Developments and New Directions for the RELAP5-3D Graphical User Interface

Dr. George Mesina
Idaho National Engineering and Environmental Laboratory
P. O. Box 1625
Idaho Falls, ID 83415-3880, USA
Phone: 208-526-8612, Fax: 208-526-6971
Email: azb@inel.gov

2001 RELAP5 International Users Seminar
Sun Valley, ID, USA, September 5-7, 2001 AD

ABSTRACT

The direction of development for the RELAP5 Graphical User Interfaces (RGUI) has been extended. In addition to existing plans for displaying all aspects of RELAP5 calculations, the plan now includes plans to display the calculations of a variety of codes including SCDAP, RETRAN and FLUENT. Recent work has included such extensions along with the previously planned and user-requested improvements and extensions. Visualization of heat-structures has been added. Adaptations were made for another computer program, SCDAP-3D, including plant core views. An input model builder for generating RELAP5-3D input files was partially implemented. All these are reported. Plans for future work are also summarized. These include an input processor that transfers steady-state conditions into an input file.

Keywords RELAP5, Graphical User Interface (GUI), 3D Visualization

1. Introduction

The purpose of a Graphical User Interface (GUI) is to provide a helpful, visual interface between a user and a computer program along with the data that it accesses and produces. In the nuclear field, the first GUI, called the Nuclear Plant Analyzer [1] was built at the Idaho National Engineering and Environmental Laboratory (INEEL) in 1986 to provide a graphical interface to RELAP5 [2]. To satisfy the ever-increasing need of the nuclear field and its analysts, simulator trainers, and plant designers, many other such interfaces have been built. These are summarized in Reference [3]. These new interfaces also take advantage of the ever-improving computer hardware and software available.

Among the newest interfaces in the nuclear field, the RELAP5-3D Graphical User Interface (RGUI) was created primarily to help analysts visualize the 3D data produced by RELAP5-3D [4]. Other purposes for its creation include: a common interface for working with RELAP5 that spans Unix and Windows; automatically-generated and accurate display of an input model; display of calculated data on that display image;

transient and replay control; and integration of numerous RELAP5-related tools. These are realized in the current RGUI product.

RELAP5-3D code users and ergonomics experts have strongly influenced the form RGUI has taken. The result has been a product that presents what nuclear plant analysts want in a very user-friendly, easy to use manner that can be used in many ways. Some of these include:

- Automatic creation of a nodalization diagram for an input file.
- Visual recognition of errors in an input file.
- Access to all minor edit quantities in an input file (and during transient and replay).
- Viewing points of interest in the plant image.
- Visualizing RELAP5 calculations via color maps, graphs, and numerical labels and tables.
- Stopping the calculation with a non-input, temporary trip condition to carefully examine an event of interest.
- Visually recognizing interesting and anomolous behavior during the transient.
- Creating graphics for inclusion in reports, papers, presentations, etc.

Because RGUI has proven to be quite useful for analysts and other code users, there have been many requests to apply the RGUI interface to other computer programs. Users at the Advanced Test Reactor (ATR) facility within the INEEL requested that RGUI be modified to work with RELAP5/MOD2. A similar request from the INEEL Severe Accident Core Damage staff resulted in an on-going project to connect RGUI with SCDAP-3D [5] and visualize a reactor vessel. The Electric Power Research Institute (EPRI) traded intellectual property with INEEL, receiving RGUI from INEEL for adaptation to RETRAN03 [6] in exchange for the Engineering Code Pre-Processor (ECPP) [7] for adaptation to generating RELAP5 input files. The INEEL Fusion Safety staff requested that RGUI be adapted to ATHENA [8]; they use it quite extensively.

These and other customer requests have resulted in an extension to the development plans for RGUI. The original plan encompassed visualization of all major aspects of a nuclear safety plant model and completion of an input model generator. In addition to this, the plan now calls for separability of RGUI from RELAP5-3D. When separated, RGUI will only replay RELAP5 calculations from restart-plot files. This capability will be further enhanced so that RGUI can replay transient calculations from the restart files of other programs such as COBRA [9], CONTAIN [10]], TRAC [11], and MELCOR [12]. Another extension is to add to the RGUI top level work area the ability to interface with a PVM master program, also called an executive, to visualize and control the solution of a complex problem that is undertaken by many different programs, such as RELAP5, CONTAIN and COBRA.

The latest developments in RGUI are discussed in Sections 2 and 3. The heat structures screen is presented in Section 2, and the reactor vessel screen in Section 3. Progress on ECPP is reported in Section 4. The extension of RGUI in new directions is presented in Sections 5 and 6. Section 5 covers other developments and the new directions for RGUI including separation of RGUI from RELAP5, adaptation to other computer programs via filters, and PVM plans.

