

Development of MARS Transient Analyzer

M.K. HWANG, K.D. KIM, J.-J. JEONG, Y.J.LEE, B.D. CHUNG

Korea Atomic Energy Research Institute
Dukjin 150, Yusong
305-353, Taejeon, Korea

Key words: System Codes, MARS, GUI, Nuclear Plant Analyzer (NPA)

Abstract

A visual environment for system analysis codes (hereinafter called "ViSA") has been developed to support code users in their input preparations, code executions, and output interpretations. ViSA provides a more convenient way for base input data generation and modification on a user-friendly basis. It also provides on-line graphical displays to give an in-depth understanding of transient thermal-hydraulic behaviors in nuclear power plants. This paper presents the main features of ViSA.

Introduction

In the past two decades, best-estimate thermal-hydraulic system codes, such as RELAP5, TRAC-PF1, CATHARE, and RETRAN (Paulsen, 1998), have been developed to capture the major physical phenomena during a wide range of incident and accident conditions in nuclear power plants. Their application, however, has been limited to only a few groups of specialists due to the requirement of high computer performance and much expertise. Rapid advances in computer technology enable the code users to run the codes on a personal computer instead of main frame computers. However, it is still difficult to use the best-estimate codes because of the complicated I/O structure. This has motivated the development of easy-to-use graphical user interface (GUI) tools for the best-estimate codes, such as SNAP (Jones, 2000) and PEGASYS (Agee, 1996). Using these GUI tools, users can utilize the system codes like conventional NPAs (Bartsoen, 1997; Maselj, 1997).

MARS code, developed in KAERI for the analysis of nuclear power plant via 1-D/3-D nodalization, provides GUI functionalities in its current stage (Jeong, et.al 1999). The GUI has not been, however, considered to provide enough capabilities to be utilized as Nuclear Plant Analyzer by the variety of its user. The GUI therefore has been extensively revised in a similar manner as is performed for RETRAN code (Kim, 2000). The primary concern of the development was to provide valuable aid to all level of MARS users, ranging from plant engineers to experts, in their input preparations, code executions, and output interpretations. The key features of the tool can be summarized as follows:

- The MARS input data can be interactively modified/generated. Because current MARS input features are maintained, the existing input data can be fully utilized.
- Visual displays can be easily created by drag-and-drop of elementary graphic display units, such as digital meters, level gauges, dial meters, LED indicators, X-Y plots, etc.
- The MARS code execution can be interactively controlled through pre-defined menu.
- It can run by itself without any additional graphics software on personal computers.

This paper presents the main features of ViSA. It is noted that descriptions in this paper are limited to the MARS application of ViSA.

ViSA Structure and Programming

As a target platform for ViSA, personal computer (Windows 95, 98, 2000, and NT) was selected because of its high accessibility and portability. For programming the ViSA package, a component-based methodology was adopted. Each component was created in modular type with well-defined interface and then the developed components were simply assembled later. This approach could reduce the rare programming parts and the complexity of the ViSA design. Because each component is independent with each other and encapsulated, it could save the time for the further software development and maintenance.

DELPHI 4 (Inprise Corp. 1998) was utilized for the ViSA development. DELPHI is an object-oriented visual programming environment for Pascal language, which is very efficient for component-based programming. The multi-thread technique was also used for CPU sharing between MARS and graphic software. The technique ensures prompt response in user's request.

Figure 1 shows a schematic of the ViSA package. It consists of three functional modules; (i) pre-processor for interactive input generation, (ii) MARS execution control, and (iii) visual analyzer for output processing. Each of the components can be used either in a standalone fashion or as an integrated form of the ViSA package. ViSA loads the MARS code as a dynamic link library (DLL) only when it is necessary (DEC, 1999). ViSA takes two sets of basic input data: a normal MARS input deck and a graphic file containing plant mimics and a nodalization diagram produced by any drawing software. The latter is not obligatory and is used for dynamic display of plant operation status.

