Author:
Shubham Agarwal
Department of Information Technology,
Indian Institute of Information Technology, Allahabad, India

Supervisors:
Prof. Dr.-Ing Monika Heiner
Dipl. Inf. Christian Rohr
Department of Computer Science,
Brandenburg Technical University, Cottbus, Germany
Abstract

Graphical User Interface Reengineering is a way of changing and modernizing the look of an application. Snoopy is a tool for designing and simulating Petri Nets. This report mainly focuses on the GUI reengineering of Snoopy and its simulation features.

The task is accomplished by dividing the simulation window into two parts namely, a control window which controls the simulation parameters, and a display window which has a plot area to display graphs. The splitting makes the simulation window more flexible for the users. The small screen users can now see the plot in full screen as the simulation controls and the generated graph are not a part of the same window. The graphical user interface is built using wxWidgets which is a C++ library that gives developers the GUI component that helps to make the application.

There is a feature of opening and closing the subcategories using collapsible panes for saving more screen space. There can be multiple plot areas created by the user and several result views (plot areas) can be open at the same time with just one click. The major design goal was to make it platform independent. wxWidgets is a library that helps us accomplish our goal as it allows us to make GUI which is platform independent.
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>A.1.2 What is inside Simulation Configuration?</td>
<td>2</td>
</tr>
<tr>
<td>A.1.3 What is inside Model Configuration?</td>
<td>3</td>
</tr>
<tr>
<td>A.1.4 What is inside Export and Import details?</td>
<td>3</td>
</tr>
<tr>
<td>A.1.5 Playing around with result views</td>
<td>4</td>
</tr>
<tr>
<td>A.1.6 Other simulation stuff</td>
<td>5</td>
</tr>
<tr>
<td>A.2 How to use result views</td>
<td>6</td>
</tr>
<tr>
<td>A.2.1 General idea of result views</td>
<td>6</td>
</tr>
<tr>
<td>A.2.2 Editing curve properties</td>
<td>7</td>
</tr>
<tr>
<td>A.2.3 Editing View properties</td>
<td>10</td>
</tr>
<tr>
<td>A.2.3.1 General Tab</td>
<td>10</td>
</tr>
<tr>
<td>A.2.3.2 Legend Tab</td>
<td>11</td>
</tr>
<tr>
<td>A.2.3.3 Axes Tab</td>
<td>12</td>
</tr>
<tr>
<td>A.2.3.4 Fixed Tab</td>
<td>13</td>
</tr>
<tr>
<td>A.2.3.5 Curves Tab</td>
<td>15</td>
</tr>
<tr>
<td>A.2.4 Working with Nodes selection dialog box</td>
<td>16</td>
</tr>
<tr>
<td>A.2.4.1 Using Regular Expression</td>
<td>18</td>
</tr>
<tr>
<td>A.2.4.2 The four buttons for switching the nodes between the list</td>
<td>20</td>
</tr>
<tr>
<td>A.2.5 Exporting your data</td>
<td>22</td>
</tr>
<tr>
<td>A.2.6 Other options</td>
<td>23</td>
</tr>
<tr>
<td>A.3 Everything you need to know about Regular expressions</td>
<td>24</td>
</tr>
<tr>
<td>A.3.1 Character class</td>
<td>24</td>
</tr>
<tr>
<td>A.3.2 Anchors</td>
<td>25</td>
</tr>
<tr>
<td>A.3.3 Quantifiers</td>
<td>25</td>
</tr>
<tr>
<td>References</td>
<td>27</td>
</tr>
</tbody>
</table>
List of Figures

1  Snapshot of old simulation dialog box of Snoopy .......... 7
2  Snapshot of Charlie. ........................................... 8
3  Snapshot of Charlie when all panes are open. ............... 9
4  Snapshot of redesigned Snoopy’s dialog box. ................. 12
5  When “Add New View” is pressed. ........................... 15
6  Giving a name to the new view. ............................... 15
7  Pressing OK button to create the new view. .................. 16
8  When Views are selected for deletion. ......................... 17
9  After the Remove views button is pressed. ................... 17
10 Time shown in seconds only .................................. 18
11 Time shown in minutes and seconds .......................... 18
12 Snapshot of Edit Dialog Box. ................................. 20
13 Snapshot of a result view with node list. ..................... 21
14 Snapshot of a result view without node list. ................. 21
A.1 Snapshot of components in New simulation dialog box. .. 2
A.2 Snapshot of expanded simulator configuration tab. .......... 3
A.3 Snapshot of expanded Model Configuration tab. ............. 3
A.4 Snapshot of expanded export import details tab. ............ 4
A.5 Snapshot of components in Result view. ..................... 7
A.6 Showing Different curves in the plot area. ................ 8
A.7 Double click on the node name in nodes list. ............... 9
A.8 Changing curve property of a curve. ......................... 9
A.9 Curve look after the properties are changed. ............... 10
A.10 Snapshot of General tab in viewer properties window. .... 11
A.11 Snapshot of Legend tab in viewer properties window. .... 11
A.12 Snapshot of Axes tab in viewer properties window. ....... 13
A.13 Snapshot of Fixed tab in viewer properties window. ...... 15
A.14 Snapshot of Curves tab in viewer properties window. ..... 16
A.15 Snapshot of components in Nodes selection dialog box. . 18
A.16 Node selection window without regular expression. ...... 19
A.17 Node selection window with regular expression. .......... 20
A.18 How to use “>” button in nodes selection dialog box. ... 21
A.19 How to use “<” button in nodes selection dialog box. ... 22
A.20 How to use “>>>” button in nodes selection dialog box. .. 22
A.21 How to use “<<<” button in nodes selection dialog box. .. 22
1 Introduction

Background Snoopy [1] is a tool for designing and simulating Petri Nets [2]. Petri nets are mathematical modelling language which are used to describe distributed System. Petri Nets also known as Place/Transitions nets are bipartite graphs in which node represents transition and places. For more understanding of basic Petri nets please refer [3]. This project is basically working on the simulation dialog box. The simulation dialog box of Snoopy is the dialog box where all the Parameters and controls are provided to the users to control its simulation. The user can change as many parameters as he/she wants to test the net. There are different types of simulators available for the user. The user can use any simulator as per users choice.

