Chapter 1

HIMAP Overview

Hierarchical Modeling and Analysis Package, HIMAP, is a Reliability Modeling and Analysis Tool. HIMAP provides modeling and analysis support through an integrated graphical user interface for Fault trees, Markov chains and Stochastic Petri Nets. Some of the reliability, availability, or performability analysis tools like EHARP (Markov analysis tool for analyzing multiple mission models), HSP (Hierarchical Stochastic Reward Net Solver) are used as underlying solvers. This manual will provide you with necessary help to get started in reliability/availability modeling using HIMAP. The manual also highlights the windowing structure of HIMAP with detailed explanations about editing features and program controls.
HIMAP provides the user with several useful features like:

- Hierarchical modeling features where the sub-models of fault trees can be linked together through special gates for efficient management of models.
- Efficient library structure for maintaining information about the elements in the fault tree model.
- Standard interfaces to develop Fault trees, Markov chains and Stochastic Petri Nets
- Standard output graphics features like postscript and 'xvgr' plots of the models and results.

The unix version of HIMAP requires that user have certain path presets in the users '.cshrc' file. The presets and their importance are explained in detail in next section. This manual is not intended to be any reference on the modeling aspects in advanced modeling paradigms like Markov chains and Stochastic Petri Nets. For more detailed modeling support, please consider one of the references listed at the back of this manual.

Contact Information
Prof. Arun K. Somani
Dependable Computing & Networking Laboratory
Iowa State University
Ames, IA 50011
Fax: (515)-294-8432 Phone: (515)-294-0442
E-mail: arun@iastate.edu
URL:http://shasta.ee.iastate.edu/ himap.html

Chapter 2

HIMAP Requirements

2.1 User Setup (Unix Version)

2.1.1 '.CSHRC' File Setup (Unix Version)

HIMAP is a X-Windows based graphic oriented reliability modeling and analysis tool. Before one starts the application, few system variables need to be set up in a user setup file '.cshrc'. This file, '.cshrc', is a C shell defaults file located in your home directory on the UNIX server. The system variables tell the program where the component libraries should be saved, where the underlying tools (EHARP, SPNP, SDM, SETS) executable are and where the files have to be printed. Parameters which need to be set up are given below:

- HIMAP_HOME → This variable represents the directory where the HIMAP binaries are stored.
- HARP_EXEC → This variable gives HIMAP the information about underlying tools (EHARP, SPNP) binaries. The user should give the appropriate path. The user has to include this in the PATH variable.
- HIMAP_GLOBAL_LIB → This variable gives HIMAP the path where the global libraries are stored. Global libraries are the component information libraries that have failure rate information and are a superset of project libraries. Library structure is explained in detail in the next chapter.
- PRINT_PATH → This variable represents where the file 'header.ps' is present. This file provides the necessary headers for the postscript files to be created when the picture files are printed.

Apart from this, the file ERROR.TXT should also be stored in the same directory where the HIMAP executable is stored. This file stores the various error messages that HIMAP will generate to guide the users.

2.1.2 System Resources

If you are using a Unix workstation, then skip this section.To run the windows version of HIMAP, a Pentium machine with 32MB RAM is recommended.

Chapter 3

HIMAP Layout - Fault Tree Editor

This section discusses the features provided by HIMAP in the fault tree modeling environment. The main window is shown below. Section 3.1 discusses the options available in the main window menu bar. In Section 3.2, the procedures for creating models, component libraries are explained.

3.1 Main Window Menu Bar

The default modeling environment in HIMAP is the fault tree modeling environment. The modeling environment can be changed by choosing the 'Change Modeling Environment' option of the Utils popup in the HIMAP main window.The main menu bar in the HIMAP window has the following options:

- MODEL: This option provides the capabilities of creating, opening, or saving model files (".mod"). It also provides features for printing the models.
- LIBRARY: This option provides the capability of creating, opening and editing the library.
- VIEW: This option provides the Zoom in and out capabilities and label printing capabilities.
- UTILS: This option provides the capability for create/change the project, create phases, change modeling environments and print results.
- TOOLS: This menu provides the interfaces to several reliability analysis tools.
- ACTIVE MODELS: This option shows all of the model files which are active in current project. This option is available only in Fault Tree modeling environment. In the windows version this option is supported under the 'Windows' menu item.

The Unix version of HIMAP has a Message area. The message area has the feedback information for the user about the current project, current picture, current component library and a status window to display any error/status messages. In the next few subsections, these window options and menus are explained in detail.

3.2 Model Menu

Model menu has the following options: New, Open, Close, Save, Save As, and Print. A new reader is strongly urged to read the section describing the model development after reading this section.

