DEVELOPMENT OF A NUCLEAR PROCESS MODELING ENVIRONMENT BASED ON THE MICROSOFT .NET FRAMEWORK

W.A.MALIK[1], M. AFNAN[1], S. WALEED[1], I. R. CHUGHTAI[2], M. ILYAS[1]
[1] *Department of Nuclear Engineering, PIEAS, P. O. Nilore, Islamabad, Pakistan*
[2] *Department of Chemical Engineering, PIEAS, P. O. Nilore, Islamabad, Pakistan*
Vvaleed@gmail.com

Abstract: *Nuclear Reactor safety analysis is a systematic study to demonstrate the limits and reliability of reactor in normal as well as accidental conditions. Reactor Excursion and Leak Analysis Program (RELAP5) is a tool for analyzing small-break LOCAs and system transients in PWRs or BWRs. It has the capability to model thermal-hydraulic phenomena in 1-D volumes. It is used for the thermal- hydraulic transient analysis in water-cooled nuclear reactors by solving one dimensional, two-phase thermal-hydraulics equations in an arbitrarily connected system of volumes. However, the preparation of the input file and subsequent analysis of results in this code is a tiresome task. The aim of this work was to develop a Graphical User Interface (GUI) for preparation of the input file for RELAP5. The GUI has been developed in VB.NET using the GDI+ Libraries. This paper deals with the development of the GUI Environment named as RELAP5 Input File Generator (RIFGen). The base classes for the development of RELAP5 Components have been designed and implemented. The major programming logic and layout of all forms is described in detail in the paper. This GUI generates complete set of cards for Hydrodynamic Components, Core Components, Time Step Control, Minor Edit Requests, Trip Input Data, Heat Structure Input, General Table Data, Plot Request data, Control System Data, Additional Plot variables, General Code Input, Couple Control Cards and other Miscellaneous Cards. These components have been tested and validated individually. RIFGen has also been validated as a whole by generating input files for several standard problems.*
Keywords: *Nuclear Power Plants, DWSIM, GUI, input file, RELAP 5, hydrodynamics, Neutronics, TMI Analysis.*

1. Introduction

Reactor safety analysis is an analytical study to demonstrate the limits and integrity of reactor in normal as well as accidental conditions. There are many codes available for reactor safety analysis but RELAP5 is the most widely used systems analysis code. There are numerous references available in the public literature describing the application of the RELAP5 to a variety of problems. It is developed by the U.S. Nuclear Regulatory Commission (NRC) for use in rulemaking, licensing audit calculations, evaluation of operator guidelines, and as a basis for the nuclear plant analyzer. Although RELAP5 was originally developed to support the analysis of postulated accidents in commercial power plants in the United States, different versions of the code have been widely distributed around the world and now are used to support a wide range of activities.

In recent years, RELAP5 has been applied most extensively to support the certification of advanced reactor designs as well as help resolve outstanding severe accidents issues. RELAP5 was important tools used by the USNRC to certify the expected performance of the AP600 passive reactor design while SCDAP/RELAP also played an important role in the resolution of concerns about high pressure failure of US reactor designs during postulated severe accident conditions. RELAP5 made important contributions to the reassessment of safety of Russian-designed reactors [1].

RELAP5 is also being widely used by regulatory and research organizations around the world to support international standard problem exercises and experimental programs. The impact of user experience, the ability of the code to predict thermal-hydraulic and severe accident phenomena, and its applicability to prototypic plant transient data have been extremely well characterized. Although plant data for accidental conditions is limited, the code has also been widely used to assess the performance of plants under design basis and severe accident conditions. In particular, RELAP5 was used extensively by the US Department of Energy and many international organizations to support the assessment of the TMI-2 accident.

2. Background

RELAP5 input file is a text file of extension ".i". The system flow, components properties and all the initial conditions are described by specific card numbers. The card numbers are actually six to eight digit numbers. For a single component these cards may range from tens to hundreds. A single problem comprising of three components may have hundreds of cards therefore a user needs to remember or browse throughout the manual to write even a single card. This is a tedious job and time consuming job. Moreover, if the solution is not converging, a lot of time and effort is required to modify this input file.

The complexity of the input file can be seen by just considering these few cards in input file. If a single card is misplaces then RELAP5 does not read the input file, and user has to trace and correct the error. This tedious job can be well reduced by a helpful Graphical User Interface (GUI) for input file generation. In the GUI the nodalization diagram is first drawn, and then the parameters of components and the initial conditions are input. This file is saved and can be modified easily. This type of GUI, on one hand, makes RELAP5 user friendly and on the other hand, saves times of input file generation and modification. In the past efforts have been made to improve user interaction with RELAP5, some of them are listed below.