Motivation There are numerous tools present in the market which can be used for performing simulation on biochemical reaction networks. This design of the simulation dialog box is motivated by a companion tool Charlie [4]. The design of Charlie is such that it can be accommodated in small screens, which is taken as the base of our design. Charlie uses collapsible panes to categorize similar properties under one heading. This is also adapted by us in our design so as to make the dialog box small and easy to use.

Outline This thesis deals with the Graphical User Interface of the Simulation dialog box and certain functionalities of the simulation dialog box. The GUI of the simulation dialog box is redesigned and implemented. The simulation dialog box has become small and more convenient to use for the users. The plot area can now be seen in full screen of the user desktop which helps users to see each and every details of the simulation results. Additional functionalities like selection of places through regular expressions, loading of curves using regular expressions, saving of curves using regular expression to save disk space are also incorporated in this work.
2 Requirement Analysis

2.1 Snoopy’s previous simulation dialog box

Snoopy’s current simulation dialog box is very static. It comprises of the plot area, different simulation configuration, model configuration and all other parameters in the same window and these are permanently shown to the user. This consumes lots of the screen space making it very difficult for the user to observe the graphs that are been plotted after the simulation.

For small screen users the buttons, text fields and all other GUI components takes there default size thus, compromising the size of the plot area as it don’t have any default size so it shrinks and have to allow other components to take the space. This leads to smaller plot area and a very con-justed graph very every curve overlap each other thus becomes very difficult for the user.

In order to perform simulation please refer [3] and [7]. The user manual was also referred. For user manual please refer [8]. For the graph based data structure used in Snoopy and modeling and simulation in Snoopy refer [9]. The tool also supports certain imports and exports for which you can refer [6].

Lets discuss some of the major components in this old simulation dialog box. The simulation dialog box is been divided into three columns.

- The first column contains all the properties that corresponds to simulator and model. Some of the stuff like changing different simulator for testing can also be done in this column. The user can also change certain properties that are specific to the simulator from this column only. There are things like the simulation timer or the simulation start button. The export and import properties are also available in this column. Thus as a final word this column contains all the properties and they are always seen to the user and it also consumes a lot of space of the screen.

- The second column just comprises of the plot area where the user can see the plotted graph. This column entirely helps user to see the graph. Here there is also a legend(the box containing the names of the curves with the color of the line with which it is plotted) associated with the graph.

- The third and the last column contains a list box where all the nodes are shown. This list box have check/uncheck features for the user so as to see or disappear the curve from the plot area. There is also a view
properties box available in this column and also a viewer type option which allows the user to see the graph in three different ways.

Figure 1: Snapshot of old simulation dialog box of Snoopy

2.2 Design of Charlie’s dialog box

Charlie is a software for analyzing place/transitions nets. It is a analyzing tool and applies standard analysis techniques of Petri net theory to determine structural and behavioral properties of Petri nets.

A distinguished feature of Charlie is its rule system which applies standard theorems of Petri net theory to computed properties in order to possibly derive further properties, which may save a great amount of computational time. All applied rules are reported, so the user can keep track of the analysis process. Charlie is a Java application, its primary focus is teaching of Petri net theory. The tool is in use for validation for natural systems like biochemical networks.
The basic of discussing about Charlie here is that the design that is used in making Charlie is very good for our purpose here. The design uses certain technique which helps us to keep the design of Snoopy small and more readable. The design is more readable as the properties that are similar to each other or resemble themselves in any manner are kept in a same heading, so that the user can easily use them. This also helps in removing the chaos of showing everything. Now, the user have the liberty to only see those properties which the user intends to change and other properties remain hidden from the user.

Figure 2: Snapshot of Charlie.
2.2.1 Design adaption in Snoopy

Charlie have a very small and compact design which is very useful in many aspects for the software that are used for designing and modeling. Charlie uses the concept of grouping similar properties under one heading. This concept of Charlie has been adapted by us in redesigning the Snoopy’s Graphical user interface. Collapsible panes are used to show or hide the group properties.

Grouping of similar properties under one heading also helps the user in saving the screen space as he/she can expand only those groups which are currently in use to the user and let the remaining unexpanded. This type of design also helps the user to concentrate only on specific sets of properties for their simulation as the other properties are hidden by the user.
2.3 Understanding wxWidgets

wxWidgets [5] is a C++ library for developing Graphical User Interface for applications. It is easy to use API for Building GUI on multiple platforms. It is a cross platform library which give the application a truly native look and feel because it uses the native API rather than emulating the GUI. The main advantage of using this library is that it is open source, free and also extensible. The version of wxWidgets used in this project is wxWidgets 3.0.2. The platforms that wxWidgets supports are:

1. wxGTK: The recommended port for Linux and other Unix variants, using GTK+ version 2.6 or higher.
3. wxMac: For delivering Carbon applications on Mac OS X 10.2 through 10.6.
4. wxOSX/Carbon: For delivering 32-bit Carbon-based applications on Mac OS X 10.5 and above.
5. wxOSX/Cocoa: For delivering 32-bit and 64-bit Cocoa-based applications on Mac OS X 10.5 and above.
6. wxX11: A port for Linux and Unix variants targeting X11 displays using a generic widget set.