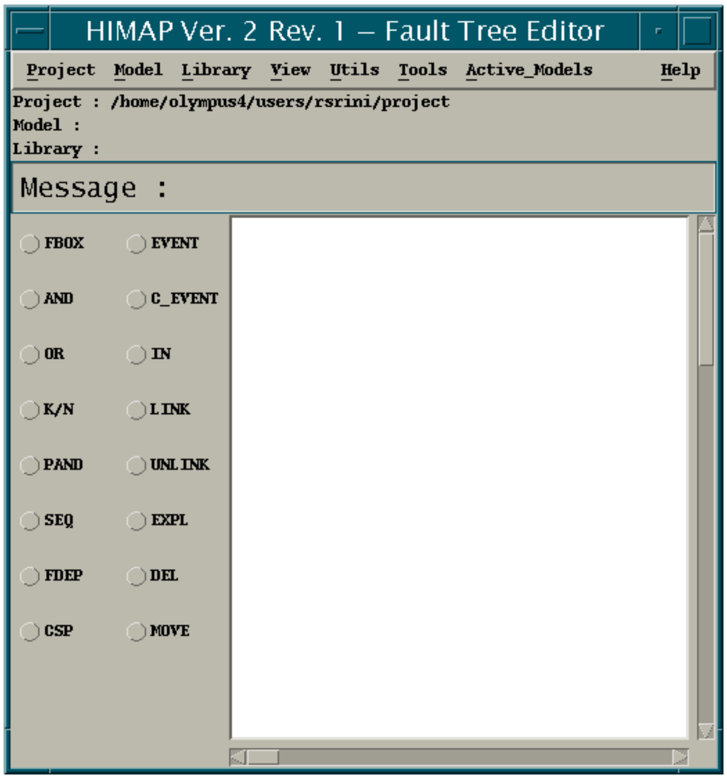

Figure 3.1: HIMAP Fault Tree Editor

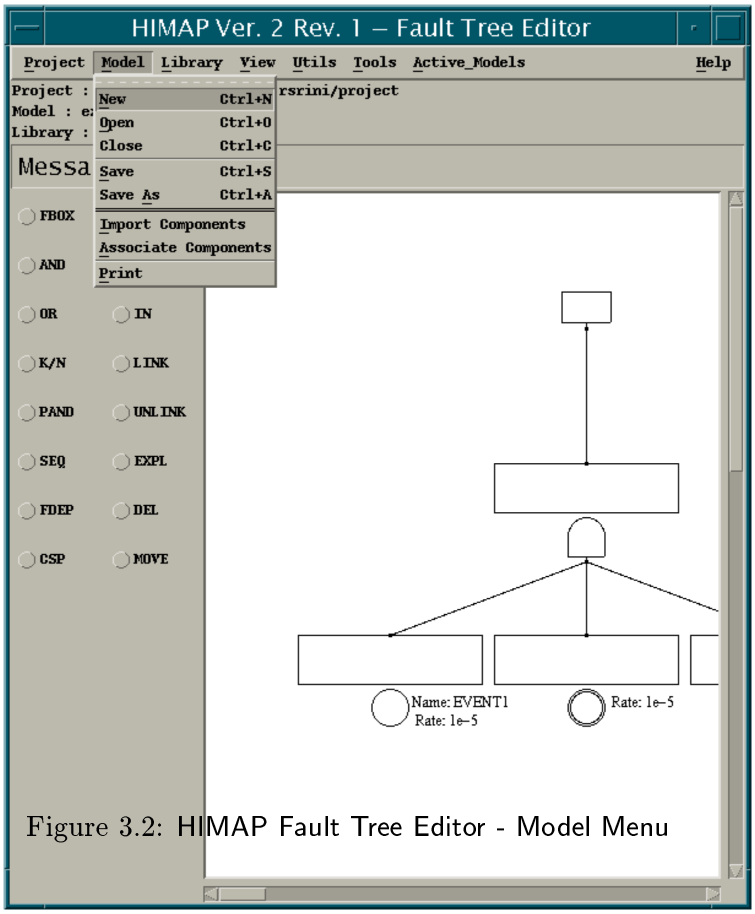

Figure 3.2: HIMAP Fault Tree Editor - Model Menu

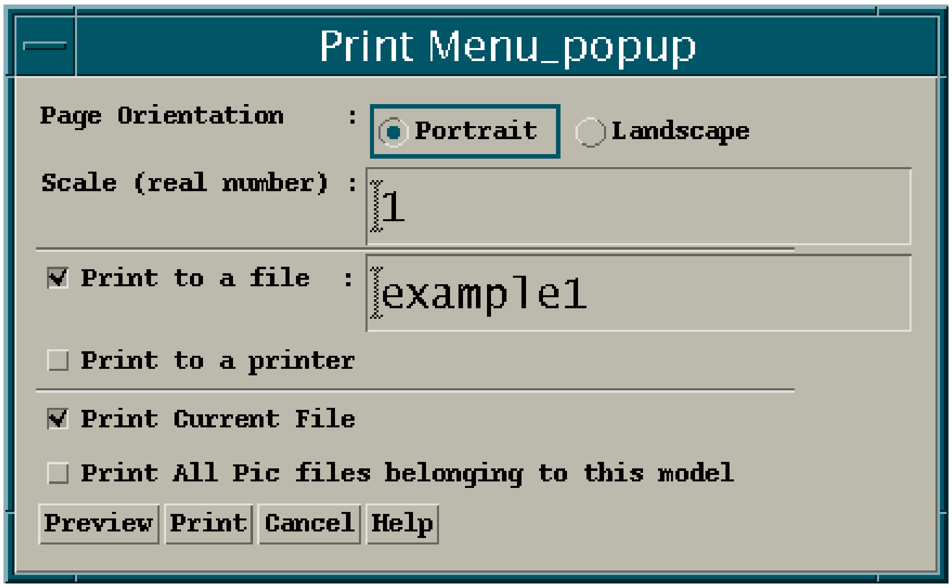

Figure 3.3: Print Dialog

- New: This option is used to create a new model file. A window pops up prompting the user to name the model.
- Open: This option is used to open a model file. A window pops up with model files listed. The user needs to select a desired file and then click 'Open' to open the file.
- Close: This option is used for closing an open model if the model has been saved. If not HIMAP gives the user the option of saving the model before closing.
- Save: This option helps the users to save the file in the disk.
- Save As: This utility allows the users to save the open model with option of giving a different name.
- Print: This option could be used for printing the model or the analysis result. Print model option would bring up the following window. Printing the model helps the user getting a hard copy of the picture file, which is being edited. The user has the option to print the file either as a Portrait or as a Landscape. The default selection here is Portrait. The option of printing the result can be used to plot output of a particular run. Selecting this option would bring up the following window, which shows all the result files which are ready to be plotted. One can select multiple files from the list shown at the left and then select 'Plot'. This will plot Unreliability of the system against time.
- Exit: As the name implies, choosing this option terminates the program. If the current file is not saved then the user is prompted to save the file before exiting HIMAP.

3.2.1 Library Menu

The library files (".lib") are necessary for the creation of the model files. The event elements are added to the model file from the library files. The Library menu has the following options: New, Open, Close, Edit.

- New: This option is used to create a new library file. A window pops up prompting the user to give the name of the library file to be created.
- Open: This option is used to open a library file. A window pops up with the library files listed. The users need to select the desired file and then click OK to open the file.
- Edit: When a library file is opened, Edit library button will bring the Edit Library Popup. Using this window, user could edit the parameters of existing element, add new elements or delete old element from the library. Please refer to section 5.16 for more details about how to create a library file.

3.2.2 View Menu

View menu has few main viewing options like Zoom in/out, Show/Hide Numbers, Show/Hide Description, Show Children, Show Parents and Show Sub_tree.

- Zoom in / out: Enlarges and shrinks the picture. NOTE: The picture can be edited effectively only when there is no enlargement or reduction. The zoom destroys the grid layout and hence it is highly recommended not to perform any graphics editing while not in 100 % zoom mode.

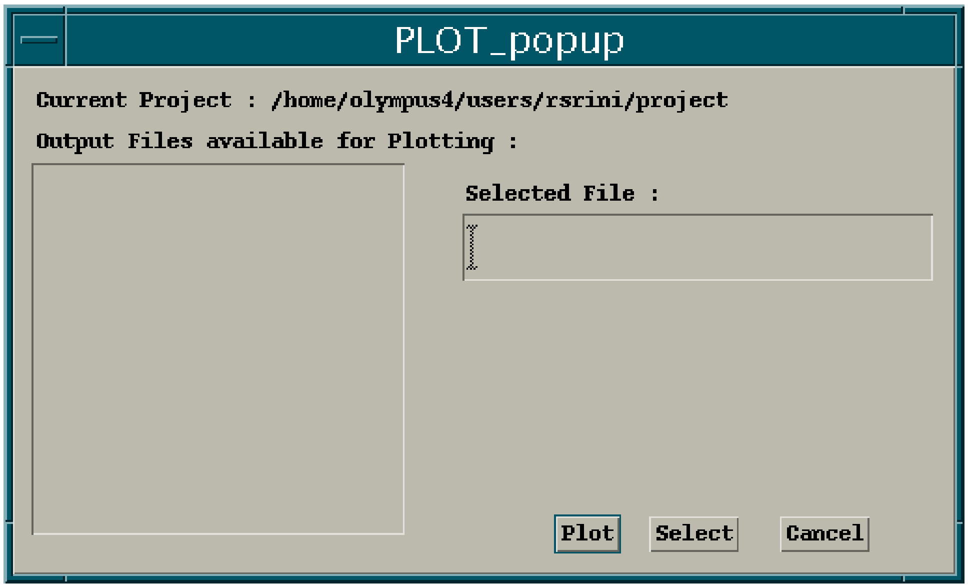

Figure 3.4: Plot Dialog

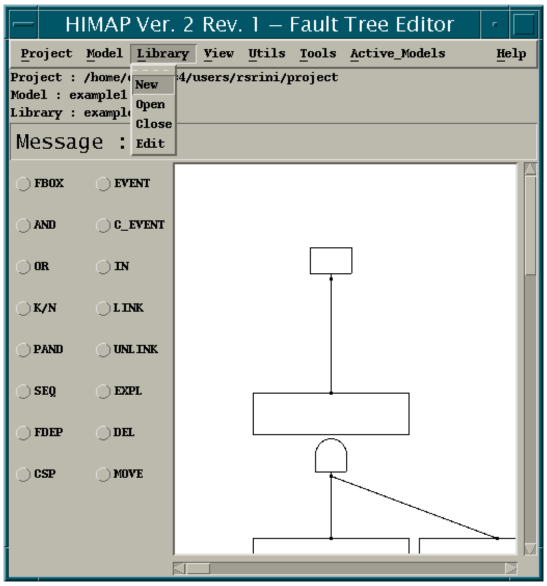

Figure 3.5: HIMAP Fault Tree Editor – Library Menu

- Show / Hide Numbers: This button performs a toggle operation of showing or hiding the user, the object ID's in the picture. The default mode is the hide mode.
- Show / Hide Description: This button performs a toggle operation of showing or hiding the description box for objects in the picture. The default mode is the show mode.

3.2.3 Utils menu

Utils menu has the following options: Create/Change Project, Change drawing Mode, and Create Phase.

- Create/Change Project: This helps the user to change the current working directory to a different working directory. All files in the current directory need to be closed before using this option. This button gives the same option as the starting Project popup window, which appears when one first starts HIMAP.
- Change Modeling Environment: This button is used when the user wants to change the mode of drawing method to the system. The default mode is Fault Tree Editor and the other modes available are Markov Chain Editor, Stochastic Petri Nets.
- Create Phase: This option helps the users to create a model for one phase of the system and it prompts the user to provide information needed for generating the phase model. When create phase is clicked, a window pops up asking about the model file name and the 2 character prefix (say pi for phase i). Given the model name, create phase creates all the .pic and .mod files hierarchially related to describe the model with (pi_) as the prefix. It changes the inside of the .mod files appropriately.

3.2.4 TOOLS Menu

This pulldown menu essentially provides the backbone for tool suite allowed in HIMAP. This button provides the link to the various analysis tools. After the mode file has been converted by global linking, the following options could be used to analyze the model:

- Solve Fault Tree
- Generate Markov Chain
- Solve Markov Chain

Solve Fault Tree

The activation of this button asks for the mode file to be analyzed. The input mode file should have been converted to a readable form for HIMAP tools by global linking. Once the appropriate mode file has been given, the user should click ok to start the analysis. In Fault Tree Editor mode, selecting this button would solve the system by a fault tree solver. In a Markov Chain editor mode, selecting this button would solve the system by EHARP solver. In Stochastic Petri Net editor mode, selecting this option would solve the system by SPNP solver. The final reliability result will been shown in a pop up window.

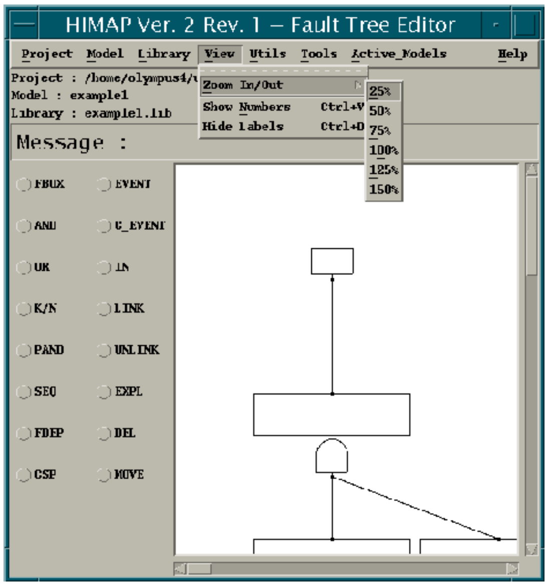

Figure 3.6: HIMAP Fault Tree Editor – View Menu

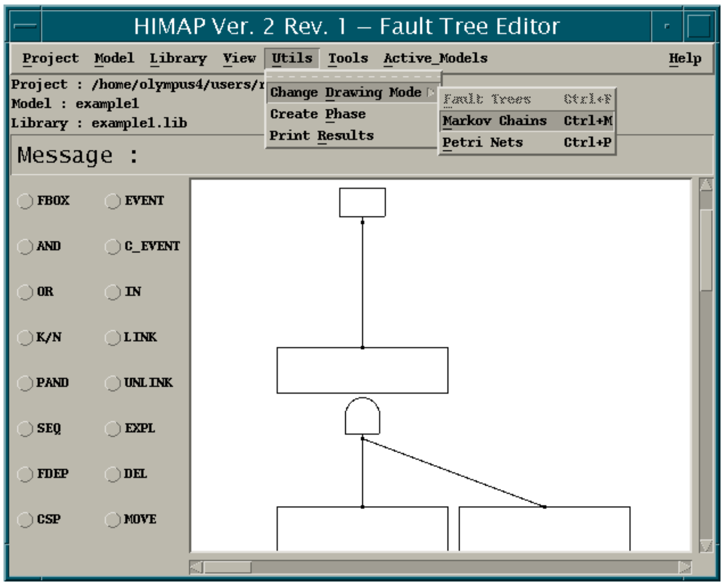

Figure 3.7: HIMAP Fault Tree Editor – Utils Menu

Generate Markov Chain (GMC)

The popup, which results on the activation of this button, shows the files that are ready to generate Markov chain. The user can appropriately click on the file, which he wants the software to run on. He can also give the inputs to the "Truncate after how many failures?" But if the user chooses not to change these values then the default values, which are shown in the window, are chosen. If the user wishes to run GMC for a number of files then the user is required to give the above details for each input file and activate the "Select" button. The user activates the Run button after selecting all the files. The Markov chain generated is automatically updated to include repair arcs. The "Cancel" Button is used to dismiss the GMC window.