In 1999, Dr. George Mesina developed GUI for RELAP-3d in Idaho National Engineering and Environmental Laboratory. He used some aspects of the older GUIs of some other codes i.e. Athena Aid, TROPIC and SNAP to establish a new friendly GUI for 3d version of RELAP5. JAVA "an objected oriented language" was used as programming language. There are good visualization aspects of this GUI but as far as complexity of the input file is concerned, the user has to browse through the RELAP5 manual to use this software [2].

In 2007, K.D. Kim from Korea and Rizwan-uddin from USA made a web based simulator using RELAP5 and LabVIEW. They used RELAP5 as the solver and LabVIEW for the graphical user interface and web casting. Their software reads the input file and shows the results graphically which can be operated remotely from another site connected to the server via the World Wide Web. But it does not have the facility to generate input file through graphical user interface [3].

In 2008, Mr. Abid Afsar Khan and Mr. Muhammad Muneeb Anwar from PIEAS started the development of the RELAP5 GUI for mod 3.2. They used C Sharp (C#) as programming language. They developed initialization form, design surface, connectors, and many thermal-hydraulic components modeling, but their project lacked modularity which made further improvements difficult [4].

All the GUIs for RELAP5 made so far are not available to use commercially so there was a need to develop a new RELAP-GUI. All the imperfections and short comings in the previously developed RELAP-GUI are taken into account and a new GUI is has been inspired from DWSIM with a more dynamic structure having the ability to extend. DWSIM is an open-source chemical process simulator featuring a rich GUI and developed using VB.NET [5].

The best features of the DWSIM open-source software are re-used to develop the GUI for RELAP5. Features like design surface, component connectors, drag-and-drop of the components to the design surface, assigning properties to the components in property grid and save/load. These features helped to make the code more dynamic and user friendly.

3. Description of GUI Environment

The developed GUI is named as RELAP5 Input File Generator (RIFGen) takes the nodalization data, component properties and initial conditions as input of RELAP. The program block diagram is shown in Figure 1. Thermal-hydraulic components, neutronics and control systems, save and load of the file, nodalization diagram and input file generation is interfaced with the main form of the program.

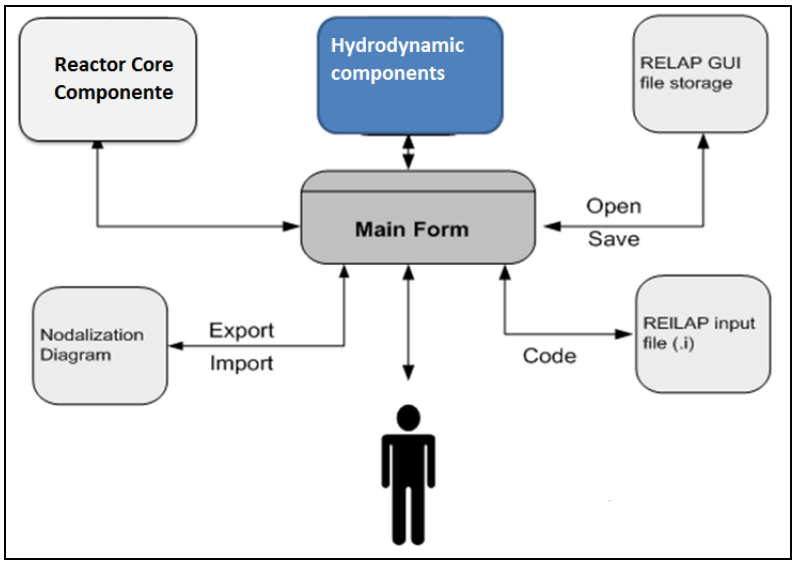

Figure 1: Block diagram of RELAP5 GUI development

3.1. Major Modules

The major modules used to develop the graphical user interface for RELAP5 are as under.

Design Surface / Flow Sheet and Graphics Collection:- The Design Surface is located in the center of the window as shown in Figure 2. Modeling of components requires the creation of a 2D graphic so that it can be rendered and manipulated digitally. Connection points are defined so that input and output connectors to different components are possible. This is the core module that has been reused from DWSIM, it has the ability to move around components, rotate them and to connect the components with each other. Components can be zoomed in or out. It uses the GDI+ API by Microsoft. Windows GDI+ is the subsystem of the Windows XP operating system or Windows Server 2003 that is responsible for displaying information on screens and printers. GDI+ is an API that is exposed through a set of C++ classes [6].

Components List:- The component list is located on the right side of the application as shown in Figure 2. It contains a list of all RELAP5 Components. These components are designed in a way that they can be added to RELAP5 in an incremental manner without disrupting the existing coding.

