Wolfgang - The Petri Net Editor

Documentation

IIG Telematics, University of Freiburg

Contents

1 Introduction

Wolfgang is a lightweight tool that allows users to easily create and edit Petri nets and check them against general and workflow specific net properties. Specifically, Wolfgang supports full PNML conform editing functionality of Place-Transition Nets (P/T-Nets [1] and Colored Petri Nets (CPN [2]) as they are defined in the SEPIA [3] package. Its simple and

[1] http://doku.telematik.uni-freiburg.de/sepia/supported-petri-net-types/pt-nets-placetransition-nets
[2] http://doku.telematik.uni-freiburg.de/sepia/supported-petri-net-types/cpns-colored-petri-nets
[3] http://doku.telematik.uni-freiburg.de/sepia/supported-petri-net-types

easy to use interface allows to quickly create Petri nets from scratch, edit them graphically in many ways, store and even export them as PDFs. Use WOLFGANG to smoothly create the Petri net you want!

This documentation was last updated on July 29, 2015 and refers to WOLFGANG release version 1.0.0, which can be found on
https://github.com/iig-uni-freiburg/WOLFGANG[4].

2 (Workflow-) Net Properties

The Petri net types presented in SEPIA[5] build the foundation for the standalone Java application WOLFGANG. The tool allows fast creation, editing and "playing-back" of P/T-Nets and CPNs. Originating from a process-oriented perspective, WOLFGANG also supports checking for typical net and workflow-specific properties in P/T-Nets and CPNs.

2.1 Net Properties

Each Petri net type defines validity and soundness properties, whereas validity refers to a correct net structure (i.e. a net specification which makes sense somehow) and soundness typically refers to workflow-specific properties of Petri nets. The soundness property implies a valid Petri net[6]. A further typical net property is the boundedness of a net.

2.1.1 Validity and Soundness

We can check for validity in WOLFGANG and get some topological information of the net. A P/T-Net is valid, if its structure makes sense, e.g. flow relations don't connect two places or vice versa. A valid P/T-Net is also sound. [7]

CPNs require flow relation effectiveness for validity. This means, that each relation must move at least one token from a place to a transition or vice versa. For the soundness property no more conditions must be met, s.th. a valid CPN is also sound[8].

Based on the fact that in both net types validity also implies soundness, the soundness check is omitted in WOLFGANG.

2.1.2 Boundedness

A Petri net is bounded if and only if, for every reachable state and every place p the number of tokens in p is bounded. The boundedness property on a Petri net is checked by building the marking graph. If the number of possible sequences is finite or smaller than the maximum possible value for the integer data type, the Petri net is bounded. Otherwise the Petri net is assumed to be unbounded. [9]

2.2 Workflow Properties

WOLFGANG allows checking structural and soundness-related workflow properties:

[4] https://github.com/iig-uni-freiburg/WOLFGANG
[5] http://doku.telematik.uni-freiburg.de/sepia/supported-petri-net-types
[6] http://doku.telematik.uni-freiburg.de/sepia/supported-petri-net-types/petri-nets-in-sepia
[7] http://doku.telematik.uni-freiburg.de/sepia/supported-petri-net-types/pt-nets-placetransition-nets
[8] http://doku.telematik.uni-freiburg.de/sepia/supported-petri-net-types/cpns-colored-petri-nets
[9] http://doku.telematik.uni-freiburg.de/sepia/reachability

2.2.1 Workflow-Net

A Workflow Net (WF-Net) is a Petri Net which has two structural properties (WF-Structuredness)[10]:

- A WF-net is a P/T-Net that has one input place (i) and one output place (o). A token in i corresponds to a case which needs to be handled, a token in o corresponds to a case which has been handled.
- Every transition and place contributes to the processing of a case. Therefore, every transition t (place p) is located on a path from place i to place o (this corresponds to strongly connectedness if o is connected to i via an additional transition).

Based on the typical properties of a Petri Net and WF-structuredness, a WF-Net is sound if and only if the following three requirements are satisfied [11]:

- Option to complete: for each case it is always still possible to reach a state which marks end place (o).
- Proper completion: if o is marked all other places are empty for a given case.
- No dead transitions: it is possible to execute an arbitrary activity by following the appropriate route through the WF-Net.

2.2.2 Colored Workflow-Net

A Colored Workflow Net (CWN) is a bounded Colored Petri Net (CPN) which additionally satisfies the structuredness properties of a WF-Net (where the token of a WF-Net is represented by the "black"-colored control flow token instead): The soundness of Colored Workflow Nets (CWN)[12] is closely related to the soundness of WF-Nets. The formal definitions of CWN-structuredness and -soundness may be found here[13].

3 Getting Started

Wolfgang is hosted as a GitHub-project and can be downloaded from https://github.com/iig-uni-freiburg/WOLFGANG [14]. The tool requires a Java VM on the target platform. Wolfgang is available as a Mac OS Application, Windows executable and as an executable JAR file which simply can be launched from the terminal via $ $java$ $-jar$ $wolfgang$-1.0.0.jar.

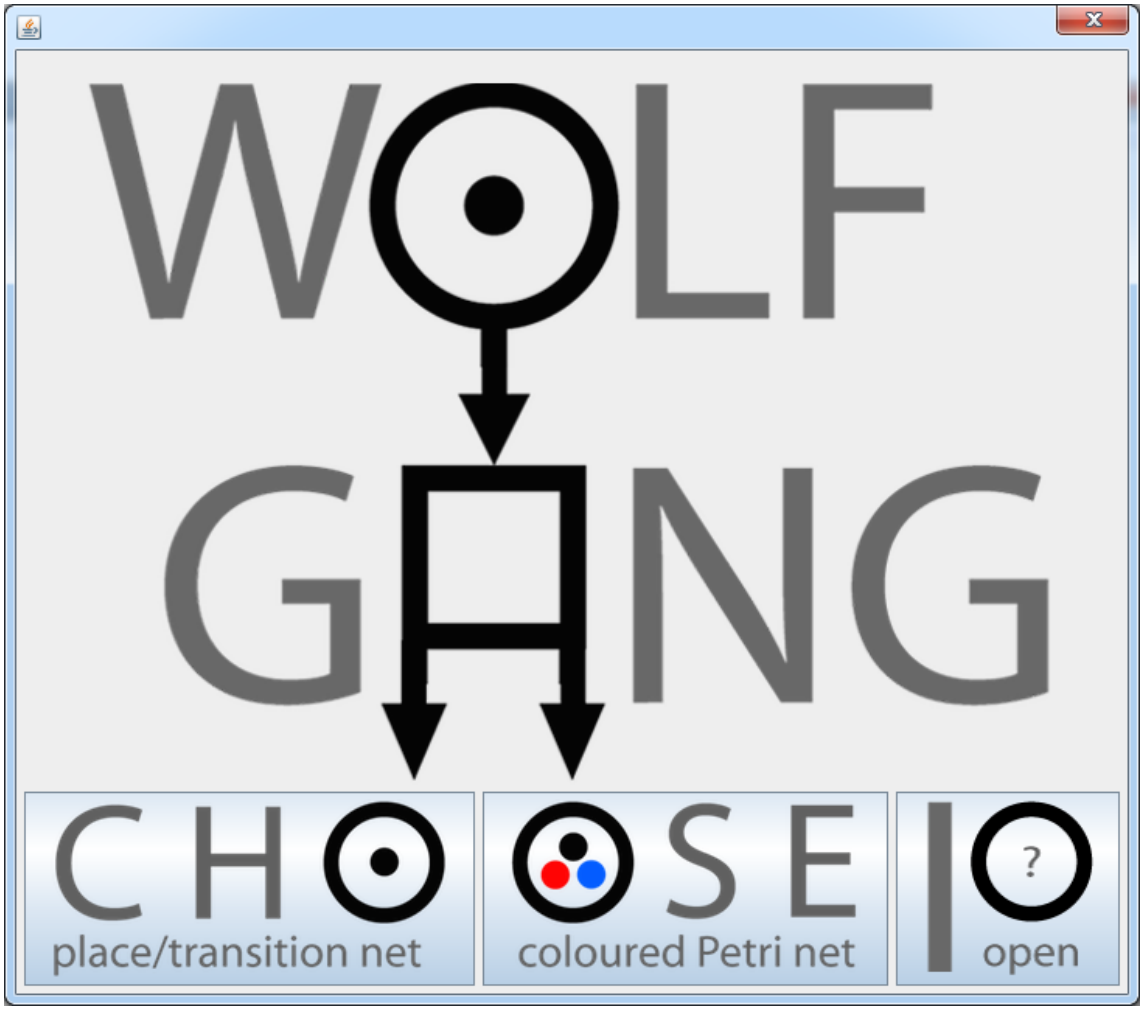
4 User Guide

Wolfgang is a Petri net modeling and editing tool which fully supports PNML Standard. The objective of the tool is to allow modelling of P/T- and CP-Nets and check them against general and workflow-specific properties. This user guide introduces the core functionality of *Wolfgang* and explores the different ways the tool offers to edit Petri nets.