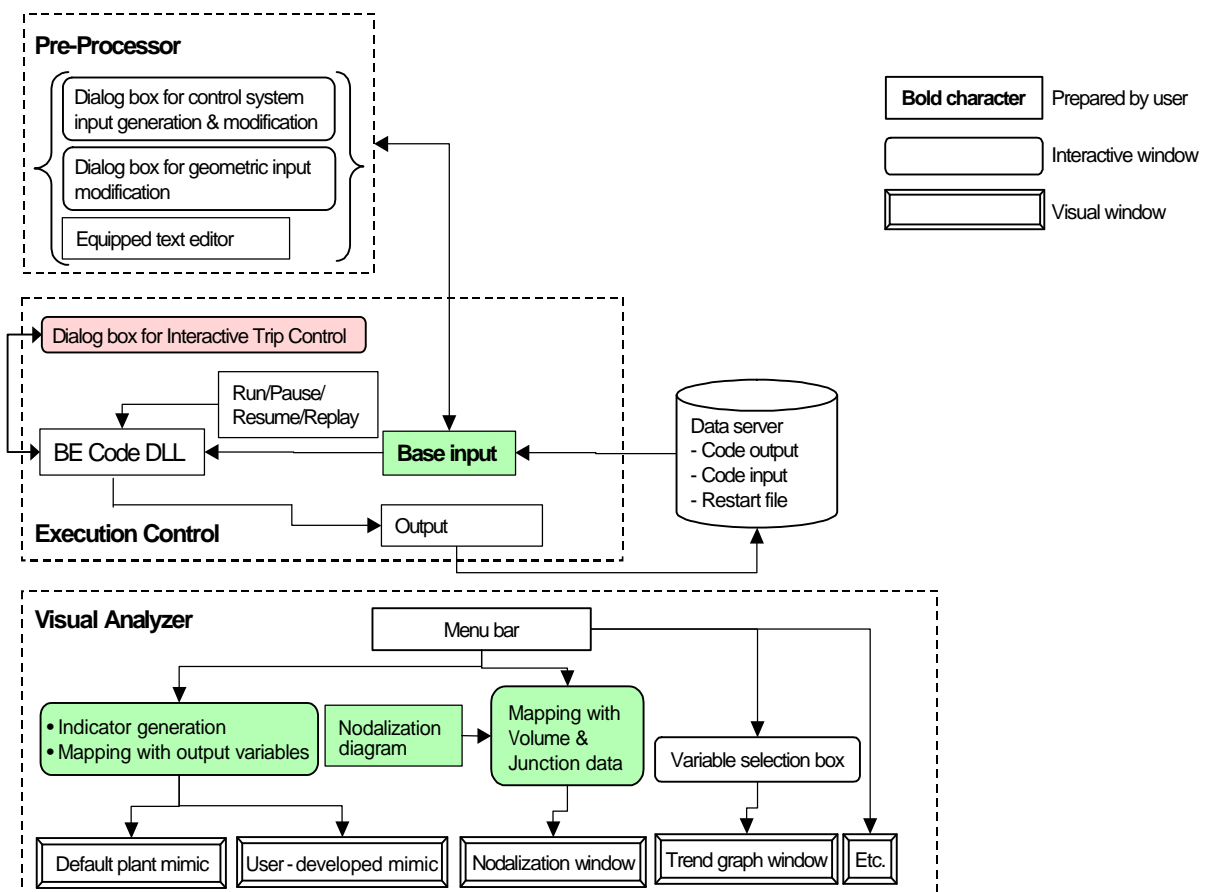


Fig. 1. ViSA package structure.

Pre-Processor

The pre-processor was designed to help the users to conveniently modify the user-prepared MARS

input data or create the control system input data. It consists of (i) control system card edit/creation windows, (ii) volume and junction input card edit windows, and (iii) a text editor for other input card editing.

Control System Edit/Create Windows

The functions of the control system edit window are listed below:

- It sorts the control system input cards in the user-prepared MARS input deck according to their functions. The users can name each functional group, which is then displayed in the top left of the control system edit window (See Fig. 2).
- When clicking one of them, a functional diagram of the selected card group is displayed as shown in Fig. 2.
- When clicking any control component shown in the functional diagram, its input properties are given in a dialog box in the bottom left. The input data can be modified through the dialog box.

The control system creation window enables the users to create a new control system input. As can be seen in the top left of Fig. 3, this window provides all the elementary control components available in MARS and, by drag-and-drop of the components, the users can create a control system, of which component properties are supplied through a dialog box in the bottom-left of the window.