Solve Markov Chain (SMC)

GMC produces the required files for SMC. When the user chooses the SMC option, a file dialog window will popup with NEW and OK buttons. The user can choose an existing .INH file and modify the file or create a new file. When the user clicks on NEW button the SMC window will popup. To run the SMC module the user has to give a Global name for the module. The user can also change the default values for the Number of missions, Number of Phases/mission and Number of checks. The following figure shows how the SMC window looks like. On activating the "Execute" button the user is prompted to specify parameters for each of the phases in the mission as shown below.

- Phase Information: The user has to specify the name for each phase. In a single-phase system, the name of the model file is also the name of the phase file. For multi-phase system, Crate phase facility can be used to generate model for each phase. The user then has to specify whether the phase is of fixed or random time intervals. If the choice for the time interval is fixed then the user has to specify the Fixed Phase Duration as well as the Reorting Interval. If the choice for the time interval is Random then the user has to specify the minimum and the maximum time for the distribution. The user also has to specify the mean phase time. "Add a New Phase" allows the user to add a phase. If it is a new phase the user needs to input the phase information or he can start from an old .IMH file and the user can Add/Delete a phase, mission or check.
  HIMAP requires that the mean phase time be a value between the maximum and minimum phase time. The user next has to specify bound information. The default is Yes. The user then has to specify whether there is a State To State Mapping. The default here is No.
  NOTE: The following two sections will appear if and only if the system is a MULTI MISSION SYSTEM and if the maintenance information window in the SMC popup has a positive value.

- Mission Redistribution Information: The user has to specify how HIMAP should distribute state occupancy probability between missions. The default here is YES. For multi-mission analysis, advanced users may be able to perform state-to-state mapping between missions by choosing S-S-Mission Map option. This is only recommended for small state space system and very specific scenarios.

- Maintenance Check Information: The user gives a name for each of the maintenance cycle. By clicking on the 'Comp' button the user can specify which components to maintain. This is done by "moving into the maint. list" action elements from the element list. The user can use the Move All into Maint. List Button if maintaining all the elements is necessary. A similar button is also provided to delete all elements from the maintenance list. The user has to specify the number of cycles after which the check has to be done. The user can now update

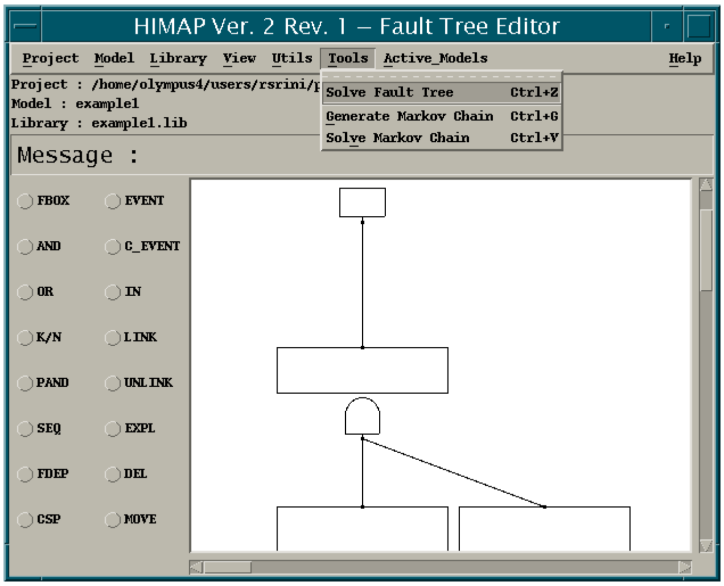

Figure 3.8: HIMAP Fault Tree Editor – Tools Menu

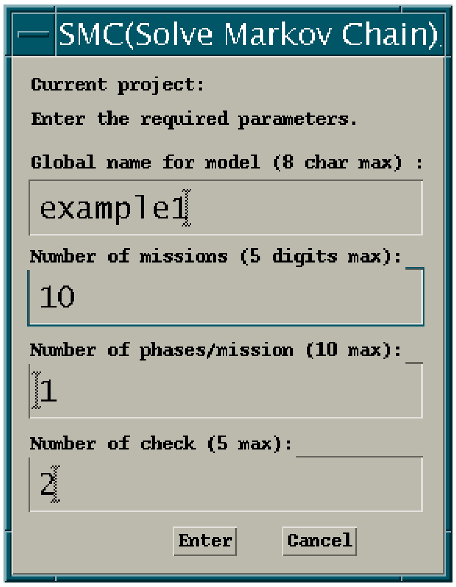

Figure 3.9: Solve Markov Chain

the values of the name of the maintenance cycle, the components and the number of cycles into the maintenance list by selecting the Update check values button.
Once these specifications are finished the user now should click on the "exec-harp" button. This will create all the files necessary, calculate the reliability of the system automatically and display the output as a X-Y plot of Unreliability .vs. Time.

3.2.5 Active Models

This option is available only in the Fault Tree drawing mode. It shows all the model files the user opened in the current project.

3.3 Creation of Model Files

To develop a model the users have to start with two files: .lib, .mod files. Once a model is created, different tools can be used to solve the model. This section will introduce these two files. Once a current project directory is selected, the user could start to develop library file ".lib" and model file ".mod".

3.3.1 Creating and Managing Model Files

There are three types of models that can be developed using HIMAP: Fault Tree, Markov Chain, and Petri Nets. The default drawing mode in which HIMAP starts is the Fault Tree Drawing mode. A user can change to different types of model development by using the "Change Drawing Mode" option of the Utils menu in the main window. Markov Chain editor and Petri Net editor will be introduced in section 4 and 5 respectively. The model file takes the position of a high level programming language. HIMAP facilitates the user to develop a model with minimum work. The user is allowed to relate the system model to the software pictorially by creating a Fault Tree, a Markov Chain or a Stochastic Petri Net. This requires the user to use some sort of mechanism to translate his ideas to HIMAP readable form. The following steps will help the user accomplish the same.
NOTE: In this manual, to simplify the explanation, → is used to refer to pulldown sub menu.
Creation: Model → New helps create a new picture file. The user will be asked for a model file name and a file "example1.mode" (in this example) in the current project directory is created. "example.pic" file, as an ASCII representation of the picture, is also created in the current project directory.

The user uses the drawing area to draw the fault tree representing the system behavior. HIMAP provides several constructs to help the user draw the Fault Tree. This section describes the various constructs available in HIMAP and how the user can make use of these constructs to develop a fault tree model.

Fault Tree Drawing Constructs

The drawing primitives have the toggle buttons to draw FBOX (top gate in a fault tree), AND, OR, K/N (combination), C_EVENT, CSP, FEDP, SEQ, PAND and BASIC EVENT. In addition, a transfer gate (IN gate) is provided to develop hierarchical models, LINK/UNLINK options to build the fault tree logic and few edit operations like 'delete' (DEL option) and 'move' (MOVE option), and to provide explanation to an object (EXPL option).

- *The FBOX :* An FBOX can be created using the FBOX button. One and only one FBOX is allowed in a picture. FBOX represents the TOP event in Fault Tree logic.