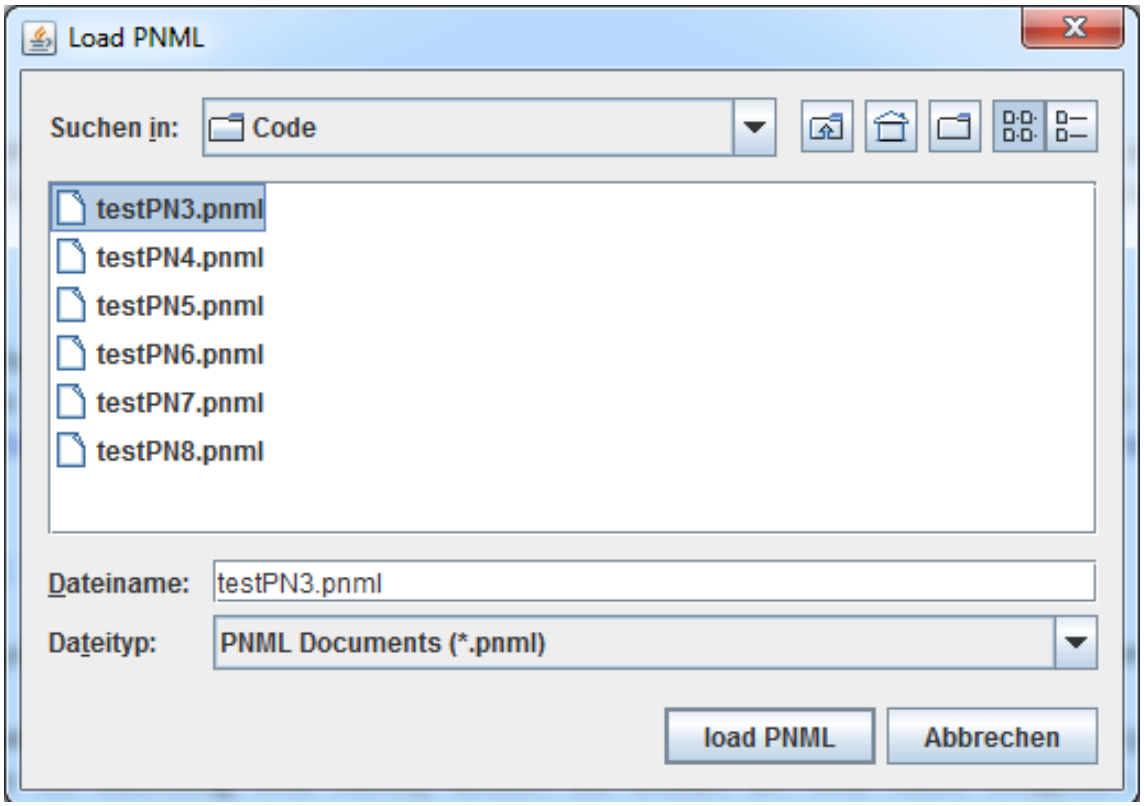
[10] http://link.springer.com/chapter/10.1007%2F3-540-63139-9_48
[11] http://link.springer.com/article/10.1007%2Fs00165-010-0161-4
[12] http://link.springer.com/chapter/10.1007%2F978-3-642-41098-7_13
[13] http://link.springer.com/chapter/10.1007%2F978-3-642-41098-7_13
[14] https://github.com/iig-uni-freiburg/WOLFGANG

4.1 Startup

The user is able to start WOLFGANG after Java was installed and configured successfully. The tool is showing its startup-screen directly after the start. This first screen is going to be discussed in the following chapter.



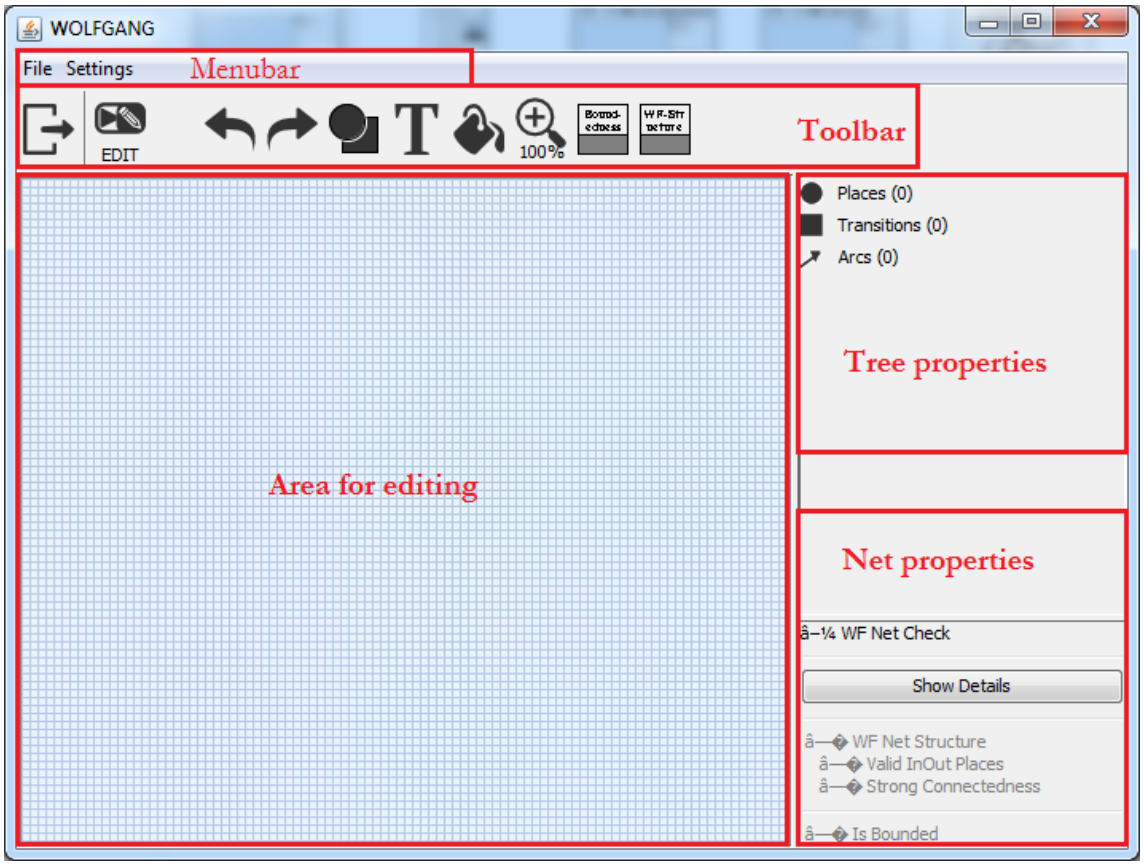
The startup-screen offers the possibility to choose between two different net types, or to open an existing net. Users can either create a place/transition net (P/T-Net) by clicking the left button or start to construct a colored Petri net (CPN) by selecting the one located in the middle. An open-dialog appears by clicking the "open" button on the right side of the startup-screen. Subsequently it is possible to open an existing P/T-Net or CPN in PNML format by selecting a net and clicking on the button "load PNML".



After one option has been chosen, the startup-screen disappears and the home-screen shows up.

4.2 The Home-Screen

This section introduces the home-screen of WOLFGANG. There are two types of home-screens, depending on the selected net type. They differ in the layout of the toolbar and the properties for checking the net. However, the differences are only relevant when it comes to editing the nets, which is shown in the next chapter. The following figure visualises the home-screen for a P/T-Net.