The output of the control system windows is text-type MARS input data, which is automatically appended (i.e. overwritten) to the existing input data.

Manual Trip Window

This window allows for the user to manually enable or disable the trip status during the execution stage. When the manual trip window is initially invoked, the trip control is in 'auto' mode, where the trip logic provided in the input deck governs the trip status. User can, however, optionally select any of the trips displayed then set to 'Manual' mode. When set to manual mode, the corresponding trip logic in the input data is not effective anymore. The trip status is simply set by 'Trip' button in the manual trip window. User can thereby conveniently examine the system transient behavior during

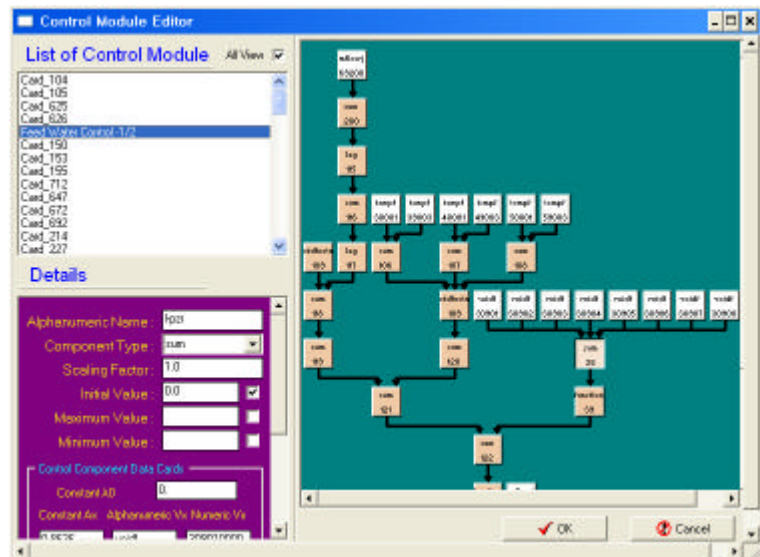


Fig. 2. Control system input modification

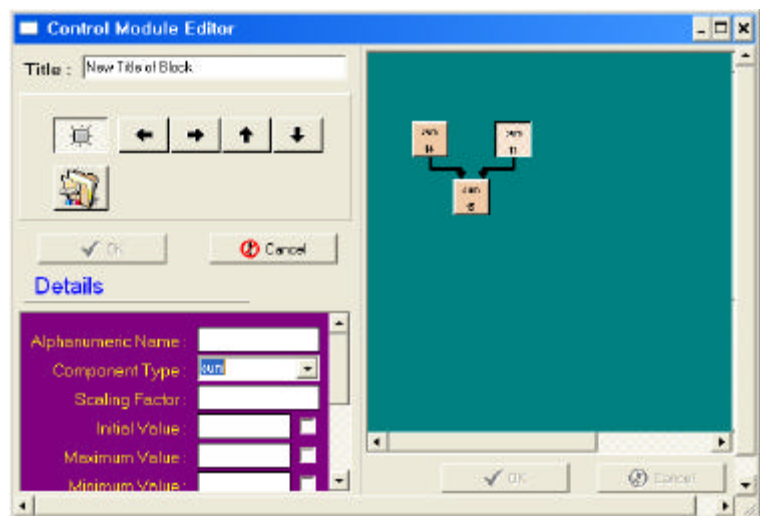


Fig. 3. Control Input Generation

the execution of the transient analyzer, by enable or disable specific trip event.

Volume and Junction Card Edit Windows

The volume and junction input data can be interactively modified through the screen. To use this feature, a nodalization diagram corresponding to the input data should be prepared first by users (see Fig. 5). Then, each volume and junction in the nodalization diagram has to be mapped with those in the base input.

As shown in Fig. 5, the volumes and junctions in the input data are automatically listed in a dialog box. Each volume in the diagram can be mapped interactively by clicking the corresponding location in the nodalization diagram and selecting the appropriate volume index from the dialog box. Each junction can also be mapped in the same manner. After then, the users can modify the input cards for volumes or junctions by just clicking the appropriate components in the mapped nodalization window. When clicking a component, a dialog box is generated, which contains a list of the component data (volume or junction data). Through the dialog box, the users can change the input data for geometry, initial conditions, and others. The modified input data is appended to the end of the base input.