Component Properties and Configuration Table:- The component properties table is located on the left of the main program as shown in Figure 2. This list contains all properties of the component (Volume, Area, Wall Roughness etc.), these properties are later used to create the Input file which is used by RELAP5 to perform calculations and simulations.

Input File Generator:- The RELAP5 input file is written in accordance with the RELAP5 user manual. An output file stream writes the input file cards and subsequent words to a file with extension ".i". These words are fetched from the collection of the components stored in the parent form.

4. Implementation

An overview of the Integrated GUI for RIFGen is shown in Figure 2. It contains a Main Toolbar on the top, Initialization Toolbar also on the top, Flowsheet in the center, Property Grid Form on the left and a Component List on the right. Plot Request, Minor Edit Requests, Control System Data Input, Trips and General Core Input forms are located at the bottom.

4.1 Initialization Toolbar

This is the initialization toolbar which is always enabled regardless of the type of simulation to be modeled. This is shown in Figure 3. The fields visible in the initialization toolbar are taken directly from the user manual for the input file generation of RELAP5. A brief description of these fields is given below.

- Problem Type**:** This is combo box that contains the type of problem as NEW; RESTART; PLOT; IN-COND; STRIP; CMPCOMS.
- Option**:** This combo box specifies INPUT CHECK or RUN.
- Type of State**:** It mentions the problem type as Steady state or Transient.
- Units**:** This is to specify units as SI or BRITISH for both input and output.