The benefits of using this platform are:

1. Code once written can be run on multiple platforms like windows, Linux, Mac
2. The applications have native feel and look.
3. Productivity is increased as it provides wide variety of classes that can be used directly.
4. Access to source codes for enhancement and troubleshooting.

The features and functionalities that wxWidgets provide are:

2. Event handling system.
3. File and directory manipulation.
4. Image loading, saving, Drawing and manipulation.
5. Date Time library.
6. Error Handling.
7. Clipboard and Drag-and-drop.
3 Redesigned Snoopy’s Dialog Box

The simulation window is one of the most important parts of Snoopy simulator. It shows users the variety of the features that the users can change so as to make a perfect simulation. More details can be found in the user manual provided in appendix. Please see Figure 4 for more details.

![Figure 4: Snapshot of redesigned Snoopy’s dialog box.](image)

3.1 Use of Collapsible panes

Collapsible pane is a panel that can store content in a compact space. Users hide or reveal the content stored in the Collapsible Panel by clicking the tab of the widget. There are three collapse panel used in the simulation window in Snoopy namely

1. Simulation Configuration
2. Model Configuration

3. Import and Export Details

3.1.1 Simulation Configuration

Simulation configuration deals with the configuration of the simulator. It gives user the flexibility of switching between the different type of simulator available in Snoopy. It also allows the user to change certain properties of the corresponding simulator that the user might have chose.

This collapse panel also have the certain properties that can help user to change the final output that is the graph. There is a property which allows user to change the interval of the x-axis in the plot. The user can decide within which interval the graph should be plotted so to make it convenient for the user to make the observations. Here the user can also change the interval splitting of the x-axis that is he can either make 5 splits or 10 splits or any number of splits in the x-axis as per the user requirements.

3.1.2 Model Configuration

Model configuration contains all the properties that are related to model designed by the user. This collapse panel allow user to play around its model and check different sets of function on the model. The user can also change the markings of the places in the model under this configuration. There are so many inbuilt function that user can use in its model directly or can define its own. It also have certain features like weight sets, delay sets, schedule sets etc depending on the type of net class user is working on.

3.1.3 Import and Export Details

This collapse panel allows the user to export and import data in different formats. There are three kinds of export available in the panel. They are as follows

1. Direct Export : This export directly export the data in csv format. The data exported is same as the data shown in tabular view.

2. Single Trace Export : This export exports the data according to the number of runs that is the number of runs used in the simulator configuration. Each run table is exported separately in one csv file.

3. Direct Trace Export : This exports just gives the raw data and it can be very large if the net is quite large.
Other stuff

- Properties button: This button is used to export the data. Once the user selects the type of export then user have to click on this button to check file path and separator of the data like coma, semicolon etc. and then press ok to export.

- Load Data: There is also an option to load data from csv files.

### 3.2 Displaying all result views

The simulation dialog box have many features. One of the feature is to display all the views that the user has created. This views are sorted according to the order they are created by the user. The names of all the views are listed in this part of the simulation dialog box. Any view that are added or deleted by the user is reflected in this list box. This list box also allows user to open the result views.

### 3.3 How to add new views

Views are very important for observations by the user as different views can be used for different observations. It certainly help the user in distinguishing different results. Adding a new view is very simple and user can follow these steps to add a new view:

1. There is a button “Add New View” in the simulation dialog. See Figure 5 Click this button.

2. A new text box will appear as soon as the button is pressed. In this text box the user can give a desired name for the new view. For more understanding see Figure 6.

3. As soon as the user press OK button after giving a name to the view, the new view will be added to the list of all views. As the view is created, the view be selected automatically now, if the user press “Show Selected View” button the user will be able to see the new view which is just created. For more understanding see Figure 7.
Figure 5: When “Add New View” is pressed.

Figure 6: Giving a name to the new view.
3.4 How to remove views

Views are essential part of observation in Snoopy. As the addition of new views are important similarly removal of old or unused views are also important. This makes the user to remove the old results and produce some new results. Removal of views are very simple. There is also an option to delete several views in one click. To remove views user must follow the following instructions:

1. There is display box in the simulation dialog of Snoopy where all the views are listed. User has to select as many views as users want to remove by pressing the ctrl button of the keyboard and simultaneously clicking on the items in the list. This will select all the views that the user intend to remove. For more understanding see Figure 8.

2. After selecting the views the user want to delete there is a button called "Remove selected views" on the simulation dialog box. If the user press this button then all the views that are selected by the user will be removed and only the unselected view will remain. For more understanding see Figure 9.
Figure 8: When Views are selected for deletion.

Figure 9: After the Remove views button is pressed.
3.5 Simulation Timer

Simulation timer is one of the main aspects of the simulation. It shows the user how much time does it take to complete the simulation of the net made by the user. It also shows the expected time of simulation while the simulation is in progress. This time is usually very helpful for the users to see how much time does their net take to simulate and according to it they can adopt some optimization to decrease the time of simulation. So, to make it more convenient for the user the time of simulation is shown in the desired units like we can see in the examples below.

Figure 10: Time shown in seconds only

Figure 11: Time shown in minutes and seconds
4 Result views

4.1 Design

The result view are kept very simple so as to allow to use it easily. The main focus of the design was to make the plot area more readable and clear for all the types of screen sizes. When the window is maximized, it is maximize in such a way that only the size of the portion containing the plot area is increased and other components remain at their default size. see Figure 13

There are various other components like buttons, text fields etc. but they are arranged in such a way that it maximizes the plot area. There is a node list placed in each and every view. This list helps the user to check or uncheck the nodes and the corresponding curve will appear or disappear.

There are also many features in this dialog box such as

1. Change viewer type: The user can switch between three viewer type namely xy Plot, histogram view and tabular view provided in Snoopy. The Changing of the viewer type can be done from this window only. There is a combo box present with the name viewer type in which the user can switch between the views.