Manual Trip Control				
No.	Description	Auto/Manual	Trip	Set point
509	SI On	<input checked="" type="radio"/> Auto <input type="radio"/> Manual		30.00
565	RCP Trip	<input checked="" type="radio"/> Auto <input type="radio"/> Manual		30.00
582	MSIV Isolation	<input checked="" type="radio"/> Auto <input type="radio"/> Manual		0.00
586	ACC-1 Off	<input checked="" type="radio"/> Auto <input type="radio"/> Manual		0.10
588	ACC-2 Off	<input checked="" type="radio"/> Auto <input type="radio"/> Manual		0.10
590	ACC-3 Off	<input checked="" type="radio"/> Auto <input type="radio"/> Manual		0.10
599	30 cm LOCA	<input checked="" type="radio"/> Auto <input type="radio"/> Manual		1050.00

Fig. 4. Dialog box for Manual Trip

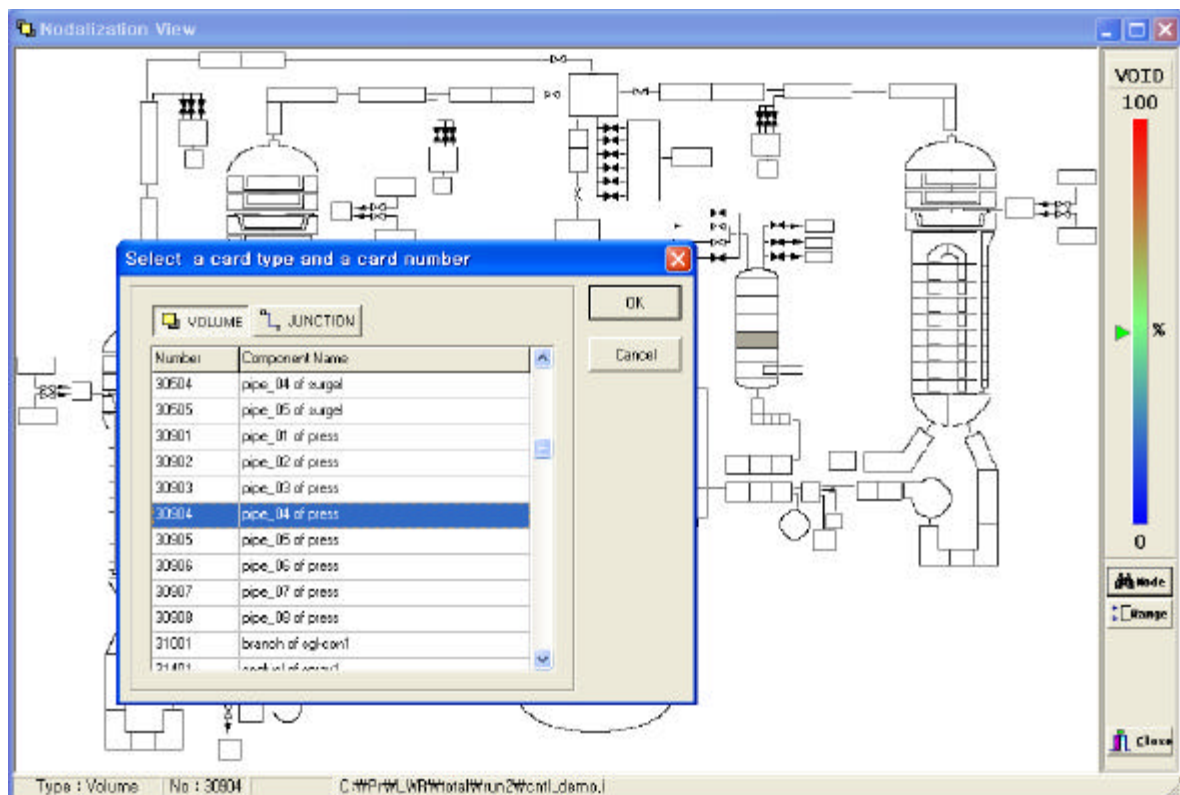


Fig. 5. Mapping of volumes and junctions.

