GRIF 2014

Petri Nets with predicates

User Manual

Version 31 March 2014

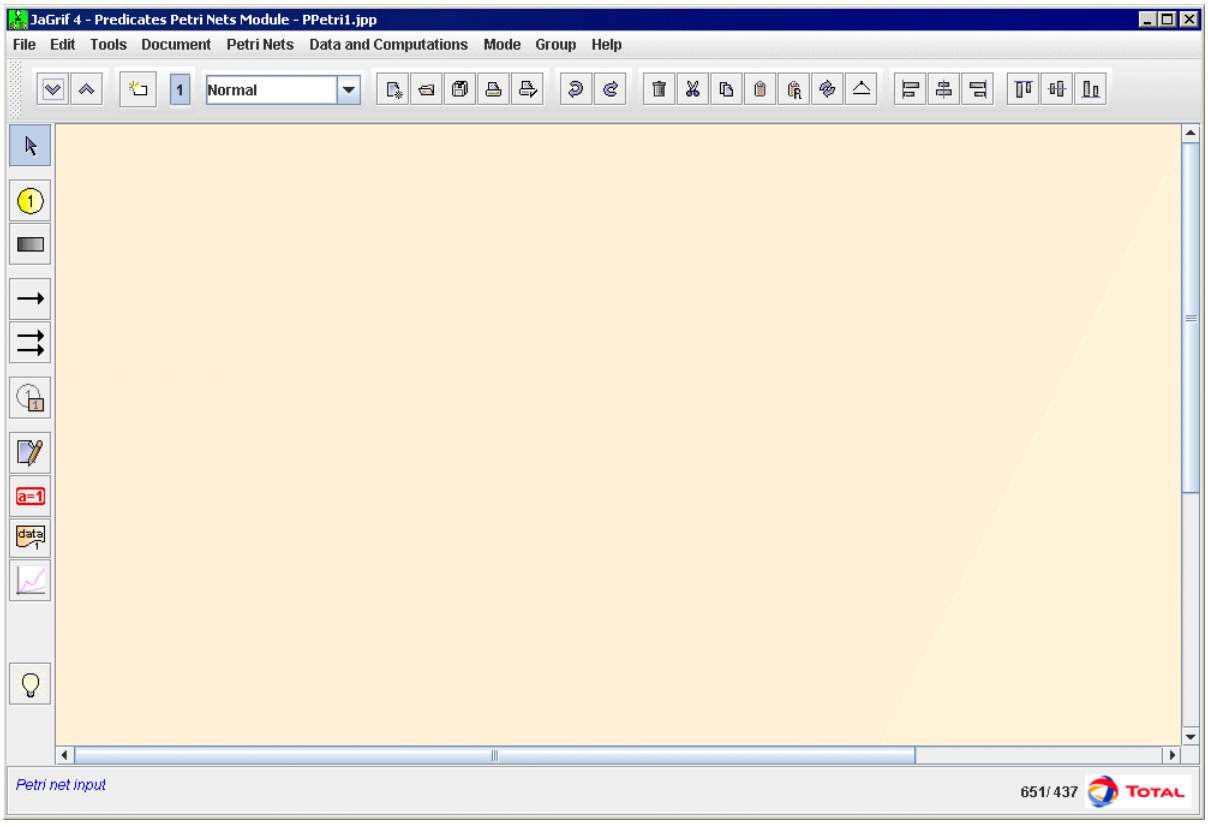
Table of Contents

1. Description of the interface

1.1. Main window of the Petri Nets with predicates module

The main window is divided into several parts:

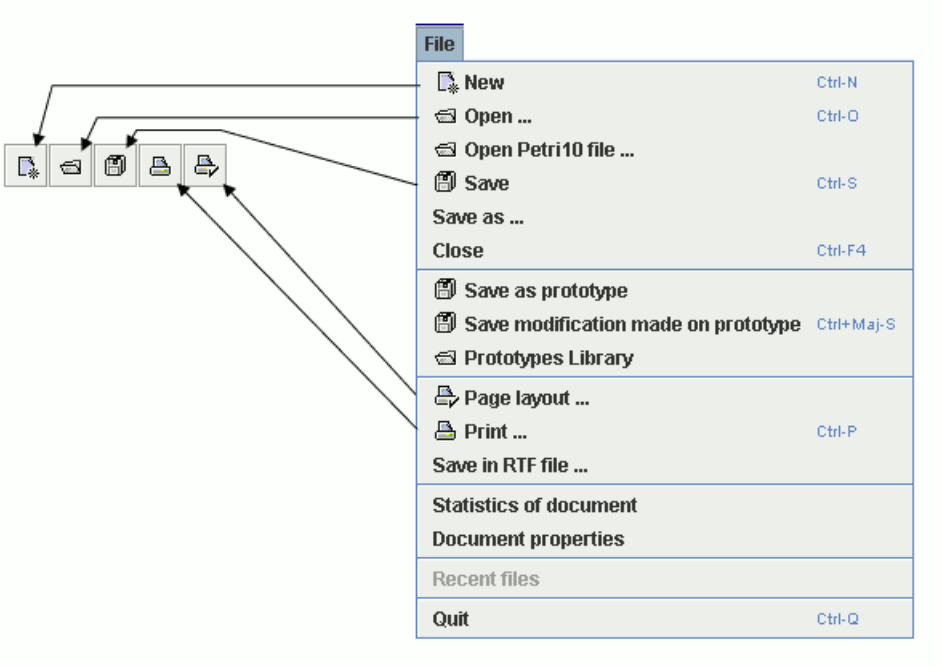
- **Title bar**: The title bar shows the names of the module and file being edited.
- **Menu bar**: The menu bar gives access to all the application's functions.
- **Icon bar (shortcuts)**: The shortcut bar is an icon bar (horizontal) which gives faster access to the most common functions.
- **Tool bar**: The tool bar (vertical) allows you to select the elements for modeling.
- **Input zone**: A maximum amount of space has been left for the graphical input zone for creating the model.
- **Tree**: A tree is "hiden" between input zone and tool bar. It enables to walk through pages and groups of the document.
- **Set of tables**: Tables are gathered in "hiden" tabs on the right.



1.2. Description of the Menus

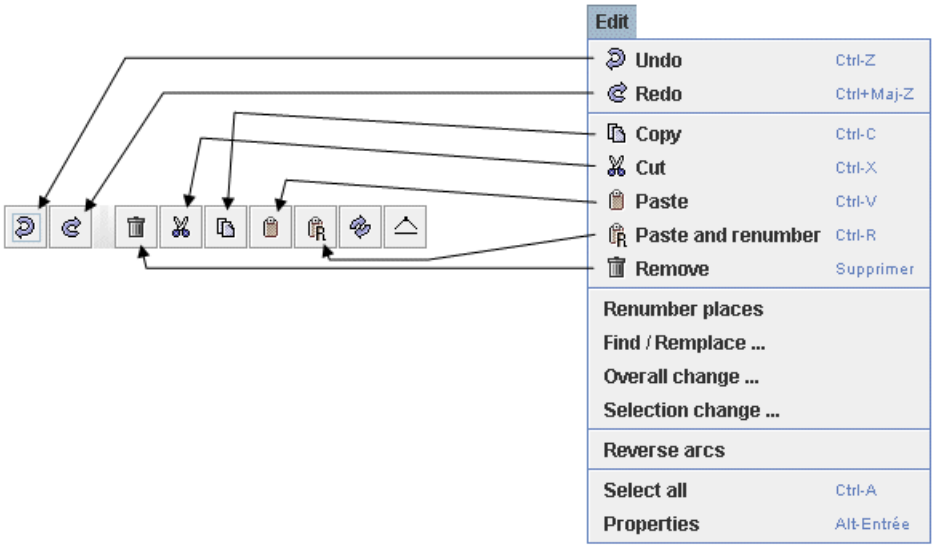
1. The **File** menu contains the standard commands used in this type of menu (open, close, save, print, etc.). The properties (name, creation date, created by, description, version) can be accessed and modified by selecting **Document properties**. The **Document statistics** provide information on the model's complexity. It is also possible to access a certain number (configurable) of recently modified files.

The icon bar just under the menus proposes shortcuts for most of the **File** commands:



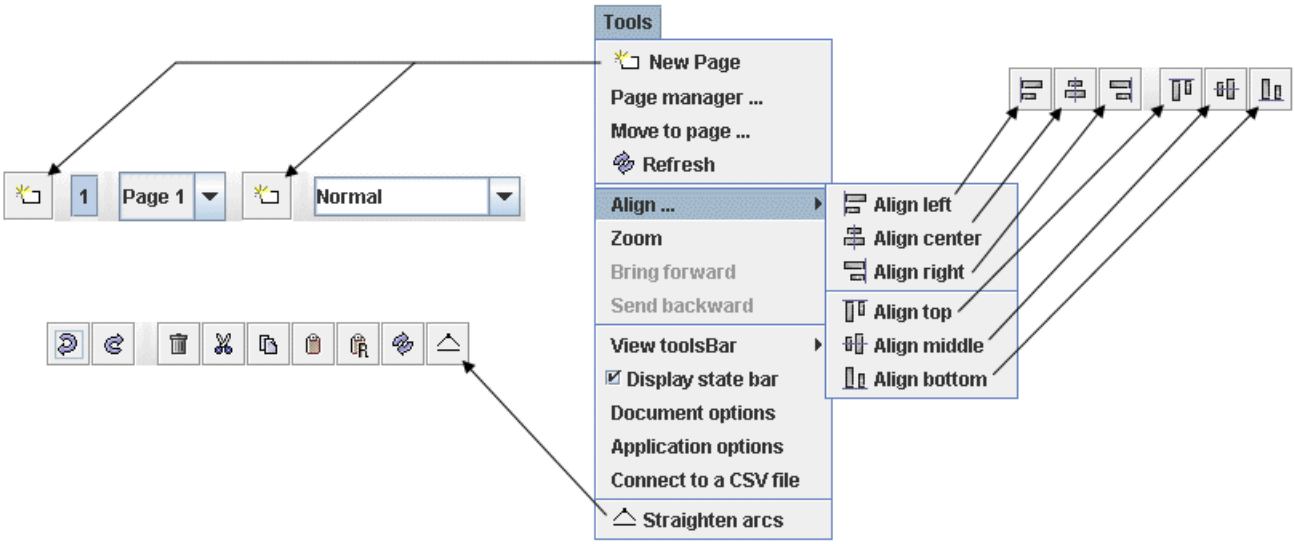
2. The **Edit** menu contains all the commands needed to edit the model being input graphically.

   The icon bar just under the menus proposes shortcuts for most of the **Edit** commands:



3. The **Tools** menu contains all the commands needed to manage the current model (page management, alignments, options, etc.).

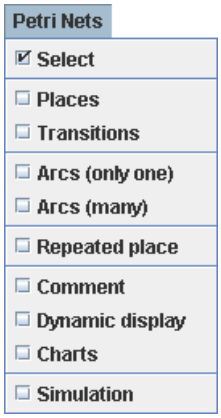
The icon bar just under the menus proposes shortcuts for most of the **Tools** commands:



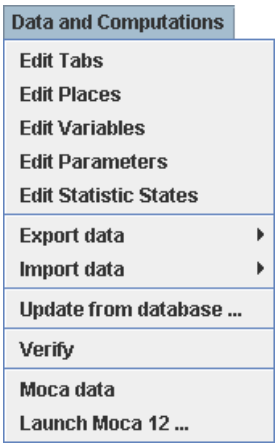
4. The **Document** menu gives access to all the documents being created or modified.

5. The **Petri Nets** menu contains all the commands needed to produce the graphical part of the current model.

   The vertical icon bar on the left of the application provides shortcuts for each of the **Petri Nets** commands (cf. vertical tool bar).



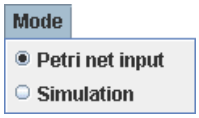
6. The **Data and Computations** menu is divided into two parts: data management (creation and management of the different parameters) and configuration/computation launch (computation time, desired computation, etc.)..



Notes:

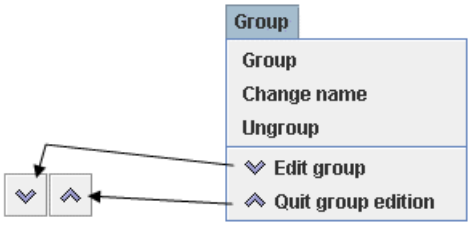
a. The **Verify** function detects any errors in the model: data without values (equal to NaN), places with an identical number, etc.
b. The **Export data** function exports tables.
c. The **Import data** function imports tables.

7. The **Mode** menu switches from **Input** mode to **Simulation** mode.



8. The **Group** menu concerns the input and management of submodels grouped into independent subassemblies.

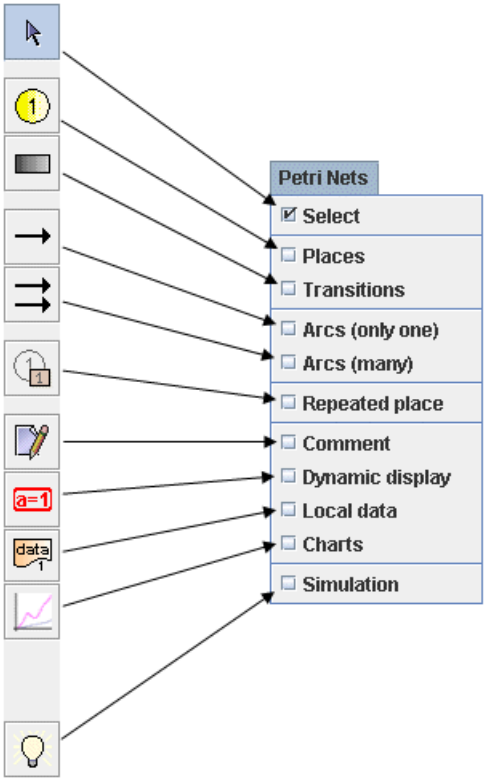
The icon bar just under the menus proposes shortcuts for two of the **Group** commands:



9. Finally, the **Help** menu accesses the on-line Help, the Help topics and to "About".

1.3. Vertical Tool bar

Each model used for operating safety has its own icons. All the Petri Nets graphical symbols are shown on the vertical icon bar on the left of the input window.



The vertical tool bar contains the following items:

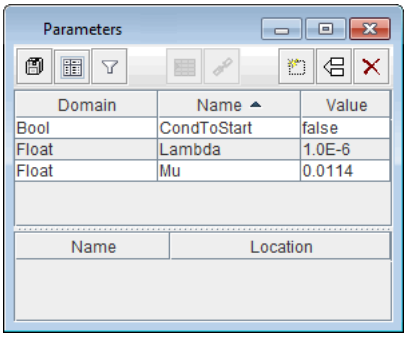
- **Places** represented by circles.
- **Transitions** represented by rectangles.
- **Upstream and downstream arcs** represented by arrows.
- **Repeated place (Shortcut)** to create links between several parts of the same model (on different pages or in different groups).
- **Comment** to add text directly on the chart.
- **Affichage dynamique** to display the value of an element in the model.
- **Local data** create variables linked only to one part of the model.
- **Charts** to draw curves representing computations on the model.

- **Simulation** to switch to simulation mode.

1.4. Data Editing Tables

1.4.1. Description of the Tables

To create or modify data (parameters, variables, etc.), tables are available in the **Data and Computations menu** and in tabs at the right of the view. All the GRIF 2014 data tables operate in the same manner.

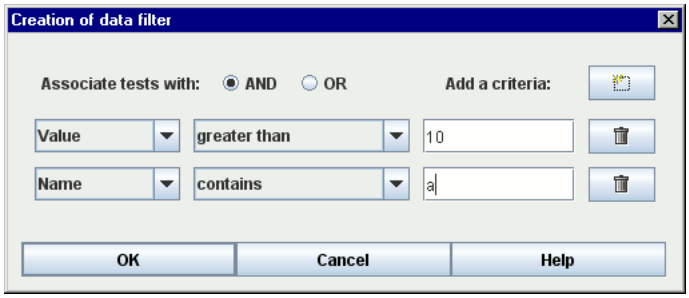


The data editing table/panel is divided into 3 parts:

- The top part containing the buttons.
- The main part containing the data table.
- The bottom part indicating what the selected data is used for.

	Saves the table in a text file.
	Opens the table in a text editor (that defined in the Options).
	Opens the column manager.
	When the **display selection** button is pressed, a click in the table leads to the selection in the input area.
	Displays the data filtering part.
	Multiple modifications made to all the selected data.
	Creates new data.
	Duplicate the selected data (ask a new name)
	Deletes the selected data (one or many).
	Enables data filtering or not.
	Defines the filter to be applied to the data.

Filtering allows you to display only what is necessary in a table. Several filtering criteria can be combined, as shown below:



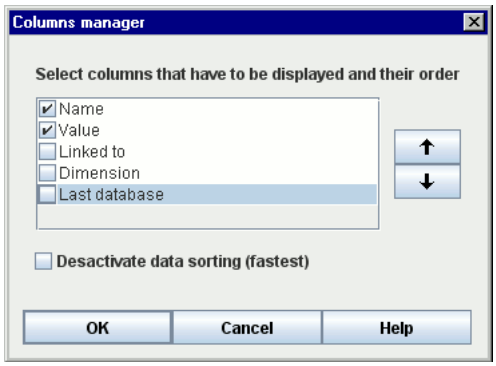
Select **AND** or **OR** to choose the type of association between each line (filter criterion). A line is a Boolean expression divided into 3 parts:

1. the first is the column on which the filter is used;
2. the second is the comparator;
3. the third is the value to which the data will be compared.

If the Boolean expression is true, the data will be kept (displayed); otherwise the data will be masked. When the filter is enabled its value is displayed between < and >.

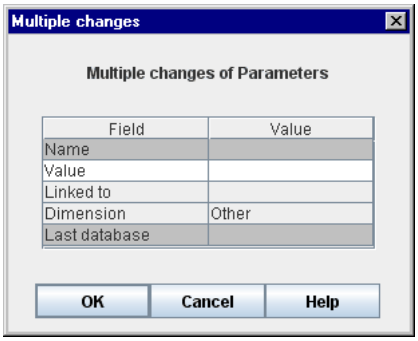
The data in a column can be sorted by double clicking the header of this column. The first double click will sort the data in ascending order (small triangle pointing upwards). The second double click on the same header will sort the column in descending order (small triangle pointing downwards).

A table can contain many columns, some columns may be unnecessary in certain cases. The "linked to database" column is unnecessary when no database is available. It is thus possible to choose the columns to be displayed and their order. To do this, click right on a table header, or click the **Columns Manager** button, the following window opens:



You can choose the columns to be displayed by selecting (or deselecting) the corresponding check boxes. The arrows on the right are used to move the columns up or down in the list to choose the order of the columns. The **Disable data sorting** check box disables the data sorting. This improves the application's performance with very complex models.

To modify data, double click the box to be modified. When several lines are selected (using the CTRL or SHIFT keys) changes can be made to all the selected data by using **Multiple changes**. A window then opens to allow you to make these changes.



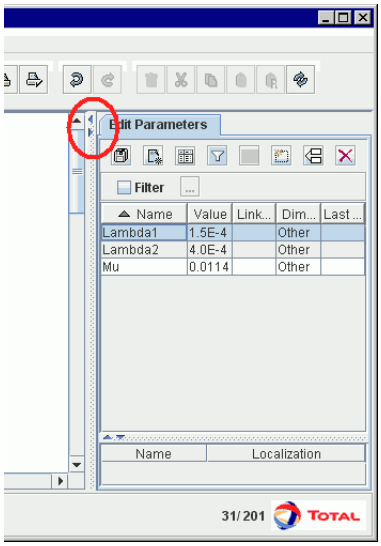
Items which cannot be modified are greyed. The white lines indicate that the selected data does not have the same value for the field in question. A new value can be entered which will be taken into account for all the selected data. The lines with no background colour indicate that all the selected data has the same value for this field (in this example the selected data is all "Float"); they can be changed to give a new value to all the selected data.

The bottom table in the data table indicates which elements in the model use the selected data. The first column of this table gives the name of these elements; the second indicates their location in the document (page, group). Clicking on a line in this bottom table opens the page where the element is located and selects the element.

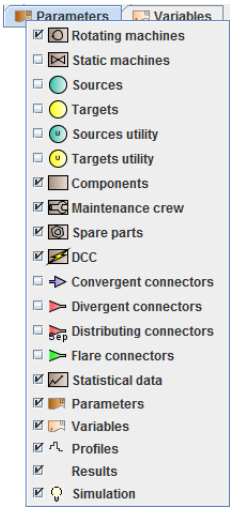
1.4.2. Table accessibility

As mentioned above, the tables can be accessed via the **Data and Computations** menu; in this case, each table is displayed in a separate window.

To avoid having too many windows open, all the tables are grouped together in tabs on the right-hand side of the application. This area can be hidden/displayed using the small arrows above the input zone.

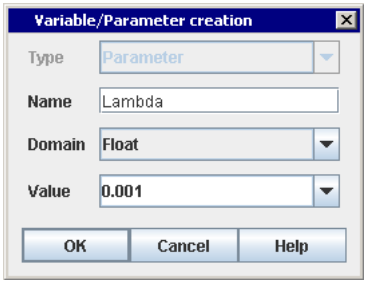


It is possible to choose the tables in this zone by right clicking on the tabs. A contextual menu appears, in which the user can select the tables s/he wishes to display.

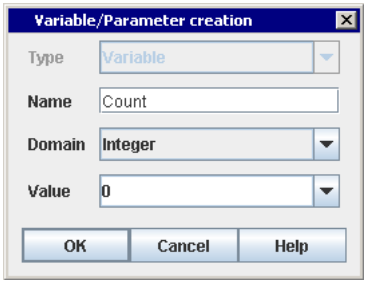


1.4.3. Data creation

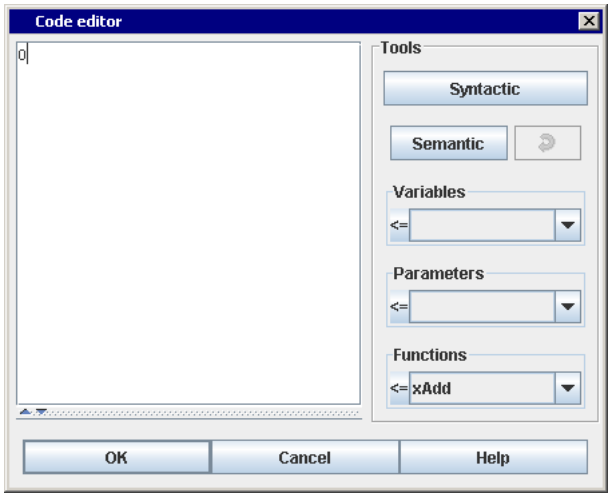
The **Parameter editor** is used to create "boolean", "integer" and "real" parameters. The following window is used to choose the name, domain and value of the new parameter. Its value must be entered later directly in the parameter table.



The **Parameter editor** is used to create variable defined with complex expression. Creation is made with the same window as parameter editor.



When variable is created, a double click in definition part opens code editor.



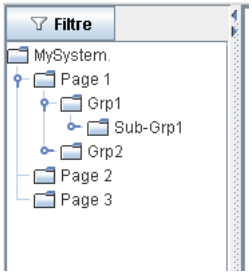
The code editor has three parts. The first is an editable text zone where you enter the code using Moca-RP syntax. Under this zone is a noneditable zone indicating any errors which may arise. The third is the **Tools** part which is a data entry aid.

The Syntax button makes a syntax change. The Semantics button checks the semantics. The errors are displayed in the bottom left part. Under the buttons there are drop-down menus giving access to the model's various data. Select the desired data then click the <= button to insert it in the code.

The **Functions** drop-down menu shows funtions that can be used in Moca (cf Moca User Manual).
The third menu display function that are available in MocaAdd.dll, for more information (cf Moca12 user manual)

A parameter can be transformed into variable and vice versa, by right-clincking on it and selectiong **Change to variable** ou **Change to paramètre**

1.5. Tree view



To help users to walk through the document (pages, groups ans sub-groups), a tree is available on the left of the application. By default, every element is displayed, you can use **Filter** button in order to select elements you want to display or not.