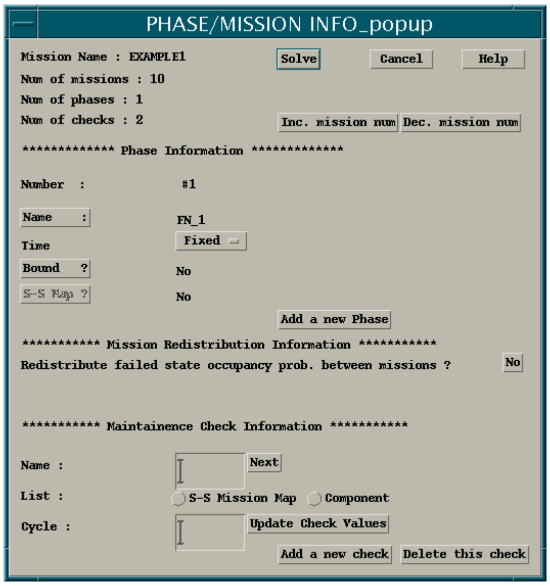

Figure 3.10: Phase Mission Information

Figure 3.11: New Model Dialog

- *The AND & OR gates :* These buttons help the user create the AND & OR gates respectively.
- *The M/N gate :* If the user decides to include a gate which is asserted in the negative logic when M out of N inputs are asserted then this button can be used to create a M/N gate. User is prompted for M and N at the time of mapping the gate. In the drawing mode, user should not worry about it.
- *The Event :* A circle in HIMAP denotes it and this represents the BASIC events in the Fault Tree. This could also represent a developed event in a Fault tree where the probability of the event is developed through reliability analysis external to the current model.
- *The IN gate :* When the user wishes to insert a picture file in a different picture file then the IN gate represents the point of insertion. This is a very critical gate for developing hierarchical models and for managing huge fault trees.
- *LINKing :* Linking different events and gates are done using the Link function. The procedure for linking two objects on the picture involves a two-click operation of clicking on the source object click on the destination object.
- *UNLINKing :* If the user wishes to unlink two objects on the picture then this function is used. This is exactly opposite to the LINK function and involve disconnecting two objects whenever user selects both source and destination object.
  NOTE: One critical difference between LINK and UNLINK operations is that LINK is directional whereas UNLINK is not. So, for LINKING, one always clicks on a source object first and then on a destination object. Whereas in UNLINK operation, one can click on any two objects and a link between the two objects is removed.
- *DELETE :* This is used when the user wishes to delete an object from the picture. The procedure to do this is select DEL Click on the object to be deleted.
- *MOVE :* This button is used when the user would like a particular object in the picture to be placed in a different position. When the objects are moved, links associated with objects are also redrawn to show the connections. The logic of the Fault tree model is not changed by this operation. The procedure to be followed is MOVE Click on the object to be moved Click on the new position of the object.
- *EXPL :* This button is used when the user wishes to show the explanation in the explanation box. Once this button is selected the Explanation popup window is made active. The user types in the explanation in the box and selects ENTER. CLEAR deletes the entire explanation and CANCEL kills the operation.
- *C_EVENT :* In order to simplify the linking of the graph one event could appear in different places of a huge picture file. HIMAP will know these events are the same if the users use C_EVENT button to draw these events and announce them to be common events in mapping.
  NOTE: All common events NEED to be in the same model file.If the user wishes to have a common event(shared event) over many models then the user has to define the shared event through a separate model.
- *CSP :* This button is used to draw the cold spare gate. All inputs to the cold spare gate must be basic events. The leftmost input represents one or more primary units that are initially on-line. All inputs to the right of the leftmost inputs represent alternate (spare) units or groups of redundant units available as pooled spares that are initially powered down (i.e., that are cold

spares). Upon the failure of any of the units that are active, replacements are selected from the set of spare units that have not yet been placed on-line. The spare units must be switched into operation in the left-to-right sequence in which they appear ass inputs to the cold spare gate.

- *FDEP :* This button is used to create functional dependency gate. It has input, called the trigger input, which can be any general event. This gate also has a number of dependent input events, which must be basic events. The output from this gate is always identical to the value of the input event.

- *SEQ :* This button could be used to create sequence-enforcing gates. This gate could have any number of inputs. The left most input could be any general event, unlike the CSP gate, (e.g., the output of any other fault tree gate or any basic event). All other inputs to the gate must be basic events. The gate has an output, which is active when all the inputs are active.

- *PAND :* This button is for the priority and gate. It has a maximum of two inputs. However, two or more priority and gates can be cascaded together to achieve the effect of a multi-input gate. The inputs of the priority and gate can be any general event (e.g., the output of any other fault tree gate or any basic event). The output event only occurs if all input events of the gate occur in the left-to-right sequence in which they appear as inputs to the gate.

NOTE: Using the New button creates a version of the image in memory only and not on disc. To create the disc version of the file one has to use the "Save" or the "Save As" buttons.

- *Open :* Model → Open helps the user to open an already existing model file from the disk (this routine searches the current working directory and displays all the files with a .mod extension). The user then can click on the required file name and the file is then opened. The names of the open model files are shown in the list on the window. It is also to be noted that the user can open more than one file at a time.

- *Save :* Model → Save is useful to save an opened picture file into disk. Information about whether a particular picture file is saved is available in message area of the main window. The picture file saved is always the foreground picture

- *Save as another name :* Model → Save As is useful in saving the picture file into a picture file of another name. Switch between different picture files: This is useful when multiple files are open at the same time and the user wants to change from one file to another. It there is any open or newly created picture files, a menu named "pictures" appears in the main window. It will give the user a shortcut of switching between any open picture files.

- *Close :* Model → Close button is used to close a particular file which the user does not want to be in memory. If the file has been saved then the file is removed from memory. Else the user is given the option of saving before removing from memory.

3.3.2 Creating and Managing Library Files

The ".lib" library files are necessary for the creation of the mode file. The event elements are added to the mode file from the library file. The library forms the main specification for HIMAP. The information in the library defines the characteristic of each element, which is going to be used in the mission. This section will first briefly introduce the library structure in HIMAP and the go into the details of how the user can create libraries.

The library structure

The library structure in HIMAP has two levels. The top level consists of the Global Library and the lower level consists of a local library. The Library serves the purpose of defining the components. The Global Library if created can serve the purpose of a database of all the components, which will ever be users, though its existence is not necessary. The local library serves the purpose of having components, which are involved in a particular reliability model. The components for a particular mission can be extracted either from the Global Library or the Local Library. The location of the Global Library however is predefined by the environment pointer HIMA_GLOBAL_LIB which is set in the users' '.cshrc' file.

Building A New Library

When Library → New is activated the user is prompted to make one of the following choices. This function is useful when the user wishes to create a new library.

Figure 3.12: New Library Dialog

Creation of Libraries :

Once the name of the local library is given a file is created in memory and is open for editing. To edit the library the user has to activate Library → Edit. This action causes the popping up of another window.
This window gives user the information about the name of the library, the location of the library and also the number of elements present in the library. The scrolled list present in the window gives information about the components in the library.

- *New button :* The New button is the route the user has to take to add components to the library. This window prompts the user to give inputs for the name of the component, the version of the component if a component of the same name already exists. The user then has to specify the parameters (failure rate and/or repair rate) for the component. Five choices are specified for the failure rate:

- Fixed Probability Value
- Probability Value from another module
- Constant Failure Rate
- WEIBULL1 failure rate
- WEIBULL2 failure rate.

The choices pop up a window, which has the fields for the corresponding parameters. The user should click on "OK" to accept the values of the fields. The "Cancel" button is provided to kill the window.

- *Edit button :* The edit button is used to modify the parameters of an existing element in the library file. The user highlights the element in the scrolled list and clicks on the edit button. This will bring the same window as the one created by new button. Delete Button: This is used to remove an existing element from the library. The user first has to highlight the element in the scrolled list and then click on the button and the element is removed.
- *Save Button :* This button saves the changes made to the library file into memory.
- *Close Button :* This button kills the window.
- *Import component(s) from another library :* Provision is also given to the user to add elements from an existing library. The user clicks on the Import component(s) from another library button. This pops up a window which lists all the global libraries present in the directory specified by the user in the '.cshrc' file. The user now has to choose the library from which the elements are to be transferred to the local library. Once this is done another window is popped up with the following details. This window the Library combination window popup helps the user transfer elements from another Library list on the left part of the window to the Local Library on the right part of the window.
- *The transfer button :* When the user clicks on the element and then clicks on this button, that element is transferred to the Local Library list from the Global Library list.
- *The View button :* This button helps the user get information about the element of his choice. The user has to first click on the element and then click on the View button.
- *The Select All button :* If the system demands the transferring of all the elements from the global list to the Local list the user can use this button for that purpose.
- *The Deselect All button :* This is useful when the user has made a mistake and wants to restart.
- *The Delete button :* Going on to the local library side, the delete button is useful when the user wishes to delete some of the elements transferred from the Global Library.
- *The View button :* This button is used to get information about the elements transferred. This works in the same way as the View button on the Global Library side.
- *The Close button :* This window is used for closing the window after the transfer.
- *The Search Function :* HIMAP also provides for a Search Window. This window is useful when there a number of elements in the Global List. The user can use this window to reach a particular element fast.