The core of WOLFGANG is located in the middle of the screen, where nets can be edited. The right section shows the graph properties and workflow-specific net properties which both are affected when adding elements to the net. These elements are going to be discussed in section 4.3

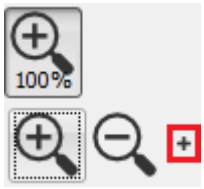
4.2.1 Toolbar

Another important component is the toolbar. The user is able to export the displayed net in PDF format by clicking initially on the first item of the toolbar and then on the icon which shows up afterwards.

A dialog appears where the location and name of the file needs to be chosen, after it can be converted and finally saved. The next two buttons trigger the common undo and redo function. The functionality can also be provoked by pressing the keystroke combination STRG+Z or STRG+Y respectively.

The last option of the toolbar which needs to be introduced in this section is the zoom function. It can be utilised by clicking first on the zoom button and then on the plus or minus icon respectively in order to zoom in or zoom out.

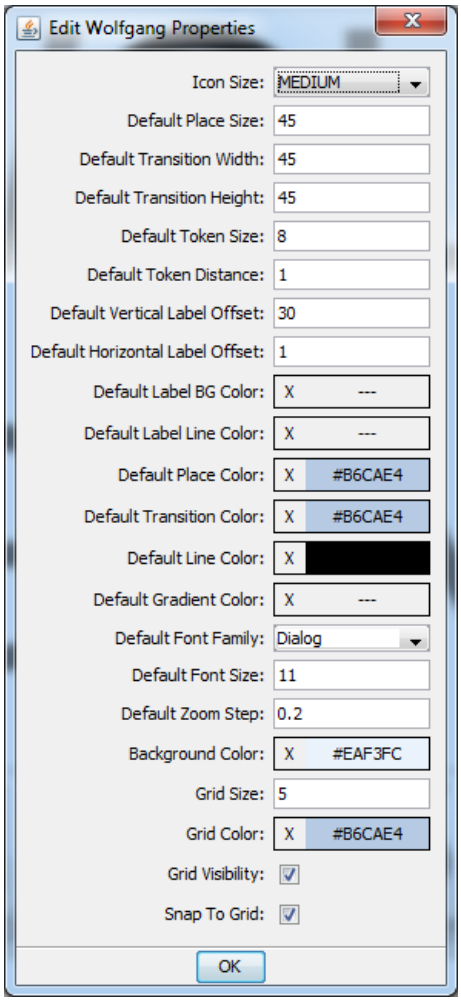
Particularly icons in the toolbar can be expanded by clicking on them. As a consequence, new buttons are shown in a horizontal list under the icon. Each list displays a "plus-sign" on its right side. By clicking the "plus-sign" the list of buttons from the toolbar is encapsulated.



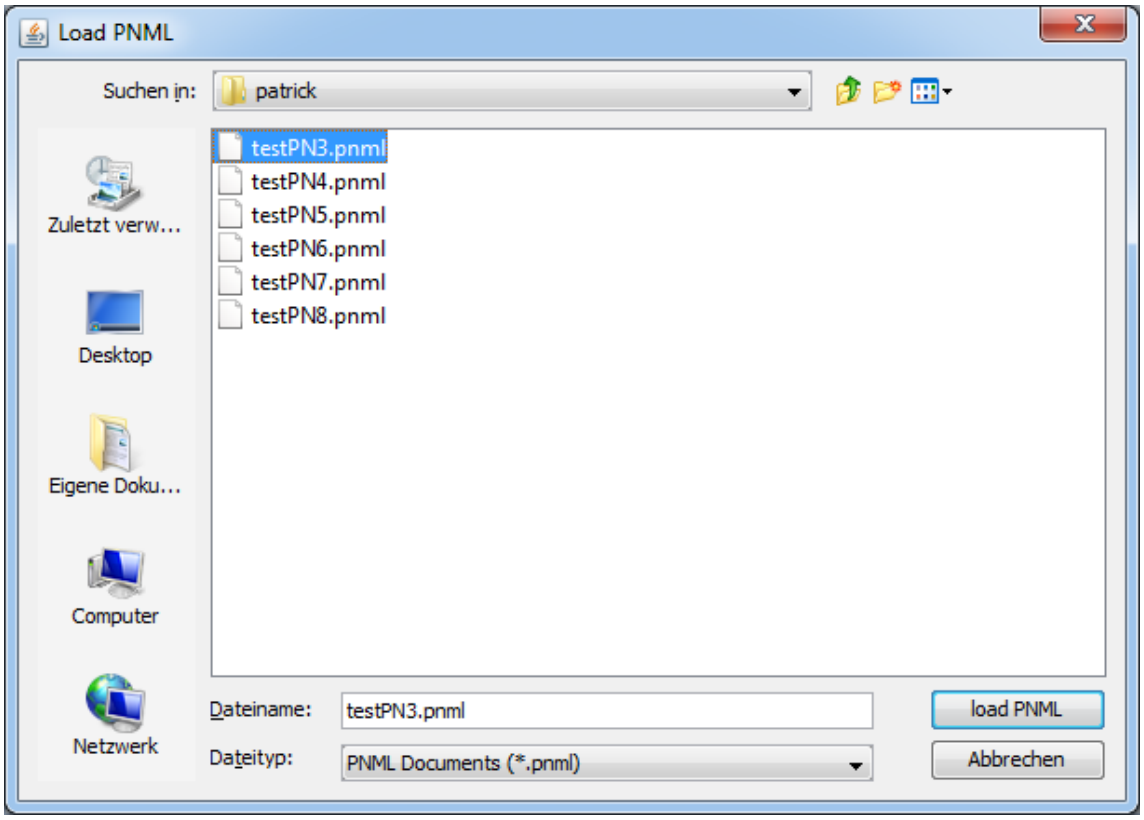
This feature is useful, when essential functions want to be utilised directly without extra clicking. All other options, which can be chosen from the toolbar, are explained in the section 4.3.

4.2.2 Menubar

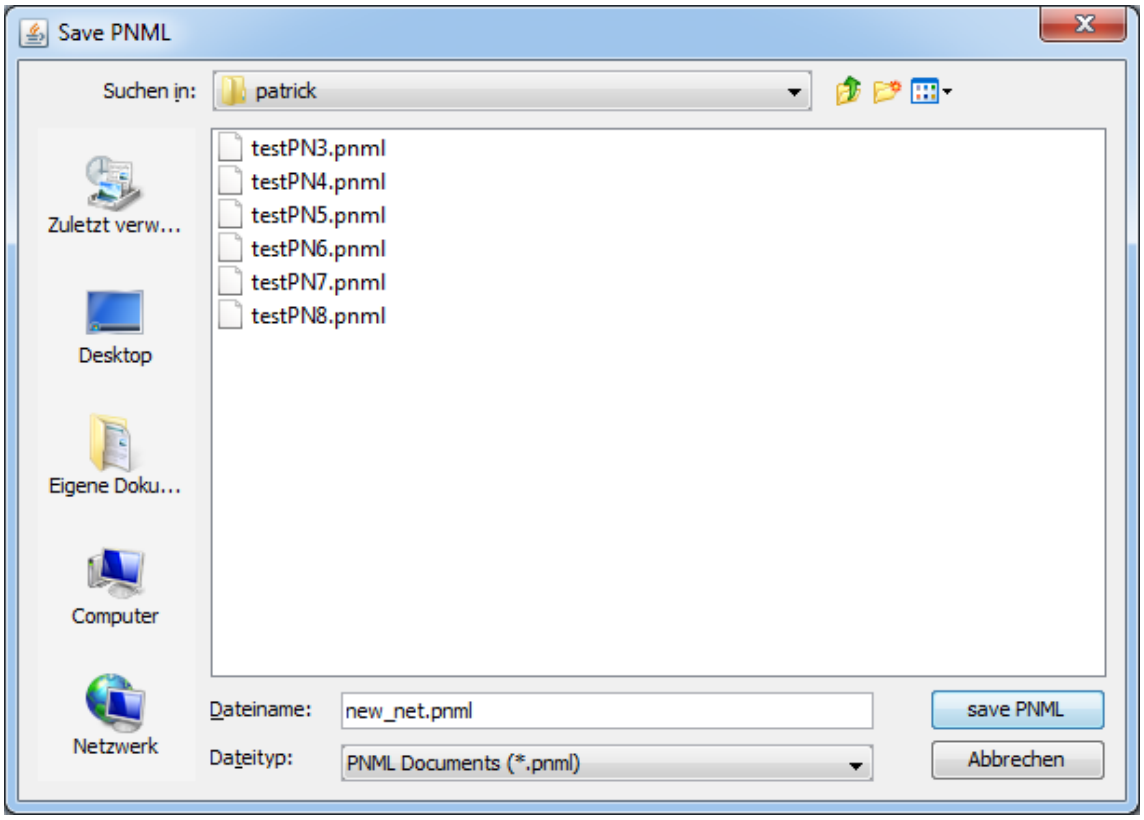
There are plenty of different options, which can be invoked from the menubar. For example, P/T-Nets and CPNs can be built by selecting File→New and choosing the fields P/T-Net or CPN. This way, several individual instances of home-screens can be initialised. Moreover, the user is able to change a variety of properties in WOLFGANG by selecting the item Settings→Edit WOLFGANG properties... in the menubar. The fields of the following dialog can be edited.