You can expand or collapse a node in a recursive way with a right click on the node.

As explained for tables on the right, you can "hide" the tree.

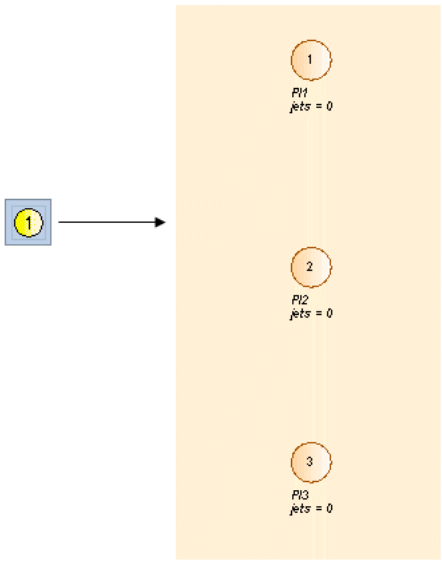
2. How to create a Petri Net

2.1. Entering the Net

2.1.1. Inputting Places

To input the different **Places**, select the corresponding symbol on the symbol bar. A new element is then created each time you click the mouse on the graphical input zone. Each of the places in the model has three parameters:

1. A **number**: Located in the centre of the places, they are automatically incremented. These numbers are the true identifiers of the places which will be used by the computation engine. This is why two places cannot have an identical number.
2. A **label**: A default label is assigned to each place ("Pli" for Place number "i"). Since each place normally has a very precise meaning for the user you are strongly recommended to assign a label which is more mnemonic than that given by default. This enables you to locate yourself better in the model and in the results file.
3. A number of **tokens**: By default it is equal to zero for each place created. In a Petri Net the presence (or not) of a token in a place normally corresponds to the presence (or not) of a specific status for one of the components in the system modelled by the Petri Net. All the tokens present at a given instant (Petri Net "marking") thus correspond to the global status of the system studied. The change in this "marking" is the dynamic aspect of the system.



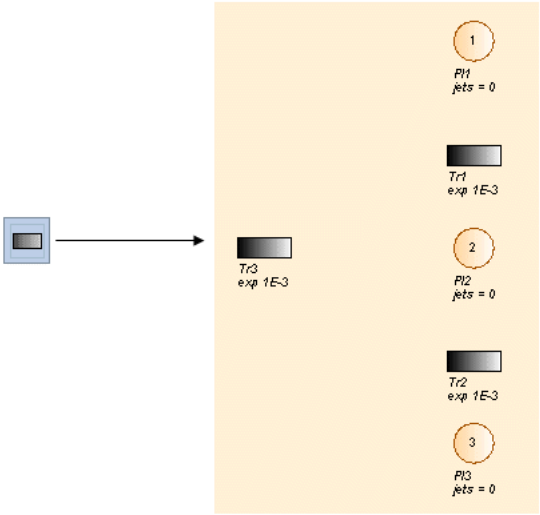
2.1.2. Inputting Transitions

To input the different **Transitions**, select the corresponding symbol on the symbol bar. A new element is then created each time you click the mouse in the graphical input zone.

In a Petri Net, the **Transitions** model the events which may happen at a given moment on the system studied (failures, tests, maintenance, etc.). Firing the transitions modifies the marking of the places to which they are linked by the arcs (upstream and downstream). This is used to simulate the system's dynamic behaviour.

When created, each transition has a default name ("Tri" for the transition entered in the ith position). Unlike the places, the transition number is unimportant since it is not used in the data file generated for the computation

engines. You are therefore highly recommended to assign them a mnemonic label (more than for the places) (cf. Section 2.2.3, “Configuring the Transitions”).



2.1.3. Inputting Upstream and Downstream Arcs

The "upstream arc" function describes part of the transition validation conditions (the other part is handled by the "guards" - cf. Section 2.2.3.3, “Guards tab”). They define the marking necessary for the upstream places to allow the transition to be fired.

The "downstream arc" function describes what happens at token transfer level when the transition is fired.

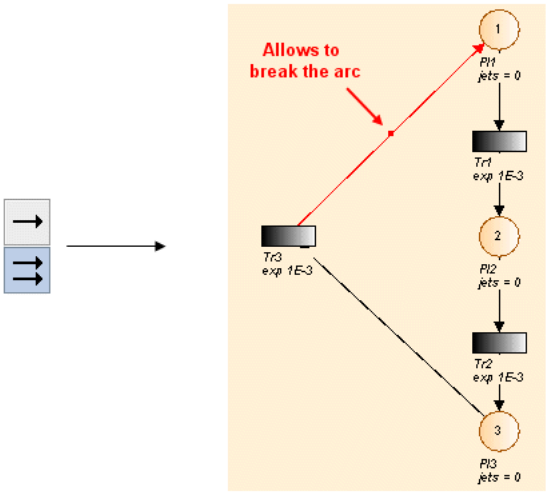
To input the upstream or downstream arcs:

1. select one of the two corresponding icons on the symbol bar:
   - the "single arrow" which is used to input only one arc at a time or
   - the "double arrow" which is used to input as many arcs as you wish.
2. select a start **Place** (respectively a **Transition**) by clicking on it with the left mouse button
3. drag the mouse (without releasing the button) to the final **Transition** (respectively the **Place**) and release the button.

It is the order "place" => "transition" or "transition" => "place" which determines the type ("upstream" or "downstream") of arc entered.

The result obtained is shown in the following figure. Upstream arcs have been drawn between place 1 and transition Tr1, then between place 2 and transition Tr2, and downstream arcs have been drawn between transition TR1 and place 2, then between transition Tr2 and place 3, etc. It must be noted that, unlike the "Reliability Networks, there are no bidirectional arcs for the "Petri Nets". However, it is often the case that a downstream arc must be drawn

between the same place and the same transition. In this case they can be superimposed and give the illusion that they are a bidirectional arc but in fact they are two separate arcs.

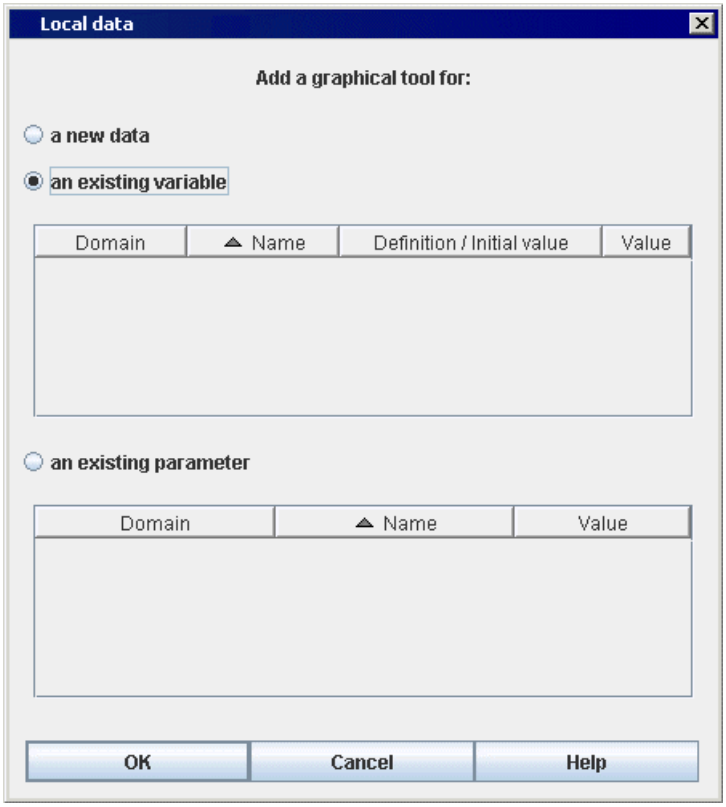


Notes:

1. For reasons of model appearance or legibility it may be useful to break down an arc into several parts. To do this, select the arc then move the small red square located in the middle of the segment.
2. It is also possible to straighten an arc using the command: **Tools - Straighten arcs** .

2.1.4. Inputting Local Data

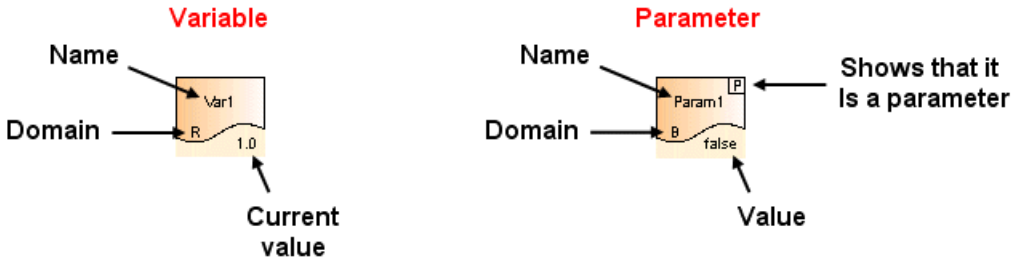
To add local data to the model, select the corresponding icon on the task bar then click left at the point on the model where the data is to be placed. A window then opens:



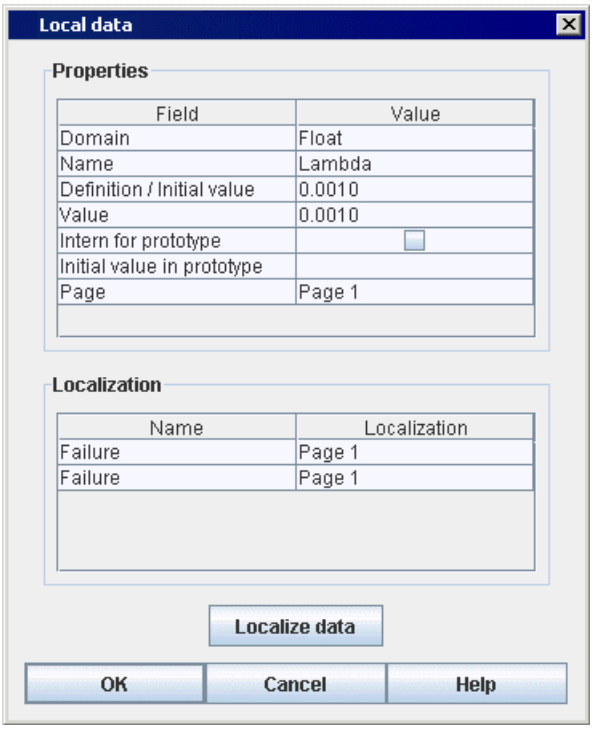
This dialogue box gives three choices:

1. Create **new data**: opens a window where then you can create new data or a new parameter.
2. Use **an existing variable**: creates a graphical representation of a variable.
3. Use **an existing parameter**: creates a graphical representation of a parameter.

Once created, local data is represented as follows:



Click right on the local data to access its properties. Some fields can be modified in this manner: Name, domain, initial value, etc.

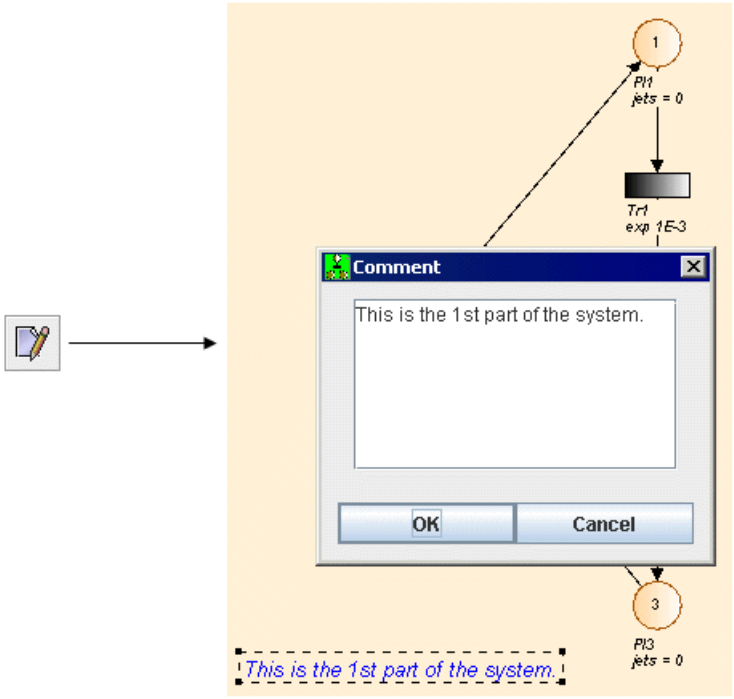


Note: the fields where the term "prototype" appears are only used to create a prototype (cf. appended document on prototypes).

2.1.5. Entering Comments

To add a comment anywhere on the chart, click the pencil icon and place yourself on a point in the graphical input zone. The **Comment** dialogue box opens where you can enter the desired comment.

Note: Character "%" is a reserved character, it must be type twice "%%" in order to display "%".



2.2. Configuring the Elements

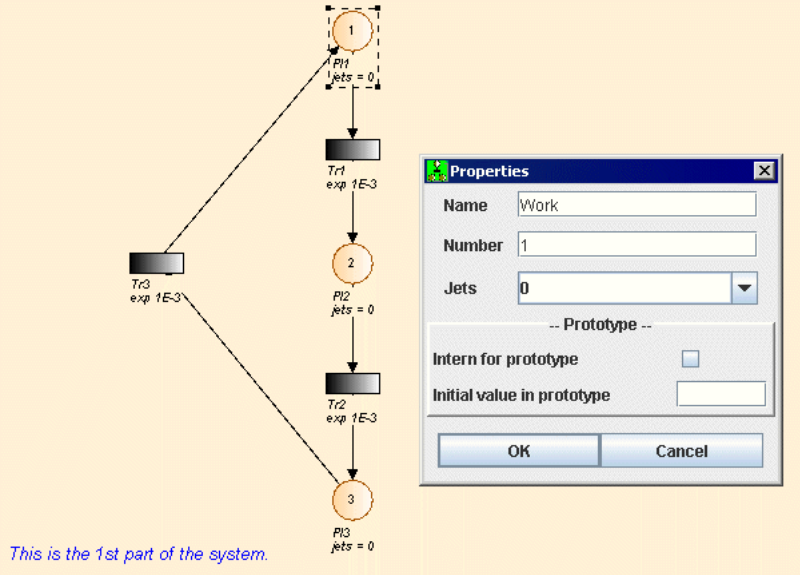
All the graphical elements can normally be edited with a double-click on them or using the **Edit - Properties** menu, or using the shortcut **Alt + Enter**.

2.2.1. Configuring the Places

The various place parameters are entered as follows:

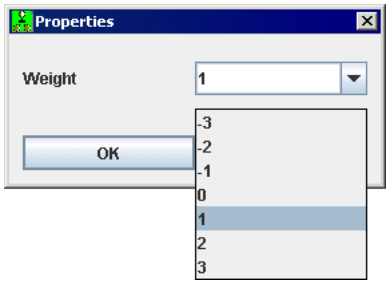
- Select the place using the right mouse button. A small panel then opens containing three fields to be filled in.
- Enter the place **Name** (optional but highly recommended). This description is "Pli" by default.
- Where necessary, change (carefully) the place **Number**. Other safer methods are described below.
- Enter the number of **Tokens** initially present in the place in question. This can be done manually but clicking the small black triangle is much easier.

Note : The **Prototype** part is only used to create prototypes (see prototypes appendix).



2.2.2. Configuring the Arcs

By default, the **Weight** of all the arcs (upstream and downstream) is "1". However, this can be modified. To do this, click right on the arc relevant concerned to display the editor shown below.



Clicking the small black triangle displays some possible values which can be selected with the mouse. For the downstream arcs, the weights are always positive and correspond to the number of tokens which will be added to the downstream place when the corresponding transition is fired. For the upstream arcs there are three cases:

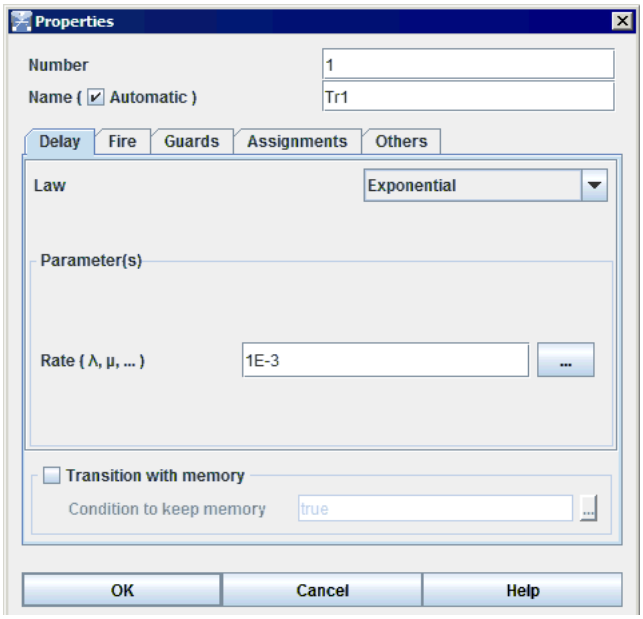
- Weight strictly positive: these are "normal" arcs which validate the transition when the number of tokens in the upstream place is greater than or equal to the weight of this arc. When the transition is fired, a number of tokens equal to the weight of the arc will be removed from the corresponding upstream place.
- Weight strictly negative: these are "inhibitor" arcs which inhibit the transition when the number of tokens in the corresponding upstream place exceeds the absolute value of the arc weight (e.g. 3 tokens for a weight of -3). This type of arc is graphically represented by a dotted line and does not modify the marking of the upstream place when the transition is fired.
- Weight "0": it is an arc which empties the corresponding upstream place when the transition is fired (whatever the marking of the upstream place before it is fired).

2.2.3. Configuring the Transitions

Entering the various transition parameters is more complex than those of places since it requires a good understanding of the MOCA-RP application's possibilities. You are highly recommended to consult the manual at this stage of the operations!

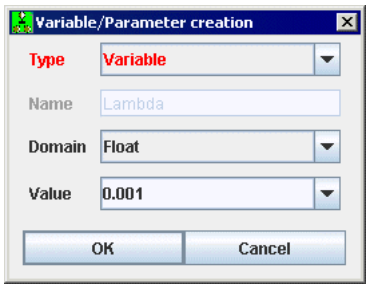
Click right to select the transition concerned. A dialogue window appears. The top part allows you to modify the **Name** and **ID** of the transition. The other part consists of 5 tabs for configuring the transition's behaviour.

2.2.3.1. Delay tab



The **Delay** tab indicates the transition delay law. Select the desired law (see the MOCA-RP User's Manual), and then enter the parameters of the law. In this example the default law is "Exponential" with 1E-3 as a parameter. For the parameters, you can enter a value, a name (a parameter) existing in the document, or a new name. In this last case the following window will appear to ask for the value and the domain. In the example, Lambda is a **Real Parameter** (invariable) with value 0.001.

It is also possible to use a **Variable** in the case of dynamic reliability.

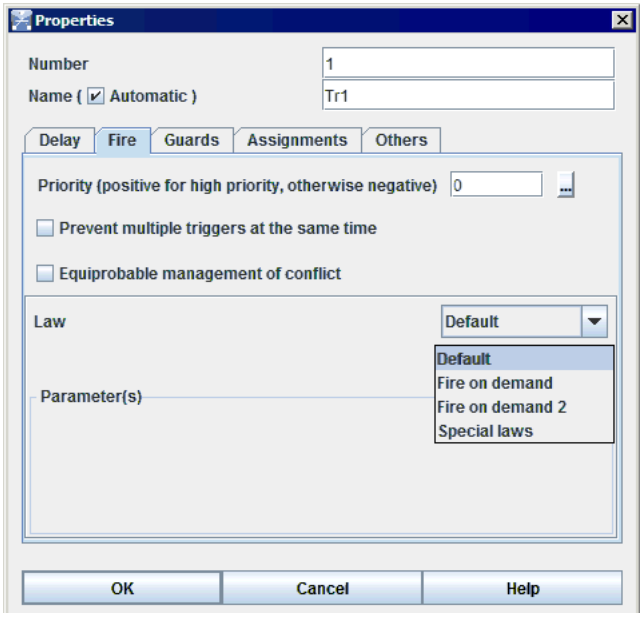


The **Transition with memory** option: Once the transition has been configured you still have to specify what happens when the transition is "inhibited" before it can effectively be "fired". This parameter is very important and two very different cases must be considered:

- Case 1: When the transition is inhibited, this indicates that the corresponding event can never happen (e.g.: an "old" component is replaced by a "new" one before it fails ==> the old component can now never fail). In this case, when the transition becomes valid again, it represents a new event (e.g.: failure of the "new" component) which has nothing to do with that which was "nipped in the bud" before it could happen. It is thus necessary to simulate a new firing delay for the transition: the transition has "no memory" of what happened previously.
- Case 2: When the transition is inhibited, this only indicates that the corresponding event is temporarily "suspended" (e.g.: a component repair is stopped due to it being night time ==> the repair will continue where it left off the next morning). In this case, when the transition becomes valid again, this only means that the delay relative to the suspended event continues. We must therefore use the delay which remained at the time when the transition was inhibited as the new delay before the transition is fired: the transition must retain the "memory" of this residual delay. The choice is made by selecting (or not selecting) the Transition with memory check box.

You may need to erase memory, do to this you can input a **condition to keep memory**. If this condition (an expression) get "false", the memory is "lost" and a new delay will be computed the next time the transition becomes valid.

2.2.3.2. Fire tab



The **Priority** check box is used to give the transition a priority level. If two transitions can be fired at a given instant, the one with the higher priority will be fired first. If the priority level is identical the transitions will be fired in the chronological order of their creation. Since 2010 version, priority can be a complex expression.

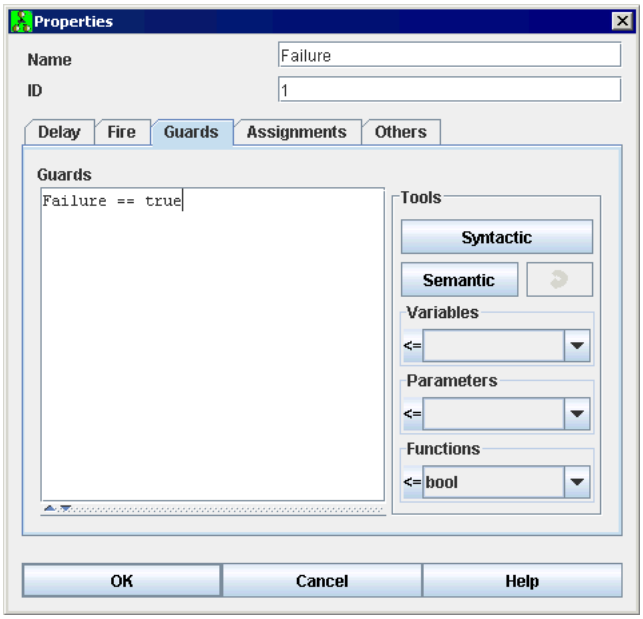
The **Prevent multiple triggers at the same time** prevents from multiple Drc 0 or IFA/IPA firing without time increase.

The **Equiprobable management of conflict** provides an equiproble firing frequency of 2 or more transitions when they are in conflict.

The **Fire** tab allows you to select the firing law for the transition:

- The **Default** law corresponds to the "normal" operation of the Petri Nets: the downstream places will be filled as defined in Section 2.2.2, “Configuring the Arcs”.
- The **On demand fire** law corresponds to the Moca-RP law of the same name: only one of the downstream places is filled after firing. The arguments of this law are the (N-1)th probabilities of happening in one of the N downstream places. The last probability is computed by the computation engine by making the 1s complement (see "Moca-RP User's Manual" for more details).
- The **On demand fire** law corresponds to the Moca-RP law of the same name: only one of the downstream places is filled after firing. The arguments of this law are the N-1 couple (probability,place). It is possible to specify probability to go in a given place. There are N-1 couple because last couple is computed by making the 1s complement, and using the place wich is not selected. (see "Moca-RP User's Manual" for more details).
- The **Special laws** can only be used in the very special case where the computation engine has been recompiled to take account of it (Cf. Moca-RP User's Manual).