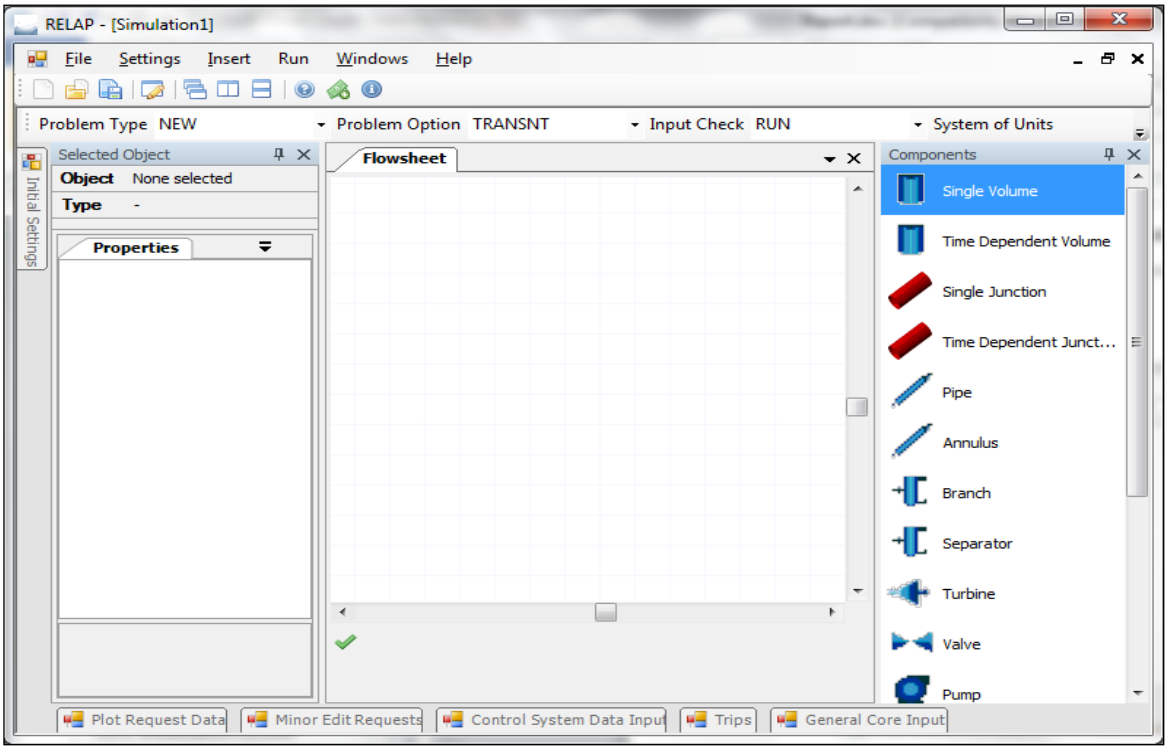

Figure 2: RIFGen Overview

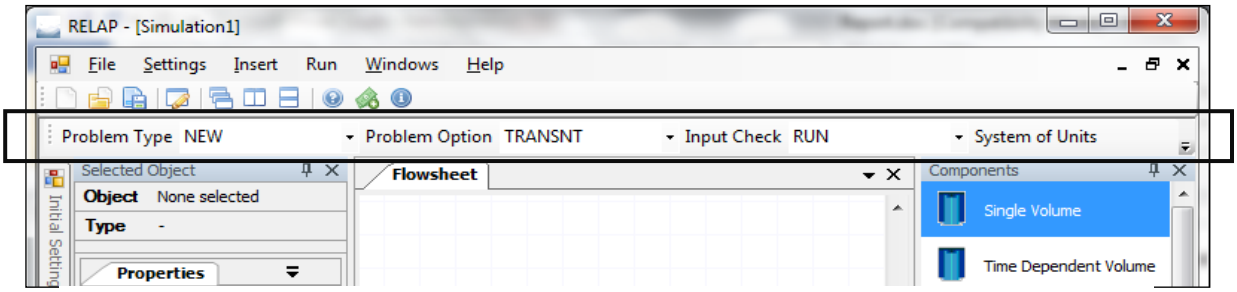

Figure 3: Main Toolbar and Initialization Toolbar

4.2. Flow Sheet

The vital module of RIFGen is the Flow Sheet, also known as the drawing canvas or design surface. GDI+ Plus based drawing canvas has been implemented so that Components may easily be dragged and dropped, repositioned, rotated or transformed into desired orientation. Graphics objects can be connected to each other using predefined connection points. These have been implemented in such a way that the process flow of the components is better visualized in the nodalization diagram. A graphics object is drawn using the curves and lines. This is done by defining points to draw a curve or a line.

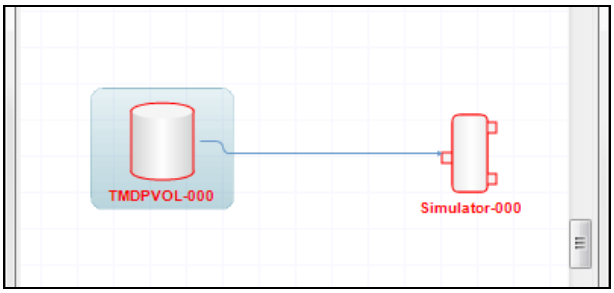

Figure 4: Example of Connectors

The graphic object also contains the tag which is used to identify the component, this identification tag is incremented automatically for example Time Dependent Volumes are tagged as TMPDVOL-000, TMPDVOL-001 and so on. This tag may be modified as per user's need. An example of connected components is illustrated in Figure 4.

4.3. Property Grid Form

This form has 2 tabs, one is the property tab and the other is the appearance tab. All component specific properties are visible in this toolbar once the component has been selected on the drawing canvas. These include properties related to the component defined in the RELAP5 User Manual. Values of these properties are formatted according to their data type. As seen in Figure 5, the Volume Flow Area is formatted as a real number formatted up to 2 decimal places. These properties are further categorized for easier user readability and navigation. Detailed description for each property is displayed at the bottom of the form. In the appearance tab the graphics object may be modified. Complex properties are further tackled in separate forms called from this parent Property Grid Form using UI Editors.

4.4. User Interface (UI) Editors

UI Editors provide a custom design-time experience for complex property types by implementing a user interface (UI) type editor.

Displaying and Editing Custom Types

When a custom type as a property is exposed, there are three ways to edit the property's value in a PropertyGrid:

- Edit the property in place as a string. This requires a TypeConverter for the custom type
- Edit the property with a drop-down UI. This is especially useful for properties that can be set with a single click.
- Edit the property with a modal dialog box. If the property is particularly complex, a full dialog box is necessary to edit it properly. This feature has been extensively used in RIFGen

To enable either single-click or modal dialog box editing, one needs to implement a UI type editor to interact with a PropertyGrid.

Drop-down Editors

Drop-down editors are ideal for types that can be set with a single click. For example, to change the "Area change junction control flag" for a Turbine with a drop-down editor as in Figure 5. The drop-down UI type editor can be accessed by clicking on the arrow button that appears next to the selected property entry in a PropertyGrid. The custom UI appears, attached to the PropertyGrid. The top of its window is positioned along the bottom of the property entry, and its width matches that of the property entry. This editor window needs to be closed after the user makes a selection. The implementation must call the DropDownControl method to position and size the UI type editor window in the design environment, and call the CloseDropDown method to close the window.

Modal Dialog Editors

Modal editors are useful for types that require a fully interactive UI. For example in Figure 6, the "Thermo dynamic states" property is handled by the modal editor. The modal UI type editor may be accessed by clicking on the ellipsis button that appears next to the selected property entry in a PropertyGrid. The

modal dialog box appears, and the user interacts with it like a typical dialog box. The implementation must call the ShowDialog method to position and size the dialog box in the design environment.