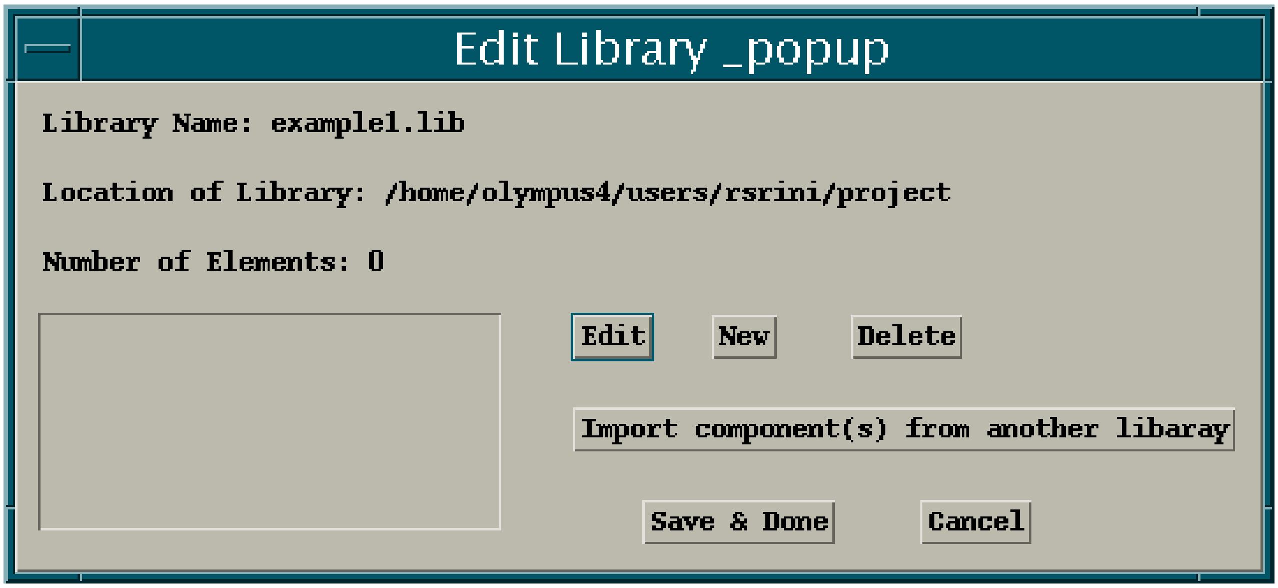

Figure 3.13: Himap Library Editor

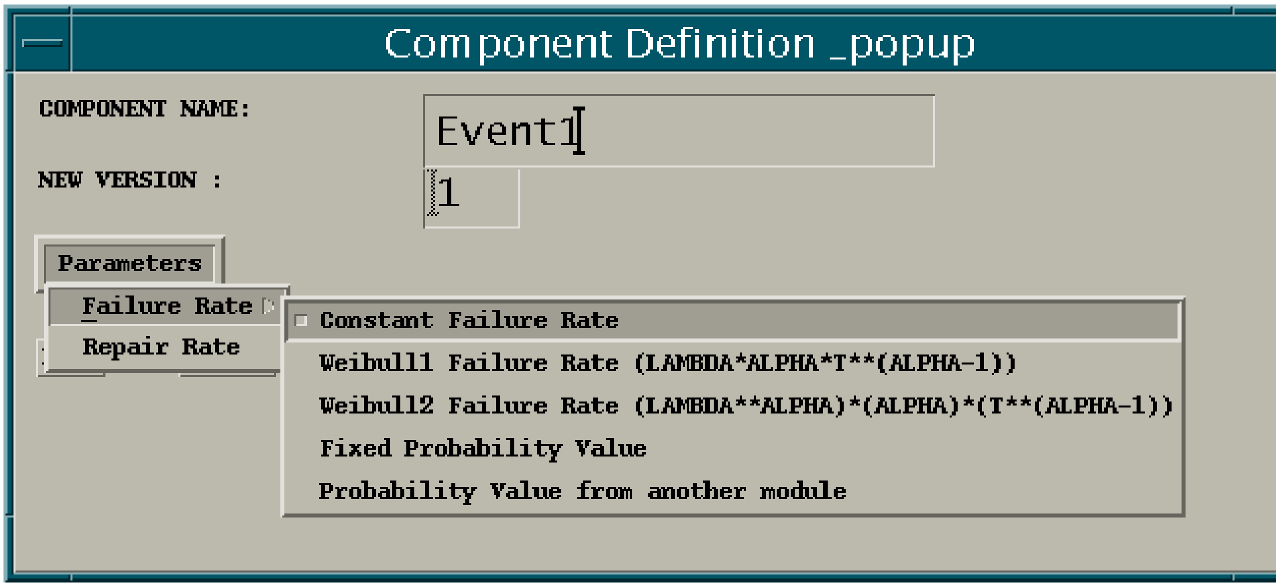

Figure 3.14: Component Definition

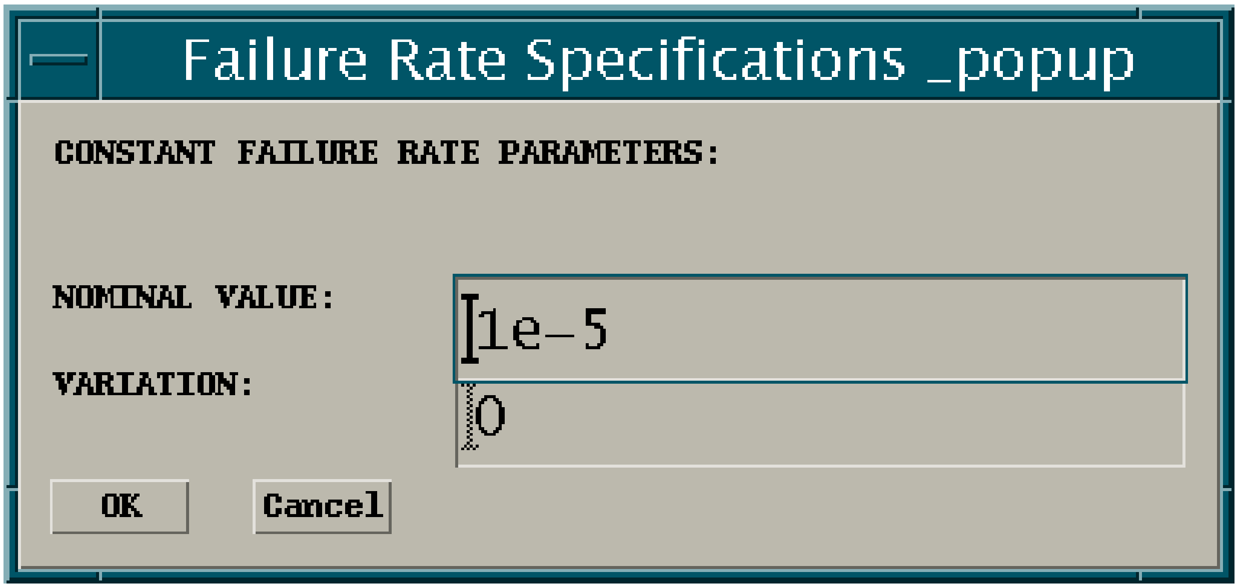

Figure 3.15: Failure Rate Specification

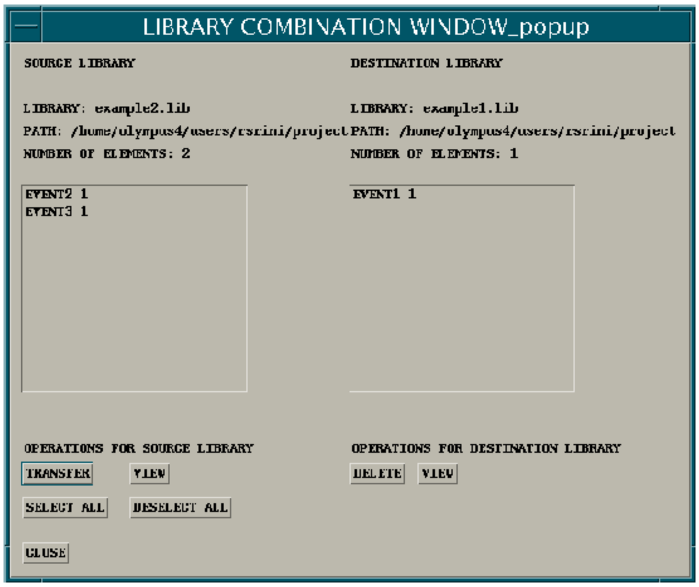

Figure 3.16: Library Combination Dialog

Change Library

Upon activation of File → Open .lib a window pops up with a list of all files with the extension '.lib' in the current working directory. The user can now make a selection to change from the present library to one of the libraries listed. The user can kill this window using the "Cancel" button.

Edit Library

When the user wants to further edit one specific library, the library has to be opened before the activation of Library Edit. The method of addition is exactly the same as the one described in the "Build New Library".

3.3.3 Associate Components

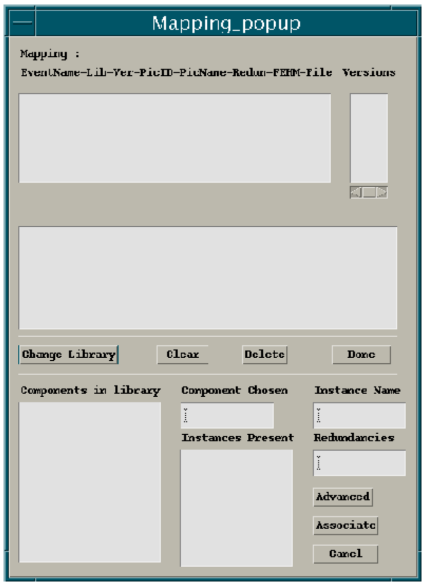

Figure 3.17: Associate Components Dialog

A .mod file serves as the link between the pic file and the lib file. In a picture file, there is no component information. It is just a pictorial representation of the system. In library file, it is just a list of events including the pertinent component information like the failure rate, repair rate and FEHM models. Mode file is the one, which acts as a bridge between the picture file and library list file. The events, which will be used in one model, are extracted from the local library file and stored in a mode file. Mode file is also responsible for component mapping from the picture file into the elements.

NOTE: It is critical that the model and picture files have the same name.

Mapping of a model(.mod) file to a picture file : With the necessary elements in the mode file, the users are now ready to map the elements in the mode file to the basic events in the picture file. The mapping procedure is used to map the basic events and special events. The basic events are

represented as circles on the picture file and the IN gate and the M/N gate are the special events. Activation of File Utils → Edit Mapping brings up the Mapping window. Details about mapping are given in the following subsections.
The Mapping popup window has a scrolled list describing the:

EventName : The EventNames are the names given to the event at the time of creation of the library by the user.

Version : The Version field here tells the user the version of the element the user wants in the fault description.

PicID : PicID refers to the id of the event in the picture.

PicName : PicName refers to the picture from which the events are being mapped.

Redun : Redundancies here refers to the number of redundancies of the event in the picture file.

FEHM : This field stores the information of the type of FEHM model the user wishes to associate to the event.

File : The File name field represents the name of the FEHM file.

