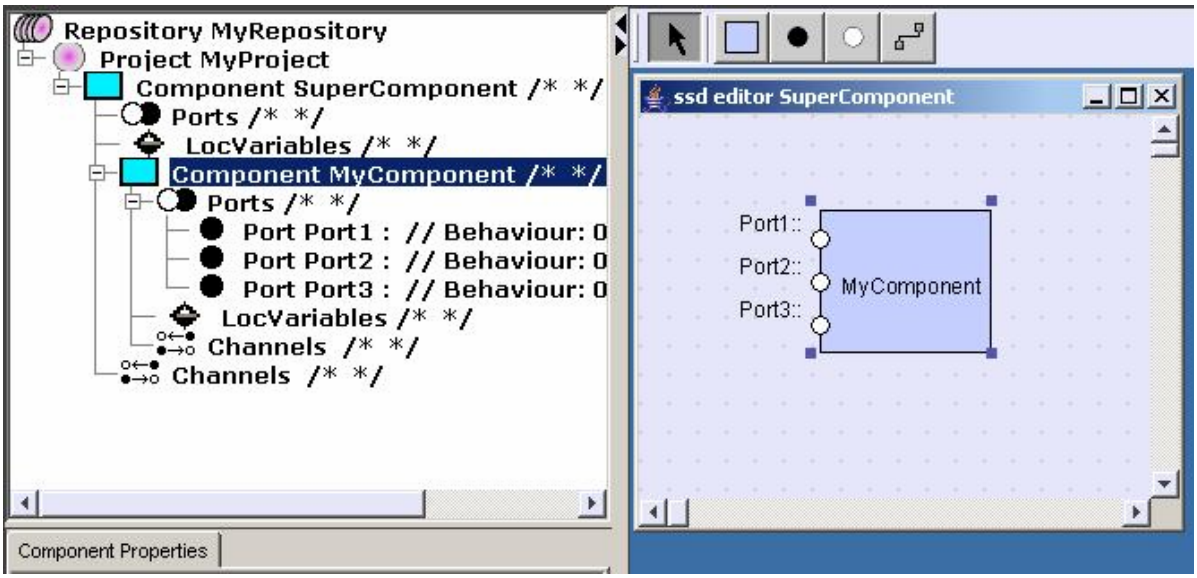
As we will see later, default values can be removed if they are not required.

Once the CME opened, the user can already start to insert labels and output values.

The use of the CME will be more explained in the following tutorial.

2.4 Using the Characteristic Map Editor :

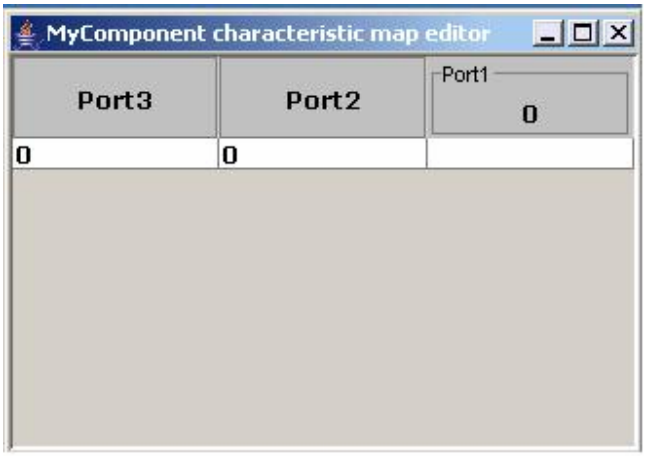
We consider that we have a component named MyComponent that has 3 input ports: port1, port2 and port3, and an output port OutputPort.



MyComponent example

All the ports have now type integer (int).

By choosing Edit characteristic map in the context menu of MyComponent, the CM editor is opened. It presents a map with all the input ports of the concerned component.



Characteristic Map editor for MyComponent

Inserting Labels

To insert a label into a dimension, use the button **Insert Label** of the toolbar. A dialogue will popup, where the user can enter the label and choose the dimension from the available ones on the list.

Inserting labels

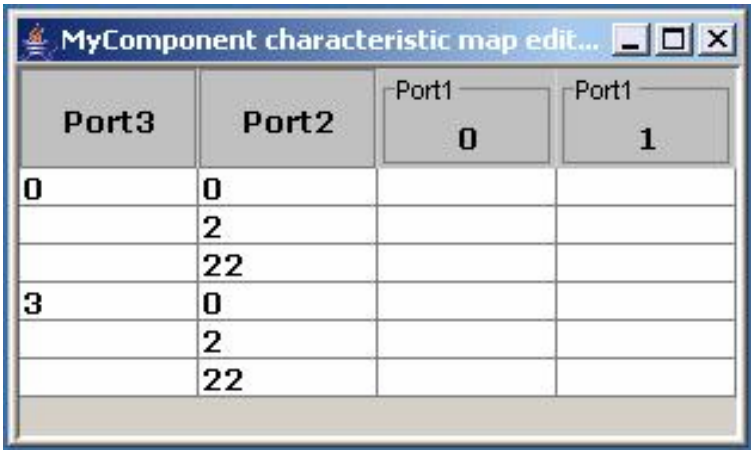
After entering the label in the text field and choosing the dimension, use the **Add** button to validate. If any information is missing, a notification message will remind the user. In case the user wants to break off the operation, simply close the dialogue.