The property window shows plenty of options which are relevant for the layout of the home-screen and the design of the nets. The menubar in the home-screen also offers the option to open existing nets. A Dialogue shows up when the user clicks on File→Open .pnml in new Window.

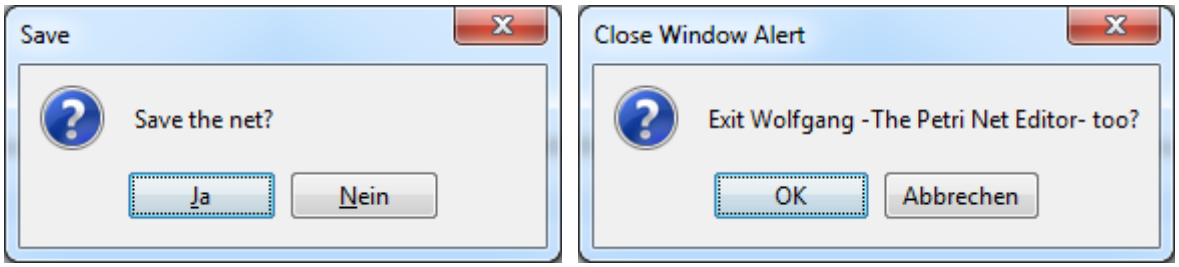


Subsequently, a P/T-Net or CPN in the standardised .pnml format can be selected. The chosen net is displayed in a new instance of the home-screen after the user clicked on the button with the caption "load PNML". WOLFGANG is able to save changes of an already as .pnml file existing net by clicking on File→Save in the menubar. There is the option File→Save as... in the case that the user wants to save the modified net in a new .pnml file. A dialogue comes up where a file name and location needs to be determined.



Afterwards the net can be stored by selecting the button "save PNML". A click in the menubar on File→Save provokes the same action as the selection of File→Save as..., if the user wants to store a newly created net, which has not been saved before.

When the user closes an instance of WOLFGANG by clicking on File→Exit or through an action of the operating system, two questions come up. The first one asks, if the net should be saved. Then, if the answer is yes, the File→Save as... action is being triggered, otherwise the question disappears. The second questions asks, if the software should quit completely or close just the active instance. A click on the "OK" button terminates the programme.



All functionalities of the home-screen have been introduced in this section.

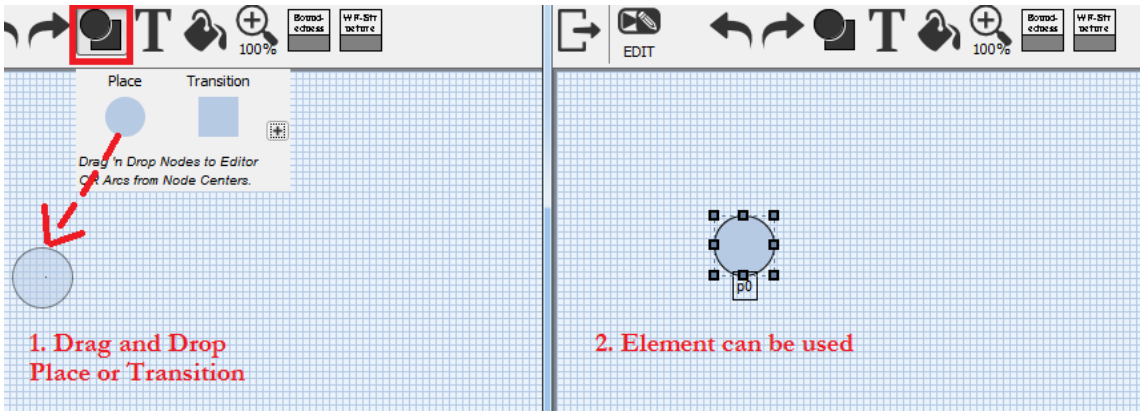
4.3 Editing nets

This chapter explains how to initially create and edit nets in general. Explicit differences between a P/T-Net and CPN are shown in the subsections 3.2??? for P/T-Nets and 3.3???

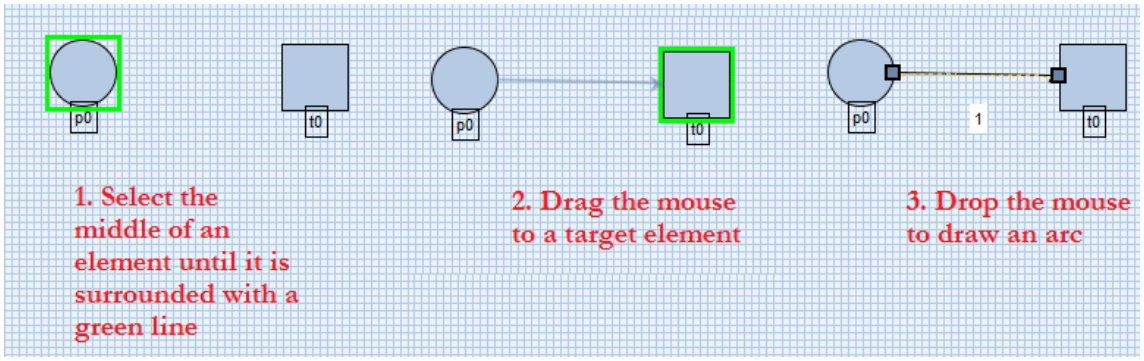
for CPNs. Various features for editing can be found in the toolbar, done with the mouse or used by a keystroke combination.

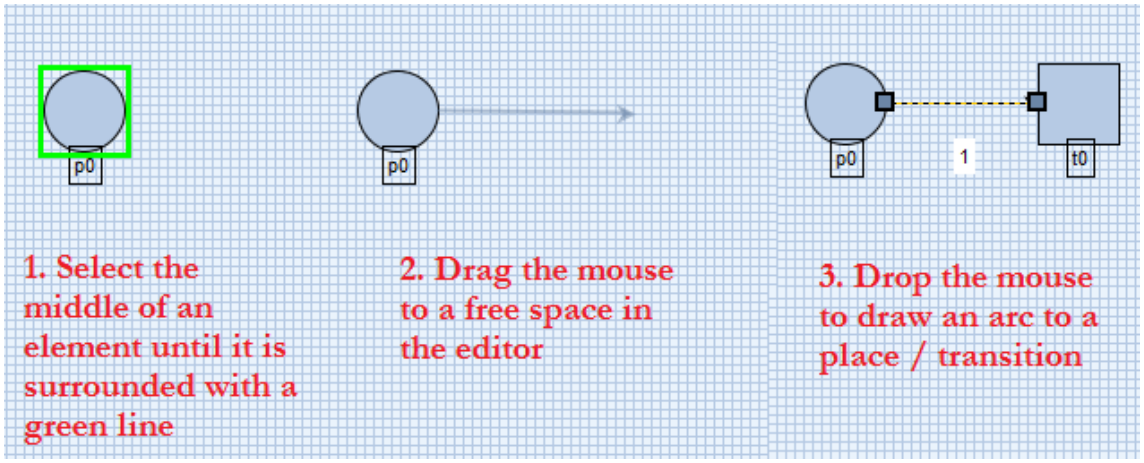
4.3.1 Creating a net

One of the most essential function is adding new places and transitions to a net. This can be done by dragging and dropping the following symbols from the toolbar to the area for editing.