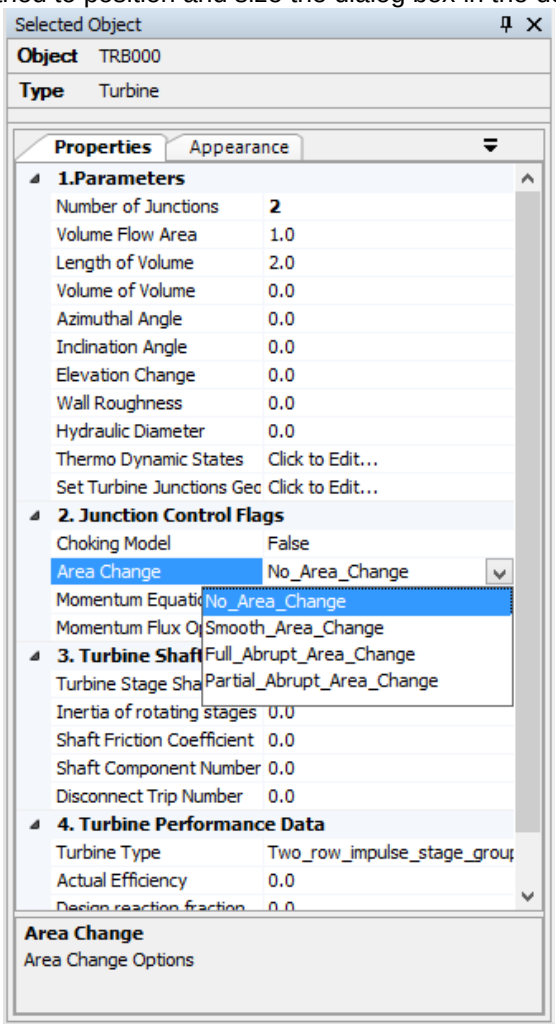

Figure 5: Property Grid Toolbox for Turbine

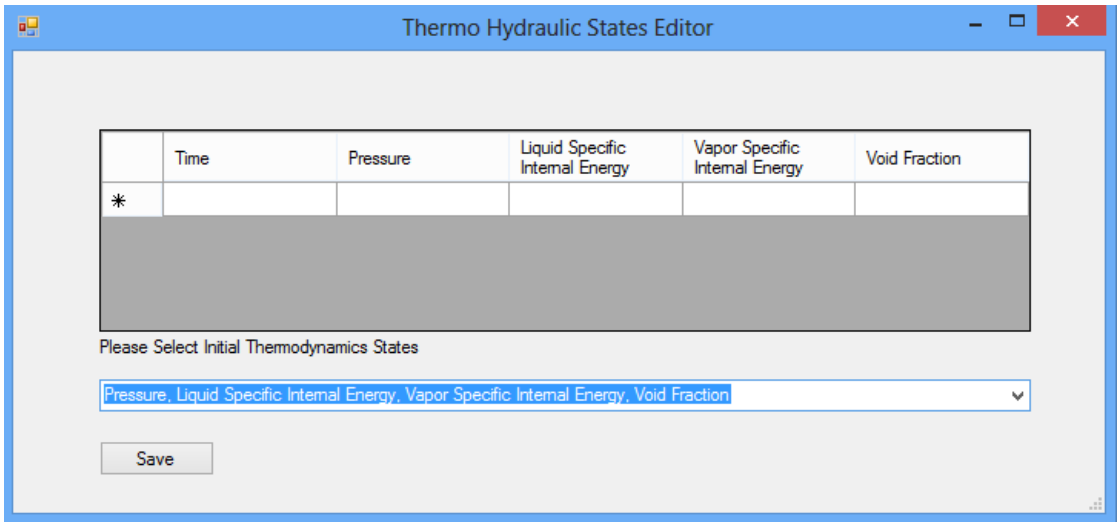

Figure 6: Modal Dialog Editor for Turbine

Implementing a UI Type Editor

To implement a custom UI type editor, the following tasks needs to be performed:

- Define a class that derives from UITypeEditor.
- Override the GetEditStyle method to inform the PropertyGrid of the type of editor style that the editor will use.
- Override the EditValue method to handle the UI, user input processing, and value assignment.

The following tasks need to be performed to add additional support for painting a value's representation in a PropertyGrid:

- Override GetPaintValueSupported to indicate that the editor supports displaying the value's representation.
- Override PaintValue to implement the display of the value's representation.
- Override the UITypeEditor constructor method if the editor should have initialization behavior.

4.5 Component List

This has been achieved by using the image list control. Component images have been populated into the image list control using Visual Studio's designer interface. The MouseDown event is used to detect the dragging of components towards the flow sheet. A partial list of components is shown in Figure 7.