Visual Plant Analyzer

MARS produces a large amount of text-based output after a transient simulation. From this kind of text-based output, it is not easy to obtain a deep insight into physical phenomena during the transient conditions. ViSA was designed to provide various graphical displays of the results during or after a transient simulation upon user's request so that the users can easily understand the plant dynamics.

To utilize the visual plant analyzer functions, users have to establish their-own visual windows. ViSA provides three types of graphical displays, i.e., (i) plant and panel mimic display, (ii) nodalization display, and (iii) trend curves. In addition, the trip information is always provided and the user-prepared plant information also can be displayed on the user's request.

Plant and Panel Mimic

The plant mimic window shows major parameters through some selected indicators as shown in Fig. 6. The background plant mimic image in Fig. 6 was generated using other drawing software and then imported in ViSA. This can be replaced with any other ones. Users can construct their-own indicators on the background image. Of course each indicator should be mapped with a corresponding MARS output variable.

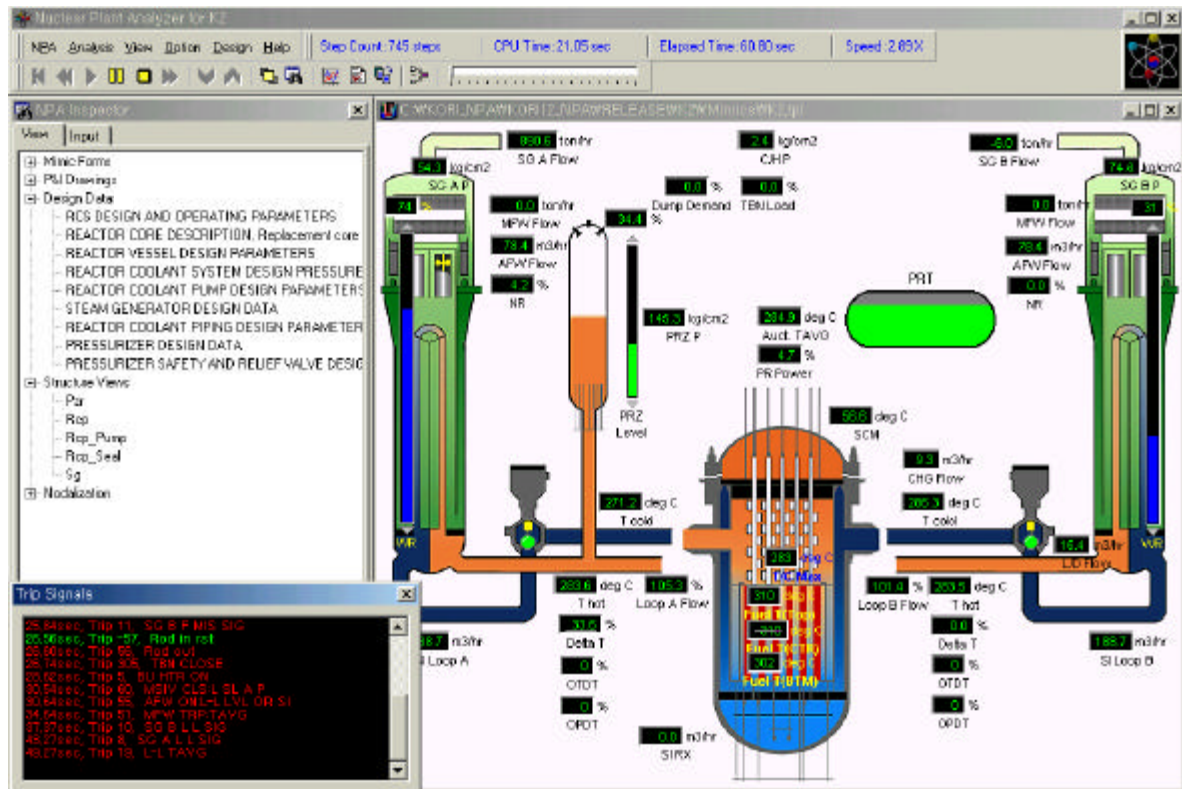


Fig. 6. Plant mimic window in ViSA (Main window).