2. Export: The user have the support to export the simulation result in two different ways. The first option is the csv export, in this the user can export the results in a comma separated files that can be easily open in excel sheet software. The other option is a image export in which a snapshot of the plot area is saved to system. There are different types of image supported in this export like bmp, png, jpeg etc.

3. Show/Hide node list: There is also an option for the user to show to hide the node list. If the user wants more space for the plot are then the user can hide the node list. as it would generate some more extra space for the graphs. See Figure 13 and 14

4. Edit Node list: This is one of the most important part of the result views. This dialog box contains all the nodes which are present in the net. As all the places cannot be loaded at once because it can hang the user system if the number of nodes are very high. So, some of the nodes are loaded initially in the result view and all the remaining nodes are available here. The user can see them in the view by selecting them from this dialog box.
There are also various options available for the user in this dialog box. The user has the option to see the type of node, whether place or transition separately. There is also an option of using regular expression for selecting the nodes. This prevents the user from doing the irritating work of selecting each node manually.

Figure 12: Snapshot of Edit Dialog Box.

5. Connect or Disconnect: This button is used to make the result view connection or disconnection with the simulator. If the user disconnects the result view, then this view can’t receive any update from the simulator, and it won’t reflect any changes made after disconnection. Once the view is connected again, it receives all the updates.

6. Close: This button is used to close the view.

20
Figure 13: Snapshot of a result view with node list.

Figure 14: Snapshot of a result view without node list.
4.2 Various properties of curves

There are various curve properties that the user intend to change. This properties actually helps the user for better understanding and readability. There are several properties that are common to all the curves and several properties that are related to each and every curve separately.

Lets talk about some of the common properties that Snoopy posses. They are like the type of line user wants that is whether lines may be simple or may contain symbol in it or only symbols can also be used. The user can also fix the style and line width for all the curves here. We actually have a very little discussion over here. For a very detailed description please to Section A.14 in the manual. Similarly the individual curve properties are also very important. The individual curve properties posses curve style, curve width and curve color which can also be seen in detailed in Section A.2.2 in the manual.

4.3 Nodes selection dialog box

The design of nodes selection dialog box is easy to understand and also easy to use. There are various important component. Some of these components are dynamic that is their behavior is dependent on certain condition while others are static that is there behavior is always same. This list is open from the result viewer by pressing the button "Edit node list".

Now, lets see all the components that reside in this nodes selection dialog box.

1. Name : This is a text field which is related to the view name. So if we want to change the name of the view after we have created it then we can do it from here.

2. Observers : Basically this field tells us what are the different kinds of nodes that are present in out net. This is a radio button box so that we can select any one of them can see all the nodes corresponding to it in the list box which we are about to discuss.

3. Output type : This actually depends on the type of observer that you have selected, based on it these are more sub category.

4. Regular expression field : This field is used to enter a regular expression as the name suggest. These regular expression is used to determine which nodes you want to see in your result viewer that is the nodes whose names matches with the regular expression will be seen in the graph.
5. The overall nodes: This is a list box where all the nodes that are not currently shown in the result view are displayed.

6. The selected nodes: This is a list box where all the nodes that are currently shown in the result view are displayed.

7. List switching buttons: These buttons are used to switch nodes between the list. As it is a bit confusing please check the section A.2.4.2 for details.
5 Regular Expressions in Snoopy

5.1 Use in nodes selection dialog box

The use of regular expression in the nodes selection dialog box is that the user can conveniently moves nodes in the two list that is "The overall nodes" list and "The selected nodes" list. For a visual of the lists please see Figure 12.

The two list are so designed that the nodes that are currently viewed in result view will be in the "The selected nodes" list(or right list), while the nodes that are not shown in result view are in "The overall nodes" list(or left list). Now the regular expression used in the dialog box is to determine which nodes user wants to see in the result view currently. This regular expression matches all the nodes that are present int the net that is in the left list or in the right list an then puts them in the right list if they match or put them in left list if they don’t match. In short all the matching nodes will be put in the right list and other will be in left list.

The regular expression is executed whenever there is a change in the regular expression field. For more details please refer the manual Section A.2.4.1.

5.2 Advantages and disadvantages

The advantages are as follows :

1. It saves a lot of space as the curves which are displayed in the result views are not stored one by one in-fact a regular expression which is currently in used is stored.

2. It helps user to select some of the nodes from a very long list just by writing a short command.

3. Loading of curves are also done using regular expression. If any result view is using regular expression for the list of nodes then that regular expression is used for loading the curves during initialization.

4. If the names of the nodes are short then regular expression are very useful.

5. Regular expression are very easy to understand and are very helpful.

The disadvantages are as follows :
1. Curve properties are not saved so every time the user curve properties are lost and default curve properties are loaded.

2. If the names of the nodes are very large then it can be a disadvantage with regular expressions.

3. User will not have the liberty to use manual selection of nodes if there is a regular expression in use. That is user can use only one option at a time either manual selection or selecting by regular expression.

4. User must know the basic of regular expression because there is no way to suggest how to write regular expression in this dialog box.

5.3 Saving and loading of Curves

This is one of the most important aspects of using the regular expressions in Snoopy. The saving and loading of curve is determined whether the user is using a regular expression or not, if the user is using a regular expression then in this case the saving and loading of curve in that view will be determined by the regular expression used by the user.

Let talk about how the curves are previously stored in Snoopy before regular expression. Before each and every curve is stored by name and all its property were written in the XML associated with it. Thus making the file very large if the number of nodes in the net are quite large. This also use quite a amount of disk space and also it is slow to load the file every time user open its net in Snoopy.