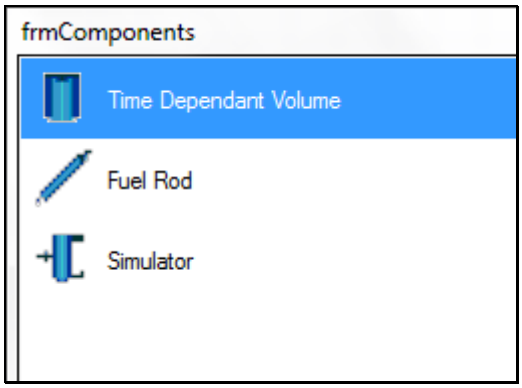

Figure 7: Partial Components List

5. File Save/Load

This has been achieved by writing export files for each object in the GUI and zipping them into a single file with the extension of ".RELAP". Objects and GUI layout is serialized and exported to binary files. This may later be used to read and load the layout back again. The properties of the components are in this way saved and may be reused later when needed.

6. Validation

The validation of the RIFGen generated input file is necessary to test whether the system produces desired results. This is achieved by generating input files for standard test problems and comparing the results.

6.1 Standard Test Problem

In our selected test problem there is a flow of water between two vessels through a vertical pipe. One vessel is a source and other a sink. These vessels are time dependent volumes and they are connected to the pipe using single junctions. Source is at higher pressure than sink and when the water flows through the pipe, pressures of the vessels changes and so does the flow. In Figure 10 , the variation of fluid flow is shown with the passage of time. Process flow diagrams and nodalization diagrams are in Figure 8 and Figure 9 respectively. The final plot can be seen in Figure 10.

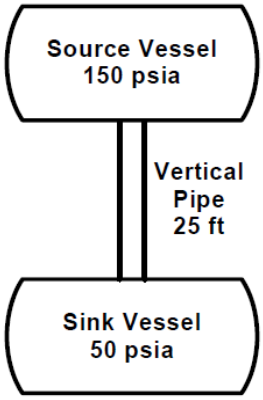

Figure 8: Process flow diagram

6.2 Sample Input File

```
=  Simple three component system
100  new  transnt
102  british  si
105  10.0 40.0 200.0
110  air
115  1.0
201  20.0  1.0e-6  0.05   3  1  50  2000
20300011  mflowj 120000000 1
20300012  mflowj 127000000 1
1100000  "source"  tmdpvol
1100101  20.0  0.0  1.0e6  0.0  -90.0  -5.0e4  0.0  0.0  0000000
1100200  003
1100201  0.0    150.0  120.0
1200000  "sngljuni"  sngljun
1200101  110010002  125010001  0.1  0.0    0.0    0000100
1200201  1       0.0           0.0           0.0
1250000  "stmpipe"  pipe
1250001  5
1250101  1.0  5
1250301  5.0  5
1250501  0.0  5
1250601  -90.0  5
1250801  0.0  0.0  5
1250901  0.0  0.0  4
1251001  0000000  5
1251101  0000000  4
1251201  3  100.0  90.0  0.0  0.0  0.0  5
1251300  0
1251301  0.0    0.0       0.0      4
1270000  "sngljuno"  sngljun
1270101  125050002  130010001  0.1  0.0  0.0 0000100
1270201  1  0.0  0.0  0.0
1300000  "sink"  tmdpvol
1300101  20.0  0.0  1.0e6  0.0  -90.0  -5.0e4  0.0  0.0  0000000
1300200  3
1300201  0.0  50.0  90.0
. End of input.
```