2. Heat Structure Visualization

Previous versions of RGUI [13-17] had only 3 major screens. The first is a work area that gives both command-line and mouse/menu access; this gives both Unix- and Windows-oriented users an interface style they are comfortable with. This screen also provides a menu of tools such as a text editor, graphing package, and input file post-processor. The second major screen helps the user set up RELAP5-3D command lines visually. The third major screen shows a geometrically exact representation of the plant model and displays the hydrodynamic data in myriad ways. This screen has many auxiliary windows, such as point-and-click plots, trips set-up, and the detailed volume window, that show data in a concise and ergonomic manner.

Beginning with RGUI version 1.5, at least two new major screens will be available. They are the Heat Structure Screen and the Reactor Core Screen. Both screens can be accessed via hot-keys and menu items from the Isometric Image Screen. *Both produce images automatically and solely from the input file data*. At the present time, neither is an actual screen with its own independent frame, as are the three major screens described above. Whenever either one is accessed, the frame of the isometric image screen and its top level menu remain, but the tool bars and hydrodynamic plant image section are replaced by a different tool bar and an image of either the heat structures or the reactor vessel. In a future release, these two will be independent screens with their own frames. This will allow a user to view, for example, the hydrodynamics and heat structures simultaneously.

The primary heat structure display is based on the heat structure geometry. A heat slab, or heat structure, represents a one-dimensional segment of conducting material located somewhere in the plant, such as in a pipe wall, fuel rod cladding, or reactor vessel wall. The heat structure geometry represents one or more such related segments. Generally, heat structure geometries are comprised of contiguous heat structures, but this is not a requirement. There are two requirements placed on all heat structures in a heat structure geometry. Some discussion of the numerical analysis of the heat conduction equation is necessary to explain the requirements and to understand the visualization.

To calculate heat transfer through a conducting material with a computer, the governing heat conduction equations and the domain to which they apply must first be discretized. The one-dimensional segment represented by a heat slab is divided into intervals; the end points of the intervals are called mesh points. Discrete variables that approximate the temperature are defined at the mesh points. The heat conduction equation is discretized into a system of discrete equations in discrete unknowns. Its solution gives the temperature at the mesh points.

A heat structure is fine for situations where temperature varies in only one direction, but for cases where temperature varies in two directions, such as through the walls of a steam generator U-tube, one heat structure is insufficient. If several contiguous heat slabs were used to model the temperature, there would be no heat exchange between heat slabs because heat slabs are governed by the one-dimensional heat conduction equation. To model two-dimensional heat flow, a two-dimensional region with a true two-dimensional governing equation is needed. A heat structure geometry is a two-dimensional, rectangular region covered with a rectangular grid of mesh points as shown in Figure 1. The different shading patterns represent different materials.

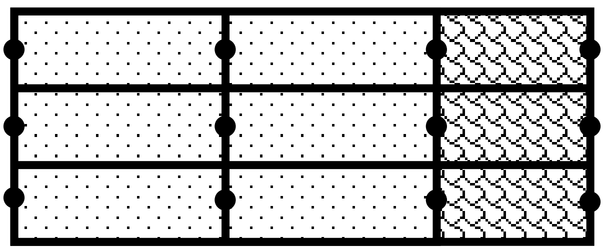

Figure 1. Heat structure geometry showing mesh and materials.

Now the requirements for a heat slab to be in a geometry can be stated. The first is that all slabs in a geometry must represents segments that have exactly the same material composition as every other slab. The second is that they all have the same number of mesh points and that corresponding slab mesh points have the same x-coordinate.

Recall that the discrete temperature at a given mesh point represents the continuous temperature variable in a region around the mesh point extending halfway to the next mesh point in each direction. This region is the domain of the discrete variable. In the display on the computer screen, *the domain of the discrete variable is colored* according to the temperature of the discrete variable its corresponding mesh point. The domain lines are dashed in Figure 2.

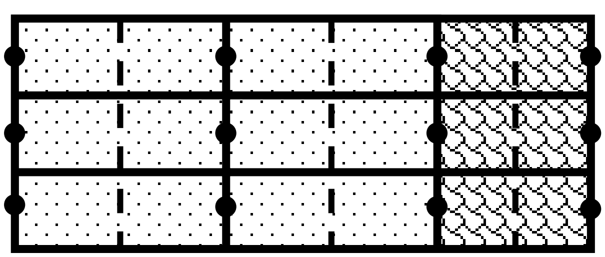

Figure 2. Dashed lines separate the domains of the discrete variables.

It is important to note that the colored domain for interior mesh points extends between dashed lines, while the colored domain of edge mesh points extends only from the edge to the nearest dashed line. To help the user viewing the heat structure display image, the domain lines are shown in black and the mesh point lines in white.