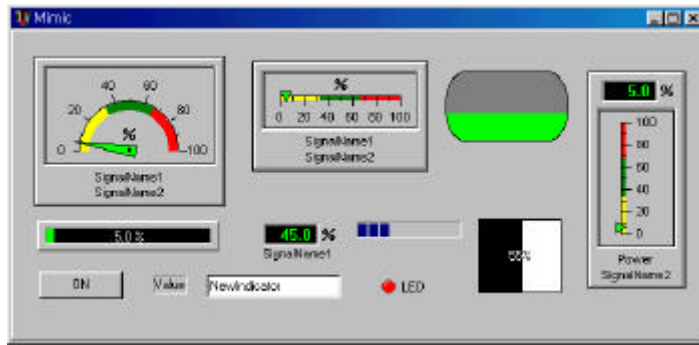


Fig. 7. Elementary indicators available in ViSA.

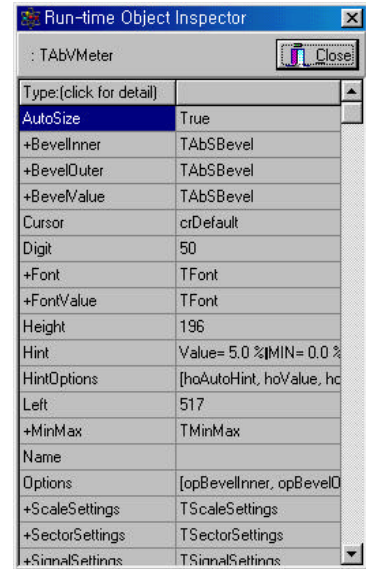


Fig. 8. Dialog box for the indicator property editing →

ViSA provides various types of indicators as shown in Fig. 7, such as digital meter, level gauge, dial meter, LED indicator, and etc. These can be added to the plant mimic by drag-and-drop operations. The properties of the indicators are specified using a dialog box called “run-time object inspector”, which is shown in Fig. 8. The indicators should be mapped one-by-one with specific output variables through “assign value” dialog box. This dialog box appears by clicking the right button of the mouse after selecting an indicator. As can be seen in Fig. 9, the dialog box lists major MARS output variables defined in the volume and junction data blocks and all minor edit variables. By clicking a variable in the dialog box, the variable is connected to the indicator.

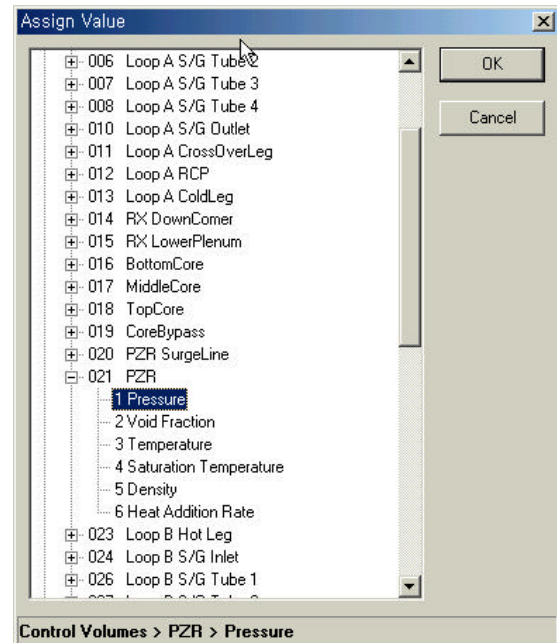


Fig. 9. “Assign value” dialog box.

The panel mimic is created to efficiently show the trip information, valve positions using LED indicators, flows, level, etc. through various meters (see Fig. 10). The panel mimic is established in the same way described for the plant mimic. Users can create additional mimics if they want.

Trip window

The trip window shows trip times and a description for each trip at the time of occurrence. This window is always given (see the bottom left of Fig. 6). The scroll bar makes it possible to examine previous trip occurrences during the transient. This feature is useful in tracking the causes of discontinuous changes in trend during the transient. For the user's convenience, two different colors are used for trip messages in order to distinguish between normal trip and reset trip.

Nodalization window

This window shows void and temperature distributions in a color spectrum (see Fig. 11). Users can see the exact value in the status bar by moving the cursor into the location.