Now things have changed and regular expression are incorporated in Snoopy. The curve saving is done by just storing the regular expression. This helps in saving disk space and also the process of loading curve is quite fast. This is because reading a file in a program is one the most costliest operation. But now only regular expression is read from the file and the curve are loaded by comparing it with the names of all the nodes and all those nodes which matches this regular expression are shown to user in the result view and those are not shown can also be seen by changing the regular expression in the node selection dialog box. For more details of regular expression in nodes selection dialog box please check the Section 5.1.
6 Summary

6.1 Achievements

- Before the simulation window which was static and big. Now, we have a simulation window which small and compact and compatible for any screen size users.

- I have never programmed GUI in C++, but using the library wxWidgets to program GUI in C++ was fun and a learning experience.

- Using regular expressions in Snoopy helps us in saving user’s disk space.

- The report is written in LaTeX which was completely new to me. So yes, I learned a few basics about LaTeX, too.

6.2 Open Problems

There are a few potential areas where this work can be extended. Some of them are:

- There is a bug while using the simulation dialog in mac. If you open any result view and after that if you expand any collapsible pane then the result view comes over the simulation dialog box and you have to restart the simulation dialog box to get rid off this.

- There is a problem in mac version of snoopy that when you are using the simulation dialog box the window where the net is drawn must be disabled and cannot be edited until the simulation dialog box is closed but in mac it is still allowed to change the net when the simulation dialog box is open and this should not be the case.

- We can add more symbols to represent curves in the graph. We can use a different library for this, I guess.

- We can use the same design of the simulation dialog box and can make the steering dialog box similar to it. So, that there is only one design for the user to remember, whether the user perform simulation on remote server or on local machine.
References


A User Manual - everything you need to know about Snoopy’s simulation dialog box

A.1 How to use simulation dialog box

A.1.1 Getting started simulation dialog box

The simulation dialog box in Snoopy [1] is the place from where the simulation is controlled for petri nets [2]. The basic components of the simulation dialog box are numbered in the Figure A.1 as you can see. We will know them in detail later but for now lets see the basic meanings of the components.

1. The first component of simulation window is simulation configuration. In this we will find all the details that are related to the simulator itself, like different types of simulators, their different properties etc.

2. The second component of simulation window is model configuration. In this we will find the all the details that are related to model like marking overview, delay set, formula check etc.

3. The third component is different types of export and import available in the simulation dialog.

4. The forth component of the simulation window is the portion where we can see all the different result views. Here we can play around them like add new views, remove old ones etc.

5. The fifth and the last component of the simulation dialog is the place where we have the start button for our simulator, where we watch them running with the progress bar and many more stuff.
A.1.2 What is inside Simulation Configuration?

Simulation configuration deals with the configuration of the simulator. It gives user the flexibility of switching between the different type of simulator available in Snoopy. It also allows us to change certain properties of the corresponding simulator that we have choose.

This collapse panel also have the certain properties that can help us to change the final output that is the graph. There is a property which allows us to change the interval of the x-axis in the plot. we can decide within which interval the graph should be plotted so to make it convenient for us to make the observations. Here we can also change the interval splitting of the x-axis that is he can either make 5 splits or 10 splits or any number of splits in the x-axis as per our requirements. Please refer Figure A.2 for more details.

Figure A.1: Snapshot of components in New simulation dialog box.
A.1.3 What is inside Model Configuration?

Model configuration contains all the properties that are related to model designed by us. This panel allows us to play around its model and check different sets of function on the model. We can also change the markings of the places in the model under this configuration. There are so many inbuilt function that we can use in its model directly or can define its own. It also have certain features like weight sets, delay sets, schedule sets etc depending on the type of net class we are working on. Please refer Figure A.3 for more details.

A.1.4 What is inside Export and Import details?

This collapse panel allows us to export and import data in different formats. There are three kinds of export available in the panel. Please refer Figure A.4 for more details. The types are as follows
1. **Direct Export**: This export directly exports the data in csv format. The data exported is same as the data shown in tabular view.

2. **Single Trace Export**: This export exports the data according to the number of runs that is the number of runs used in the simulator configuration. Each run table is exported separately in one csv file.

3. **Direct Trace Export**: This exports just gives the raw data and it can be very large if the net is quite large.

Other stuffs

- **Properties button**: This button is used to export the data. Once the user selects the type of export then user have to click on this button to check file path and separator of the data like comma, semicolon etc. and then press ok to export.

- **Load Data**: There is also an option to load data from csv files.

![Figure A.4: Snapshot of expanded export import details tab.](image)

### A.1.5 Playing around with result views

Result views are essential parts of simulation. In this field we can play around with result views. The list box that you see in the Figure A.1 is actually containing the names of all the views that has been created so far. This list box is re-sizable and adjust itself according to the size of the simulation dialog. It also has an option to select multiple views at a time. For window and Linux users pressing ‘ctrl’ button on the keyboard along with clicking the name on the view will select multiple views. For mac users ‘cmd’ button is just replaced with ‘cmd’ button.

More things that can be done are as follows:
• **Addition of new views**: This is very simple to do, just click on "Add a new view" button on the simulation dialog box then type in the name of the new view and press "Ok" button. New view will be created.

• **Removal of views**: Removal of views is even simpler. There is an option of removing multiple views at a time. Start with selecting all the views that you want to delete, after selection just press the "Remove views" button on the simulation dialog box.

• **Show selected views**: Here it is an option to open multiple views simultaneously. Similar to removal of views just start by selecting the views that you want to open and then press "Show selected views".

• **Show all views**: This is an option to open all the results views. These views will open in separate windows.

**A.1.6 Other simulation stuff**

There are some other exciting stuff in the simulation window. They are as follows:

• **Start simulation**: This is the key to simulation. To start simulation just press "Start simulation" button on simulation window.

• **Progress bar**: This bar gives the idea of the simulation been completed so far. If the bar is half full then it means that 50% of the simulation is done.