2.2.3.3. Guards tab



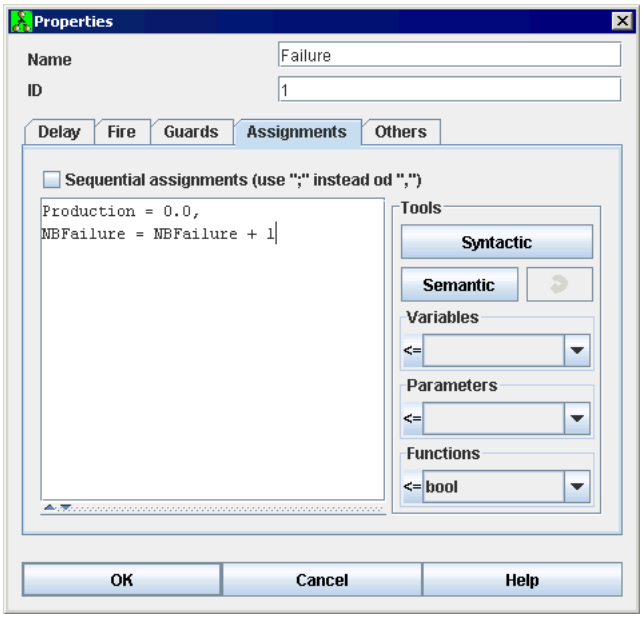
The **Guard** tab consists of a code editor where you can enter one (or more) guard(s) for the transition. A guard is a Boolean expression. The transition can only be fired if the guard is true.

The code editor has three parts. The first is an editable text zone where you enter the code using Moca-RP syntax. Under this zone is a noneditable zone indicating any errors which may arise. The third is the **Tools** part which is a data entry aid.

The Syntax button makes a syntax change. The Semantics button checks the semantics. The errors are displayed in the bottom left part. Under the buttons there are drop-down menus giving access to the model's various data. Select the desired data then click the <= button to insert it in the code.

The **Functions** drop-down menu shows funtions that can be used in Moca (cf Moca User Manual).

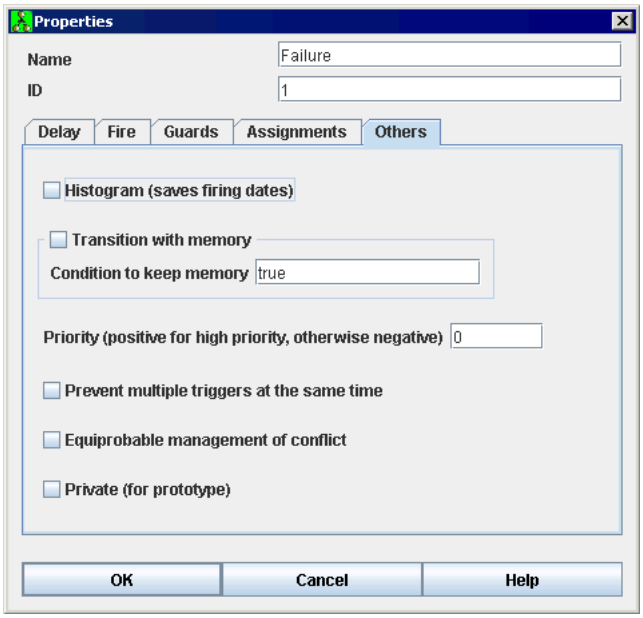
2.2.3.4. Assignments tab



The **Assignments** tab contains a code editor (identical to that of the Guards tab) for entering the assignments. The assignments are "played" when the transition is fired.

In Moca engine, usual behavior is the following: assignments are separate with "," and they are processed in parallel. You can ask for sequential processing of assignments using **Sequential assignments** check-box. In this case, each assignment must end with a ";".

2.2.3.5. Others tab



The **Others** tab contains 4 options:

1. Selecting the **Histogram** check box tells MOCA-RP to record all this transition's firing instants and to print them later.
2. The **Private** check box is only used to create prototypes (cf. prototypes appendix).

2.2.3.6. Addition of guards

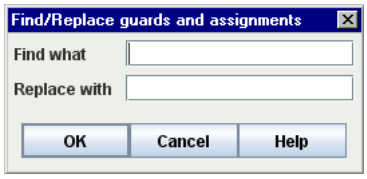
Once the transition's guards configured, it is possible to add one or more other guards. This functionality is available in the transitions table (**Edit transitions** tab, located in the right side of the application). To add one or more guards to one or more transitions, all we have to do is to select transition(s) to modify, then do a rigth click and finally select the **Add guard - With a "And"** menu or the **Add guard - With a "Or"** menu. **Add guard - With a "And"** menu will add guard(s) to the selected transitions by doing a "logical And" with the existing guards. **Add guard - With a "or"** menu will add guard(s) to the selected transitions by doing a "logical Or" with the existing guards. Gaurds to add are entered thanks to a code editor.

2.2.3.7. Addition of assignments

Once the transition's assignments configured, it is possible to add one or more other assignments. This functionality is available in the transitions table (**Edit transitions** tab, located in the right side of the application). To add one or more assignments to one or more transitions, all we have to do is to select transition(s) to modify, then do a rigth click and finally select the **Add assignment** menu. Assignments to add are entered thanks to a code editor.

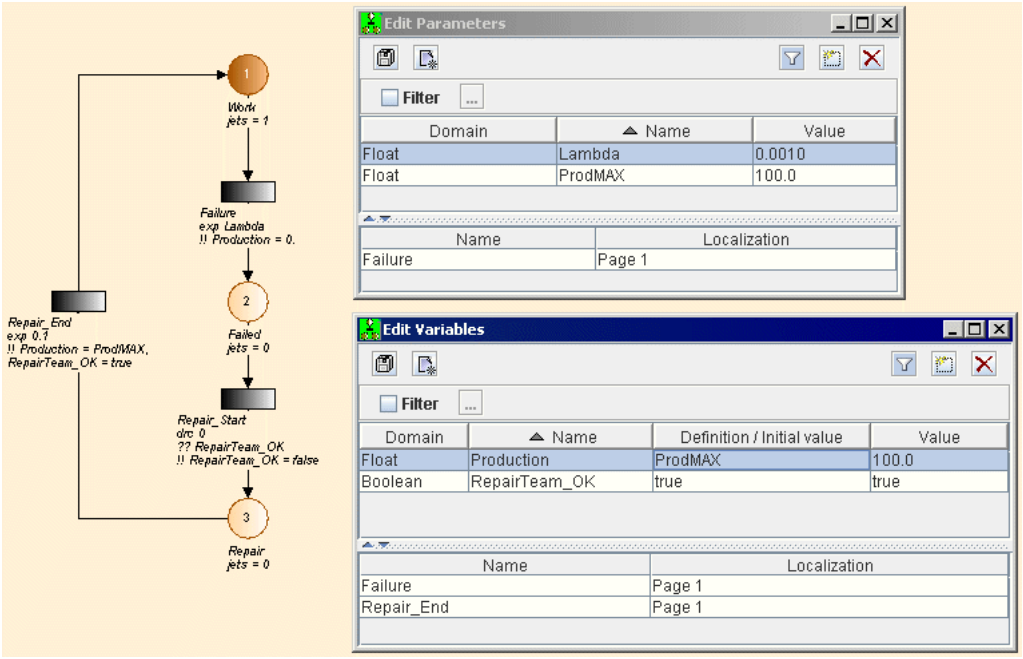
2.2.3.8. Find / replace on guards and assignments

Once the transitions configured, it is possible to perform a **Find / Replace** operation on guards and assignments. This functionality is available in the transitions table (**Edit transitions** tab, located in the right side of the application). To find and replace a string in the guards and assignments of one or more transitions, all we have to do is to select transition(s) to modify, then do a rigth click and select the **Find / Replace** menu and finally fill the following window :



- **Find what :** string to replace.
- **Replace with :** replacement string.

2.3. Petri Net example



The small Petri Net above represents the behaviour of a piece of equipment repaired by a maintenance team which is not necessarily available when the equipment fails.

This net has three places:

- **Work**: operating (place 1)
- **Failed**: failed, awaiting repair (place 2)
- **Repair**: being repaired (place3)

And three transitions:

- **Failure**: failure of the equipment
- **Repair_Start**: the equipment will be repaired
- **Repair_End** : the equipment is repaired and restarts

Here is how the model can be used to simulate the behaviour of a real piece of equipment

1. The **Work** place initially contains a token and the result is that the Failure transition is the only transition *valid* at the initial instant.
2. It will be fired when the component fails (delay fired randomly according to the exponential law assigned to this transition). The effect of this will be to remove the token from the **Work** place and to place one in the Failed place. In addition the **Production** variable will be reset to 0.
3. Since the arrival of the token in the **Failed** place is not sufficient to validate the **Repair_Start** transition, we must wait until the **RepairTeam_OK** variable (the message) input (or guard) to this transition becomes TRUE.

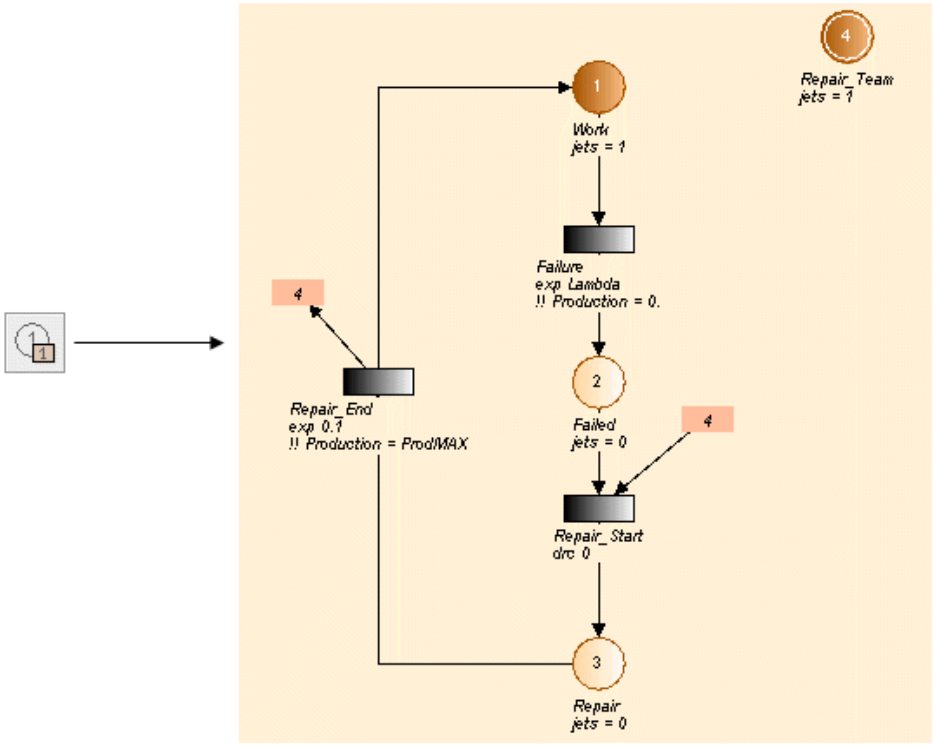
4. When the team of repairers is available (**RepairTeam_OK** message changes to TRUE) then the repair will immediately start since the delay law for this transition is a Dirac law with a zero delay.
5. When the **Repair_Start** transition is fired, the token is removed from the **Failed** place, a token is placed in the **Repair** place and the **RepairTeam_OK** message changes to FALSE (meaning "Repairers Unavailable"). Therefore, if another piece of equipment fails it must wait until the team of repairers is free before it can be repaired.
6. The arrival of the token in the **Repair** place validates the **Repair_End** transition which will be fired at the end of the repair delay (delay fired randomly according to the exponential law assigned to this transition).
7. When the **Repair_End** transition is fired, this removes the token from the **Repair** place, places a token in the **Work** place and again changes the **RepairTeam_OK** message to TRUE. The **Production** variable takes the **ProdMAX** value (100). Therefore we have returned to the initial state and the equipment is ready for the simulation of its second failure, etc.

We have taken exponential laws but any other type of laws could have been used (e.g.: the log-normal law for the duration of the repair). Also, the Dirac law has enabled us to mix a deterministic phenomenon with random phenomena without hesitation. Therefore, although it is very simple, this small model already gives an idea of the power of the Petri Nets associated with the Monte Carlo simulation.

2.4. Using repeated places (or shortcuts)

The concept of a **shortcut** (or repeated element) was introduced in the Petri Nets with predicates module for four main reasons:

- To link together portions of the model;
- To avoid graphicaly complex model, and keep readability;
- To simplify the use of the **Group** function (cf. below);
- To highlight what is essential and what is not.



In order to create a **repeated place**, select the **repeated place** icon in the toolbar, then click on place (here place number 4). The **repeated place** (or shortcut) is displayed as a number if place has no token, or as a colored rectangle if place contains tokens.

Then, the repeated place can be selected to be put where tou want. You link the shorcut to a transition with an arc. Operation can be repeated many times (2 times on figure).

The behavior of this Petri Net is the same as the one describe in above paragraph. Messages have been replaced by **Repair_Team** place. For the net describing equipment behavior, the **Repair_Team** place is auxiliary.

2.5. Page and group management

The use of shortcuts allowed us to obtain two Petri Nets which have no graphical link between them. They communicate only by **shortcuts**. This can be used, for example, to place each subpart on a different page:

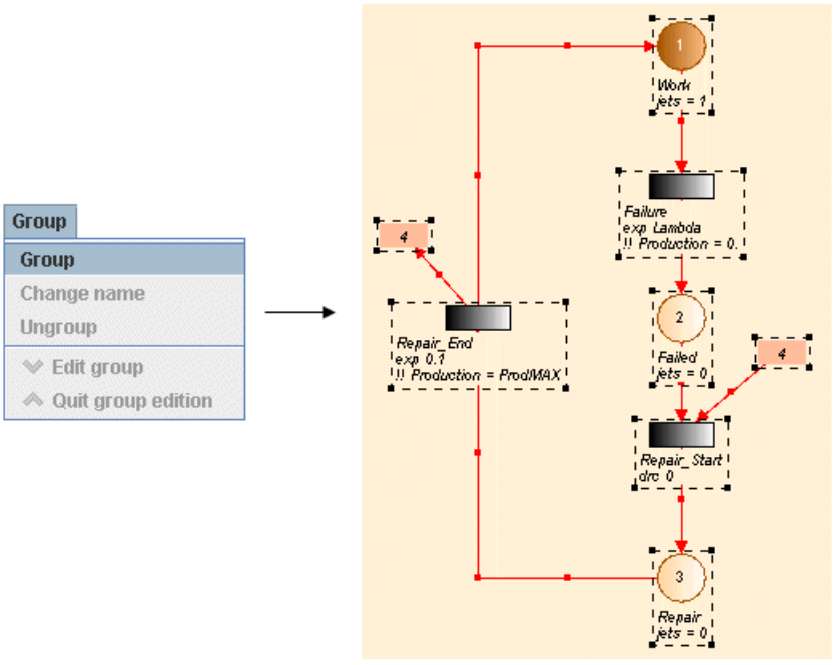
1. Create a new page by clicking the corresponding icon in the icon bar (or use menu **Tools - New Page**). A page number 2 is thus created.
2. Return to page 1 by selecting the page using the page selector in the ideographic command bar (or use menu **Tools - Page manager**).
3. Select the part to be moved.
4. Open menu **Tools - Change page**.
5. Select page 2 and click **OK**. The part selected is transferred to page 2 but it continues to communicate with page 1 via the **shortcuts**.

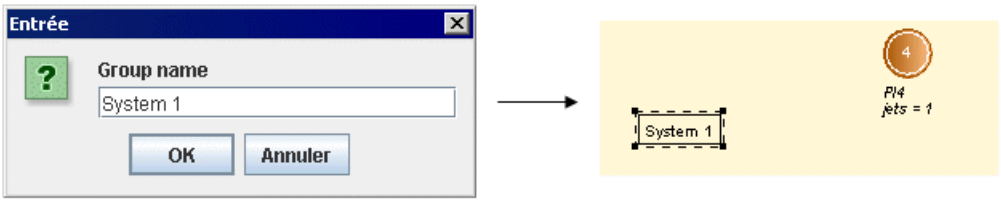
Note: For large models the division method described above is very useful.

Another possibility for entering large Petri Nets is to use the **Group** concept. This is made possible by the **shortcuts** and the fact that the data is global for a document. This allows quite separate subparts to be created:

1. Select a subpart.
2. Use menu **Group - Group**. A dialogue box then opens asking for the name to be given to the group being created.
3. Enter the desired name and click **OK** (e.g.: "System 1"). The group is created: the subnet is replaced by a rectangle assigned with the chosen name.

You can also create an empty group with **Group - New Group** menu or group tool in the left toolbar.





Each group can then be **edited**, **renamed** or **ungrouped** using the commands in the **Group** menu. The group can also be edited with a click right or using the "cursor down arrow" on the left of the page manager. In Edit mode, the submodel can then be modified as you wish. When the modification is terminated you return to the previous figure by exiting group editing by menu **Group - Quit Group Edition**, or using the "cursor up arrow" on the left of the page manager. It's also possible to choose a picture for a group by using **Group - Change Picture** menu.

Note: Groups can be grouped recursively.

2.6. Data Entry Aids

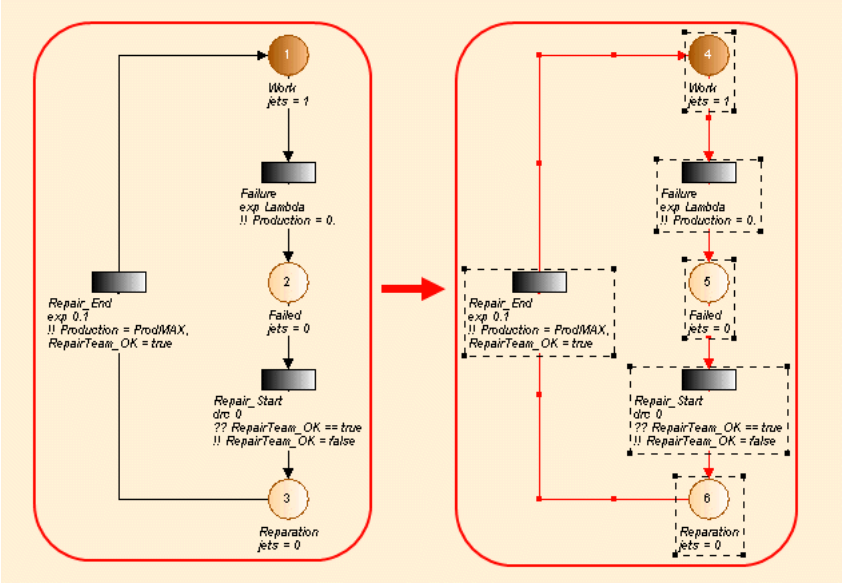
To simplify model creation the Petri Nets with predicates module has different data entry aids to automate time-consuming operations.

2.6.1. Copy / Paste / Renumber (without shortcut)

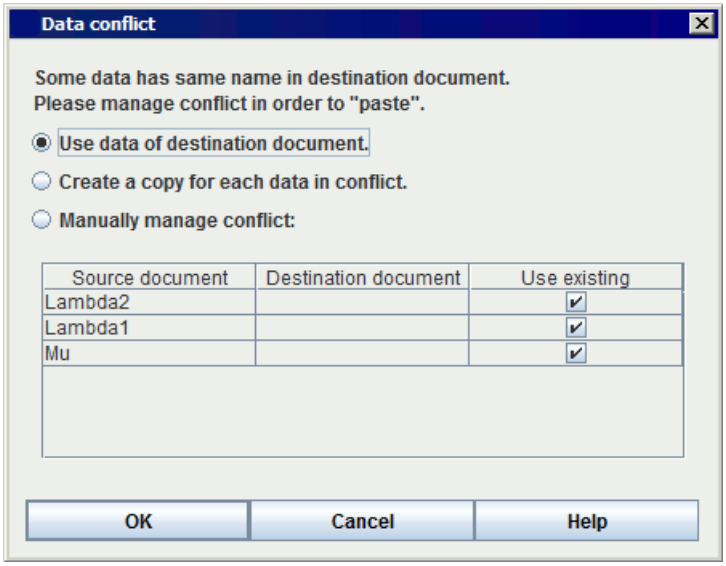
To assist with the entry of the repeated parts of the Petri Nets "Copy / Paste and Renumber" mechanisms have been provided. This operation is carried out in 6 steps:

1. Select the part to be copied.
2. Click the **Copy** icon, or use menu **Edit - Copy** or the shortcut Ctrl + C.
3. Click the **Paste and Renumber** icon, or use menu **Edit - Paste and Renumber** or the shortcut Ctrl + R.
4. A window appears where you choose the start number for the renumbering.
5. The previously selected part is copied and the copy is selected.
6. Move the copy to the desired location.

We then obtain the net shown in the figure opposite/ below: Places 1, 2 and 3 of the original have been transformed into 4, 5 and 6 for the copy.



When copying to a new document, any data conflicts are handled in the following window:



This window shows all the data which has the same name in the source document and the destination document. There are three choices:

1. Use data of destination document, this will replace the occurrences of the data in the source document by the data with the same name in the destination document.
2. Create a copy for each data in conflict, this will replace the occurrences of the data in the source document by a copy with a name with the suffix "copy".
3. Manually manage conflict, this allows you to choose whether you use the existing data or not, depending on the data. You can also specify the name of the copy by double clicking on the box in the "destination document" column. The names in this column are normally masked when the **Use existing** check box is selected, since it is the data which is already in the destination document which will be used.

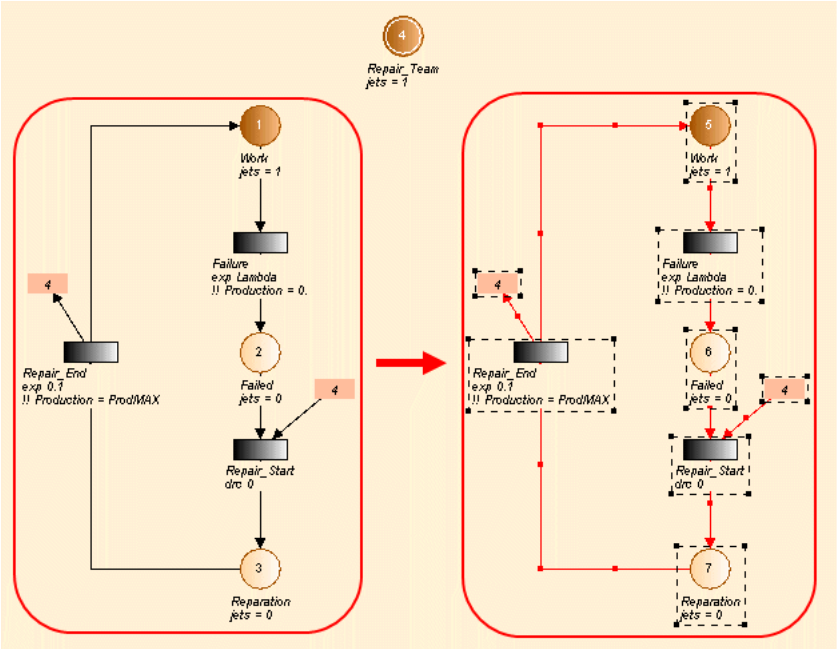
2.6.2. Copy / Paste / Renumber (with shortcut)

The "Copy / Paste and Renumber" command creates new "instances" i.e. new subcharts similar to the subchart copied:

- Same graphical structure;
- Same probability law;
- Same guards and assignments;
- Same labels but different places and transitions from those copied.