Below the list of the events in the library, the window has the information about:

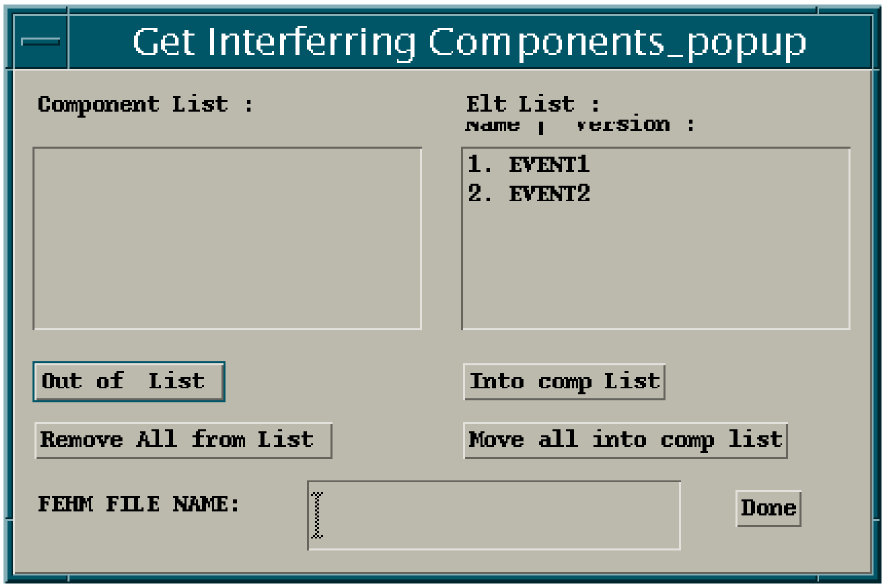

Figure 3.18: Interferring Components Dialog

The FEHM Models : The various FEHM models that HIMAP can handle are given next.

The user can click on any one of these models if associating a FEHM model is necessary. When the user clicks on any of the FEHM model desired, user is asked to give the information about the interfering components. The 'Interfering Components' are the list of components which might interfere with the fault recovery of this component.

FEHM File Name : The next field is used if a FEHM model is to be associated. This field is used to give the name of the FEHM file name. If there is a FEHM file name present and the file does not exist then HIMAP appropriately informs the user by popping up a window. In this example, we used 'aries' FEHM model and a new FEHM file 'aries.ari'. This window has three options:

CREATE a new FEHM File : This enables the user to create the FEHM file. HIMAP pops up windows and the user has to enter the parameters needed to create the file. Since, we selected 'ARIES" FEHM model, the popup window provides all the necessary input parameters for this FEHM model.

Since, number of recovery phases is listed as '3', then when user chooses 'MORE', another window is displayed asking for the recovery effectiveness probability in each of these three phases. This window is shown below with default parameters.

DEFAULT FEHM File : This option creates a default FEHM file with the default options which are popped up. Though the default options are set the user can minimize the effort needed to create a new FEHM file by choosing this option and changing only a minimum number of parameters.

CANCEL : This button is used to kill the window.

For more detailed understanding of FEHM models, please refer to the HARP manual within IRAP toolkit manual.

Rest of the mapping window has information for:

Redundancies : If the event in the picture has more than one redundancy then the user can use this window to give the value of the redundancy.

Continuous Repair Info : The information regarding the repair of the event is desired here. The 3-options are **Always, Selective** where a window pops up asking user about the start and end redundancies and **No Repair**.

Selective repair provides an option as to when the repairs are to start and at which point repairs need not be performed anymore. For example, if there are three CPU's in a machine, then one can provide the selective repair strategy such that repairs are only done when two CPU's fail. In that case, 'To start repair' filed will have redundancy value as two and 'To end repair' will have the redundancy value of three. Note that, these repairs make sense only when there are redundancies associated with a component.

User can also assign a Priority to the event. Greater the number, higher the priority. The priority is used in case of assigning repair men. As seen in Repair Groups Popup, the window is divided

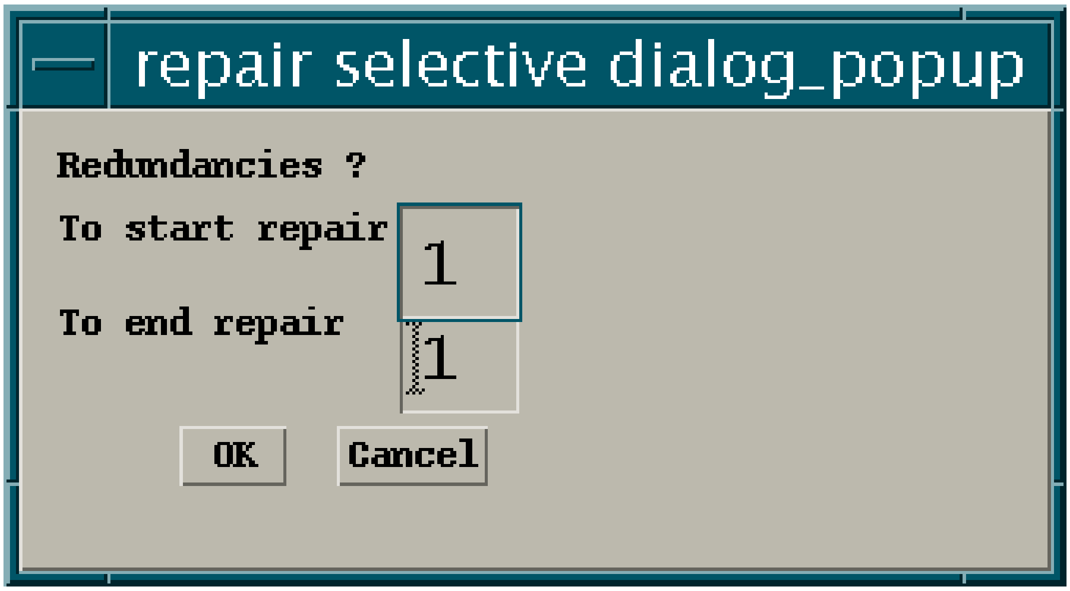

Figure 3.19: Repair Info

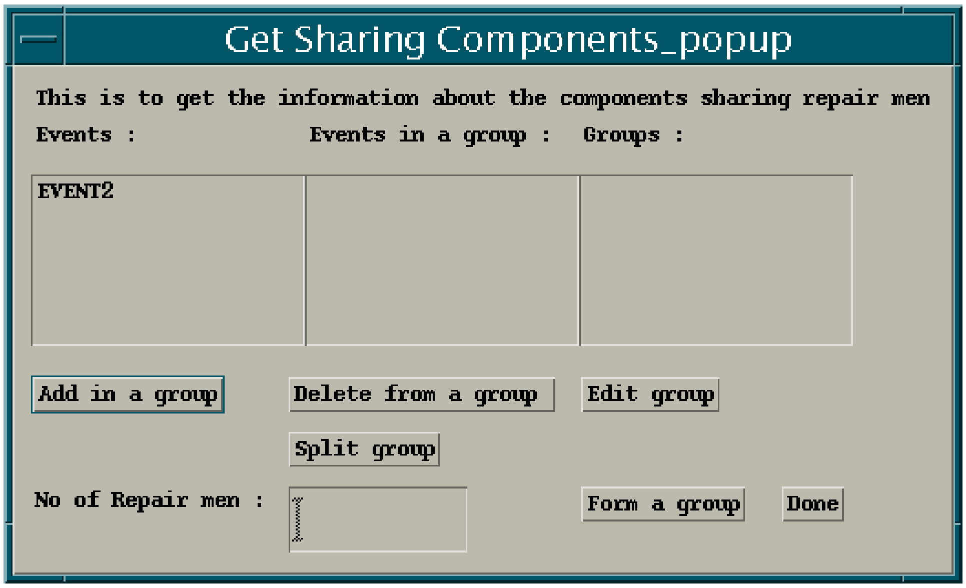

Figure 3.20: Sharing Components Dialog

into 3 compartments.

The list on the extreme right shows the total groups, the center compartment displays the events in a group. The leftmost compartment has the remaining events i.e. events which are not been grouped. When user clicks on Add in a group after clicking on an event of the leftmost compartment, the event is added to the current group i.e. the event is removed from the leftmost compartment and put it in center compartment. When user clicks on Delete from a group after clicking on an event in the center compartment, the event is removed from the center compartment and put it in the leftmost compartment. When Assoc. button is clicked, a new group is formed of all the events in the center compartment. Done button just closes the window. The user has to note that after the mapping is done the user has to save the map file using the save option described in the Management of MODEL files section.

The other scrolled list, shown at the bottom of the mapping window, is to show the user what special event has been mapped from the picture file. The scrolled list gives the following information: PicID, PicName, the type of special event, and the associated detail which is the M/N value for the M/N gate and the name of the inserted pic file in the case of an IN(transfer) gate.

The procedure for mapping to create the mod file is as follows:

Mapping basic events

- Click on the event or special event(in the drawing area) to be mapped
- Click on the appropriate event(in the mapping window) which is displayed in the list
- Choose the version of the element if there are multiple versions(DEFAULT: 1)
- Choose a FEHM model if needed(DEFAULT: no model) - If a FEHM model is chosen then give the FEHM file name
- Give the number of redundancies the event has in the picture (DEFAULT: 1)
- Click on DONE after associating all the events and special events

Mapping special gates

- Click on the special event; such as an IN/Transfer gate or a M/N(combination) gate.
- In the window which pops up, give the M/N value if the special event is a M/N.
- If an IN gate needs to be mapped then user needs to give the following information:
  - The name of the model being imported.
  - The instance name for the model.
  - Whether the gate is shared (useful when the same instance is imported at different points in the system.

The right column in the IN gate mapping window gives the names of the existing instances of the model which being imported in the present model.

The three buttons which control the mapping are (shown at the bottom of the mapping window):

ASSOC. Button : After the user is done with associating each event, the user should click on the Assoc.

Clear Button : This function is useful when the user wishes to change a mapping. The procedure to use this function is as follows: Click on the element in the element list which is to be cleared and then click on the Clear button. This option works for both ordinary events as well as special events.