• **Simulation timer**: This is the physical time taken by the simulator to simulate. This timer is very intuitive. It displays time in only those units which are necessary. This timer shows the current elapsed time by the simulation.
A.2 How to use result views

A.2.1 General idea of result views

Result views are the essential part of simulation in Snoopy. It is very important to know about the result views. For more about simulating in snoopy please refer [?] and [4]. There is also a user manual for simulation [5]. Lets start with the basic of result views and know about the components that reside in result views. They are as follows: Please see the Figure A.5 for a look of result view.

1. **Plot area**: The most important parts of the result view as it shows the result of simulations. This is the first component of the result view where we can see all the graphs that are the result of our simulation.

2. **Nodes list box**: The second component of the result view is also important as it helps in making our results more appropriate. This list holds all the nodes names which you can currently see in your plot area. There is also a option in the bottom of the list to select all of them or deselect all of nodes in one click.

3. **Viewer type**: The third component of the result view. There are three option available in the view type which are actually very beneficial to in different situation. Histogram view, xy plot and tabular view. Also, there is a edit button to change some of the property of view type like the size of the chart, title of the chart etc.

4. **Export**: The forth component of the result view is its export ability. There are two types of export available in the result view. Image export and csv export. For image export you can change the size of the chart in viewer type properties and image export doesn't work for tabular view.

5. **Nodes Selection window**: The fifth component of the result view is the nodes selection window which is open by pressing the “Edit node list” button in the result view. This window give us the option to see all the nodes that are available in our net and allow us to select them for the result on the graphs.

6. **Other options**: The sixth component is a long list of button available in the result view and are there for some minor task ant they are discussed in the Other option section in this manual.
A.2.2 Editing curve properties

This is one of the most interesting part of the result view. We have the liberty to change the individual curve properties. These helps us in identifying different curves easily also we can associate a different color to each and every node. Different types of representation also allows to make our curve interactive, Lets us now discuss the different properties that we can changes for each curve.

- **Curve color**: There is a whole color wheel to select a color or you can also provide the RGB values so as to select the color.

- **Curve width**: We can set the line width of the curves so plotted. there is a option of choosing a width from a range of 1-20. The higher the number you choose the broader the line gets.

- **Curve style**: This allow us to to set a style a of line for the curves that are plotted. There is a option of choosing from five different line styles. They are Solid, Dot, Long dash, Short dash and Dot dash.
3 Steps to change the individual curve property:

1. Go to the result view and on the nodes list so displayed at the right of the plot area, double click on the node name for whom you want to change the curve property. For easy understanding see Figure A.6

2. A dialog box will appear with the current property of the curve. Now change the property as you desire and press the "Ok" button. For details see Figure A.7 and A.8

3. As soon as the "Ok" button is pressed you will see that the curve have changed to those property which you just set. For details see Figure A.9

Figure A.6: Showing Different curves in the plot area.
Figure A.7: Double click on the node name in nodes list.

Figure A.8: Changing curve property of a curve.
A.2.3 Editing View properties

The view property is open by pressing the ”Edit” button under the heading viewer type in the result view. The window so open after pressing this button have five tabs in it. We will be discussing them one by one as we progress. The tabular view has only one tab in it that is the general tab. So, lets start with the first tab that is General tab.

A.2.3.1 General Tab

The general tab provides very basic information and it is common to all the viewer type. The general tab has the following options available. For more details please see Figure A.10

- **Title**: This field is used for the title of the chart. We can change the title of the chart from here.

- **Window Width**: This field is used to set the Window width as per our requirements. We can change the window width from here.

- **Window Height**: This field is used to set the window height as per our requirements. We can change the window height from here.
• **Export Image Width**: This field is used to set the width of the image we want to export. We can change the export image width from here.

• **Export Image Height**: This field is used to set the height of the image we want to export. We can change the export image height from here.

![Result Viewer Properties](image)

Figure A.10: Snapshot of General tab in viewer properties window.

**A.2.3.2 Legend Tab**

Legend is the list box along with the graph that contains the names of the nodes along with the symbol with which the node is being represented in the graph. There are several properties that we can change according to our convenience. The different properties with which we can play around are written below. For more details please see the Figure A.11

- **Show/Hide Legend**: This check box helps us to whether we want to see the legend along the side of the graph or not.

- **Horizontal position**: This option gives us the liberty to move our legend in the horizontal position of our plot area. There are three horizontal positions available as of now. They are left, right, and center.
• **Vertical position**: This option gives us the liberty to move our legend in the vertical position of our plot area. There are three vertical positions available as of now. They are top, bottom, and center. Along with the horizontal position, we can have our legend in nine different locations in our plot area.

• **Sort curves**: There is also an option to sort your curves by names and also if you want you can let it unsorted. There is a option to sort them in ascending and descending order.

Figure A.11: Snapshot of Legend tab in viewer properties window.

### A.2.3.3 Axes Tab

The Axes tab as name suggest it is related to the axis of the graph. There are several options available for the axis as we can see them in the Figure A.12 and also they are detailed below:

• **Show X axis**: This check box will help us to see or not see the X axis in the plot area or graph.
• **X axis title**: This allows us to change the title of the X axis as per our requirements. Generally the X axis is time.

• **Show Y axis**: This check box will help us to see or not see the Y axis in the plot area or graph.

• **Y axis title**: This allows us to change the title of the Y axis as per our requirements. Generally the Y axis corresponds to the final markings available in the nodes after simulation.

Figure A.12: Snapshot of Axes tab in viewer properties window.

**A.2.3.4 Fixed Tab**

This tab also works on the axis of the plot area but it is somewhat different. This tab actually helps us in focusing a part of the graph or one can say zooming in the graph. we can set the values of the axis as per our requirements and the plot area will be fixed to these axes values and we can see the result between these values only which is actually a smart feature to have a close observation of your result. There are several properties that we can
change. We can see them in Figure A.13 and also details of them are as follows:

- **Fixed X axis Adjustment**: This is a check box to ask us whether we need a fixed sets of values for the plot area in the X axis.