When creating a new instance it is therefore necessary to make the distinction between the places repeated "internally" (specifically belonging to the subchart copied) and the places repeated "externally" (attached to a place not belonging to the subchart copied). Only the places repeated "internally" must be renumbered. The places repeated "externally" must normally not change numbers.

Remarque: Only the "shortcuts" of the places present in the selection to be renumbered (differentiated from the others by a double circle) are actually renumbered. Therefore the corresponding "input" shortcuts (represented by small rectangles) are renumbered in their turn. The result is that the "shortcuts" on the places not belonging to the selection are not renumbered.

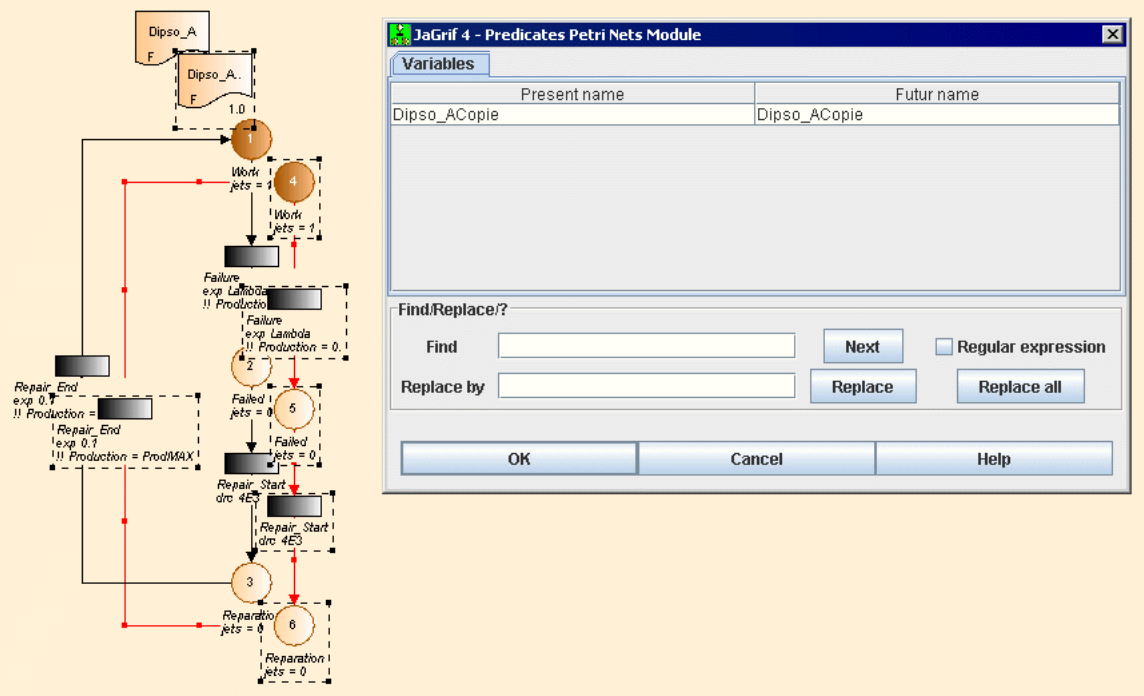


The above figure explicitly illustrated the mechanism. Repeated place number 1 (internal to the original subchart) has been renumbered as 5 on the copy. However, the numbering of the corresponding shortcuts to place 4 (external to the copy) has not changed.

You can navigate between an element's different shortcuts, using menu **Tools/Navigate to shortcuts**. A window opens and displays the list of shortcuts. Clicking on a shortcut automatically positions the view on this shortcut. You can return to the original element by clicking on its name at the top of the window.

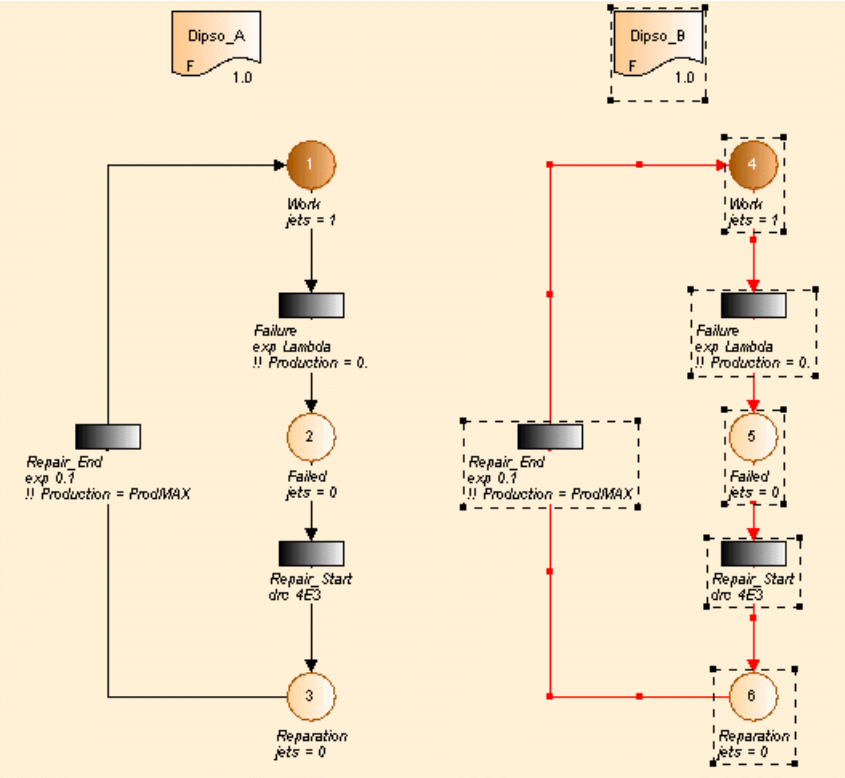
2.6.3. Copy / Paste / Renumber (with local data)

When you wish to perform a "Copy / Paste and Renumber" operation by including local data in the copy, a dialogue box will open during the "Paste" step. This window allows you to rename the copy of the local data.



The above figure shows that the name assigned by default to the copy of the local data is "original name" + "Copy". This name, which is not necessarily well-adapted, can be modified.

Once the copy is made, new local data is created. Where necessary, if the definition of the original local data is linked to the associated net (included in the copy), then the definition of the local data copy the will remain the same with respect to the net copy.



The above figure clearly shows that new local data has been created. It is called **Dispo_B**. **Dispo_A** is equal to the number of tokens present in place no. 1, **Dispo_B** is equal to the number of tokens in place no. 4.

Note: The main advantage of the local data is that of being able to perform "Copy / Paste and Renumber" operations such as those described above.

2.6.4. Ordinary Copy/Paste

In addition to the "Copy / Paste and Renumber" command there is an ordinary "Copy / Paste" function. It is used to make a single copy without renumbering. We thus obtain double elements which, from a formal viewpoint, is incorrect but which must be temporarily tolerated to simplify data entry.

Where possible, the "Copy / Paste and Renumber" function must be used in preference to the simple "Copy / Paste" function to minimise the risk of errors. But when it is used you must take the necessary precautions to re-establish the correct numbering to eliminate the duplicates.

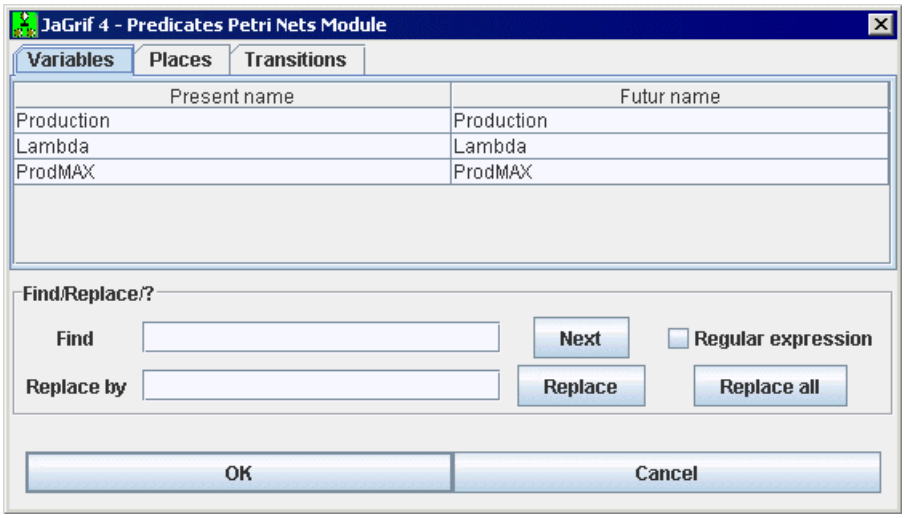
2.6.5. Overall change

When creating the Petri Nets it may be necessary to change a large part of the elements in the models: changing the names, numbers, etc. The "Replace all" function in the **Edit** menu allows you to perform overall changes:

- Use the **Edit / Overall changes** function.
- Choose the type of elements to be modified among available tabs.
- The "Find / Replace" part changes a character string present in one or more variable labels, place labels or transition labels. It is replaced by the string entered in the "Replace" part.
- The "Renumber" part only concerns the places. It is used to change place numbers. You indicate a **Start** number then specify a constant **Step**, or **Add** a constant value to the current numbers.

- Click **OK** to return to the chart. The changes are validated.

Note: The name changes and renumbering can be done manually if the necessary precautions are taken (avoiding duplicates, etc.). You click the **Future number** or **Future name** column and enter the change. Do not forget to validate it with the "ENTER" key.



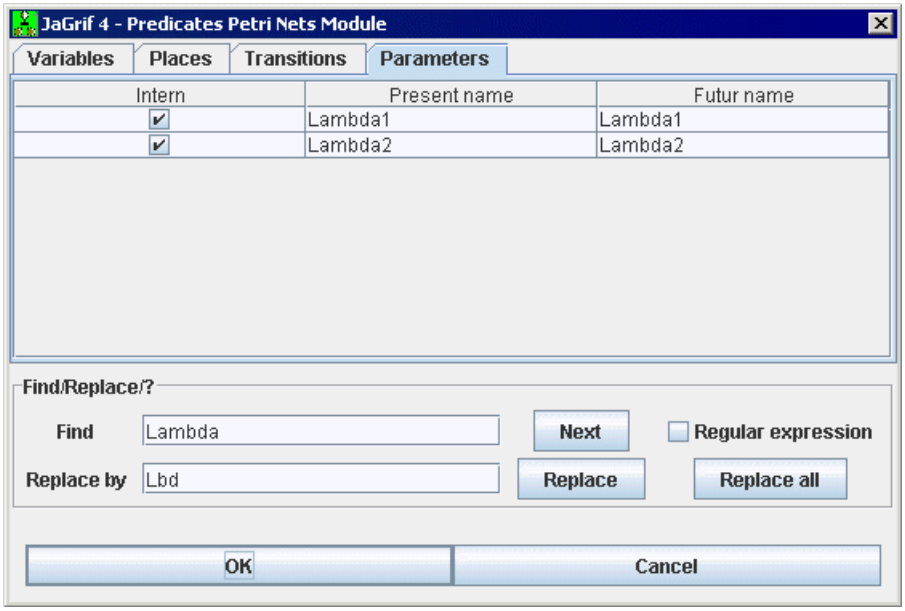
2.6.6. Selection change

The "Selection change" function is equivalent to a "Replace all" but only applied to the selected elements. The only difference is that we must make the distinction between "internal" and "external" variables/parameters.

- "Internal" variable/parameter: only used within the selection

- "External" variable/parameter: used within the selection but also used elsewhere in the model.

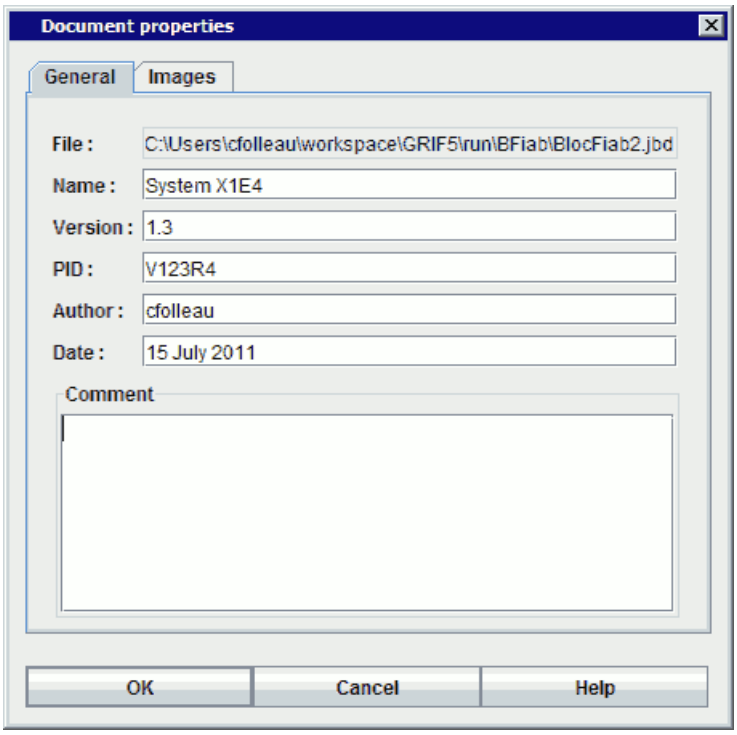
Only the "internal" elements can change their name. If a variable or a parameter is recognised as being "external", the check box in the **Internal** column must be selected before it can be modified. The change will only affect the selected part. Everything outside the selection will remain unchanged.



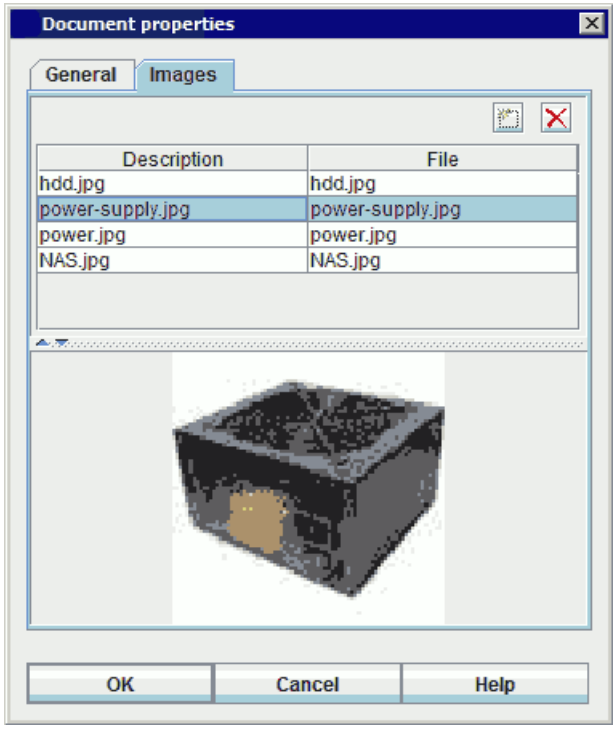
In the above example only the name of parameter "Lambda2" will change (in the "Replace selection" sense) since it is "internal" to the selection. A new parameter called "Def2" (with identical value) will be created and will replace "Lambda2" in the model. The other parameter, which is not "internal", will remain unchanged.

2.6.7. Document properties / Images management

File - Doucument properties menu enable to save information about document : name, version, comment, ... These informations are available in **General** tab.



Images may be very useful to represent sub-system. GRIF 2014 enables to save images that can be used in different parts of software (groupes, prototypes, ...). Images management is made in **Images** tab.



To add a new picture into document, use  icon. A double click in **File** column enables to select an picture (jpg, gif or png). A double click in **Description** column enables to give a name or a description to selected image.

Once in document, picture can be linked to a groupe with **Group - Picture change** menu.

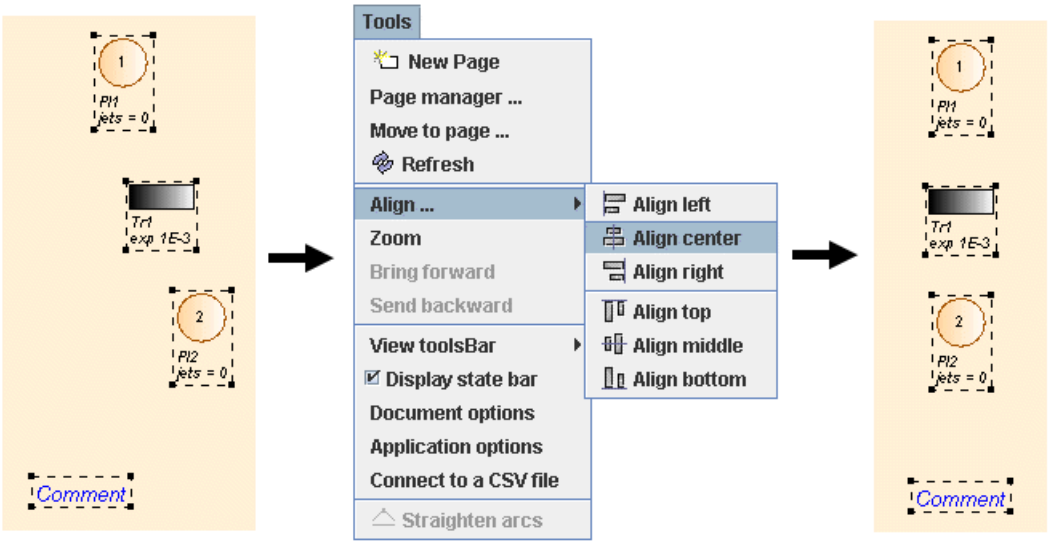
Images are saved indide document, pay attention to picture size. Because images are inside document, you have to re-add picture if picture is modified erternaly.

2.6.8. Alignment

To improve the legibility of the model the selected elements can be aligned vertically or horizontally. To do this, use the **Align** command in the **Tools** menu.

The following figure shows how the command works. For example, to align selected places and transitions vertically, proceed as follows:

1. Select the elements (places, transitions, comments, etc.) to be aligned;
2. Go into the **Tools** menu and select the **Align** function;
3. Choose the type of alignment: **Align center**;
4. Click left on the mouse.



Similarly, to align elements horizontally select the type **Align middle** which aligns the ordinates while keeping the abscissa constant. The principle is the same as that described above.

2.6.9. Multiple selection

It may sometimes be useful to select several elements located in the four corners of the input zone. To simplify this type of selection click on each of the desired elements one by one while holding down the Shift key on the keyboard.

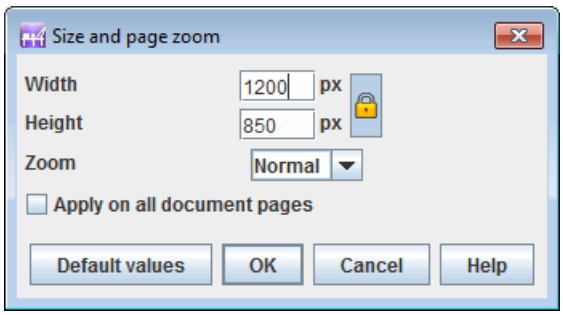
2.6.10. Selecting connex (adjacent) parts

It is sometimes difficult to select an additional part of a model. To simplify the selection process, select a graphical element then use menu **Select connex part** in the **Edit** menu. The additional part can be selected directly by clicking on the element while keeping the Control button pressed.

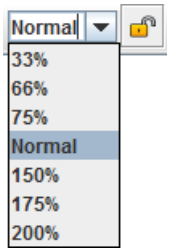
2.6.11. Zoom and page size

When creating a model, if the page size is not big enough, it can be changed using the menus : **Increase page size** (**Control**+**Keypad +**), **Reduce page size** (**Control**+**Keypad -**), **Page size** (**Control**+**Keypad /**) under the **Tools** menu.

The **Page size** menu allows the user to edit the page dimensions directly.



Page zooms can be modified either by using the toolbar menu:



Or by selecting the display and using **Control+mouse wheel scroll up** to zoom or **Control+mouse wheel scroll down** to zoom out.

The padlock on the toolbar is used to apply the zoom to the current page or to all pages in the document.

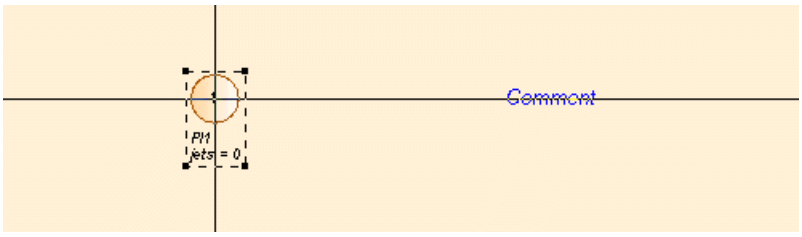
	The zoom applies to all pages in the document.
	The zoom is applied only to the current page.

Note that if an element is selected on the page, the zoom will centre the page on that element.

2.6.12. Cross hair

To be able to create an ordered and legible model quickly, the **cross hair** can be used to align the different elements with each other (but less accurately than the **Align** function in the Tools menu). The **cross hair** is enabled (or disabled) in the **Graphics** tab of the **Option** menu.

The following picture show how to quickly align two element of the model.



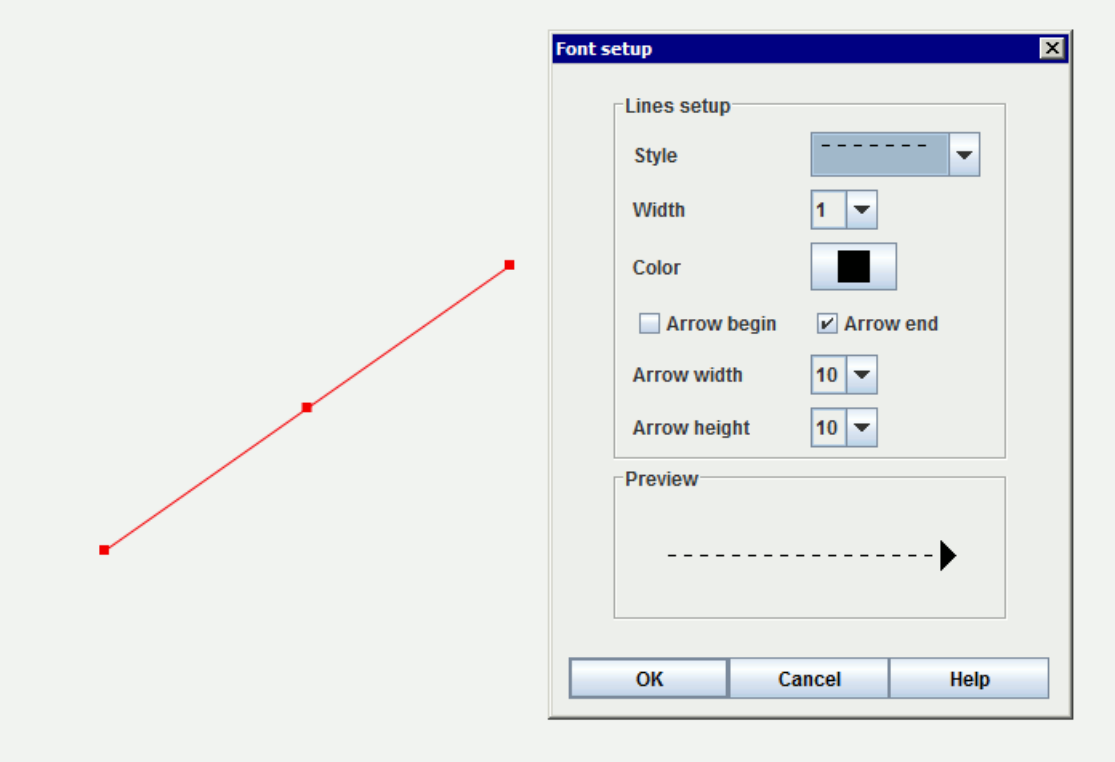
In order to align horizontally, select **Align au middle** which align keeping constant abscissa.

2.6.13. Gluing/Associating graphics

When objects are where you want, you can glue a set of object by right-clicking and selecting **Glue**. This command create a group (a graphical one, not a hierarchical one) with selected objects, so that moving one moves the others.