Furthermore, it is possible to draw arcs in order to connect one or more places with one or more transitions. Dragging and dropping the mouse from the middle of an element when it is surrounded with a green line allows the user to draw an arc directed to the counterpart of its source element. If no target element exists, it is feasible to just drag and drop an arc from a source. A new place or transition will be created respectively.

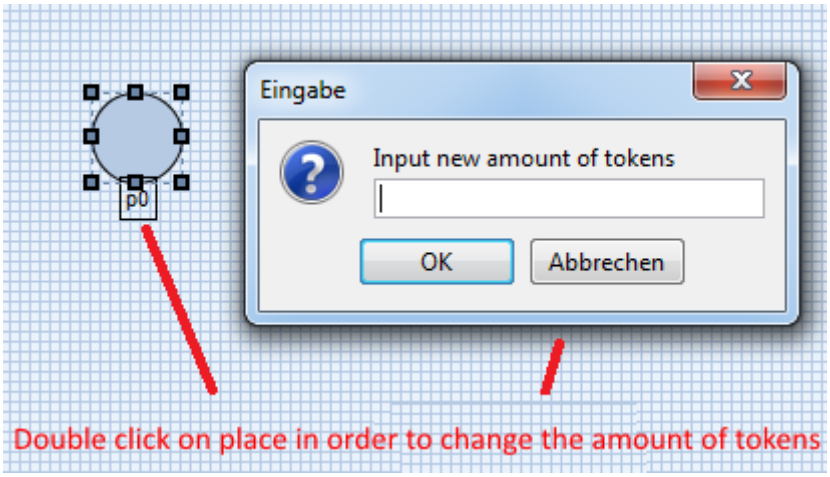




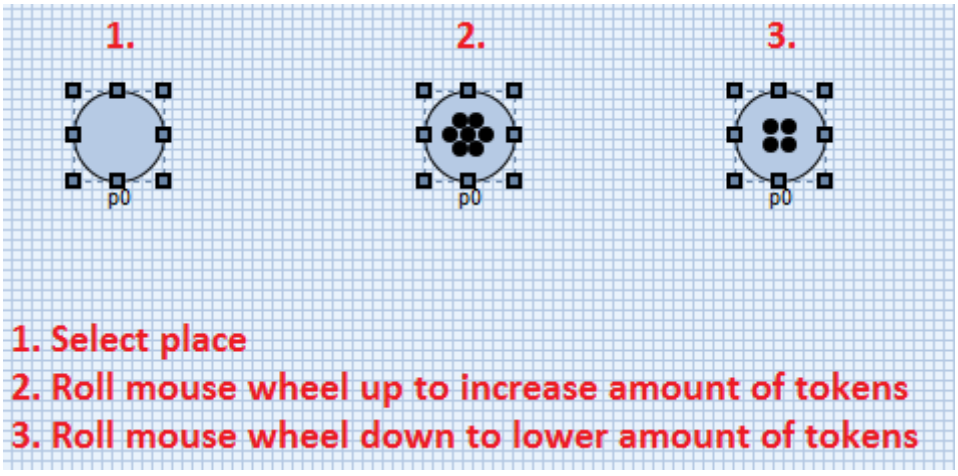
It is possible to just pull the arc to a different place or transition, in the case that the source or target node needs to be changed. The next chapter explains various special functions of a P/T-Net.

4.3.2 P/T-Net specific editing

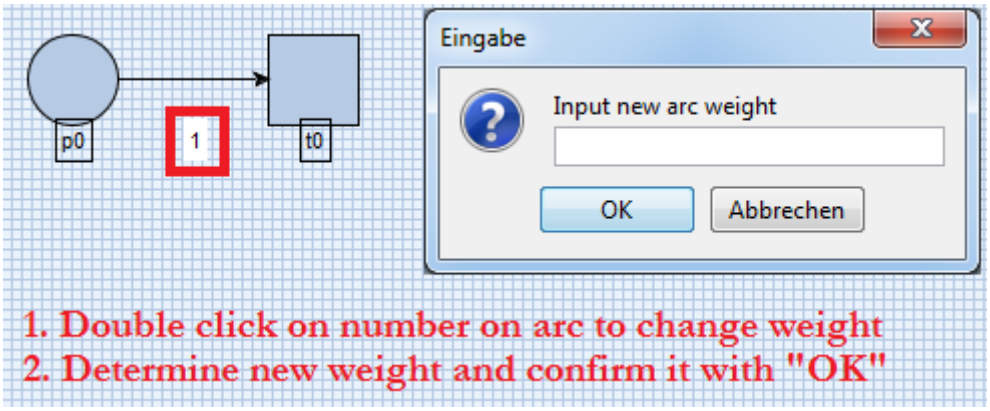
There is a way to change the amount of tokens in a place, by double clicking on it. This feature is different for P/T-Nets and CPNs. The following figure displays the dialog which pops up in a P/T-Net.



Another way to change the number of tokens is to select a place and roll the mouse wheel up or down to increase or decrease the amount.



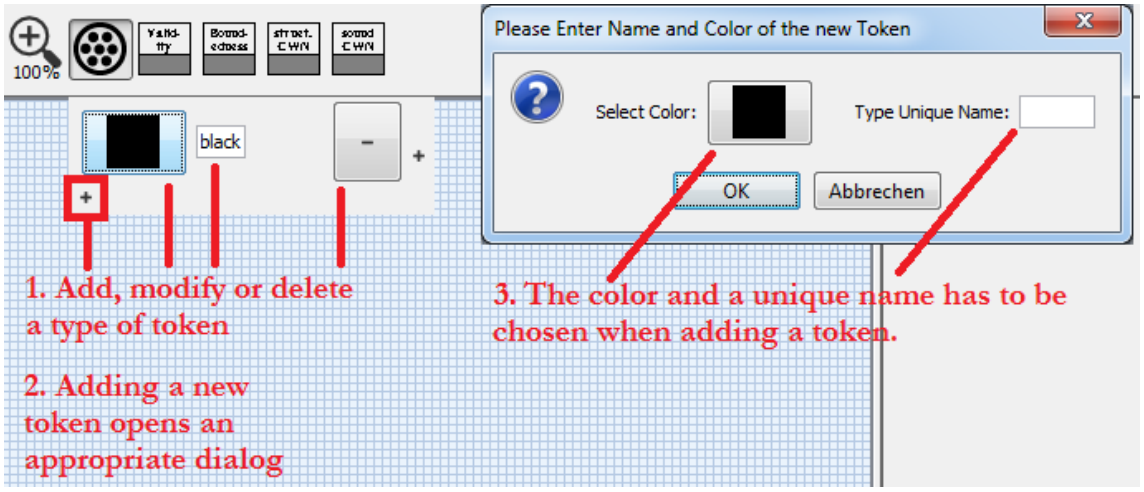
The last option to influence a net's functionality is to modify the constraints of an arc. The constraint types in P/T-Nets and CPNs differ: In the case of a P/T-Net, this can be done by double clicking the arc weight number located in the middle of the appropriate arc.



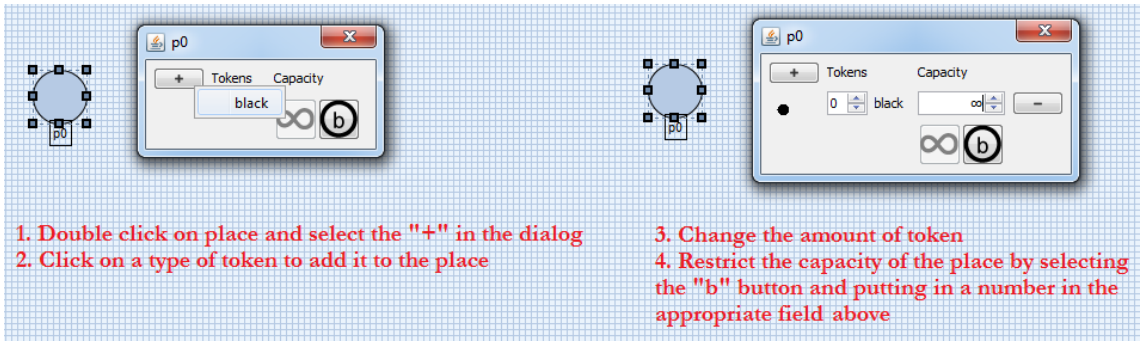
The number determines how many tokens are consumed from source places and produced in destination places when the connecting transition is fired. The following subsection exemplifies, how to change the amount of tokens, modify the constraint of an arc and trigger other useful features regarding a CPN.