- **Min. X value**: This helps us in giving the start value or the min value for the x axis, then these values will become the starting of the X axis.

- **Max. X value**: This helps us in giving the end value or the max value for the x axis, then these values will become the ending of the X axis.

- **Fixed Y axis Adjustment**: This is a check box to ask us whether we need a fixed sets of values for the plot area in the Y axis.

- **Min. Y value**: This helps us in giving the start value or the min value for the Y axis, then these values will become the starting of the Y axis.

- **Max. Y value**: This helps us in giving the end value or the max value for the x axis, then these values will become the ending of the Y axis.
### Curves Tab

This tab gives the general property that are followed by the all the curves in the graph. We have certain property can used as a general way to represent all the curves. we can also set the individual property for this please refer Section A.2.2. For more understanding please see the Figure A.14 and also the details of different properties are as follows :

- **Show lines**: this check box allows us whether we want to represent our curves using a line or not.

- **Show symbols**: this check box allows us whether we want to represent our curves using a symbols or not. We can use this in combination of the lines so that we can see the curves that are shown using a symbolic line that is curve re represented by a line which also have symbol over it.

- **Line width**: We can set the line width of the curves so plotted. there is a option of choosing a width from a range of 1-20. The higher the number you choose the broader the line gets.
- **Line style**: This allows us to set a style of line for the curves that are plotted. There is an option of choosing from five different line styles. They are Solid, Dot, Long dash, Short dash and Dot dash.

![Curves tab in viewer properties window](image)

Figure A.14: Snapshot of Curves tab in viewer properties window.

### A.2.4 Working with Nodes selection dialog box

We can get the first impression of this window by looking at Figure A.15. This dialog is actually meant so that the net can load faster that is if there is a net with large number of nodes then this dialog comes into play. Well let discuss the basic elements in this dialog and then we will move further up to some of the very important features of this dialog. So, let's start

1. **Name**: This is a text field which is related to the view name. So if we want to change the name of the view after we have created it then we can do it from here.

2. **Observers**: Basically this field tells us what are the different kinds of nodes that are present in our net. This is a radio button box so that
we can select any one of them can see all the nodes corresponding to it in the list box which we are about to discuss.

3. **Output type**: This actually depends on the type of observer that you have selected, based on it these are more sub category.

4. **Regular expression field**: This field is used to enter a regular expression as the name suggest. These regular expression is used to determine which nodes you want to see in your result viewer that is the nodes whose names matches with the regular expression will be seen in the graph.

5. **The overall nodes**: This is a list box where all the nodes that are not currently shown in the result view are displayed.

6. **The selected nodes**: This is a list box where all the nodes that are currently shown in the result view are displayed.

7. **List switching buttons**: These buttons are used to switch nodes between the list. As it is a bit confusing please check the Section A.2.4.2 for details.
A.2.4.1 Using Regular Expression

This is the most interesting part of this dialog box. Regular expressions are actually very helpful in this kind of situation in which we have a large list and have to select some of them. Here in this case we are also using them for the similar purpose. Here we have a got a list a nodes namely ”The overall nodes list” and we have to move some of the nodes to another list namely ”The selected nodes list”, so that we can see all these nodes in our result view and can analyze them.

Before we start please have a careful look to the Figure A.16 and A.17. Now, lets see how this regular expression work in our case.

1. As you can see the Figure A.16 there is no regular expression in use. The left list contains ”Retail”, ”Resu” and ”_P.4.” nodes while the right list contains ”Result” and ”_P.3.” nodes.
2. Now let’s add a regular expression ”Re” (as an example see Figure A.17). As we finish writing our regular expression we will see some shuffling in both the lists. The nodes named ”Retail” and ”Resu” have moved from left to right list. The nodes named ”_P_3_” has moved from right to left list and the nodes ”_P_4_” and ”Result” remains in the same list. See Figure A.17.

3. The item that corresponds to the regular expression have moved to the right list, while the nodes that don’t are in the left list.

4. We can see that all the nodes whether in the right list or the left list are all shuffled according to the regular expression.

As a final conclusion if there is a regular expression written in the field that the right list will only contain those nodes which corresponds to the regular expression and the left list will contains those nodes which don’t.

![Figure A.16: Node selection window without regular expression.](image-url)
A.2.4.2 The four buttons for switching the nodes between the list

Here is the confusing part of this dialog box. If you are using a regular expression then these buttons are not relevant to you but if you are not using a regular expression then only these buttons can save your lives. These buttons are used to moved the nodes from one list to another.

Let’s see how these buttons works:

- **Button with ”>” sign on it**: This button is used to transfer the selected nodes from the left list to the right list. See Figure A.18.

- **Button with ”<” sign on it**: This button is used to transfer the selected nodes from the right list to the left list. See Figure A.19.

- **Button with ”>>>” sign on it**: This button is used to transfer the whole left list to the right list see Figure A.20.
• **Button with ”<<” sign on it**: This button is used to transfer the whole right list to the left list see Figure A.21.

Figure A.18: How to use ”>” button in nodes selection dialog box.

Figure A.19: How to use ”<” button in nodes selection dialog box.
A.2.5 Exporting your data

This is also a feature which is used more commonly than any other feature. You can see this feature visually in the bottom left corner of the Figure A.5. This feature is the export feature. For using you should first select which export you want in the combo box and then by pressing the export
button your data will be exported in that export type. There are two exports available: csv export and image export. Image export can be done in various types of extensions like .jpg, .png, .bmp etc.