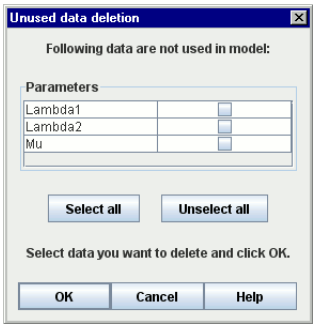
2.6.14. Line

To be able to draw a line, polyline or arrow, the **Line** can be used. Draw the line and edit properties of line to make an arrow.



2.6.15. Table Cleaning

Data may not be used anymore, it can be used usefull to delete every unused data. To facilitate removal, use **Data and Computations / Unused data deletion** menu.



This window displays unused data. Select data you realy yan to delete and click OK.

2.6.16. Prototypes

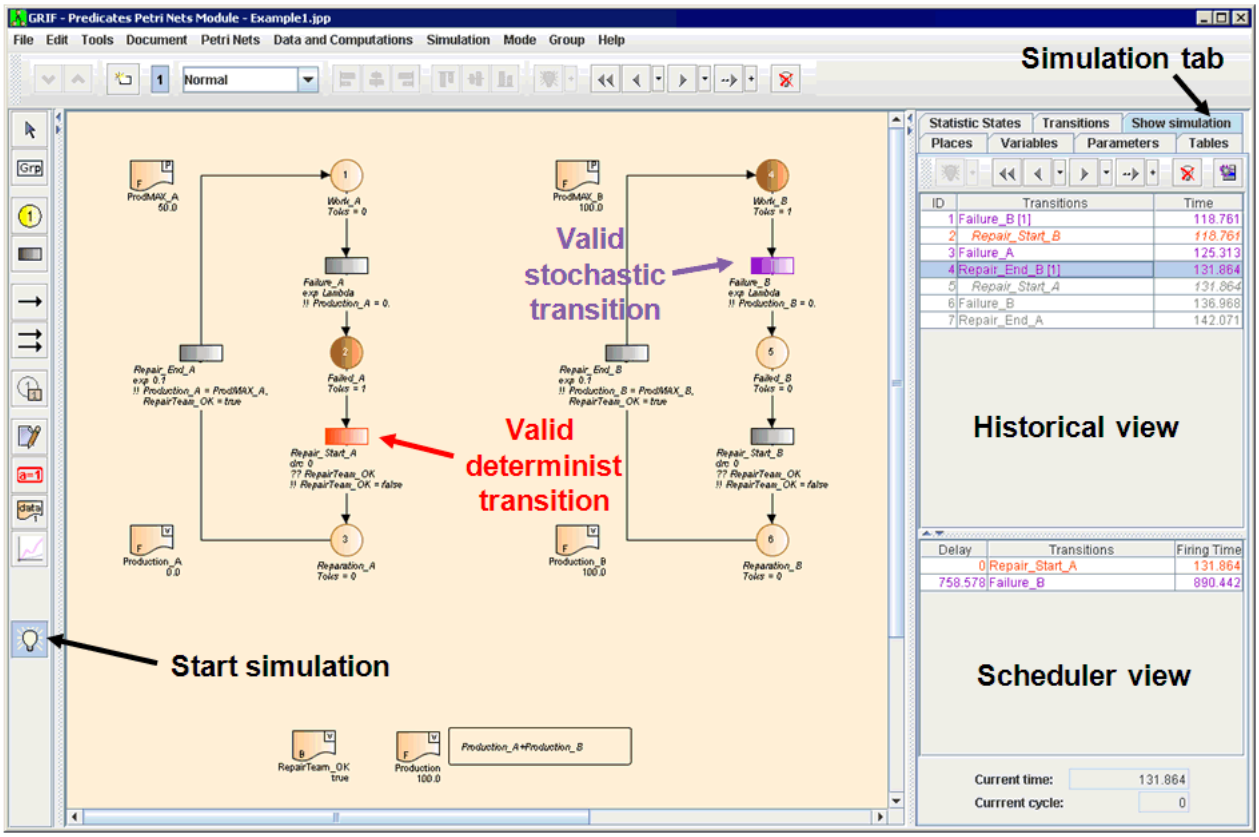
cf. appendix about prototypes

3. Simulation interactive

3.1. Introduction

One of the most important characteristics of the GRIF interface is that it allows the user to manually simulate the behaviour of the network s/he has just created. It is therefore easier to understand, debug or explain a model

When the simulation is launched, transitions can be fired to understand how the model reacts, to go back a stage, to replay a sequence of a given event, to view the status of the components, the quantities of flow circulating, the value of each variable, etc. at any time.



3.2. Simulation panel

The interactive simulation panel has four parts :

- Right at the top, a **toolbar** groups together the functions that enable a user to start, stop, configure and play the simulation.
- Just below is a **history** of the fired transitions.
- After that is the **scheduler**, which contains the list of the fireable transitions ranked in order of firing date.
- Right at the bottom, a panel displays the **current time** and the **current cycle**.

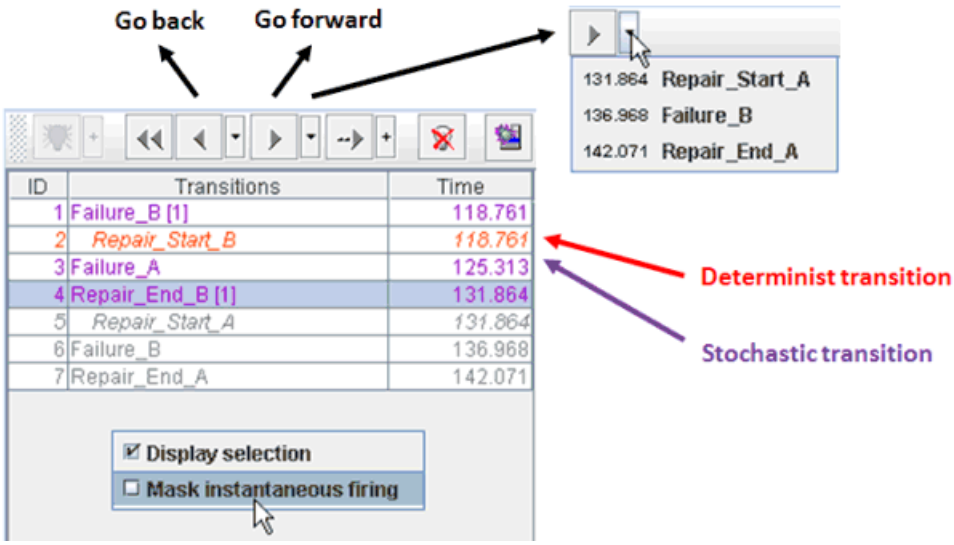
The toolbar contains the following functions :

	Gives access to the interactive simulation mode.
	

	Opens a simulation configuration window. This point is covered in the section Section 3.5, “Configuring the simulation”.
	Goes back to initial state.
	Goes back one step.
	Rolls back a fired transition. The simulator will go back to the step before the selected transition.
	Replays a step in the fired transition history.
	Replays the fired transition history up to the selected transition.
	Plays the next transition in the scheduler.
	Goes forward in the simulation to a certain time or to the firing time of a given transition. The simulator will stop one step before the transition or time selected.
	Exits simulation mode
	Modifies simulation options during the simulation. See section Section 3.5, “Configuring the simulation”.

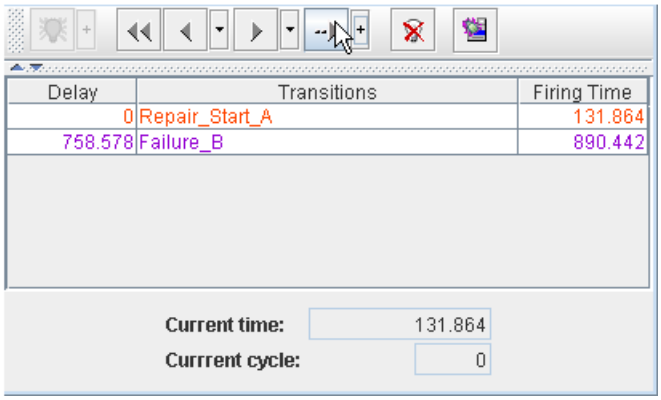
3.3. Simulation history

The simulator contains a panel that displays the history of fired transitions. The user can "explore" the history or take things back a step to modify the running order. A right click on a transition in the history displays a contextual menu that enables the user to display/hide the instantaneous transitions among other operations.



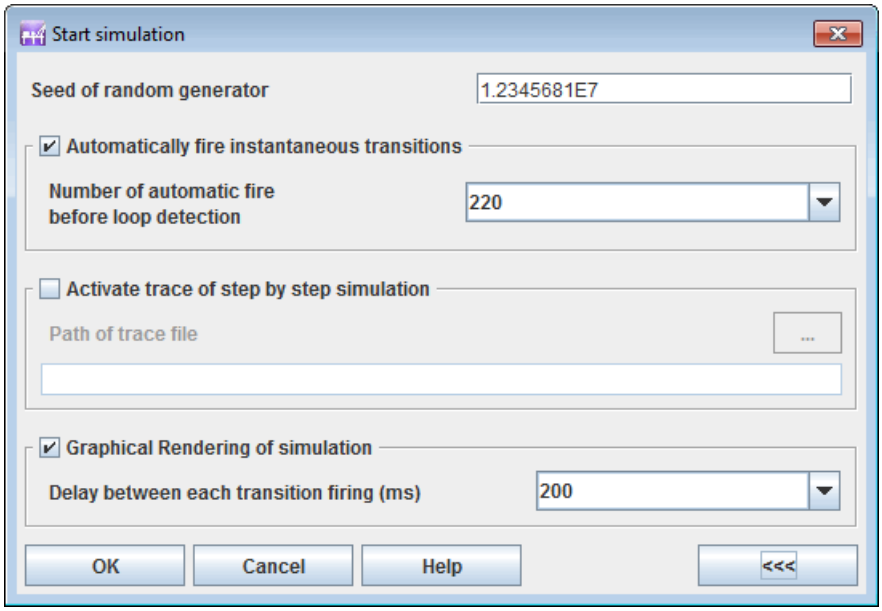
3.4. Simulation scheduler

The simulator includes a panel which displays the scheduler of the transitions to be fired. You can display a transition from the scheduler in the model view by right-clicking on it.



3.5. Configuring the simulation

The simulation options can be accessed using the (+) button located next to the start simulation button, in the simulator toolbar. They can be configured when the simulation is started.



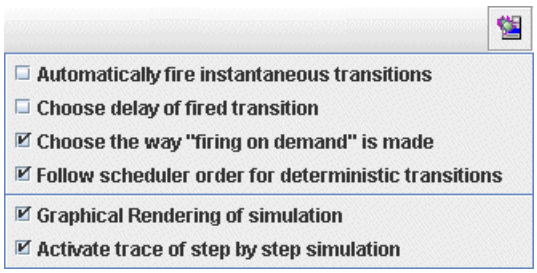
The **seed** to be used by the simulator can be entered here.

The tick box **Automatically fire instantaneous transitions** automatically plays the transitions in Dirac delta function 0. In this case, the user must indicate how many fire operations the simulator can do before considering that there is a loop in the simulation.

The tick box **Activate trace of step-by-step simulation** saves all the transitions fired during the simulation in a file.

The tick box **Graphical rendering of the simulation** enables the user to follow the step-by-step progress of the simulation in the model input area. S/he can then enter the length of time between each transition firing to slow down or speed up the simulation.

Other options can be modified during the simulation



The option **Choose delay of fired transition** enables the user to choose the date on which the transition will be fired (applies only to stochastic transitions).

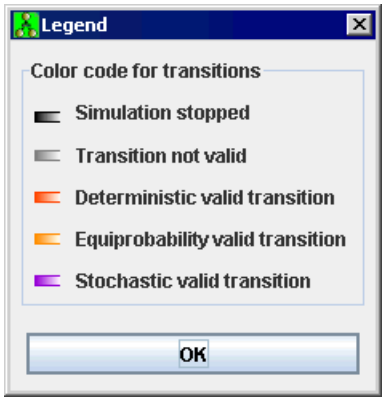
The option **Choose the way "firing on demand" is made** enables the user to automatically fire or not the transitions that use the firing on demand rule. If the option is ticked, a dialogue window will prompt you to select the location that will receive the token if this kind of transition is fired.

If the option **Follow scheduler order for deterministic transitions** is unticked, it is possible to fire deterministic transitions before the date initially planned.

3.6. Colour code / Legend

During a simulation, the transitions active at a given time are displayed in specific colours according to the type of transition.

Below are the colour codes used in the simulator panel for the transitions.

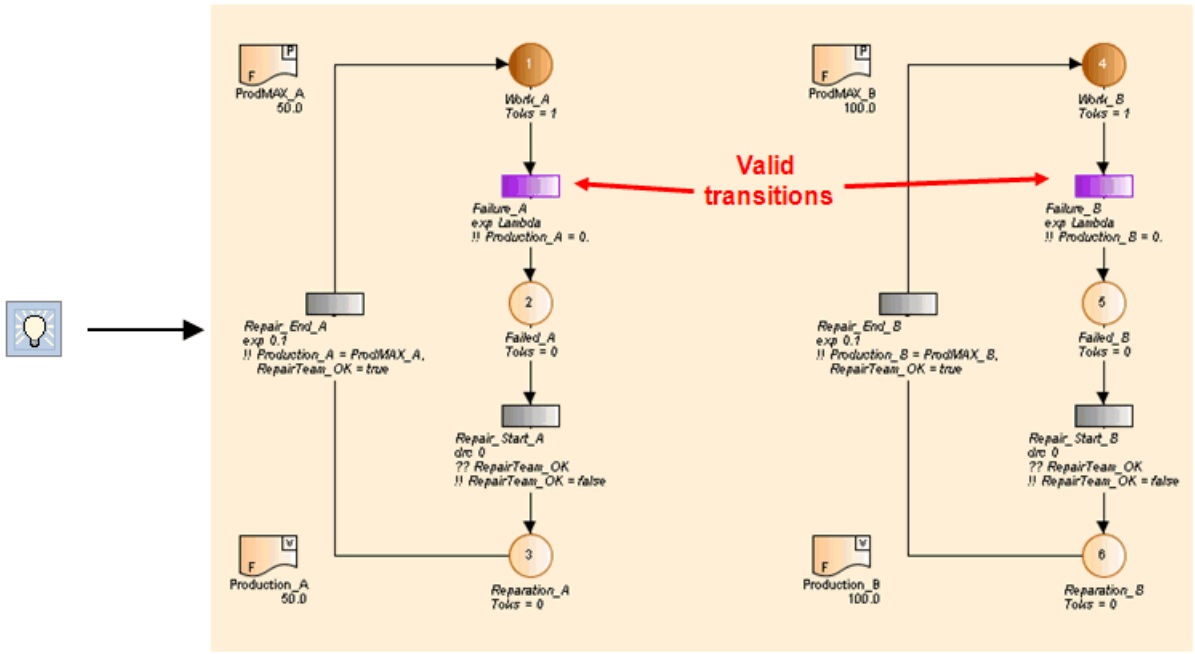


3.7. Transition firing

There are several ways of starting a simulation :

- Use the menu **Mode - Simulation**.
- Click on the corresponding icon in the vertical toolbar.
- Click on the corresponding icon in the simulation panel toolbar.

Once the simulation mode has started, the "valid" transitions in the initial status of the Petri net appear in the scheduler.



In the following picture, two transitions are valid at initial state. There are the failure of right equipment and the failure of left equipment. The two **Failure** transitions are valid because there is one token in their upstream place (**Work**).

This simulation is the expected one: at initial state (system is perfect), the only thing that can happen is the failure of one of the two components.

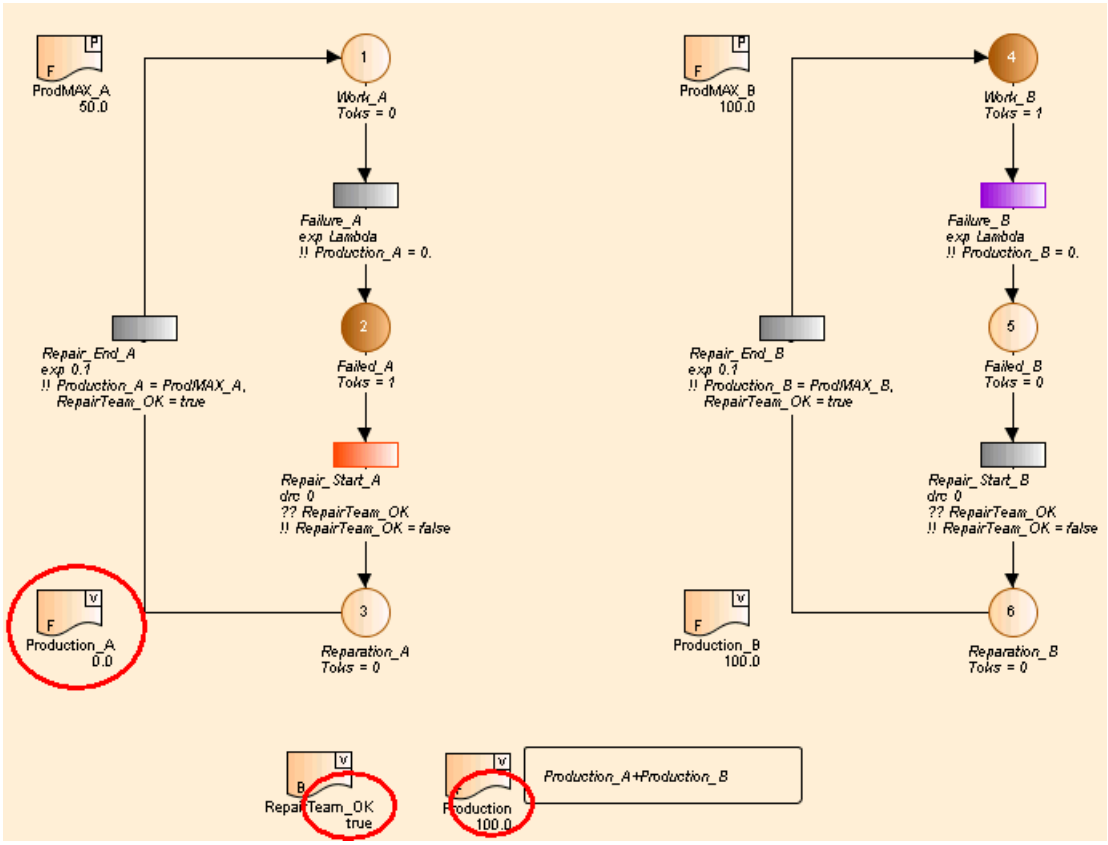
The operation which is of major interest is to be able to "manually fire" the valid transitions:

- choose one of the valid transitions;
- click left on this transition.

The result of this is to:

- remove the token from the **Work** place (which inhibits the **Failure** transition);
- add a token in the **Failed** place.

In the example, given that the Boolean variable **RepairTeam_OK** is initially **TRUE**, the **Repair_Start** transition of the component which failed (that on the left) will be valid. As for the other component, its **Failure** transition will remain valid.

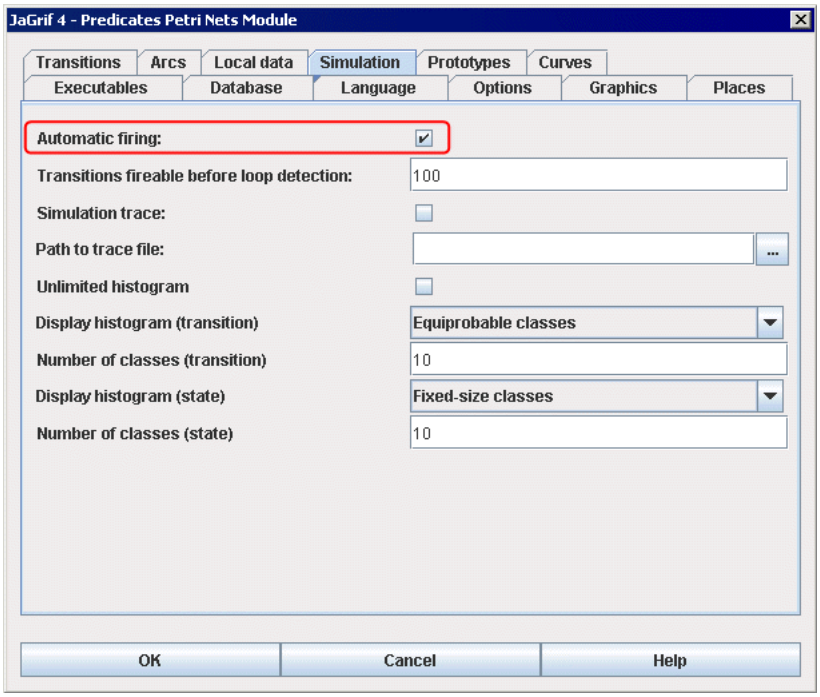


3.8. Automatic firing of transitions with zero delay

The automatic firing of the transitions in simulation mode automatically fires the transitions which have a zero delay (Dirac law with parameter 0) when they are valid. In the case where several Dirac law transitions are in "conflict", the transitions are fired according to their priority, then in chronological order of their creation on a page, then in increasing order of pages. This is how the simulation works when the computations are launched.

Remarque: The transitions with zero delay of a group are fired after those of the page where this same group figures.

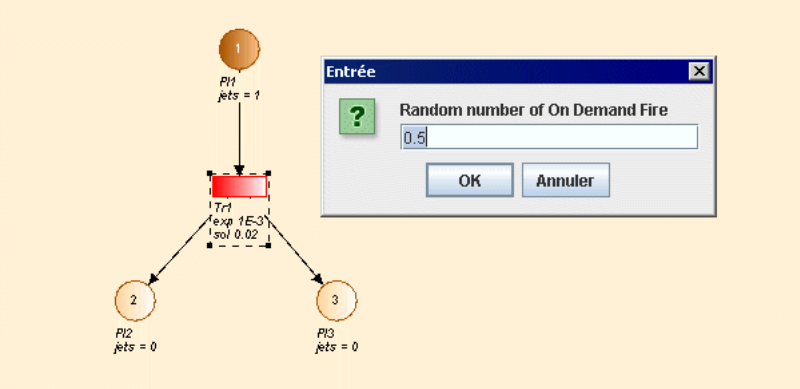
To enable or disable this function go into **Application Option - Simulation**:



3.9. Probability firing for transitions with fire on demand

In simulation mode, when we wish to fire a transition with "fire on demand", the token(s) must only move to a single place downstream. When you click on this type of transition a window is displayed where you have to enter the probability manually (it is normally a probability which is determined by a simulation).

Note: The default value of this probability is 0.5.



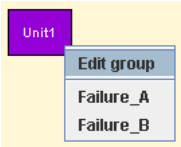
In the above example, if 0.5 is entered, the token will go into place no. 3.

3.10. Simulation with groups

There are two ways of firing one or more valid transitions that belong to a group:

- Enter the group to fire the transitions by clicking on it directly (left click on the group or right click then **Edit group**).

- Or run a fire operation simply using the list of valid transitions in the group and "sub-groups" (right click on the group).



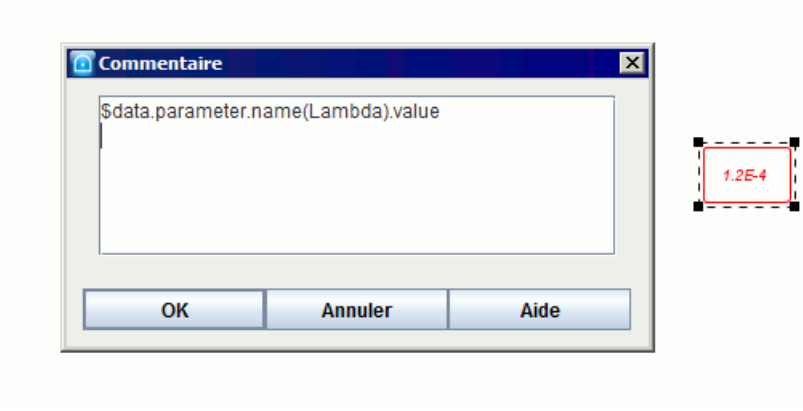
If we go back to the previous example with two components, then the **Failure** transition was effectively the only valid transition.