4.3.3 CPN specific editing

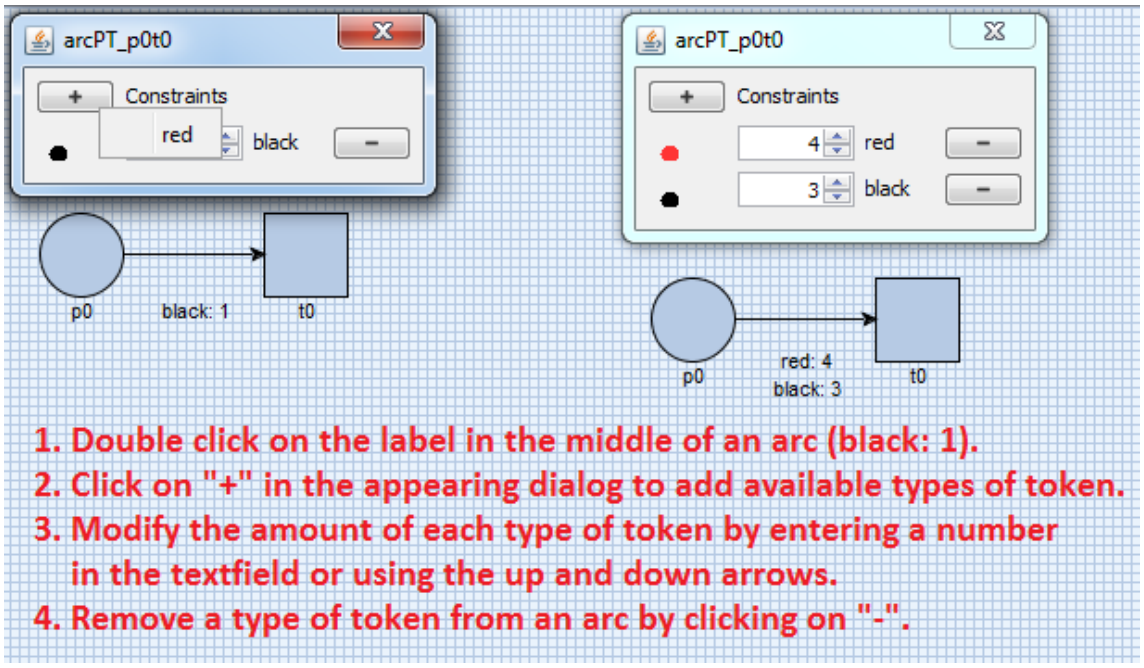
A CPN is only useful, when different types of tokens are needed. Otherwise a normal P/T-Net is sufficient. A click on the proper icon in the toolbar provides a possibility to create new types of tokens.



The option of changing the amount of tokens in a place offers more detailed options when creating or modifying a CPN.



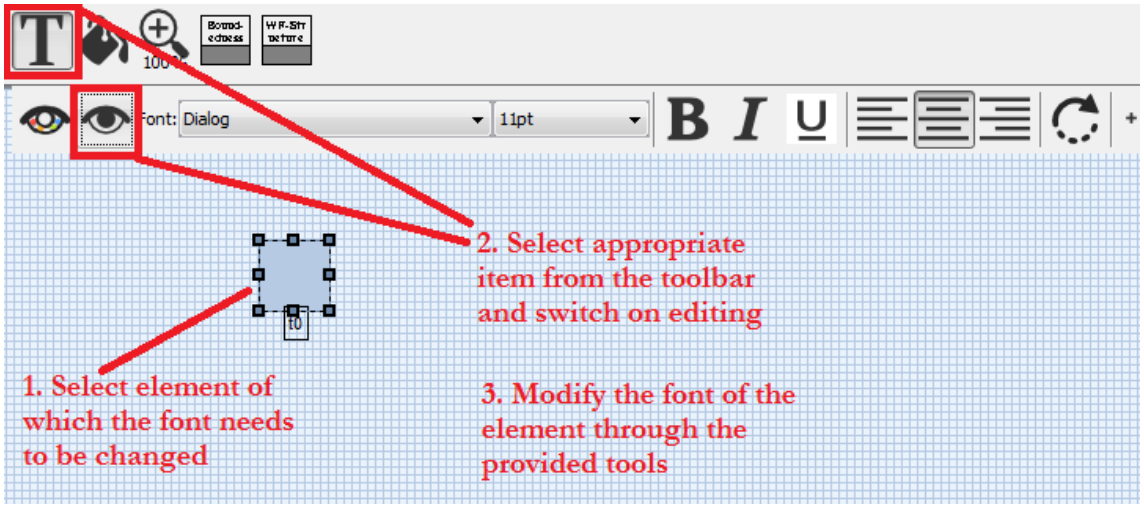
The following figure illustrates how to modify the constraint of an arc.



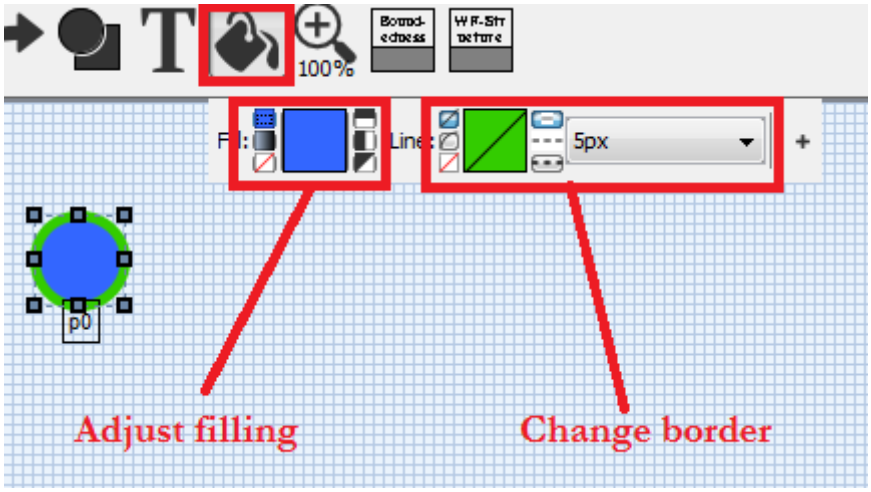
The sections 3.2 and 3.3 showed features, which are different in P/T-Nets and CPNs. The subsequent chapters discuss options which are available for both types of nets.

4.3.4 Modifying the representation of a net

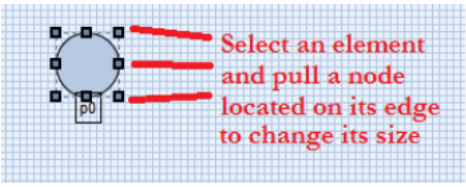
Once elements have been added to a net they can be adjusted. For example it is possible to change the font of an element, by selecting it and then choose the appropriate item from the toolbar. Subsequently it is necessary to enable the provided options.

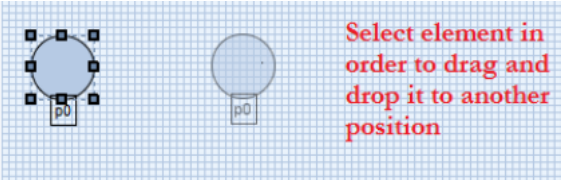


The fill colour and style as well as the border of an element can also be modified by selecting the associated icon from the toolbar.