A.2.6 Other options

There are other features that we must look so as to have complete understanding of result views. These features are very small but extremely useful. Please refer the Figure A.5 and see the component no. 6 for better understanding, while the features are listed in details below:

- **Show/Hide node list**: This is done by pressing the button called "Show/Hide node list". These features are very useful for observation as it gives the plot area more space.

- **Refresh**: This is for manually refreshing the result of your result view from the simulated result. As this thing is automatically taken care of but sometimes it happens that the simulated results are not communicated to all the result view, so it is better to give a option for manual refresh.

- **Disconnect**: This button disconnects the result view from the simulator and blocks any further updates to this result view. We can also reconnect this result view and get the updated results.

- **Change X axis**: This is an option to map X axis with different units, so as to see the results in a different manner.
A.3 Everything you need to know about Regular expressions

The best way to know the regular expression is to know all the rules of the regular expression with example. For more syntax please refer [6].

A.3.1 Character class

A character class matches any one of a set of characters. Character classes include the language elements listed in the following table.

<table>
<thead>
<tr>
<th>Character class</th>
<th>Description</th>
<th>Reg-ex</th>
<th>Match</th>
</tr>
</thead>
<tbody>
<tr>
<td>[character_group]</td>
<td>Matches any single character in character_group. By default, the match is case-sensitive.</td>
<td>[ae]</td>
<td>“a” in “gray”, “a”, “e” in “lane”</td>
</tr>
<tr>
<td>[^character_group]</td>
<td>Negation: Matches any single character that is not in character_group. By default, characters in character_group are case-sensitive.</td>
<td>[^aei]</td>
<td>“r”, “g”, “n” in “reign”</td>
</tr>
<tr>
<td>.</td>
<td>Wildcard: Matches any single character except \n.</td>
<td>a.e</td>
<td>“ave” in “nave”</td>
</tr>
<tr>
<td>\w</td>
<td>Matches any word character.</td>
<td>\w</td>
<td>“I”, “D”, “A”, “1”, “3” in “ID A1.3”</td>
</tr>
<tr>
<td>\W</td>
<td>Matches any non-word character.</td>
<td>\W</td>
<td>“”, “.” in “ID A1.3”</td>
</tr>
<tr>
<td>\s</td>
<td>Matches any white-space character.</td>
<td>\s</td>
<td>“D” in “ID A1.3”</td>
</tr>
<tr>
<td>\S</td>
<td>Matches any non-white-space character.</td>
<td>\S</td>
<td>“” in “int _ctr”</td>
</tr>
<tr>
<td>\d</td>
<td>Matches any decimal digit.</td>
<td>\d</td>
<td>“4” in “4 = IV”</td>
</tr>
<tr>
<td>\D</td>
<td>Matches any character other than a decimal digit.</td>
<td>\D</td>
<td>“”, “=”, “” in “I”, “V” in “4 = IV”</td>
</tr>
</tbody>
</table>
A.3.2 Anchors

Anchors, or atomic zero-width assertions, cause a match to succeed or fail depending on the current position in the string. The meta-characters listed in the following table are anchors.

<table>
<thead>
<tr>
<th>Assertion</th>
<th>Description</th>
<th>Reg-ex</th>
<th>Match</th>
</tr>
</thead>
</table>
| ^         | The match must start at the beginning of the string or line. | \d\{3\} | "901" in "901-333-"
| $         | The match must occur at the end of the string or before \n at the end of the line or string. | \d\{3\}\$ | "333" in "901-333"
| \A        | The match must occur at the start of the string. | \A\d\{3\} | "901" in "901-333-"
| \Z        | The match must occur at the end of the string or before \n at the end of the string. | \d\{3\}\Z | "333" in "-901333"
| \z        | The match must occur at the end of the string. | \d\{3\}\z | "333" in "-901333"
| \G        | The match must occur at the point where the previous match ended. | \G\((\d)\) | "(1)", "(3)", "(5)" in "(1)(3)(5)(7)(9)"

A.3.3 Quantifiers

A quantifier specifies how many instances of the previous element (which can be a character, a group, or a character class) must be present in the input string for a match to occur. Quantifiers are listed in the following table.

<table>
<thead>
<tr>
<th>Quantifiers</th>
<th>Description</th>
<th>Reg-ex</th>
<th>Match</th>
</tr>
</thead>
</table>
| *           | Matches the previous element zero or more times. | \d*\d | ".0", "19.9"
| +           | Matches the previous element one or more times. | be+ | "bee" in "been", "be" in "bent"
| ?           | Matches the previous element zero or one time. | rai?n | "ran", "rain"
| { n } | Matches the previous element exactly n times. | \{d\{3\} | ".043" in "1,043.6", ",876", ",543", and ",210" in "9,876,543,210" |
| { n ,} | Matches the previous element at least n times. | \{d\{2,\} | "166", ",29", ",1930" |
| *? | Matches the previous element zero or more times, but as few times as possible. | \{d*?\{d | ".0", ",9.9" |
| +? | Matches the previous element one or more times, but as few times as possible. | \{be+? | "bee" in "been", ",be" in "bent" |
| ?? | Matches the previous element zero or one time, but as few times as possible. | \{rai??n | ",ran", ",rain" |
| { n , m } | Matches the previous element at least n times, but no more than m times. | \{d\{3,5\} | "166", ",17668", ",19302" in ",193024" |
| { n }? | Matches the preceding element exactly n times. | \{d\{3\}? | ",043" in "1,043.6", ",876", ",543", and ",210" in "9,876,543,210" |
| { n ,}? | Matches the previous element at least n times, but as few times as possible. | \{d\{2,\}? | "166", ",29", ",1930" |
| { n , m }? | Matches the previous element between n and m times, but as few times as possible. | \{d\{3,5\}? | "166", ",17668", ",193", ",024" in ",193024" |
References