When transitions are fired within groups, it may prove difficult to determine the modifications that have been made. This is why it is useful to use the dynamic fields.

3.11. Dynamic fields

It may be useful to observe the change in the different parameters of the model. It is also usefull to see a result next to its corresponding system. To do this, use dynamic fields by selecting the corresponding icon on the vertical tool bar:

The dynamic fields are a type of "improved comments". They can be used not only to enter words or phrases but also to insert model values or results.



If you want to display informations about a data of the model, you must use the following syntax:

$data.'type of data'.'field used o search data'('value that the field must match).'information you want to display for the selected data'

We can analyze the above windows as follows: I am looking for a "parameter" which "name" is 'lambda", and I want to display its "value". When you type the first letters, a completion system helps to type script without error.

If you want to display a result of the result-bank, the syntax is the following:

$result.bank('path in the bank').target('target result').'what you want to display'.'at what time'

We can analyze the next picture as follows: I am looking for a result which path in the bank is "default-Moca", I want results for "TS3 for 'available' valriable" and I want its value for the "last" time. If last is replaced by time(10) we obtain value at t=10.

You can also display a summary of result. Replace 'what you want to display' by **summary**. In this case, **summary** is the last word of this script.

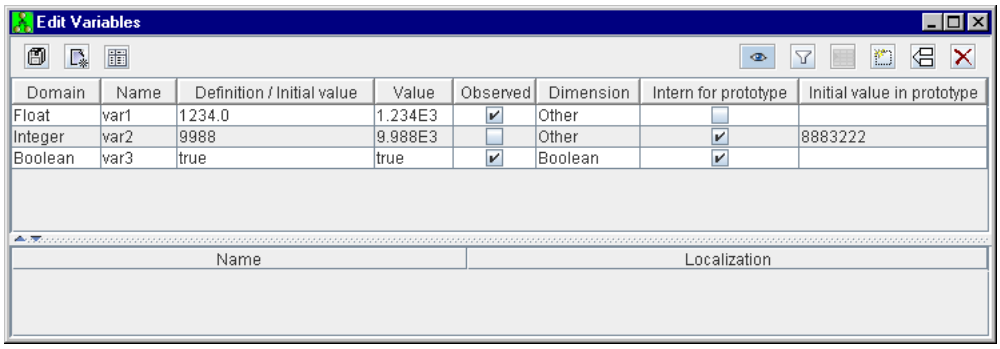
A button has been added in 2013 version, it is a script generator for model data.

4. Statistics and Setup of Variables

In addition to the mean marking of the places and of the mean number of times each transition is fired, the simulation can compute a certain number of additional statistics. Statistical results can be obtained on any of the model's variables or combination of variables. To do this, a variable must be declared as "observed". When a variable is "observed", a statistic state (Moca meaning) is created for computation.

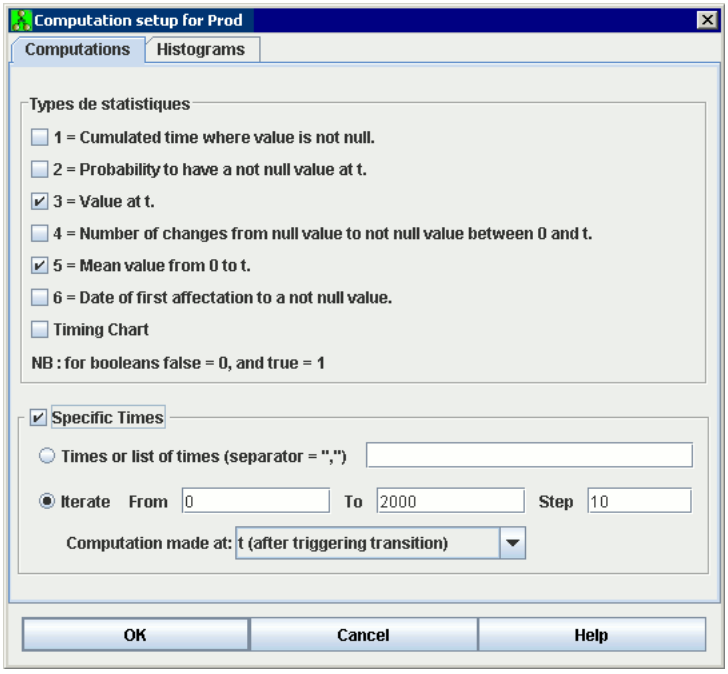
4.1. Definition of statistic states

A statistic state is defined like an "Observed" variable. We must initially define the statistic states we wish to observe. To do this, we have to edit variables from the model : either thanks to the **Data and computations - Edit Variables** menu, or thanks to the **Edit Variables** tab. Then, all we have to do is to set **Observed** property of a variable in order to make it a statistic state.



4.2. Configuration of statistic states (or variables)

Once variable is observed, we have to configure them by specifying types of computations and computation times to be carried out on them, ... To do this, we have do a right-clic on the variable and select **Configuration of computation**.



The window to edit observed variables (statistic states) is made of two tabs. The first tab is for computation, the second one is for histograms. These two tabs enables configuration of **Types of statistics**, **Computation times** and **Histograms**.

4.2.1. Types of statistics

Types of statistical computations are the following :

- **1 - Cumulated time where value is not null :** This is the mean time in which the statistic state is different to 0 on a history.

  Purpose: Mainly used to compute mean availabilities during a history.
- **2 - Probability to have a not null value at t :** This is the probability that the statistic state is different to zero at the end of the history.

  Purpose: Among other things, used to compute the mean availability at the end of a history or compute the reliability (to find if the failure state - which is an absorbent state - is present at the end of a history).
- **3 - Value at t :** This is the mean value of the statistic state at the end of the history.

  Purpose: This type of computation can be used to compute the occurrence of specific event during a history.
- **4 - Number of changes from null value to not null value between 0 and t :** This is the mean number of times during a history that the statistic state changes from a zero value to a non-zero value.

  Purpose: This type of computation can be used to compute the occurrence of specific events during a history.
- **5- Mean value from 0 to t :** This is the mean value of the statistic state on the duration of the history.

  Purpose: Among other things, used to compute the production availability.
- **Date of first affectation to a not null value :** This is the mean instant from which the value of the statistic state changes from zero to a value different to zero.

  Note: The "uncensored data" field gives the number of histories for which the simulation has been able to retrieve a value. For this mean statistical result to have a meaning, we must verify that a value has been retrieved for each history (uncensored data = number of simulated histories).

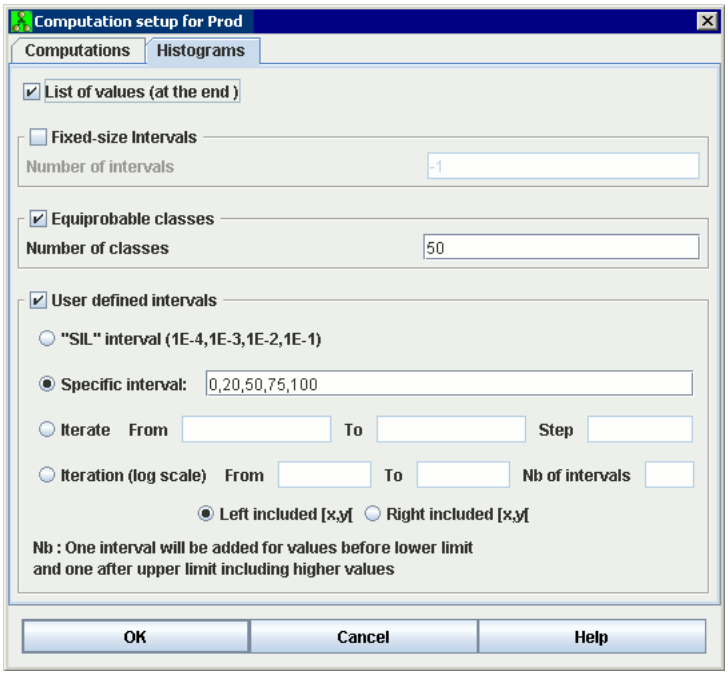
  Purpose: Used to obtain information about the mean instant when a system fails for the first time (reliability computations, estimation of the mean trouble-free operating time, etc.).

4.2.2. Computation times

Two possibilities are available to define computation times :

- **List of times :** the computations will be performed for the values of t given in this list. Separator is comma.
- **Iterate Form A to B step C :** the computations will be performed for values of t ranging from A to B with a step of C. You can also choose is computations are made before or after transitions triggering.

4.2.3. Histograms



Computations made previously gives mean value for many histories. **Histograms** enable to know how values are distributed during histories. (Cf. Moca User manuel for further information)

- **List of values :** provides value at the end of each history.
- **Fixed size intervals :** provides the way value are distributed by cutting intervals of values in X intervals which size is equal.
- **Equiprobable classe intervals :** provides intervals whose probability to contains a value at the end of a story id the same.
- **User-defined Intervals**

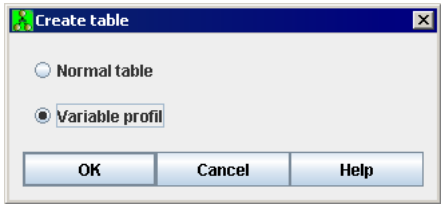
  Bounds of intervals can be defined as follows :
  - Automaticaly defined limits for SIL
  - Manual definition of limits (separate by commas)
  - "Iteration" : user choose lower limit and upper limit, and the size of intervals.
  - "Iteration (log scale)" : user choose lower limit and upper limit, and the number of intervals. Size of intervals will be computed in order to have same-size-intervals, on a logarithmic scale.

  Moreover, two limits are added at minus infinity, and plus infinity, in order to have a chart containing every history of the simulation.

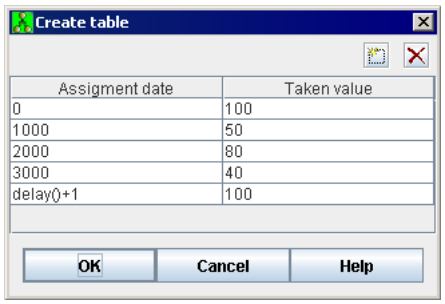
  When limits are chosen, user has to choose between "left inclued" or "right included". Nb : IEC 61508 specifies "left included" intervals for SIL

4.3. Tables and profil of variables

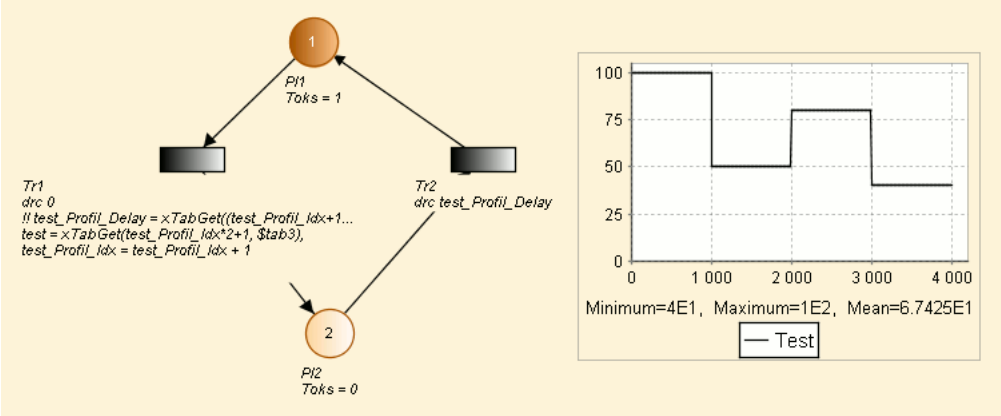
You can create data-tables (or global-tables) with **Tables** table. A global-table is a list of arguments which can be used in functions needing several arguments.



When you create a global-table (**New** button), you have to choose either **Normal table** or **Profile of variable**. In the fisrt case, the table will be one column wide, in the second one, the table will be two column wide.



A table for "variable profiling" lets you define the modifications of the value of the variable during simulation. In the left column you must enter a date (beginning with 0) and in the right one, enter value that will be assigned to the variable at this date. For a good profiling, the last date must be greater than history duration. In order to be sure that this date is greater, your are advised to use the following formula : "delay()+1" (wich return history duration + 1). Then you need to create a PetriNet which modifies variable depending on profiling table. To do that, right-click on the profil table and use the function **Create a profiling net**. You will be asked the variable to profile, then the Petri Net will be created on the current page.



NB: Profil is not linked to the variable. This is only the Petri Net that has been generated, which is modifying variable according to the profil. That why you must not modify this little Petri Net.

5. MOCA computations

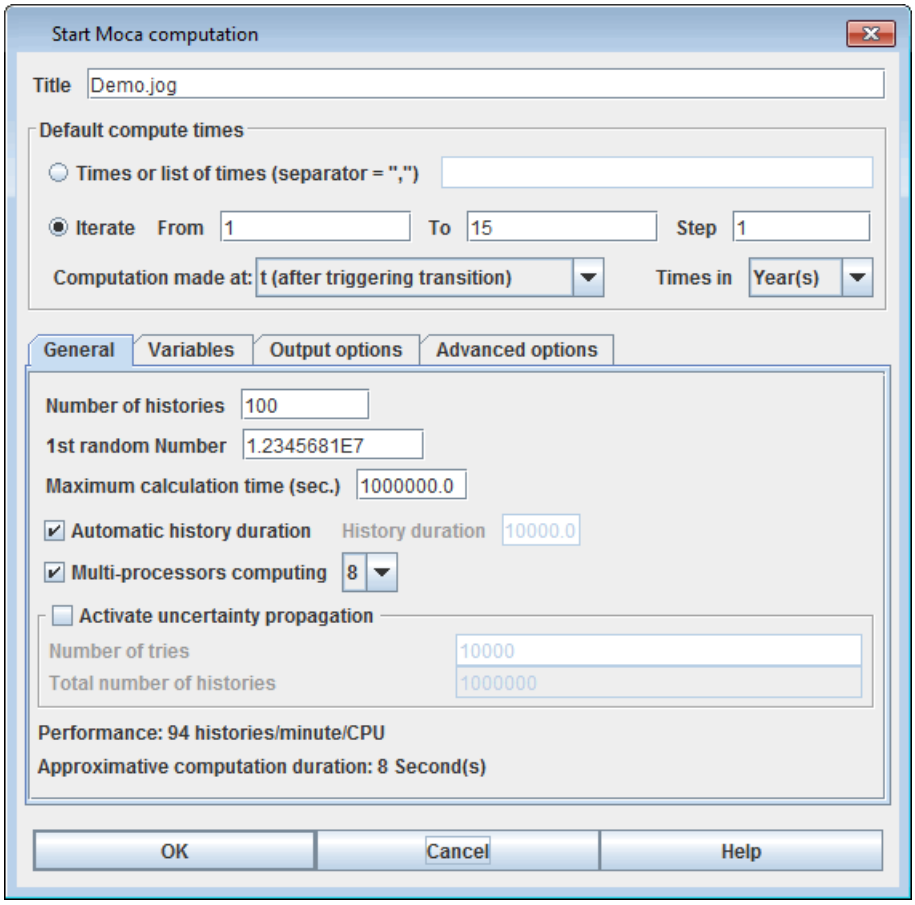
The computations using MOCA-RP V12 are performed in three main steps:

- general configuration of parameters;
- the launch itself;
- reading the results file.

5.1. Configuring the computations

The computation configuration window can be accessed in two different ways: either via menu **Data and Computations - Moca Data** or via **Data and Computations - Launch Moca 12 ...**. The difference between the two is that, in the second case, the configuration step is directly followed by the computation launch step.

The configuration window which opens is called **General Information**:



This configuration window is divided into five parts:

1. **Title**: allows you to give a title to the results file.

2. **Default computation times for statistic states**:

   - **Iterate From** A **to** B **step** C: the computations will be performed for values of t ranging from A to B with a step of C.
   - **List of times**: the computations will be performed for the values of t given in this list.
   - **Computation made at**: by default, computations are made immediatly after trantion triggering, but you can do computation at t-Epsilon (just before triggering), or at both.

- **Unit**: default times unit is "hour". You can choose a unit that will be used for computation times. N.b. results are always in hours.

3. **General**:

   - **Number of histories**: Number of histories (NH) to be simulated (each history has a time t indicated below).
   - **First random number**: It is the seed of random number generator.
   - **Maximum computation time** (MT): The computations are stopped and the results are printed even if the requested number of histories has not been reached.

     Note: the unit of time (MT) is the second.
   - **Automatic history duration**: If this box is checked, GRIF will compute history duration using computation time of variables and statistical states. If not, user can choose a specific **History duration**
   - **Multi-processors computing** Enables (or not) the multi-processor computing (when available).
   - **Activate uncertainty propagation** Enables (or not) the uncertainty propagation computations (two-stage simulation): in this case we must specify the number of sets of parameters "played" (the real number of histories thus simulated will be the "number of sets of parameters x number of histories to be simulated" and will be displayed in the "Total number of histories" field).

4. **Variables**: This tabs reminds comuting configuration of variables. If document contains some statistical states, another tab is available.

5. **Output**: used to configure the output.

   - Prints the description of the Petri Net in the results file (or not)
   - Prints the results file allowing it to be loaded using a spreadsheet application (such as EXCEL)
   - Prints the censored delays (or not)
   - Number of outputs during simulation. If 2 outputs, there will be an output at NH/2 and at NH.

6. **Advanced options**: used to configure the advanced options.

   - You can choose the limit of transitions fired at the same time before loop detection.

5.2. Reading the results (New GUI)

Since GRIF 2010, results are displayed in a windows with many tabs and tables.

5.2.1. Moca Results

Moca results atre displayed in a window containing 6 main tabs : variables, places, transitions, XML, stantard output, info.

5.2.1.1. Tab of Variables

The **Variables** tab contains every information computed for each variable (or statistical state).

- **Value** : Contains every value of a variable for every type of statistic.
- **History (at the end of histories)** : contains historical values for each computed statitic.
- **Fixed size Histogram** : Contains histograms computed by Moca (cf chapter about histograms)
- **Equiprobable classes Histogram** : Contains histograms computed by Moca (cf chapter about histograms)
- **User defined Histogram** : Contains histograms computed by Moca (cf chapter about histograms)
- **Timeline** : Contains a timeline for each variable. Times are automatically computed by Moca.

5.2.1.2. Tab of Places

It contains sojourn duration and mean mark for each place of Petri Net.

5.2.1.3. Tab of Transitions

It contains firing frequencies for each transition, and firing history for each history.

5.2.1.4. Other tabs

Other tabs display "raw" results. XML tab contains XML output of Moca, it is the file used to retrieve data. This file can be used for further post-threatments.

Standard output display the standard output of Moca (available only afer computing).

Info tab contains usefull information about computation (simulation time, number of histories that have been done ...)

6. Curves

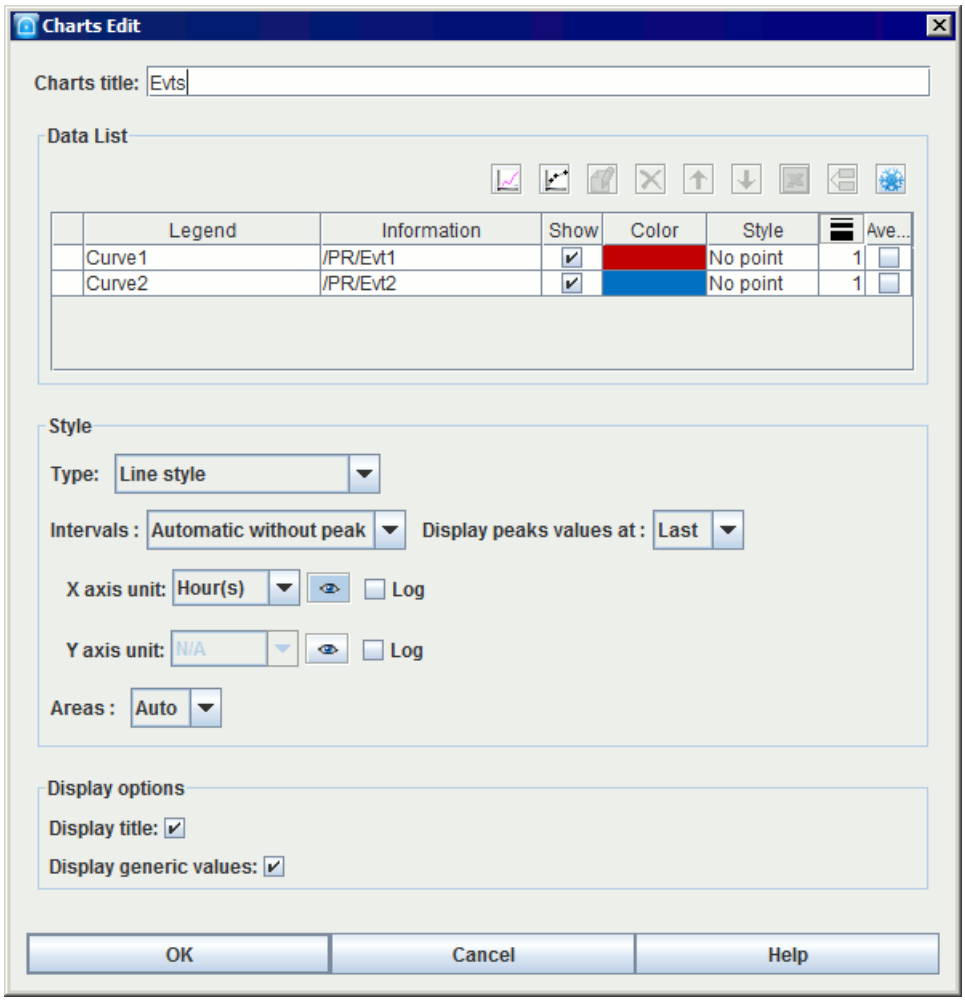
The curves can be drawn to study the model and the results better. To do this, click left on the corresponding icon on the vertical task bar then draw a box. This box will be the space assigned to displaying the curve(s). Initially it is only a white box with two axes without graduation.

Charts icon:

We must now define the curves to be drawn. To do this, click right on the box to display the **Charts Edit** window.

6.1. Edit curves window

The edit curves window is the same for all the GRIF modules.



The window is divided into several sections:

1. **Charts title**: enters a title for the graph.

2. **Data list**: this part includes a table with several columns in which the different curves on the graph are listed (name, description, display, curve colour, curve style, curve thickness, display average). A number of different buttons are available above this table.
   -  : Selects a result of computations to display. It sends the user back to the Select results window to add a curve plot to the graph (see. Section 6.2.1, “Curves from data in result-bank”).
   -  : Compares several results from different calculations for the same data. It sends the user to the Compare results window to add a curve plot to the graph (see Section 6.2.2, “Comparative curves from data in results bank ”).
   - **Edit**: edits the plot of the selected curve.
   - **Delete**  : deletes the plot of the curve selected on the graph.
   - **Up**  : moves up the plot of the curve selected in the list.
   - **Down**  : moves down the plot of the curve selected in the list.
   - **Save**: saves the list of points calculated to plot the selected curves in .csv format. This export does not contain the generic values. To obtain an export with the generic values, right click on the curve and select **Individual export**.
   - **Duplicate**: creates a new curve identical to the curve selected.
   - **Freeze**  : freezes the display of the curve, which will no longer be updated automatically according computation results.

   For each curve, the user can specify the colour, point style, line thickness and display.

3. **Computation options**: enables the user to enter settings for the computation (optional depending on the module).