Done Button : After the mapping is done, this function serves the purpose of closing the window. If there is an event requiring repairs, a window pops up when Done button is clicked asking user information regarding Groups of events sharing repair men and number of repairs.

Open of an existing model file : Activation of File Open .mod helps the user open an already existing map file.

NOTE: Only one mode file can be opened at a time.

Save and Save As a different name : File → Save .mod helps the user transfer the mode file in memory to disk. File → Save As .mod helps the user save the open mode file in a different name

Closing of a model file : Used for closing an open mode file if the file has been saved. If not HIMAP gives the user the option to saving the file before closing.

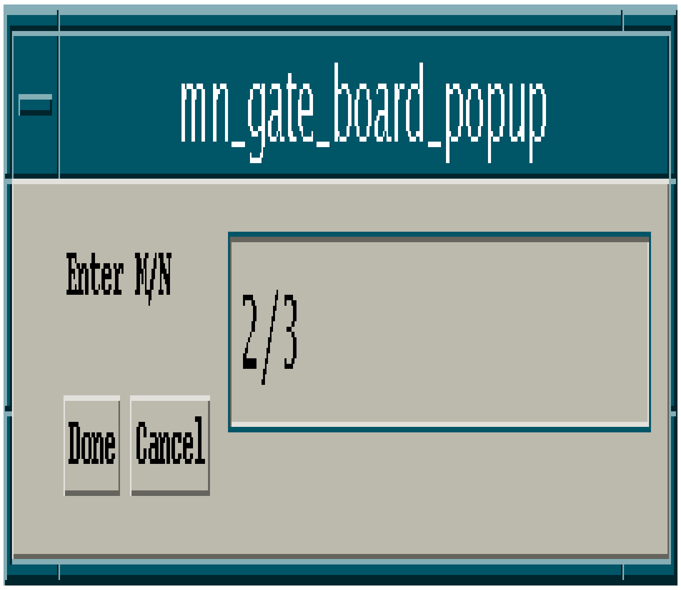

Figure 3.21: K/N Mapping

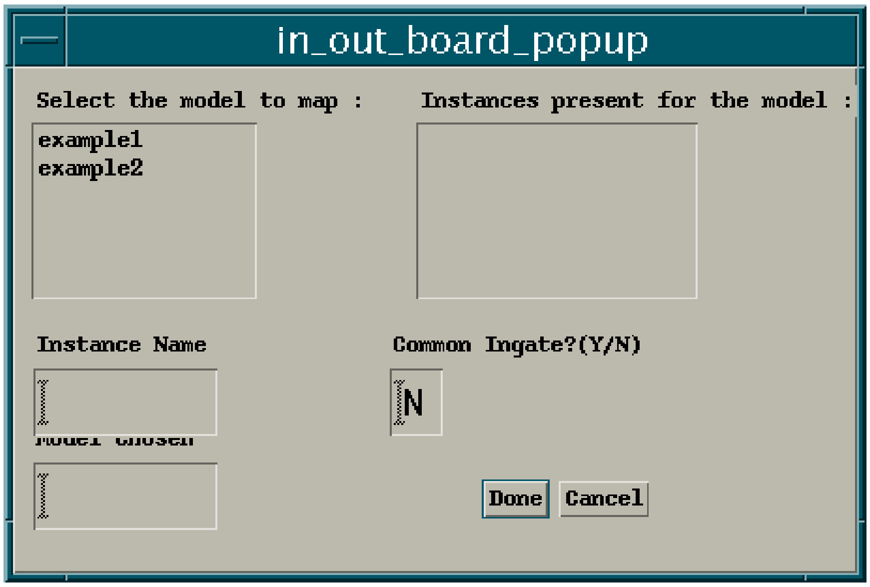

Figure 3.22: INGate Mapping

Chapter 4

HIMAP Layout - Markov Chain Editor

4.1 HIMAP Layout (Markov Chain Editor)

This chapter covers the details of how to create and manage Markov Chain Files. When activating Utils → Change Drawing Mode → Markov Chain, the mode is changed to Markov Drawing Mode. To open the actual drawing area the users need to activate Model → New to create a new image file. What follows now is the brief description of the buttons used in this mode.

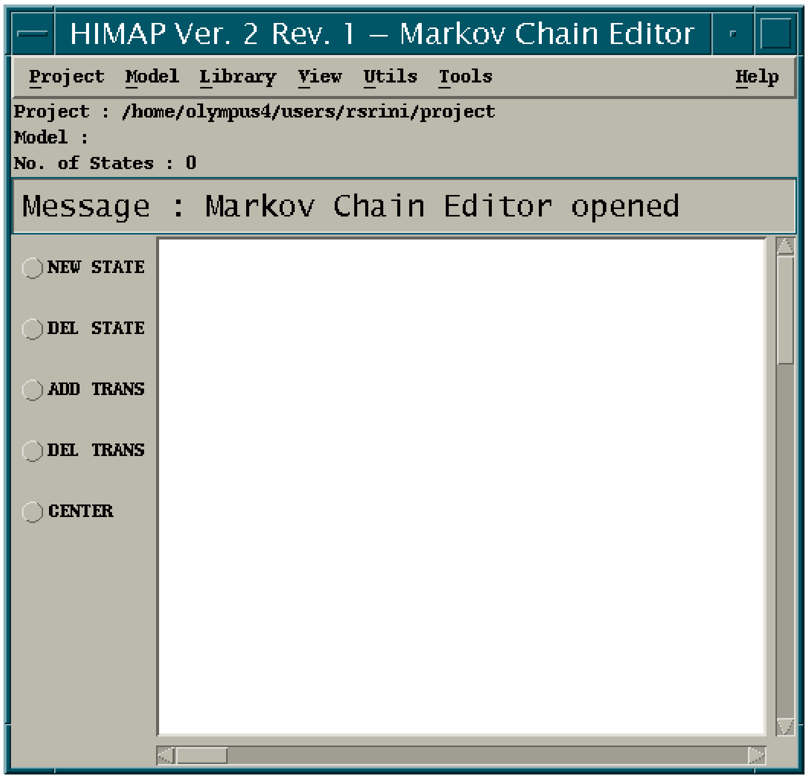

Figure 4.1: HIMAP Markov Chain Editor

4.1.1 Creating and Managing Markov Chain files

The symbolic textual Markov Chain description file (.MAT) is created by GMC (Generate Markov Chain). HIMAP is capable of reading this .MAT file and drawing the Markov Chain accordingly. It also supports drawing a Markov Chain on its own. As number of states in a Markov Chain can exceed a limit beyond which a drawing is not that clear, two types of drawings are supported by HIMAP. One which gives the overall view of the Markov Chain, the other where only one state is shown with its input and output arcs. Any change in one drawing is reflected on the other. Main File menu in the Markov chain input mode is shown below.
The following controls help user to draw and edit Markov chains.

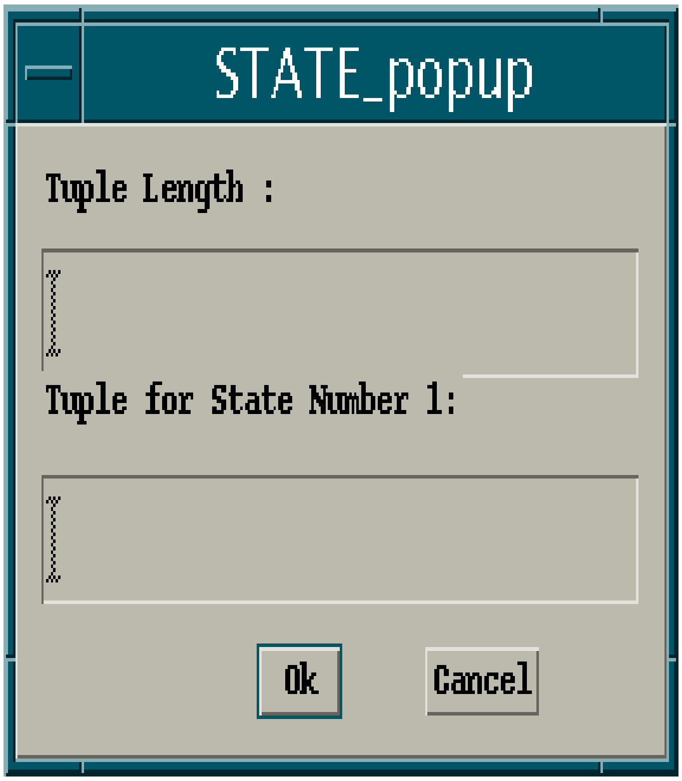

Figure 4.2: State Dialog

- *Open :* Model → Open when clicked will search the current directory for all the files with extension .MAT and display them in alphabetical order for the user to choose one of them. The user can choose any file from the list by clicking at any of the files listed.
  NOTE: Markov Chain Editor does not allow the opening of more than one file.
- *Create a New file :* Model → New when clicked asks user for a file name. Given a file name, this creates a file .MAT (name in CAPITAL letters) in the current working directory. Also it asks user for the Number of Components (tuple length) this system will model and the initial state tuple value. For example, if you are trying to model the system failure of a box which contains two CPU's, then the tuple length is two corresponding to the two CPU's and the initial tuple value if 1, 1 as both CPU's are initially assumed to be operational. So, in general, tuple length is the number of component (N) in your system and the initial tuple is list of N

1's each separated by a comma and a space. The drawing will show the first state with all the components working.

- *Save :* Model → Save will save the current file.
- *Save As :* Model → Save As, when clicked will ask the user for the name of the file in which he/she wants to save the file. If the file already exists, it will let the user have an option of cancelling the command or destroying the previous file.
- *Close :* File → Close .pic will close the file and clear the drawing area. But if the user has made any modification in the file, it will give user an option of saving the file before closing.