Another viewer aid is a so-called a material bar beneath the heat structure geometry grid. This separates the material depiction from the temperature color map. When the two are mixed, as indicated by Figure 2, the material shading can be difficult to discern.

Perhaps the most important aspect of the heat structure display for the user is the access to and display of the boundary condition information. On either side of the heat structure geometry sit two pairs of boundary columns. They are broken into cells corresponding to the heat slabs. The innermost cell identifies the control volume connected to the adjacent heat structure on that side and shows the volume temperature via color. The outer cell shows the value of a boundary quantity of the user's choosing.

If there is no boundary volume connected to a given heat slab on one side, its adjacent boundary cell in the volume column is black and has no volume number while its entry in the variable column is empty. See Figure 3.

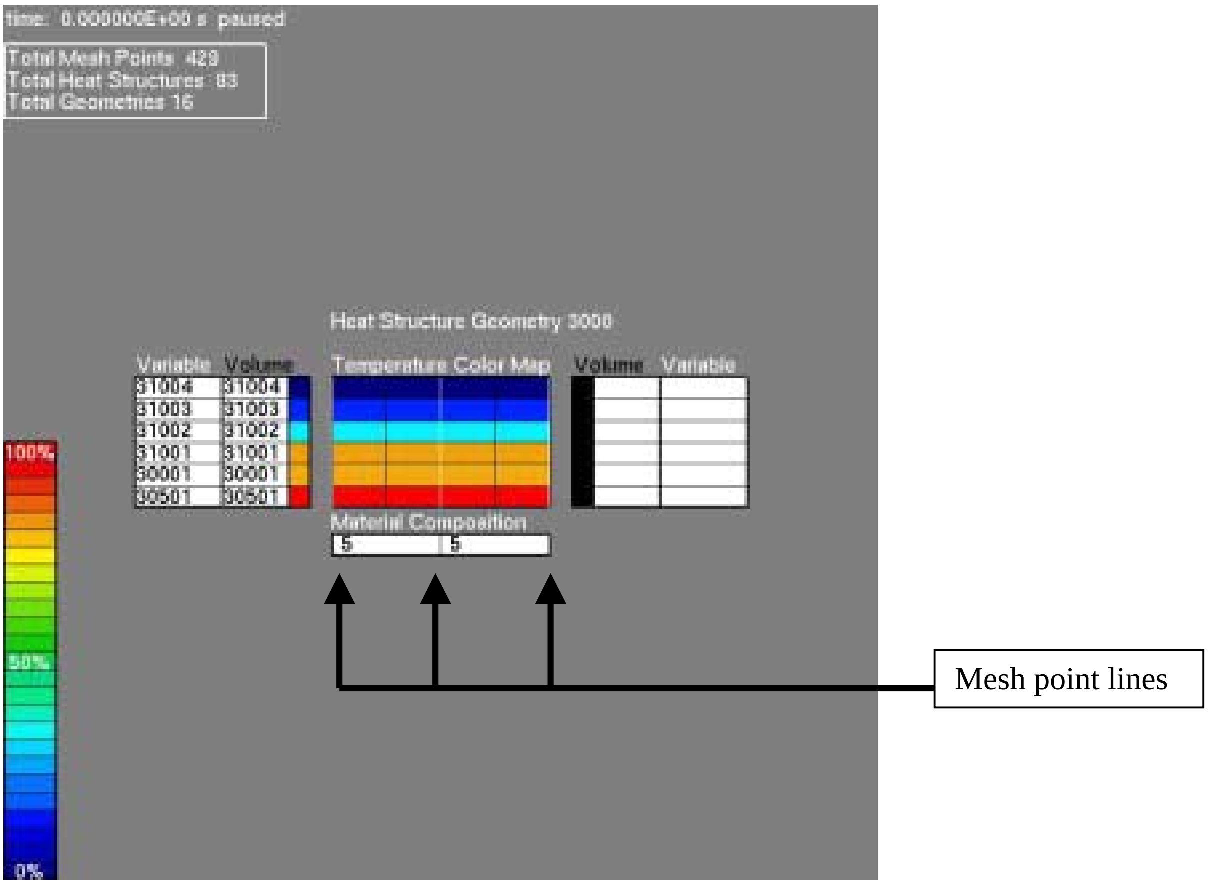

Figure 3. Heat Structure Geometry with only left-side boundary volumes.

Other useful information is displayed on the heat structure screen, such as a color key, the range of the color map, the heat structure geometry being shown, etc. The user accesses the heat structure and controls the display via the usual mouse/menu method. No help is yet available for this screen. However, where possible, the same conventions apply on the heat structure screen as on the isometric image screen. For example, to shift the image to the left on either screen, the user uses the same keystroke, namely control-x.