4. **Style**: this section concerns the curve display.
   - **Type of style**: specifies the type of all the curves on the graph (line, histogram, etc.). N.B. In the case of a histogram, the bars that exceed the display area will be displayed in shading to show the user that s/he must change the display intervals so that the entire bar can be displayed.
   - **Intervals** defines the display limits for the curve. **Automatic without peak**: the graph will not display the "peaks" of specific cases of exceptional values which would make the graph illegible. Even if the peaks are not displayed on the graph, the user can display their values using the choices proposed in the option **Display peak values**.
   - **X and Y intervals**: specification of the display interval on the X and Y axes (default intervals or intervals defined by the user). The last function enables users to "zoom in" on the most interesting parts of the graph.
   - The axis unit can be selected according to the type of computation result. For example for units of time, the user can enter hours, days, months or years.
   - The button determines whether or not the unit is displayed on the graph.
   - The **log** boxes are used to activate the logarithmic scale on the axis in question. N.B. 0 cannot be represented on a log scale, remember to enter a start value that is strictly positive (e.g. E-10). If 0 is entered, the log scale will begin at an arbitrary value E-15. Where the computation engine allows, the **trust interval** can be displayed by ticking the corresponding box.
   - **Areas**: distinguishes a range of values on a coloured background.
   - In the histogram style, a box can be ticked to create a cumulative histogram.

5. **Display options**: activates the **Display title** function (display graph title) and the **Display generic values** function (display min, max and average for each curve).

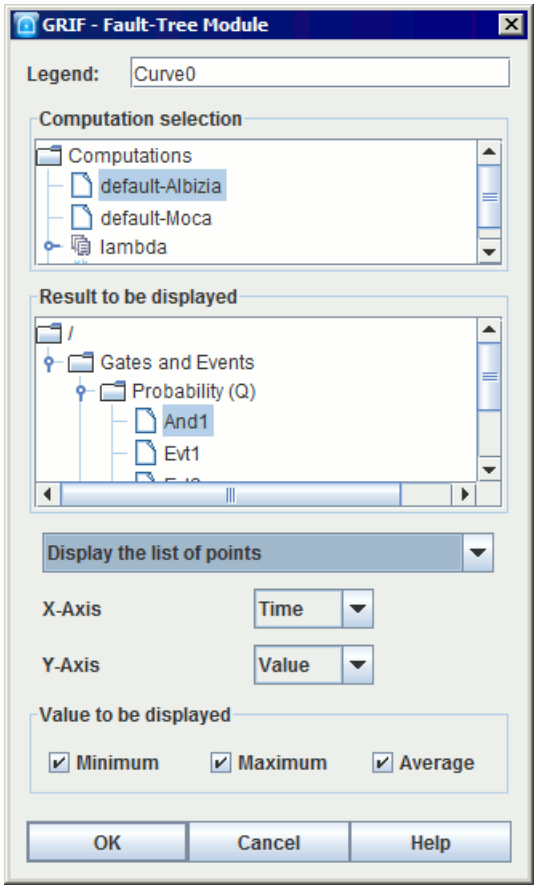
When a curve is edited, the edit curve window often includes 3 parts: the times at which the computations are performed, what has been calculated and the extra information (generic values) that should or should not be displayed below the curve.

Comment: it is sometimes necessary to refresh all the graphs in a document. This can be done using the **Tools / Refresh** command, or the keyboard shortcut F5 or the  icon.

6.2. Selection of results window

6.2.1. Curves from data in result-bank

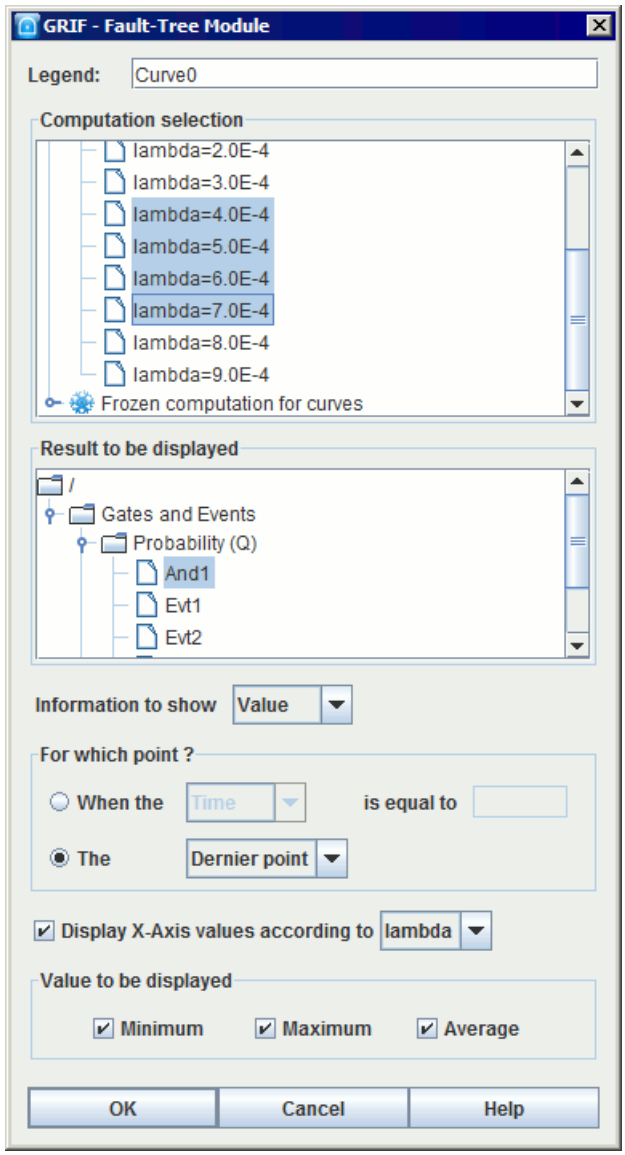
When you click the **Add** button in the **Data list** part you reach a window for curves setup. Each curve displays data stored in the result-bank. The following window helps users to specify how to retrieve data.



- **Legend**: legend of the curve.
- **Computation selection**: select the computation in the result-bank.
- **Result to be displayed**: each computation contains many results. Select the one you want to be drawn.
- Vous avez la possibilité d'afficher la liste des points en spécifiant les données voulues sur chacun des axes ; ou vous avez la possibilité d'afficher les temps passés dans les zones.
- **Axes**: When a result is selected, select what must be in X-Axe and what must be in Y-Axe.
- **Value to be displayed**: Then you can display addition informations about the result (min, max, moyenne ....)

6.2.2. Comparative curves from data in results bank

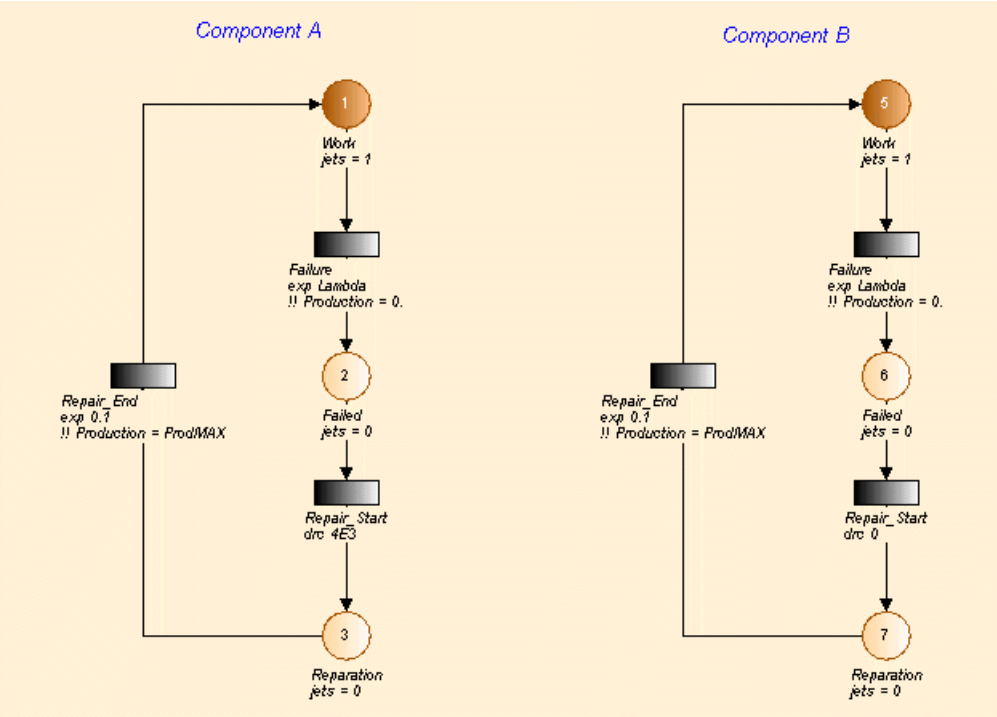
When the user clicks on **Compare** in the **Data list** section, a window opens enabling the user to specify the curve to be plotted. S/he can then choose whether or not to display a result from several different computations. The following window is used to indicate the way in which the information is to be recovered.



- **Legend**: curve legend.
- **Computation selection**: the user can select the different computations to be used from the results bank. Hold down the Ctrl button to select several different computations.
- **Result to be displayed**: a computation often contains several results and this file tree structure is used to specify the result that the user wants to display.
- **Information to show**: indicates which data are to be displayed on the y-axis.
- **For which point?**: indicates which point of the computation is to be compared to the others.
- **Display x-axis values according to**: indicates which data must be displayed on the x-axis.
- **Value to be displayed**: finally, certain extra data can be displayed (min, max, average, etc.)

6.3. Examples of curves

Here are the Petri Nets from which the different curves will be drawn.

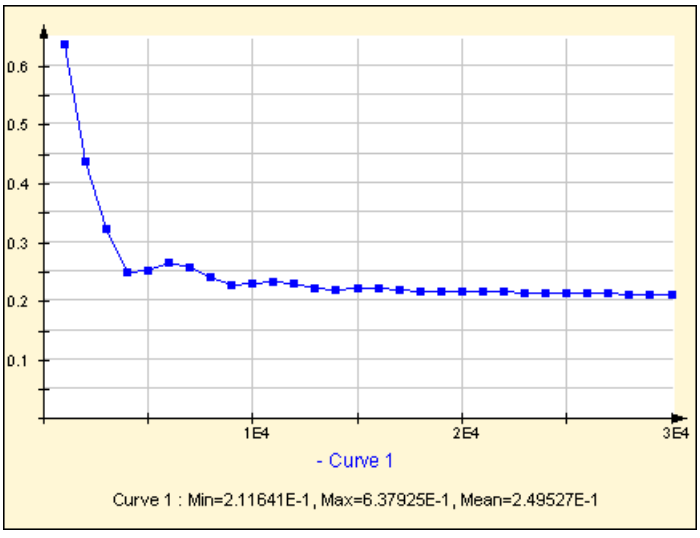


There are two components, A and B:

- A can be repaired with a delay before repair modelled by a Dirac law transition.
- B can also be repaired but with no delay before repair.

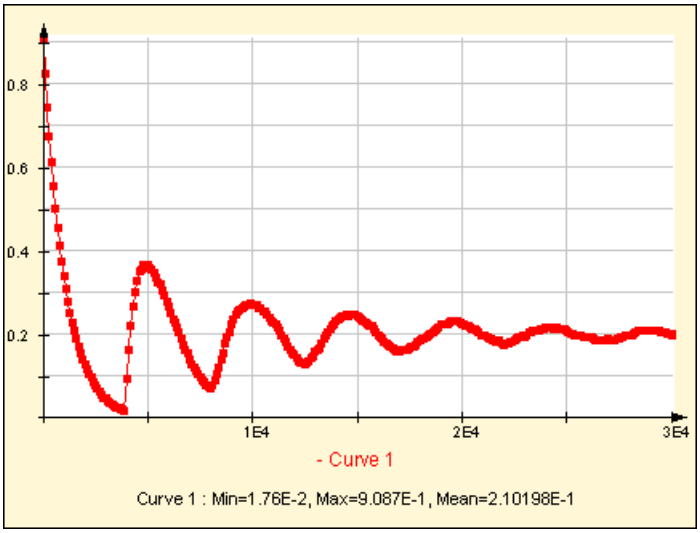
The simulation will be carried out for 10,000 histories of 30,000 hours.

6.3.1. Availability



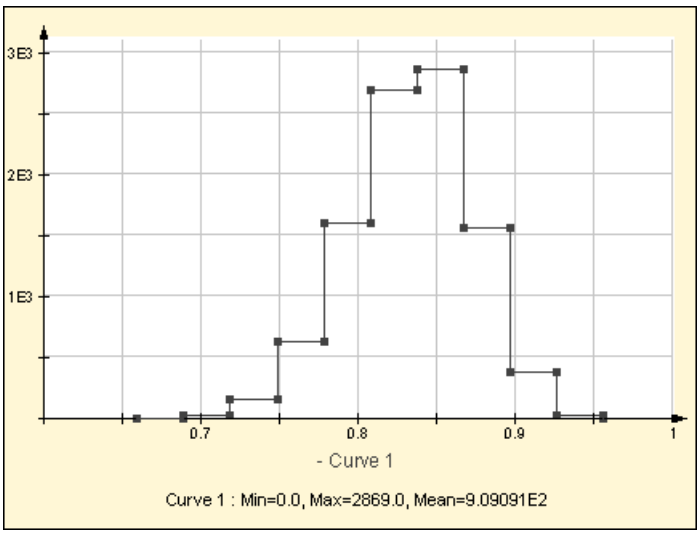
This curve represents the change in mean availability of component A over time. The points have been computed at regular intervals (1 every 1000 hours).

6.3.2. Timing Chart



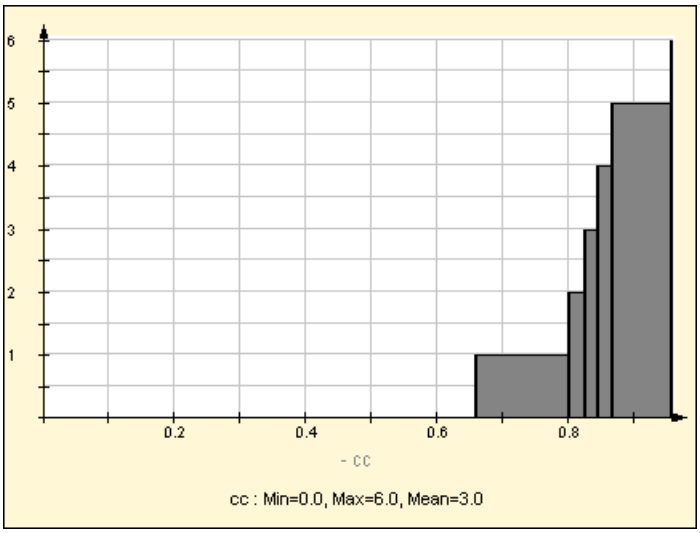
This curve also represents the change in instantaneous availability of component A over time (marking of place no. 1). In this case the points have been computed according to the curve variation. This "captures" the discontinuities better and thus makes the curve "smoother" and more precise.

6.3.3. Fixed Size Histogram



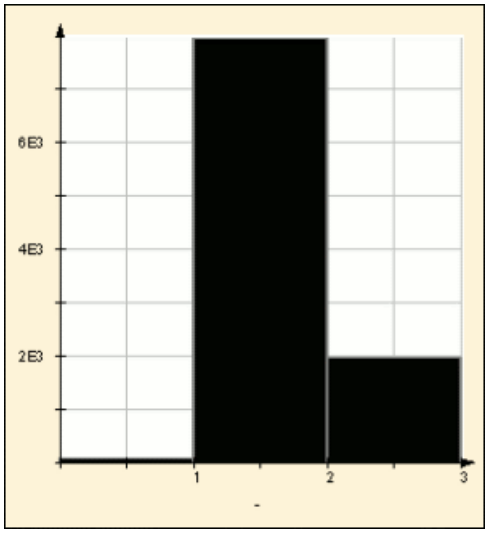
This histogram was produced from the mean value of the availability of component B (marking of place no. 5). The 10,000 results from the 10,000 simulated histories have been stored in 10 classes of the same interval.

6.3.4. Equiprobable Classes Histogram



This histogram was produced from the mean value of the availability of component B (marking of place no. 5). Five equiprobable classes were requested. The probabilities that component B's mean availability is in one of these classes is identical.

6.3.5. Defined Interval Histogram



The 10,000 results from the 10,000 simulated histories have been stored in defined intervals. We have defined "[0,1,2,3" (equivalent to [0,1[[1,2[[2,3[cf. Moca User Manual for more informations) We have computed number of working components at the end of histories. We see that 2 components works at the end of histories most of the times. Intervals corresponding to S.I.L can be defined in order to see in wich S.I.L component is for different stories.

Bounds of intervals can be defined as follows :

- Automaticaly defined limits for SIL
- Manual definition of limits (separate by commas)
- "Iteration" : user choose lower limit and upper limit, and the size of intervals.
- "Iteration (log scale)" : user choose lower limit and upper limit, and the number of intervals. Size of intervals will be computed in order to have same-size-intervals, on a logarithmic scale.

Moreover, two limits are added at minus infinity, and plus infinity, in order to have a chart containing every history of the simulation.

When limits are chosen, user has to choose between "left inclued" or "right included". Nb : IEC 61508 specifies "left included" intervals for SIL

7. Databases

In each GRIF module a connection can be established to a database. It is possible to have two different types of connections:

- connection to a CSV file;
- connection via a JDBC link.

7.1. Connection to a CSV file

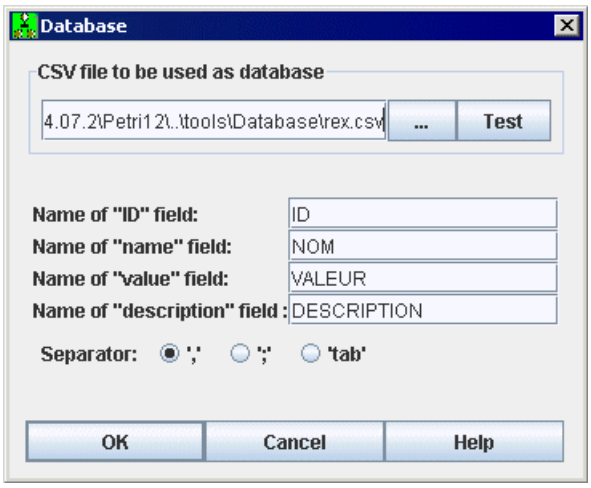
7.1.1. Form of the database

This type of connection is the simplest to make. A CSV file has the extension ".csv". It is a simple text file where the different fields are separated by commas, tabs or semi-colons. It is the simplest form of database.

```
ID,NOM,VALEUR,DESCRIPTION,DIMENSION
000001,Lambda,0.001,Exemple de Lamda,RATE
000002,Mu,0.01,Exemple de Mu,RATE
000003,Gamma,0.5,Exemple de Gamma,PROBABILITY
000004,ProdMax,1000.0,Exemple de Production maximum,OTHER
```

7.1.2. Connection

To connect GRIF to this database, go into menu **Tools - Connection to a CSV file**. A dialogue box is then displayed:



This window is divided into three parts:

- You must initially enter the path leading to the CSV file. To do this, there is an explorer available (... button). A Test function is used to check the connection.

- You must then enter the names of the four fields of the CSV file.

- Finally, specify the types of separators used in the CSV file.

Note: A CVS connection must be made with a CVS File (which means generated from only one sheet of a EXCEL File).

7.2. Connection via a JDBC link (example with ODBC connector)

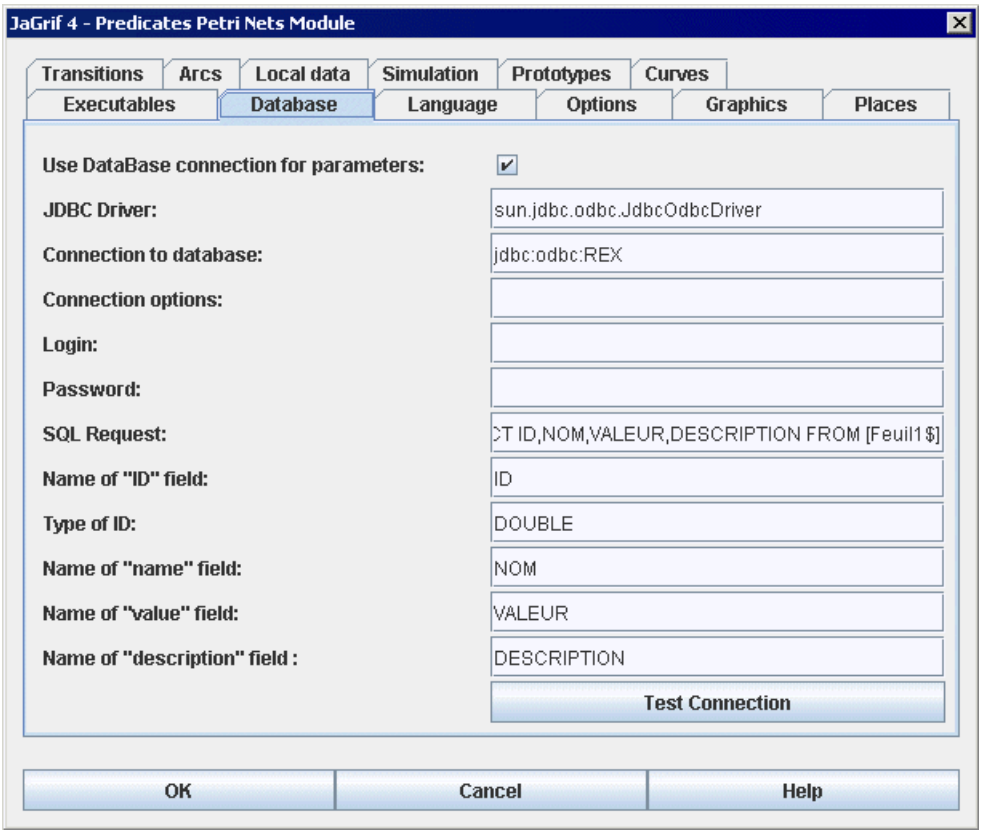
7.2.1. Form of the database

The database can initially be in the form of an EXCEL or ACCESS file. Then, using the operating system, an ODBC system data source must be created. In the case of WINDOWS, for example, this operation is performed in menu "Control Panel - Administration tools - data source (ODBC)". Here is an example of an EXCEL database:

	A	B	C	D	E	F
1	ID	NOM	VALEUR	DESCRIPTION	DIMENSION	
2	1	Lambda	0.001	Exemple de Lamda	RATE	
3	2	Mu	0.01	Exemple de Mu	RATE	
4	3	Gamma	0.5	Exemple de Gamma	PROBABILITY	
5	4	ProdMax	1000.0	Exemple de Production maximum	OTHER	
6						
7						

7.2.2. Connection

To connect GRIF to this ODBC database, go into menu **Tools - Application options - Database**. A window is then displayed which must be filled in as follows:

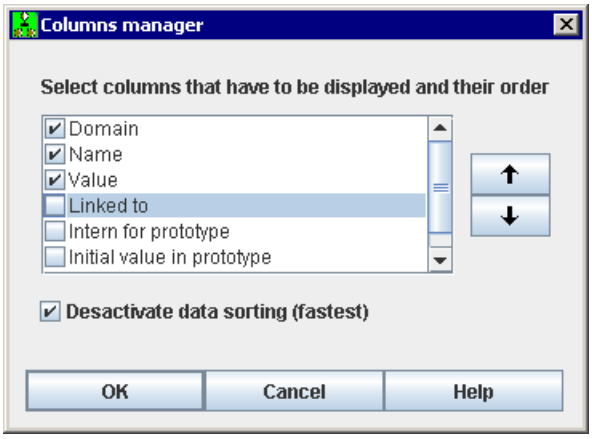


Notes:

1. **sun.jdbc.odbc.JdbcOdbcDriver** is the driver name
2. **jdbc:odbc:REX** specifies that "REX" is the name of the ODBC link
3. The fields **Connection options**, **Login** and **Password** are unnecessary here.
4. **SELECT ID,NOM,VALEUR,DESCRIPTION FROM [Feuil1$]** is called the query where Feuil1 is the name of the EXCEL sheet containing the data.