4.2 Editing Markov Chains - Global View

The following buttons help user draw a Markov Chain:

- *The STATE Button :*

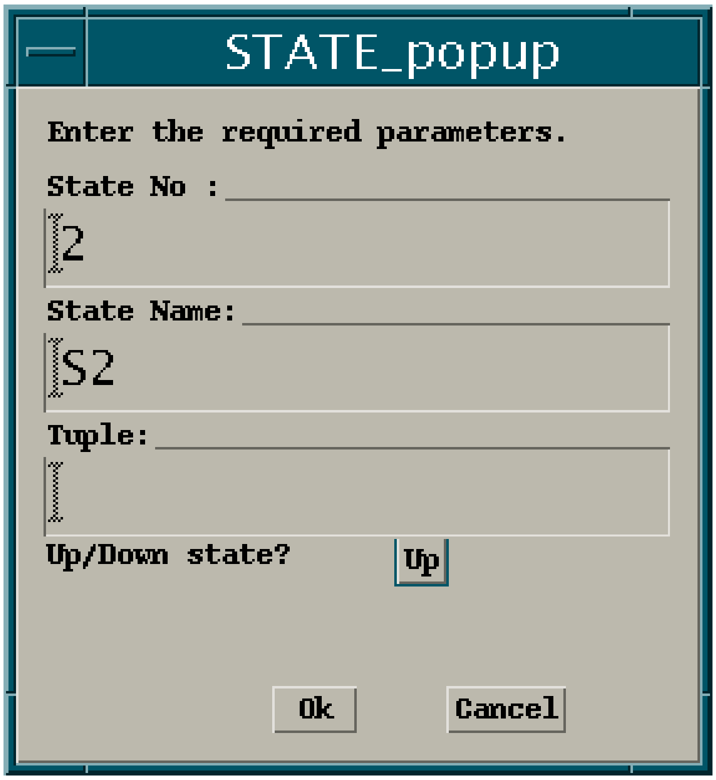

Figure 4.3: New State Dialog

When a user clicks on the drawing area after clicking on the STATE button, a window is displayed asking user for the State number, State Tuple and the State Label. If the State Number or the State Tuple already exists, the program returns with an error. Otherwise a circle with the state number is drawn on the drawing area. Note the State Number should be in sequence.

- *The DEL STATE Button :* When clicked on any state in the Markov Chain drawing, a window pops up letting the user know the IN and Out states of the state user wants to delete, also about the hanging states (a state with no input arcs) this deletion might cause. User has an option of either continuing with the command or canceling it. If there are any hanging states, they are drawn as though they are also the initial states.

- *The ADD TRANS Button :*

Figure 4.4: Transition Rate

The button is used to add an arc from a state to another state. When clicked on two states, a directed arc is drawn with first state being an output state and second clicked state being an input state. Rate of this transition is requested from the user. It can only be a double value. No symbolic values are allowed.

- *The DEL TRANS Button :* When clicked on two states, an arc is deleted if there is an arc from the first state to the second state. If this action is leaving any state hanging, the user is given a chance to cancel the command.

- *The CENTER Button :* When clicked on a state, the state is displayed with all input and output arcs on the other drawing area. In this example, detailed information about state number eight is shown.

The following buttons help user draw a Markov Chain on the drawing area which displays only one state (called Central State) with its input and output states:

- *The INFO Button :* This button is the most important in this find of drawing. When clicked, a window pops up which has a scrolling window showing the state numbers and the tuples associated with each state. Also there are four text windows for entering State Number, State Name i.e. Label, Tuple and Rate. These will become clear as we proceed with the other commands.

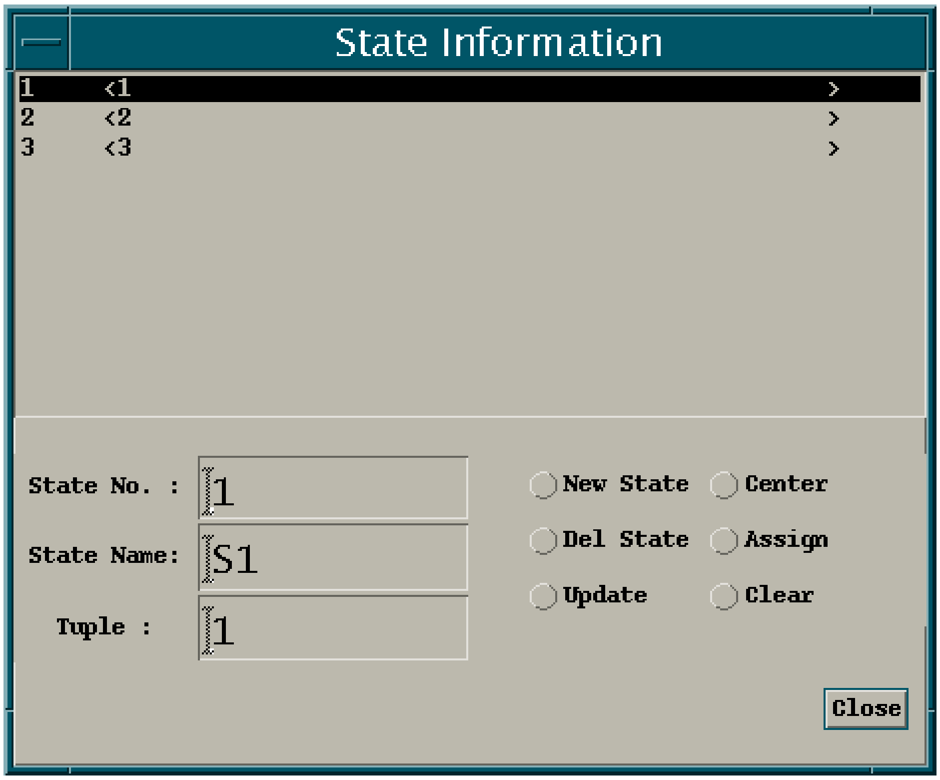

Figure 4.5: State Information

- *The Center Button :* The highlighted state in the scrolling window will be displayed with its input arc and output arcs when clicked on this button.

- *The Update Button :* This button is used to add states, also to change any other field associated with the state i.e. if user wants to change the tuple or the label, he/she can put the values in the text windows and click on the button. If the state number does not exist, a state is created with the given tuple and the label. A test is done to see if the tuple already exists or if the state number given is in sequence.

- *The Assign Button :* This button has two functions. The first function will become clear while explaining the drawing area buttons. The other function is to assign a rate or change the rate to any arc in the drawing shown in the State view. The Clear Button When clicked, the text windows are cleared.

- *The Clear Button :* Clears the tuple.

- *The Del State Button :* The highlighted state in the scrolling window is deleted when clicked on the button. Also the states are renumbered.

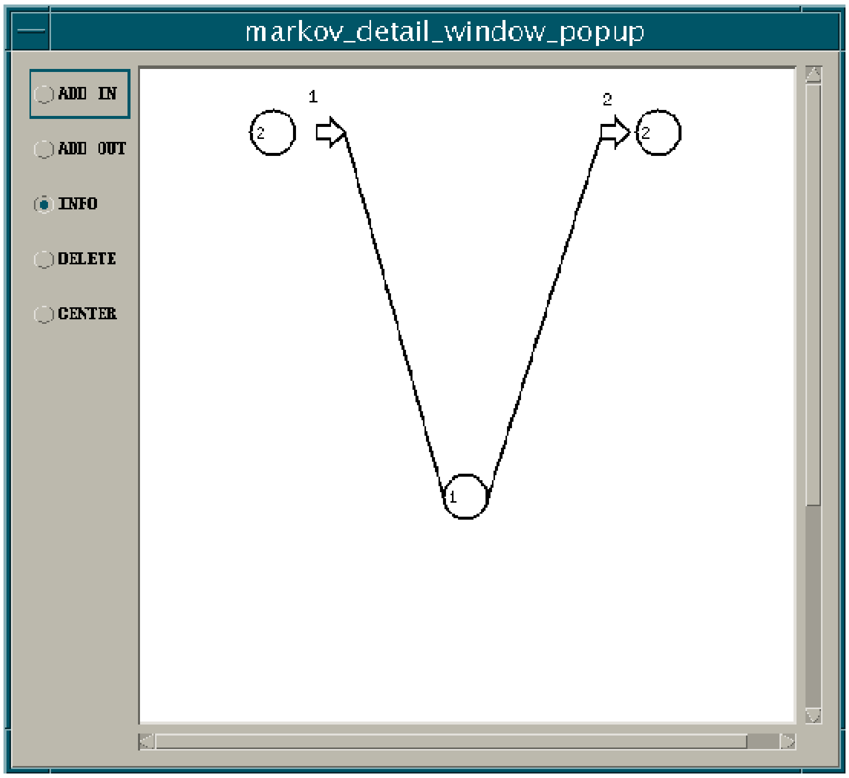

Figure 4.6: Center View

- *The ADD IN Button :* When his button is clicked, an input arc to the central state is drawn (dotted). In order to assign state number, tuple, label and rate to this arc, Assign button of Info window is used. User needs to fill up the text windows , click on the Info button, click on the circle and then click on the Assign button.

- *The ADD OUT Button :* The function of this button is the same as the ADD IN button except that an output arc to the central state is drawn when clicked on this button.

- *The DELETE Button :* The user can delete an arc as well as a state using this button. When the user clicks on the circle after clicking on this button, the state is deleted and when user clicks on the arrow, the arc is deleted.

- *The CENTER Button :* The user can make any state a central state by clicking on the state after clicking on this button.