The new labels will be automatically inserted in the right position between others of the same dimension, so that they are sorted by value. It is not possible to add a label that

already exists in the dimension. In this case, a notification message “Invalid input” will popup, and the dialogue will be closed.

Since a port can have a specified type (integer, float, double, etc…) , the entered label must be formatted according to the port. Otherwise it will be considered as an invalid input and will not be admitted.

Proceeding like described above, we add the label 1 to Port1, labels 2 and 22 to Port2, and label 3 to Port3



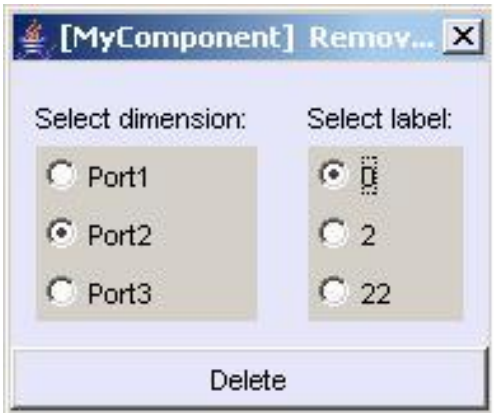
Characteristic Map editor for MyComponent after inserting labels

Note that it is impossible to add a label to a Boolean dimension, and thus no boolean dimensions will be listed. This applies also to Removing labels.

Removing Labels

The added labels cannot be edited or moved manually. The columns that contains the labels are not editable. The only way to edit a label is to remove it and eventually replace it with another one.

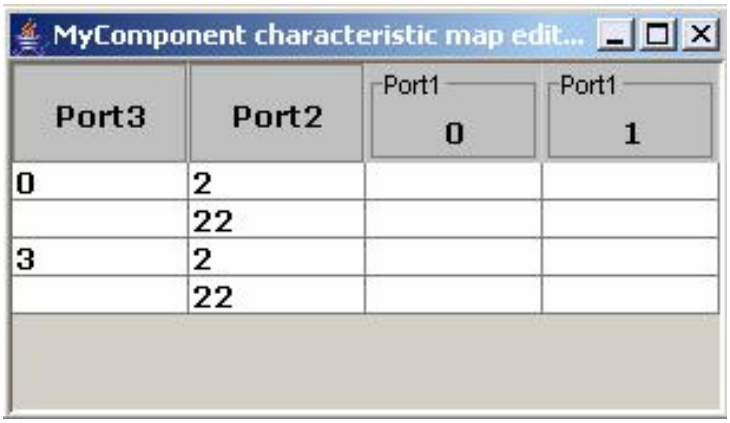
To remove a label from a dimension, use the button **Delete Label** of the toolbar. A dialogue will popup, where the user can enter the label and choose the dimension from the available ones on the list.



Removing labels

We proceed by removing the label 0 from Port2.

Now our map looks like this:



Characteristic Map editor for MyComponent after removing label 0

Labels can be removed from a dimension only if it has more than one label.

That is why we have added labels 2 and 22 before we removed label 0.

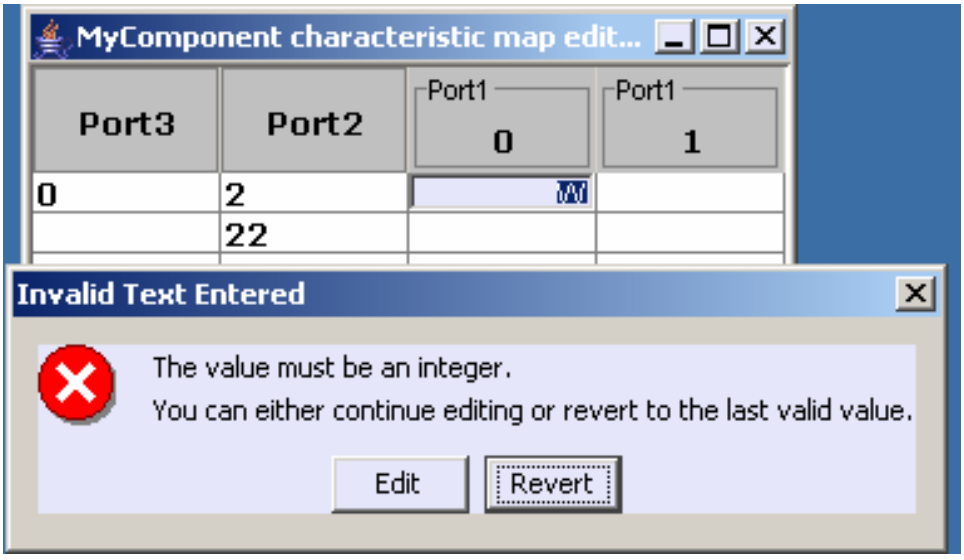
The dialogue in the next picture would have appeared if we decided to remove a label before adding the second one to the first dimension.