7.3. Operation

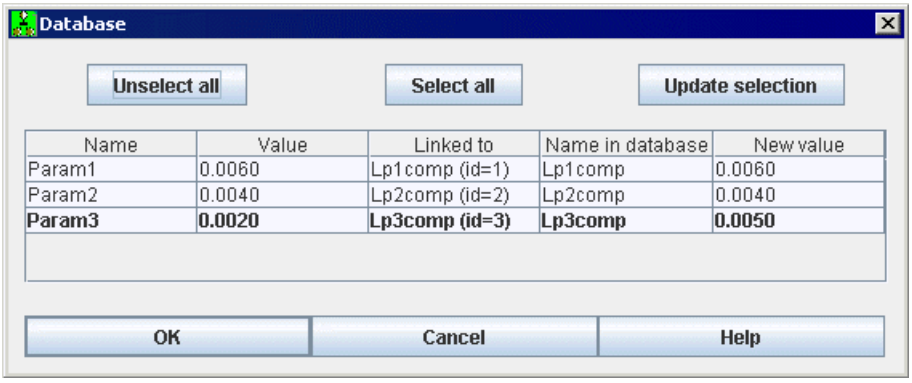
The aim is now to link some of the model's parameters to the database. To do this, start by displaying the column **Linked to** in the parameters table (click right on the top of the columns).



Then double click in the **Linked to** column to display the database items. When one of these items has been selected, click **OK** to validate the connection. The parameter then takes the value of the item to which it is now connected.

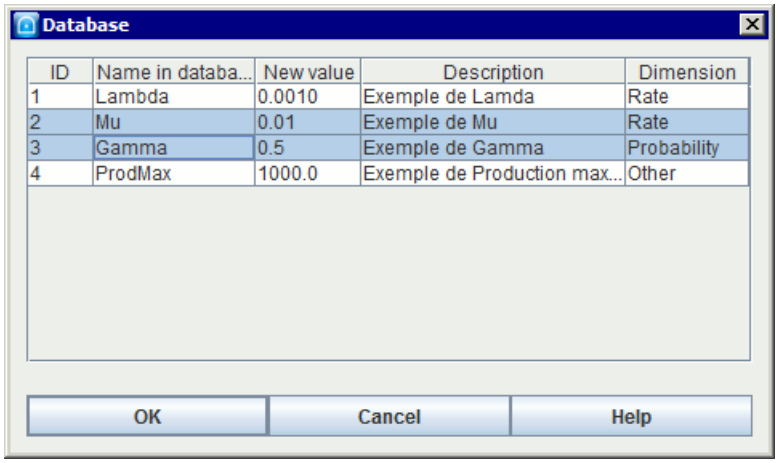
Note: The button called **Remove link to database** at the bottom of the table breaks the link between the parameter and the database item.

If some of the database values to which GRIF is linked are modified, the parameters connected to this database can be updated. To do this, select **Data and computations - Update from database...** and display the **Database window**.



The parameters whose values are not up to date are automatically detected and indicated in bold. One or more of these parameters can then be selected for updating using the **Update selection** button. Two other buttons are available to simplify the selection process: **Unselect all** (deselects all the table's parameters) and **Select all** (selects all the table's parameters).

Remark: you can also directly copy parameter from database using menu **Data and Computation/Database/Copy parameters from database**. A window is displayed, you can select parameters you want to copy into document. Parameter will be automatically linked to the right paremeter in database.

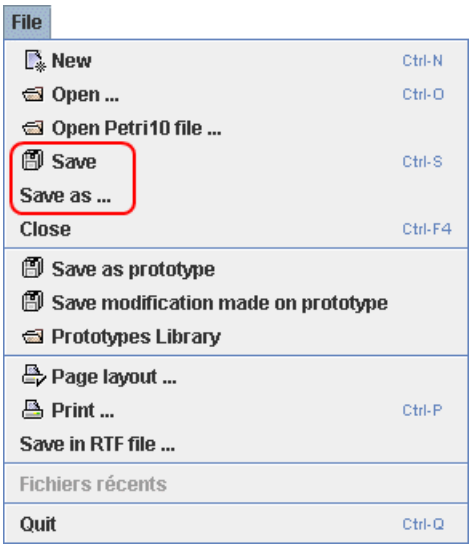


8. Save

Here is a summary of all the data which can be saved from a same model.

8.1. Model

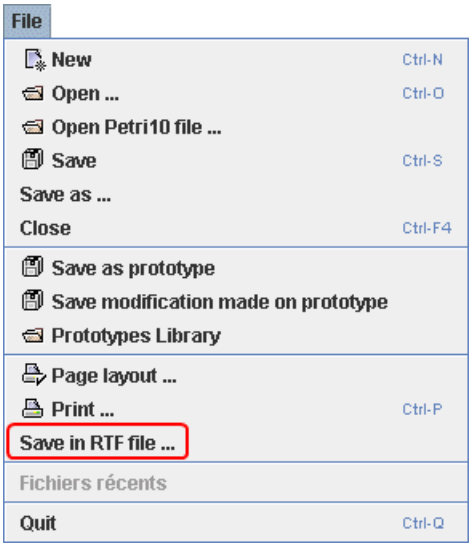
It is obviously possible to save and reload the models which are made. To do this go into menu **File - Save** or into **File - Save as....**



The **Save as template** document menu enables the user to save the document as a template in the Module template folder. New files can then be created using this model and the action **New (from template)...**

8.2. RTF File

A model can also be saved in RTF format. This allows the saved model to be reloaded in WORD to insert the graphical part of the model in any document. To do this, go into menu **File - Save in RTF file...**.



Note: There is another way to insert model in a report. Select the part of the model, copy it, and paste it in Microsoft WORD or other software.

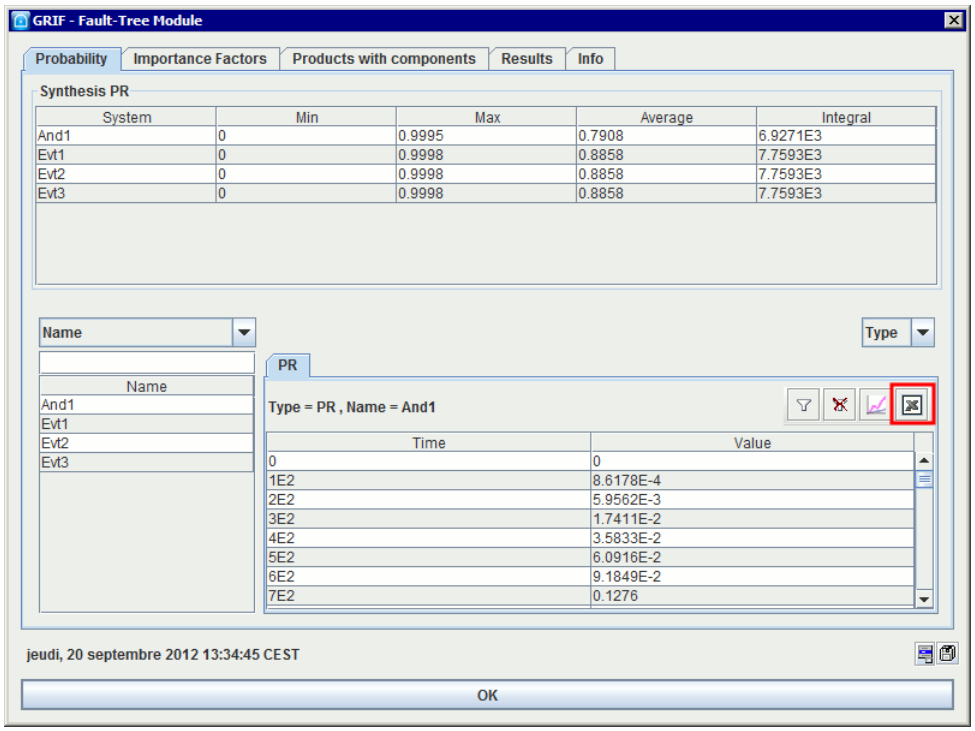
8.3. Input data

When the input data for the computation engine is generated, it can be saved. This type of file has the ".don" extension. These files can therefore be modified using a text editor then reloaded to launch computations on them (for example). This action should be made only by advanced user.

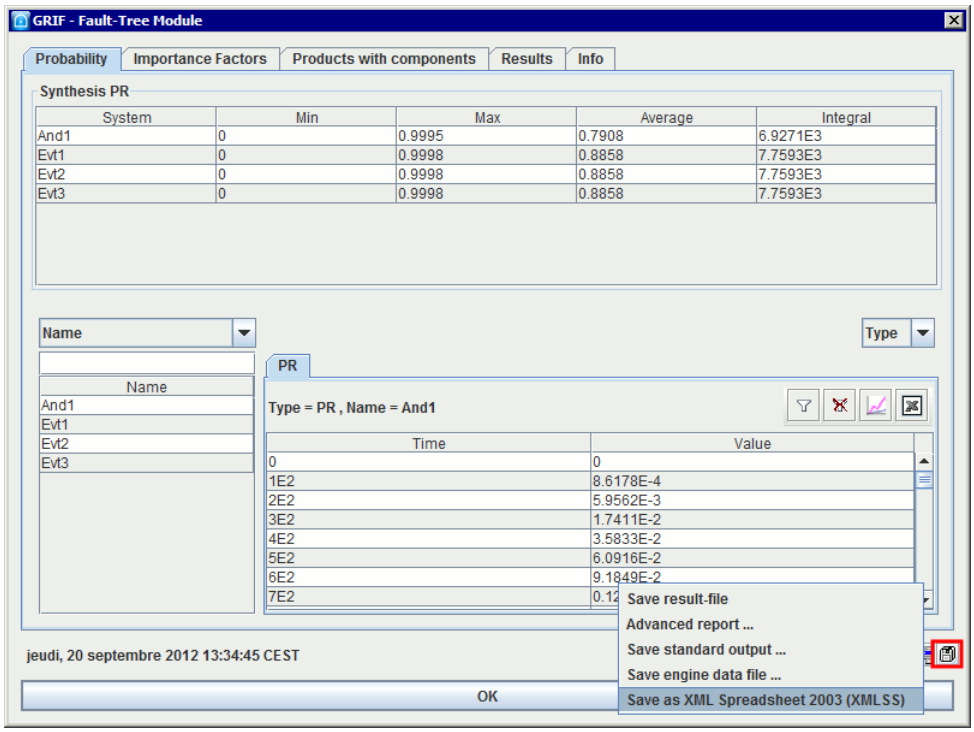
8.4. Results

Computation results can be saved in different formats :

1. Export of a table in particular in .csv format:



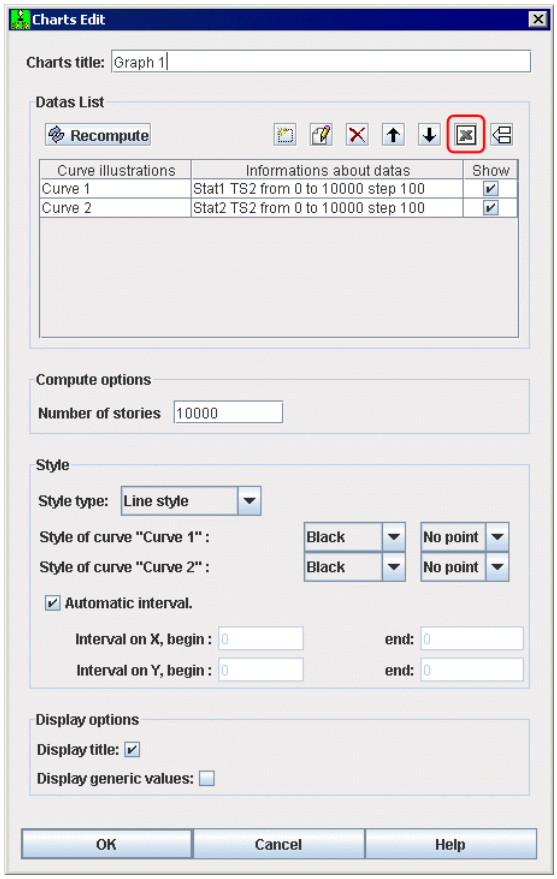
2. Export entire set of results:



- **Save result file** : saves the contents of the Results tab (.xml format).
- **Advanced report** : generates reports using style sheets.
- **Save standard output** : saves the contents of the Info tab (.txt format).
- **Save engine data file** : saves the data file sent to the computation engine (.txt format).
- **Save as XML spreadsheet 2003 (XMLSS)** : saves all the results tables in an XML format compatible with Microsoft(r) Excel 2003 and later versions.

8.5. Curves

For each curve drawn, the points which have been computed in CSV format can be saved. This list of points can then be used to draw new curves or to perform further computations.



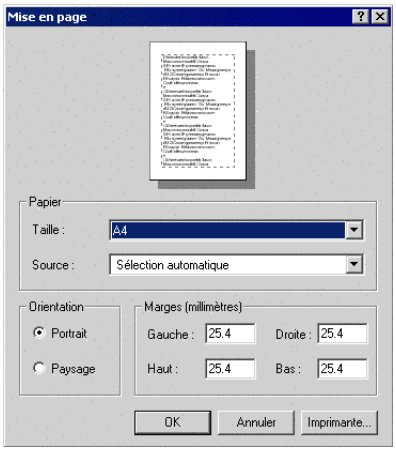
8.6. Tables

The input tables designed for complex and elaborate computations can be imported and exported. To do this, go into menu **Data and computations - Import data** or into **Data and computations - Export data**.

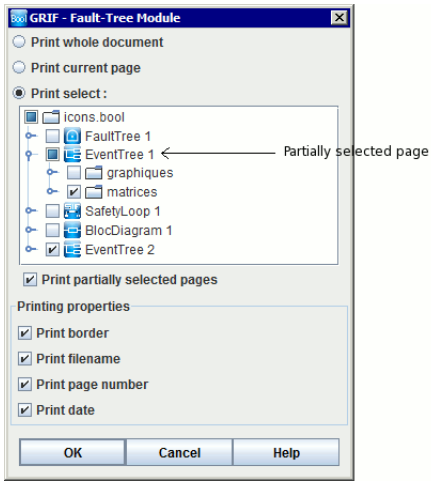
9. Printing

For printing, you have several commands at your disposal in the **File** menu **File**:

- The **Page setup** function function allows you to choose the page orientation, the size of the margins, etc.



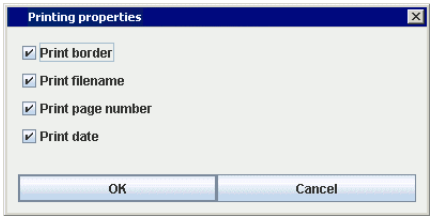
- The **Print** function allows you to export .pdf document pages. Graphics are exports in a vectorial format in order to scale its whithout deterioration.



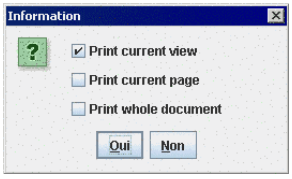
The print window appears and user can selected pages to print and configuration.

- **Print whole document** : Allows to print whole document.
- **Print current page** :Allows to print the current page.
- **Print select** : Allows to print the selected pages. The **Print partially selected pages** allows you to print pages marked by a blue square.
- **Print border** : Print a border on each page.
- **Print filename** : Print the filename on the top left corner of each page.
- **Print page number** : Print the page name and number on the bottom of each page.
- **Print date** : Print the date on the top right corner of each page.

- The **Save in RTF file...** function initially gives access to a window called **Printing properties**. Then to another called **Information**. And thirdly, a window is displayed allowing you to choose the folder in which the RTF file is to be saved.



When you select the **Save in RTF file** function, the first box to appear is that shown above. You can then select your preference: Print border, Print filename, Print page number and/or Print date.



Secondly, an **Information** window appears. It allows you to indicate whether you wish to print the current view, print the current page or print the whole document.

10. Options of GRIF - Petri Nets with predicates

Tools - Application Options menu opens a window containing the following tabs:

10.1. Executables

Executables tab enables to specify path to external executables :

- **Editor path** : Specifies text editor path.
- **Mail client** : Enable you to set the mail client to use
- **Automatically open PDF files** : Specifies if PDF reports must be opened with generation.
- **Style-sheet from XML to DocBook.** : Style-sheet allowing converting from XML report to docbook file.
- **Style-sheet from XML to HTML.** : Style-sheet allowing converting from XML report to HTML file.
- **Style-sheet from DocBook to PDF.** : Style-sheet allowing converting from docbook file to PDF file.
- **Moca-RPC path** : Specifies path of Moca 10.

10.2. Database

Database tab enables to configure database connection :

- **Use DataBase connection for parameters** : Select if database must be use.
- **Name** : Database name will be put into parameter during its update. It enables to know from which database parameter has been lastly updated.
- **JDBC Driver** : Enter name of JDBC driver to be used(sun.jdbc.odbc.JdbcOdbcDriver, oracle.jdbc.driver.OracleDriver, ...).
- **Connection to database** : Database Url.
- **Connection options** : Connection properties.
- **Login** : Login to be used to connect to database.
- **Password** : Password to be used to connect to database.
- **SQL Request** : Request that have to be executed to retrieve data from database.
- **Name of "ID" field** : Name of field containing data ID.
- **Type of ID** : Type of ID field (INTEGER, FLOAT, VARCHAR(32), ...)
- **Name of "name" field** : Name of field containing data name.
- **Name of "value" field** : Name of field containing data value.
- **Name of "description" field** : Name of field containing data description.
- **Name of "dimension" field** : Name of field containing data dimension.
- **Test Connection** : Name of field containing data description.

10.3. Language

Language tab enables to choice language :

- **Language** : Language changes are taken into account when option windows is closed.Available language are French and English.

10.4. Options

Options tab enables to tune application behavior :

- **Save the options of the current document as default options in the application** : Save options of current doc as application default options.
- **The application manages the default options of the documents. Apply the default options to the current document** : Apply -Application options- to current document.

- **Delay of automatic document saving (in minutes)** : Delay of automatic document saving (in minutes). A null value disables automatic saving.
- **Number of undo** : Specifies number of possible undo/redo.
- **Number of recent files** : Specifies number of files in recent files list.
- **Window display** : Enables separate tables (external) or linked tables (internal).
- **Columns to be resized in tables** : Enables to specify the columns on which space will be taken for resizing.
- **Manage new names to avoid name conflict** : Tries to avoid name conflict, creating new objects whose name is unik (when pasting for example).
- **Add "Copy" suffix for copy-paste** : If Enabled, a "copy" suffix will be added to the name of pasted objects.
- **Synchronize view with tables** : Select objects in tables (on the right) when they are selected in view.
- **Synchronize view with explorer** : Select objects in explorer (on the left) when they are selected in view.
- **Ask for configuration of observed variables** : Ask for computation setup when variable becomes observed.

10.5. Graphics

Graphics tab enables to modify GUI look :

- **Use Windows look and feel** : Use the look and feel of your operating system instead of java look and feel (GRIF restart is needed).
- **Element Zoom** : Changes graphics size.
- **Comment size** : Changes comment font size.
- **Group size** : Changes group font size
- **Activate cross hair** : Activate cross hair which enables object alignment.
- **Activate smoothing for texts** : Activate anti-aliasing (smoothing) for texts, it can slow the display.
- **Activate smoothing for images** : Activate anti-aliasing (smoothing) for images, it can slow the display.
- **Activate tooltips** : Activate tooltip-system.
- **Configuration of automatic layout** : Let you configure the direction of automatic layout done with F7.
- **Part that will be laid out** : Let you define if the the layout is made on upstream or downstream part when you press F7.
- **Draw places and transitions according IEC 62551** : Draw places and transitions according IEC 62551. Transitions will be drawn depending on their law.

10.6. Digital format

Digital format tab enables to customize digits display :

- **Display of parameters** : Specifies the display of parameters (number of digits, ...).

10.7. Places

Places tab defines places display options. :

- **Label size** : Specifies size of places font.
- **Display name** : Enables to display place name or not.
- **Display number** : Enables to display place ID or not.
- **Display number of tokens** : Enables to display number of tokens or not.
- **Display name on shortcuts** : Enables to display the name of reference place on each shortcut.

10.8. Transitions

Transitions tab defines Transitions display options :

- **Label size** : Specifies size of transitions font.
- **Display name** : Enables to display transition name or not.
- **Display number** : Enables to display transition ID or not.
- **Show HISTO characteristics** : Enables to display HST flag of transition or not.

- **Display firing options (PRIO, EQP ...)** : Enables to display firing options of transitions.
- **Display SEQ flag if needed** : Enables to display SEQ flag for transitions with sequential assignments.
- **Display guards** : Enables to display guards of transition or not.
- **Display assignment** : Enables to display assignments of transition or not.
- **Display law** : Enables to display law of transition or not.
- **Use a default law** : Enables to use a specific law as default law.
- **Default law** : Enables to define the default law to be used.
- **Maximum number of characters by line** : Specifies maximum number of characters per line.
- **Lexical check before computing** : Activate lexical check of transitions before each computation.

10.9. Arcs

Arcs tab defines arcs display options :

- **Label size** : Specifies label font size.
- **Link arrow width** : Specifies arrow width.
- **Link arrow height** : Specifies arrow height.
- **Display arc weight** : Enables to display arcs weight or not.
- **Show arrows links to the foreground** : Displays the direction of links on the foreground. Otherwise, the arrows will be drawn behind.

10.10. Local data

Local data tabs enables to change local data display :

- **Label size** : Specifies label font size.
- **Display data name** : Display data name.
- **Display data domain** : Display domain of definition.
- **Display data definition** : Display variable definition.
- **Display data value** : Display data value.

10.11. Simulation

Simulation tab enables to set simulation up :

- **Automatic firing of dirac 0** : Automatic firing of transition with Dirac 0 law.
- **Automatic firing when simulation starts** : Automatic firing of transition when simulation starts.
- **Transitions fireable before loop detection** : Specifies maximum number of fire before loop detection.
- **Limit duration for "fast forward" (s)** : Specifies the number of seconds before the detection of a loop during the "fast forward".
- **Graphical Rendering of simulation** : Activate or not graphical rendering of simulation or each modification.
- **Keep history during "fast forward"** : It enables history-saving of triggered transitions,the ones automatically triggered when usingGo until time/transition functions
- **Delay between transition fire (ms)** : Delay between two automatic fire.
- **Simulation trace** : Enable simulation trace.
- **Path to trace file** : File where simulation is saved.
- **Choose the way to fire transition with "on demand" type** : Displays a dialog box when firing "on demand" transition.
- **Follow scheduler order for deterministic transitions** : Only the first deterministic transition of scheduler can be fired.
- **Choose delay of fired transitions** : Displays a dialog box to choose delay of fired transition.
- **Uncertainty propagation for interactive simulation** : If not checked, the value of variable with uncertainty will be the average. For example, 1.5 for a unif(1,2).
- **Unlimited histogram** : Enables full histogram.
- **Display histogram (transition)** : Enables to choice type of histogram for transitions.
- **Parameter (nb classes, nb and size of steps ...)** : Specifies parameter for transition histograms.

10.12. Prototypes

Prototypes tab contains options of prototypes. :

- **Verify compatibility with Stochastic bloc module** : Verifies compatibility of generated blocks with Stochastic bloc module
- **Path of the directory of the prototype library** : Directory where are saved prototypes of Petri12 library.

10.13. Curves

Charts tab enables to change charts drawing :

- **Set graphics borders** : Add borders to charts.
- **Set generic values borders** : Add borders to generic values under charts.
- **Display grid** : Display grid on curves area.
- **Display legends** : Display legends under curves.
- **Drawing zone transparency** : Activate curves area transparency.
- **Graphic transparency** : Activate charts transparency.
- **Title size** : Specifies charts title font size.
- **Generic values size** : Specifies generic values font size.
- **Point size** : Specifies point size on curves.
- **Coordinates size** : Specifies coordinates font size.
- **Legend size** : Specifies legends font size.