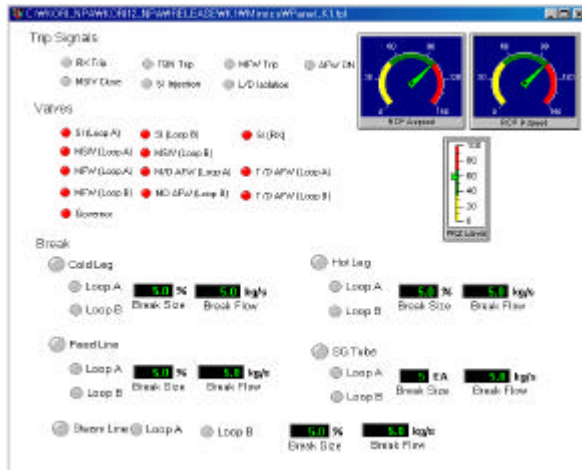


Fig. 10. An example of the panel window.

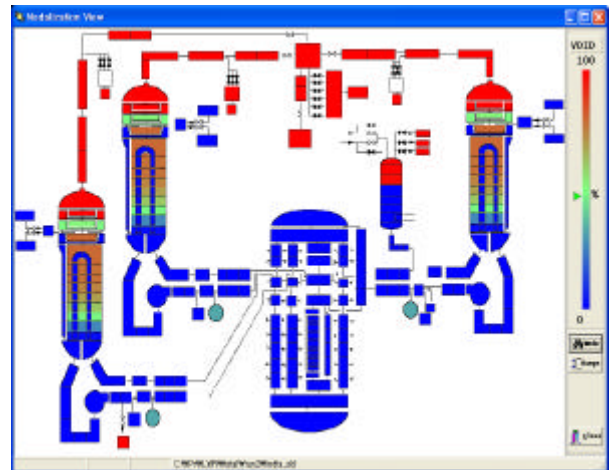


Fig. 11. An example of the nodalization window.

Trend windows

The trend window shows on-line X-Y graphs for user-selected variables. The user can select major volume and junction properties and minor edit variables through the dialog box (see Fig. 12). Multiple variables can be drawn in a graph window (see Fig. 13) and additional trend windows can be created by user's request.