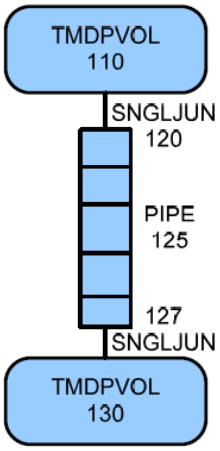

Figure 9: Nodalization diagram

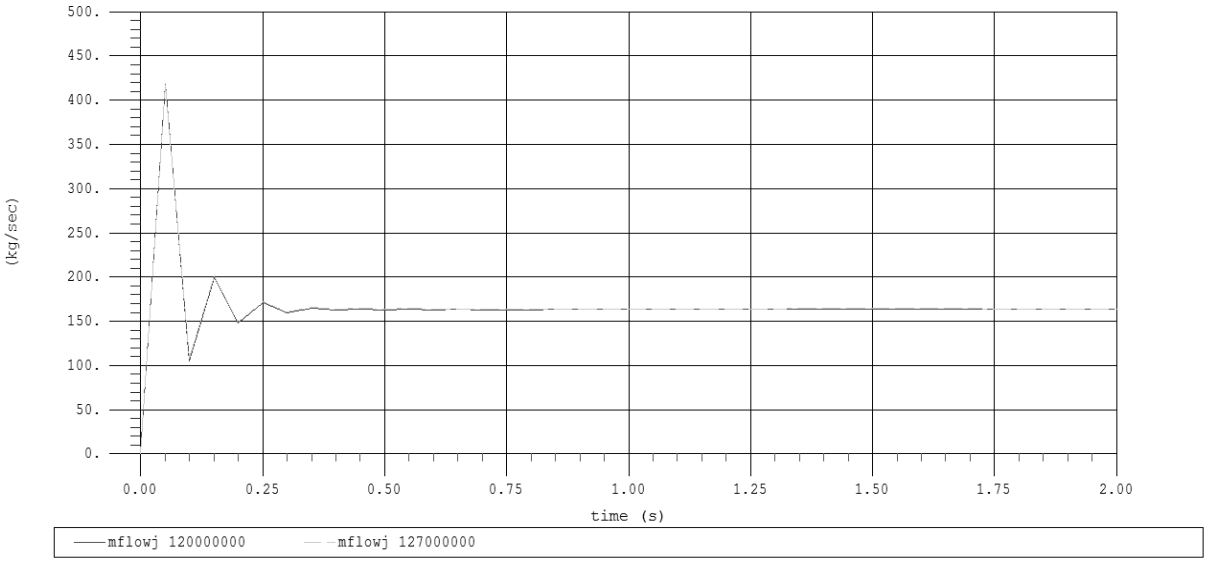

Figure 10: RELAP5 plot for standard problem

6.3 Nodalization diagram using RIFGen

The above sample test problem has been remodeled using RIFGen and its nodalization is displayed in Figure 12. This model has been used to ultimately generate the input file for RELAP5.

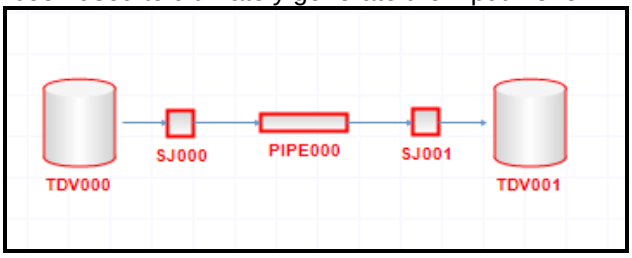

Figure 11: Nodalization diagram using RIFGen

6.4. RIFGen Generated Input File

100 new transnt
101 run

```
102 british si
105 10.0 40.0 200.0
110 air
115 1.0
201 2.0 1.0e-6 0.05 3 1 50 2000
20300011 mflowj 002000000 1
20300012 mflowj 004000000 1
0010000 "TDV000" tmdpvol
0010101 20.00 0.00 1000000.00 0.00 -90.00 -50000.00 0.00 0.00 0000000
0010200 003
0010201 0.00 150.00 120.00
0050000 "TDV001" tmdpvol
0050101 20.00 0.00 1000000.00 0.00 -90.00 -50000.00 0.00 0.00 0000000
0050200 003
0050201 0.00 50.00 90.00
0020000 "SJ000" sngljun
0020101 001010002 003010001  0.1 0 0 0000100
0020201 1 0 0 0
0040000 "SJ001" sngljun
0040101 003050002 005010001  0.1 0 0 0000100
0040201 1 0 0 0
0030000 "PIPE000" pipe
0030001 5
0030101 1.00 1
0030102 1.00 2
0030103 1.00 3
0030104 1.00 4
0030105 1.00 5
0030301 5.00 1
0030302 5.00 2
0030303 5.00 3
0030304 5.00 4
0030305 5.00 5
0030401 0.00 1
0030402 0.00 2
0030403 0.00 3
0030404 0.00 4
0030405 0.00 5
0030501 0.00 1
0030502 0.00 2
0030503 0.00 3
0030504 0.00 4
0030505 0.00 5
0030601 -90.00 1
0030602 -90.00 2
0030603 -90.00 3
0030604 -90.00 4
0030605 -90.00 5
0030801 0.00 0.00 1
0030802 0.00 0.00 2
0030803 0.00 0.00 3
0030804 0.00 0.00 4
0030805 0.00 0.00 5
0031001 0000000 1
0031002 0000000 2
0031003 0000000 3
```

```
0031004 0000000 4
0031005 0000000 5
0030201 0.00 1
0030202 0.00 2
0030203 0.00 3
0030204 0.00 4
0030901 0.00 0.00 1
0030902 0.00 0.00 2
0030903 0.00 0.00 3
0030904 0.00 0.00 4
0031101 0000000 1
0031102 0000000 2
0031103 0000000 3
0031104 0000000 4
0031201 003 100.00 90.00 0.0 0.0 0.0 5
0031300 0
0031301 0.00 0.00 0.00 1
0031302 0.00 0.00 0.00 2
0031303 0.00 0.00 0.00 3
0031304 0.00 0.00 0.00 4
```