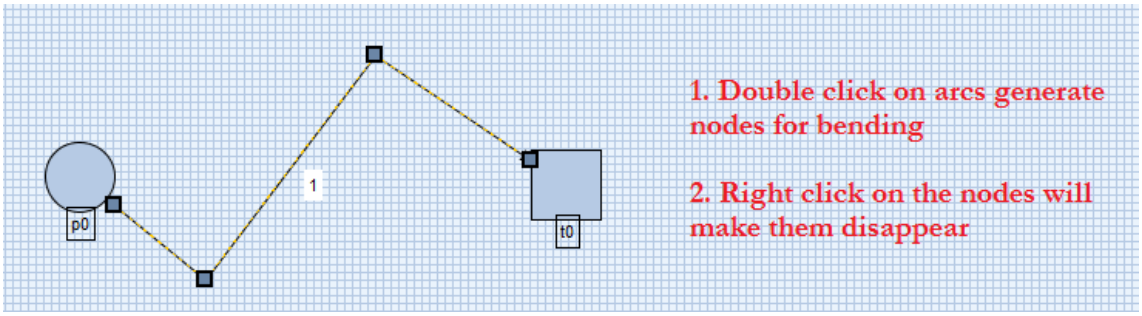


The size of a transition or place can be modified if it is too small or too big. Therefore the object firstly needs to be selected and secondly a node on the edge of it has to be pulled. The position of one or more elements can be changed by selecting the targets and then drag and drop them to another location on the editor.





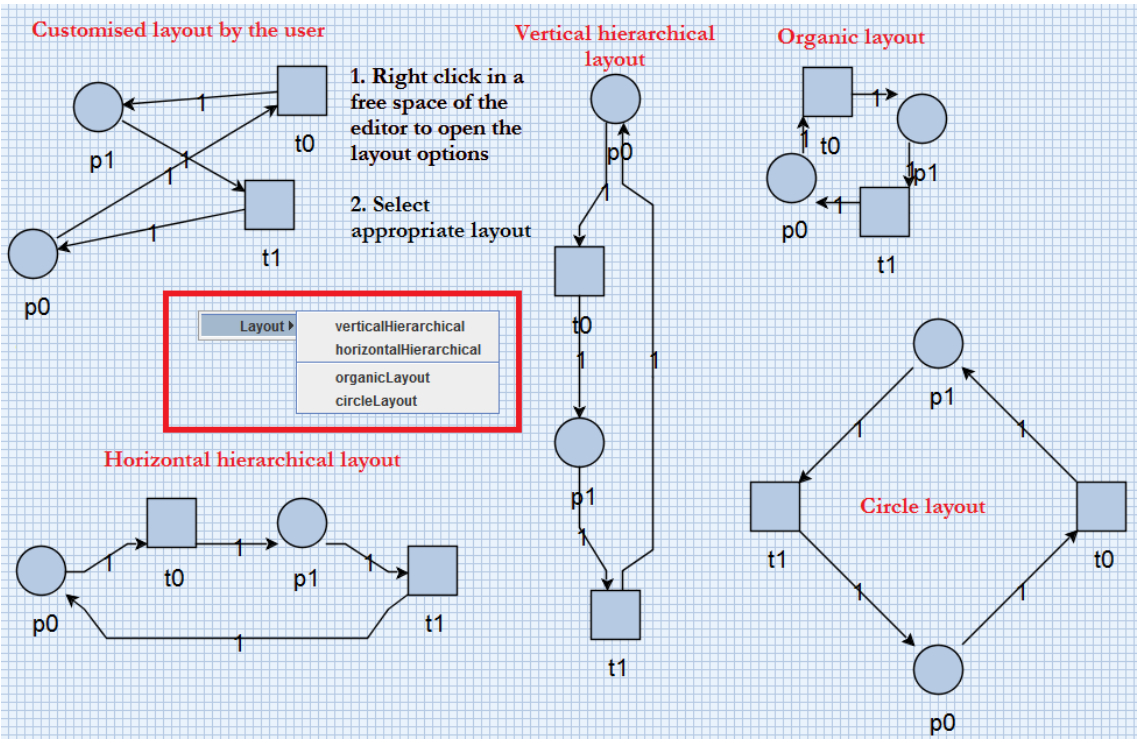
Arcs can be bent by double clicking on them. A node will appear, which can be utilised to alter the angle. This process can be repeated in order to generate more nodes.



There are also four options to change the layout of an arbitrary net

- Horizontal hierarchical layout
- Vertical hierarchical layout
- Organic layout
- Circle layout

Their purpose is to represent any net in a consistent way. The different layouts can be selected by right clicking in a free space of the editor. A list pops up which shows all available layouts.



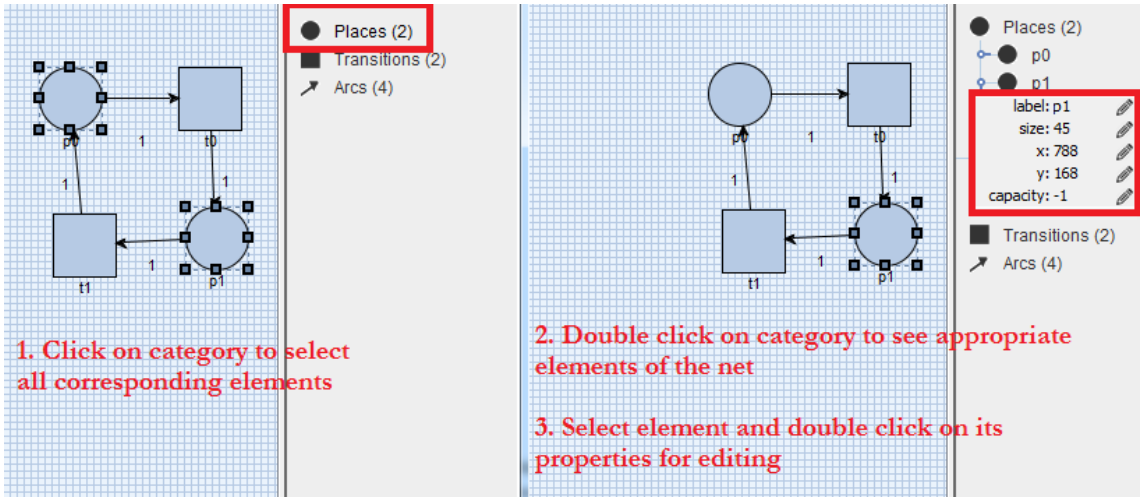
4.3.5 Editing using the keyboard

There are plenty of integrated key commands and combinations to change a net. As a consequence nets can be created and edited in a more efficient way.

- CTRL+c: Copies a selection of elements.
- CTRL+x: Cuts a selection of elements.
- CTRL+v: Pastes the cut or copied selection of elements.
- CTRL+z: Undo previous action.
- CTRL+a: Selects every element in the net.
- CTRL+d: Deselects every element in the net.
- CTRL+e or CTRL+f: Selects all arcs.
- CTRL+p: Selects all places.
- CTRL+t: Selects all transitions.
- CTRL+l: Opens a dialog to save the net in PDF format.
- Delete key: Deletes the elements which are selected.
- Arrow keys: Moves the selection of elements in a direction.
- CTRL+Arrow key: Draws an arc from the selected element to a new place or transition respectively.

4.3.6 Editing using the property tree

The last opportunity to modify a net is by using the property tree located on the right side of the editor. All elements which are utilised in the net are grouped by their category. Main categories are places, transitions and arcs. A click on a category selects all elements of the particular type in the net, whereas a double click expands a list that shows every member of the category in a tree view. Another click on a specific element in the tree opens its properties. These can be edited by double clicking on their values.