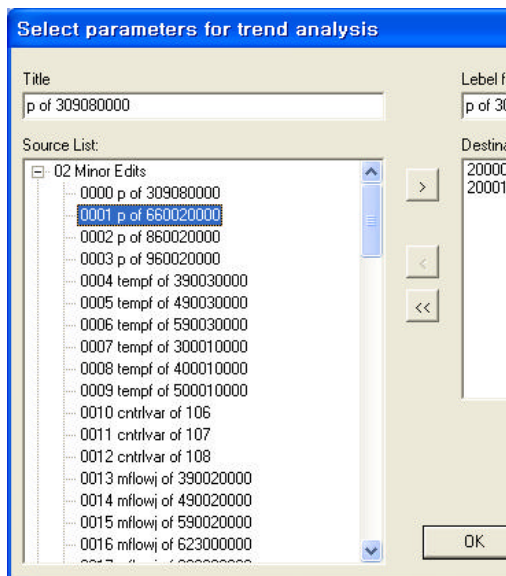


Fig. 12. Dialog Box for graph variable selection

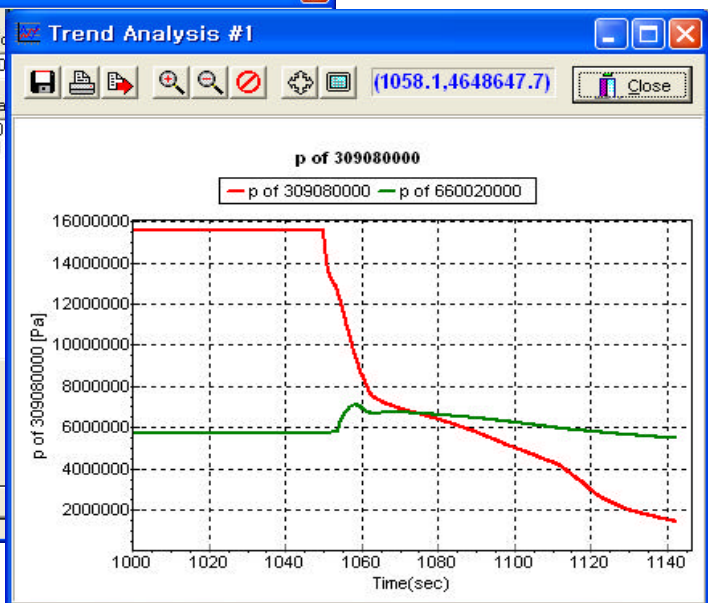


Fig. 13. An example of trend graph

MARS Execution Control

Users can start the MARS module, and pause / resume / terminate the execution by clicking the speed button in the menu bar on the top of Fig. 6. The buttons for pause / resume / termination are inactive before a problem starts. The pause and termination buttons becomes activated after a problem starts.

Since ViSA uses the MARS code without modification, a real-time calculation cannot be achieved

always. The replay function of ViSA provides reproduction of the pre-calculated results by selecting fast/slow/real time mode on the top window in Fig. 6. The animator for execution status indicates the running mode, i.e., whether the current results are from replay or MARS execution.

Other Features

Input modification history

Whenever users modify the input data, modification history is recorded, by commenting the original input lines, appending the modified input lines, and, finally, adding modification time in front of the modified input lines.

Permission for input modification

Since ViSA was developed for all level of MARS users, input modification may need to limit to some specific users for the maintenance of the input data set. ViSA provides the feature to check the authorization for input modification by comparing the password.

Concluding Remarks

To support the MARS code users in their input preparations, code executions, and output interpretations, a visual environment has been developed. The major advantages of ViSA can be summarized as follows:

- High accessibility, portability, and low cost of the target machine.
- Component-based design and general interface structure (easily portable to any other system codes or transient analysis codes).
- Easy-to-use interactive input preparation on a user-friendly basis.
- Various graphical representations of the output.
- Plant-independent features working with any MARS input and mimics.
- Integrated database for plant transient analysis.

The first application of ViSA was in Kori 1 & 2 nuclear power plants. ViSA could greatly improve the ease-to-use MARS and the easy-to-understand plant dynamics. The user's feedback is currently being taken into account for the further improvement of ViSA. Future developments will be focused on interactive control functions to simulate user's actions during transient simulations so that MARS can be used as a plant-independent NPA. In addition, ViSA will be generalized for other system codes and time-dependent analysis tools.

References

- Agee, L.J., 1996, "Status of EPRI Software," Presented at Korea Electric Power Corp., Electric Power Research Institute.
- Bartsoen, L., et al., 1997, "Nuclear Plant Analyzer: An Efficient Tool for Training and Operational Analyses", Proc. 2nd CSNI Specialist Meeting on Simulators and Plant Analyzers, Finland.
- DEC, 1999, Visual Fortran 6.0, Digital Equipment Corporation. Inprise Corp., 1998, "Borland Delphi Professional", Version 4.0.
- Jeong, J-J, Ha, K. S., Chung, B. D., Lee, W. J., "Development of A Multi-dimensional Thermal-Hydraulic System Code, MARS 1.3.1, "Annals of Nuclear Energy 26(18), 1161-1642(1999)
- Jones, K.R., 2000, "Symbolic Nuclear Analysis Package," Proc. 2000 ANS/ENS International Mtg, Embedded Topical Mtg. #2, "Best Estimate" Methods in Nuclear Installation Safety Analysis, Nov. 12-16, 2000. Washington, D.C.
- Kim, K.D., 2000, "Development of GUI for MARS," Proc. 5th MARS Users Group Meeting, Dec. 15, 2000, Korea Institute of Nuclear Safety, Taejeon, Korea.

Kim, K.D., et al., 2000, Development of a GUI Based RETRAN Running Environment for Kori NPP Units 1 & 2, KAERI/TR-1644/2000, KAERI.

Maselj, A., et al., 1997, "NPA Applications: Development in the Nuclear Safety Authority Framework", Proc. 2nd CSNI Specialist Meeting on Simulators and Plant Analyzers, Finland.

Microsoft Corp., 1999, Microsoft HELP Workshop.

Paulsen, M.P. et al., 1998, "RETRAN-3D: A Program for Transient Thermal-Hydraulic Analysis of Complex Fluid Flow Systems," EPRI NP-7450, Electric Power Research Institute.