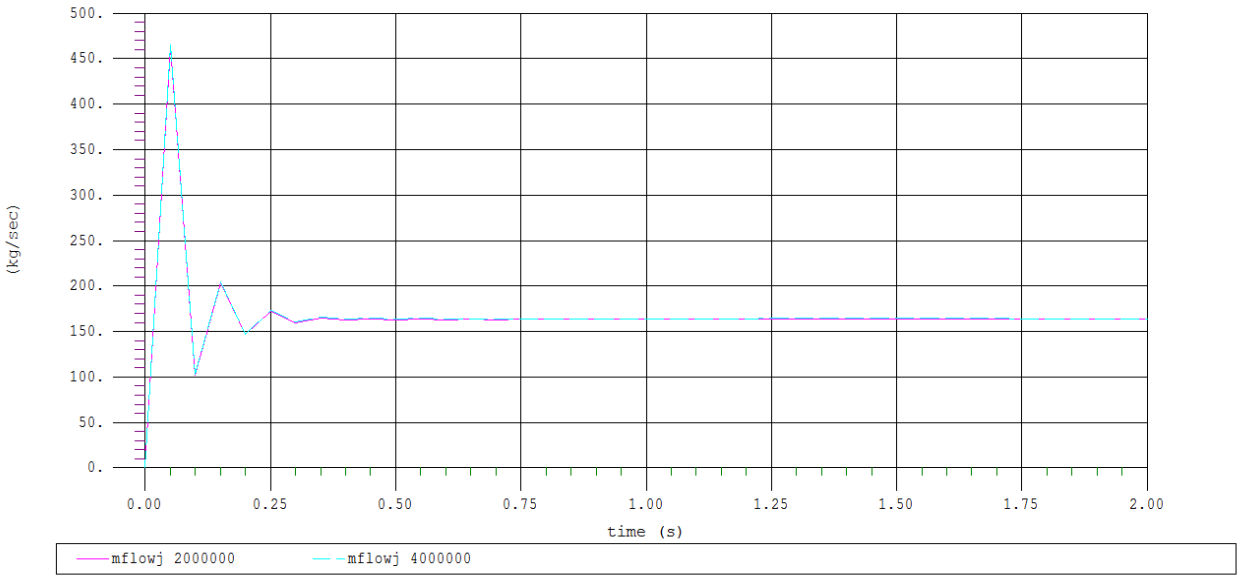

Figure 12: RELAP5 plot of RIFGen generated input file

7. Results and Conclusion

The GUI for the preparation of the input file for RELAP5 has been developed using VB.NET as the programming language, using GDI+ as the main module for drawing purposes. It is capable of reducing work of hours to few minutes. Its results have been validated for a simple 3 component system. RIFGen exhibited desired results and this can be seen in the Figure 10 and Figure 12. Both graphs are 100% correlating. The published open-source code is available on http://relap.codeplex.com under the GPLv3 license.

Acknowledgements

The author acknowledges Dr. Imran Rafique Chughtai and Dr. Muhammad Ilyas for their kind help, guidance, suggestions and support through the development of this project. Credit also goes to Pakistan Institute of Engineering and Applied Sciences and ACRE (Advance Computational Reactor Engineering) Lab for providing very conducive educational environment.

References

[1] Daniel Wagner Oliveira de Medeiros, “DWSIM - Process Simulation, Modeling and Optimization”, Version 2.1, Revision 0, January 2012 Source Code Guide.

[2] CAPE-OPEN Methods and Tools Guidelines by “Industrial and Materials Technologies.

[3] RELAP/MOD CODE MANUAL, VOLUME I: CODE STRUCTURE, SYSTEM MODELS, AND SOLUTION METHODS, NUREG/CR-5535, JUNE 1995.

[4] SCDAP/RELAP/MOD3.2 CODE MANUAL, NUREG/CR/6150, REV.1, JULY 1998.

[5] E. A. Harvego, et al., "Developmental assessment of the SCDAP/RELAP/MOD3.2 code" 6th Inter. Conference on Nuclear Engineering, ICONE-6, May 1998.

[6] James J. Duderstadt and Louis J. Hamilton, “Nuclear Reactor Analysis”, John Wiley & Sons, New York/London/Sydney/Toronto, 1976