Removing labels not possible

Entering Values:

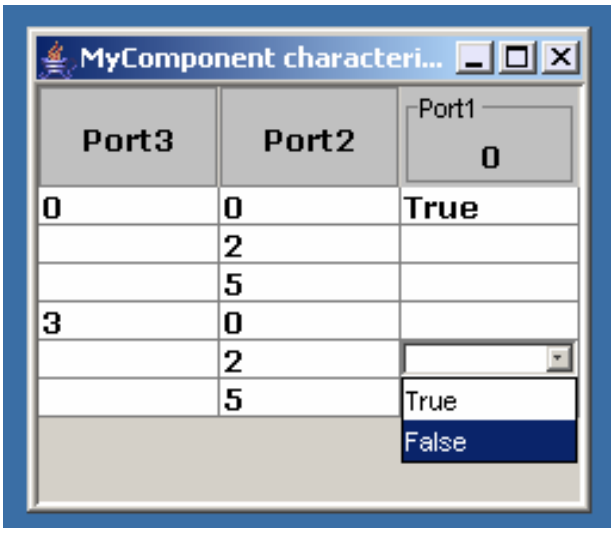
The output values are to be entered manually by the user. This has to be done in the editable area of the table. This area consists in the columns of the first port labels. The output values have to be entered in those columns. These are the only editable columns in the table. Since the output port can have a specified type, only data from that same type are allowed.

To edit a cell, it has to be clicked. A formatter will then check whether the entries are from the required type and eventually prompt the user to correct his value.



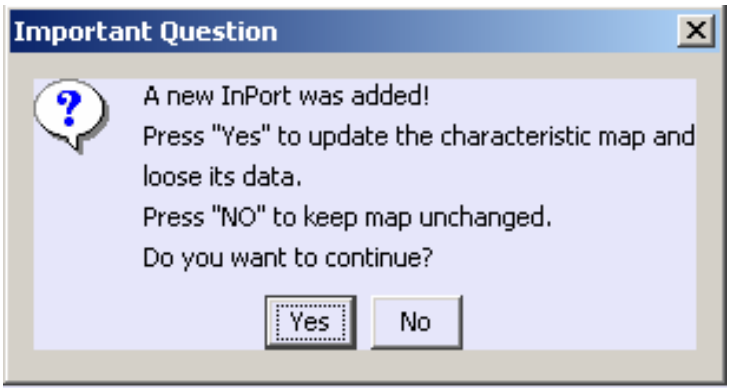
Editor for values and formatter messages

If the output port is Boolean, the user will have to choose on of the values True or False from the combo box.



Combo box for selecting a value

After entering some values in the table, we keep the CME opened and we add a new InPort using the ssd editor. As soon as we have done that, this dialogue is displayed:



Message after adding new in port

Now, this is due to the fact that the data we already entered doesn't make sense anymore, as the new port input (labels) must be taken into consideration. But this gives also the user the possibility to keep his map unchanged, in case he decides to undo his last operation.

The CME automatically detects renaming of its component, as well as renaming, adding or removing ports and modifying their types. The CME will notify the user and give him

the possibility to allow an automatic updating of the map, or just inform him that the old CM is no longer valid.

Message after adding new out port

2.5 Saving and loading maps

The editor saves the map automatically when it is closed.

If a CME of a component is opened, it loads automatically the saved map.

Otherwise, if no map has been defined yet, the default map will be loaded and initialized with default labels.

If the CME is closed while some output are still blank, the next time it will be loaded, they will be replaced with a “null” value.

If after closing the CME, a port renaming is performed, it will be automatically taken into account and the column header of the renamed port will have show the new name.

Also if a port has been added to or removed from the component, this will cancel the old saved map and a new default map will be created.

The same is done when a port type has been changed.

2.6 Release notes

One of the requirements for the CM editor was preserving consistency of characteristic maps by detecting changes performed on components and ports in other diagram editors, such as ports renaming and type changing.

This version of the CM editor has to be further developed to completely meet that requirement.

Indeed, at this stage, as long as the CM editor is opened, the data will be always consistent and every port renaming, port type change or port remove will be signaled and handled. But, after the editor is closed, there will be no consistency check until the map will be loaded again. So consistency-check for the saved map data (into the comment of a certain component) is only performed when a map is being loaded. However, this is only

a check for port names and number, and NOT for port types. Hence, if any port type has been changed while the CM editor is closed, the data in the map will very probably become inconsistent.

One solution to this problem (for now) will be, if a port type change has occurred, to open the CM editor, change the port type to any other type, and then change it back again.