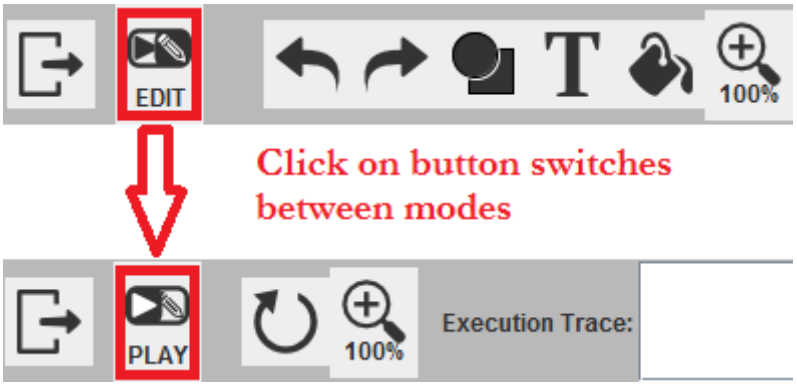
The following options can be modified through the property tree:

- Place
  - Caption of the label
  - Size
  - Position on x-axis
  - Position on y-axis
  - Capacity (only displayed for P/T-Nets): The maximum number of tokens that can reside in the place. A capacity of ∞ means unboundedness, i.e. the number and kind of tokens is not restricted.
- Transition
  - Caption of the label
  - Width
  - Height
  - Position on x-axis
  - Position on y-axis
- Arc
  - Weight (only displayed for P/T-Nets): Number of tokens to be consumed/produced over the arc

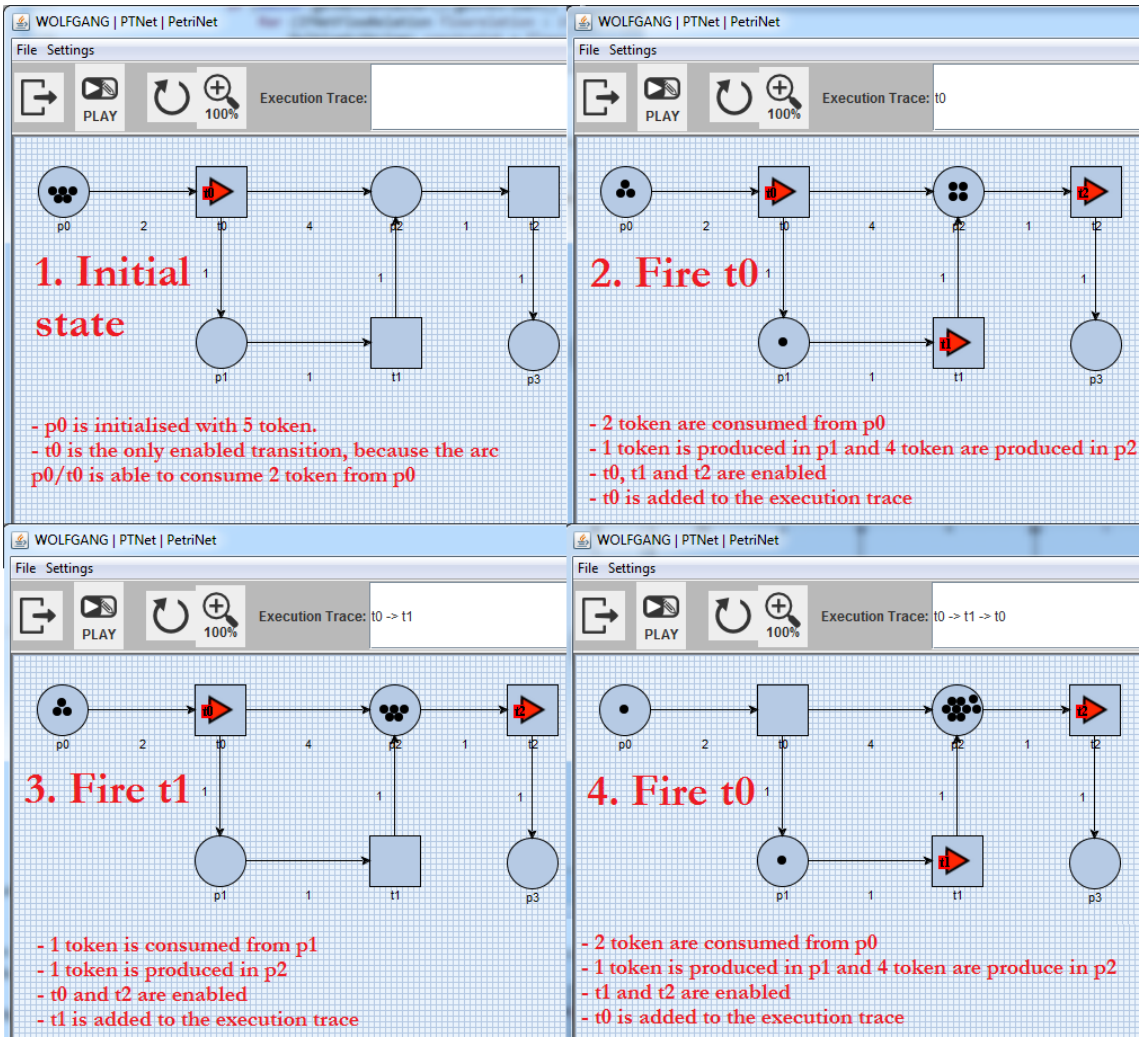
All settings of a net can be changed during "'edit-mode"'. Moreover, there exists a "play-mode" for executing nets, which is going to be explained in the next section.

4.4 Executing Petri Nets

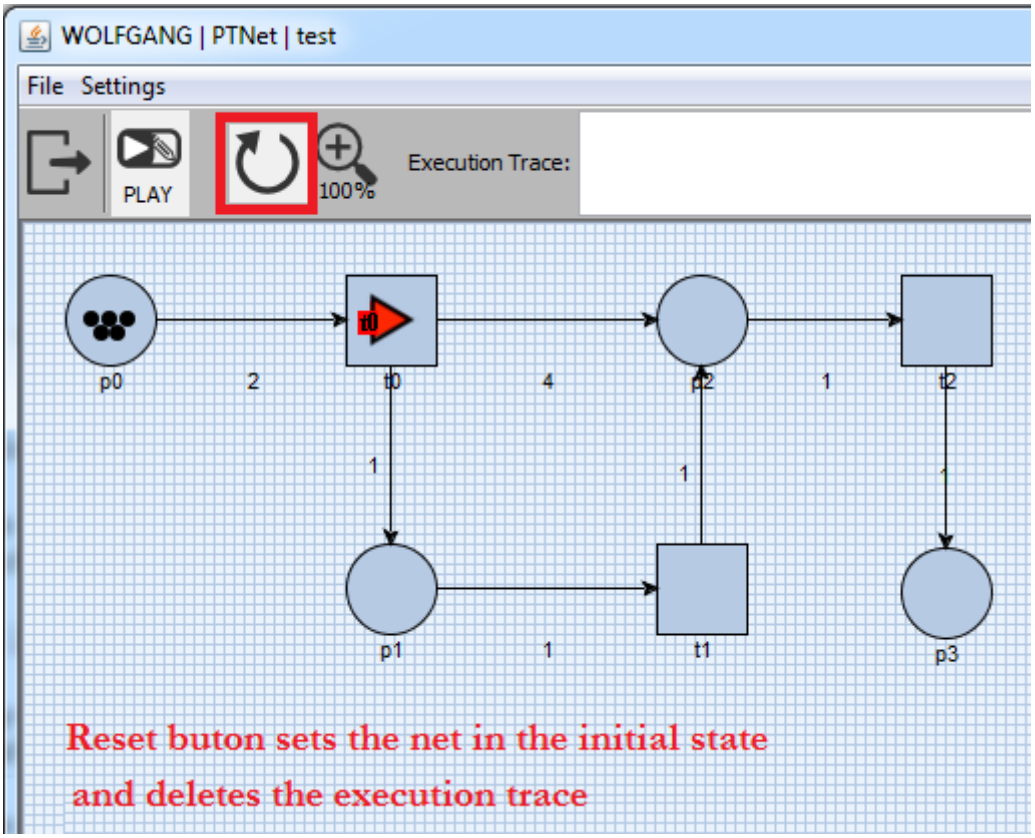
A Petri net is ready for executing, once it has been created and configured. In this case "executing" means that the user is allowed to fire enabled transitions manually in order to traverse the net. A transition is enabled, if there are at least as much tokens in the source place as an incoming arc consumes, and if the target place is able to gather all the tokens which are produced by the outgoing arc. One way to switch to the play-mode is by clicking the dedicated button in the toolbar.



No changes of the net are allowed in play-mode. Hence, elements cannot be moved and keystroke combinations do not work either. Furthermore, the editing buttons for adding and modifying elements disappear from the toolbar. However, a field called "execution trace" shows up instead. The sequence of fired transitions is stored in this textbox. An enabled transition is highlighted through a red arrow inside, which signals the user that a click on it triggers the consume/produce action. The following example illustrates how this process works.



A net can be reset by clicking the appropriate button from the menubar, which is only visible in play mode. The execution trace is going to be emptied and the net returns into the initial state before any transition has been fired.


WOLFGANG | PTNet | test
File Settings
PLAY
100%
Execution Trace:
p0
2
t0
4
p2
1
t2
1
1
1
p1
1
t1
p3
Reset buton sets the net in the initial state
and deletes the execution trace