3. Reactor Vessel Screen

The primary purpose of the reactor core screen is to display conditions in the core during a severe accident. It is intended to show the full range of severe damage starting with a static picture and animating the ballooning of the cladding, rupture, melting, relocation within the core, and formation of the corium pool in the lower plenum. Therefore, the reactor vessel screen focuses solely on a cross-sectional slice of a reactor; see Figure 4.

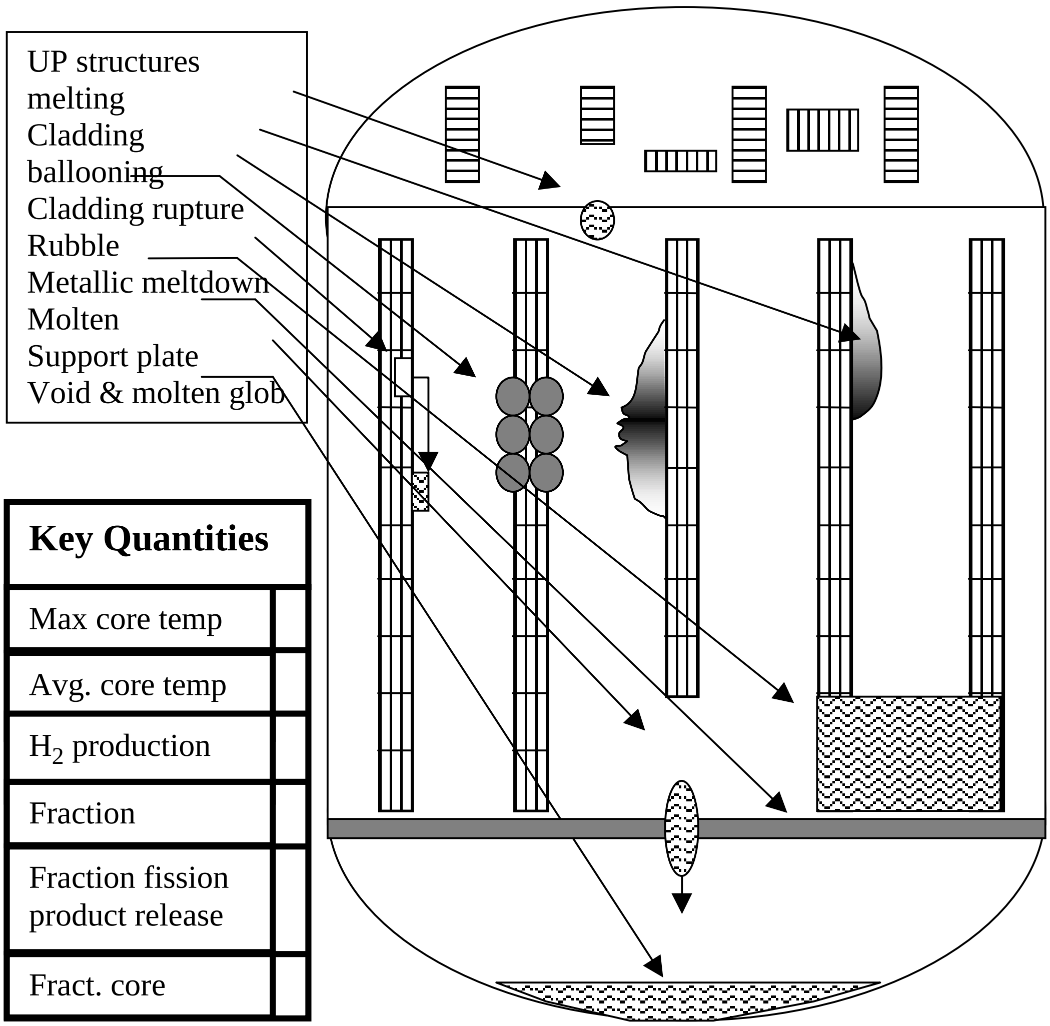

Figure 4. Sketch of proposed animated Reactor Vessel Screen.

The basic element of the reactor vessel screen is the so-called SCDAP component which can be a fuel rod, control rod, shroud, etc. There are similarities between the RELAP5 heat structure geometries and the SCDAP components in that both are discretized as a two-dimensional grid of mesh points. Efficiency and simplicity in the programming was obtained by using the same coding to generate the color-mapped temperature mesh for both.

However, the rest of the reactor vessel display image is very different from the heat structure display. Within the core, the SCDAP components appear completely within a control volume. There may, in fact, be several components within a volume.

When the screen first appears, the components are shown with the proper height to width ratio relative to the vessel. Therefore they appear as short and extremely thin black lines. The user has the option, through the tool bar, to artificially change this aspect ratio to any desired value to make the fuel rods wide enough to properly. The components, when first displayed, are half-components; they are shown only from their centerlines to the right edge. The entire vessel is drawn from its center to right edge also. The user can change the view to show whole rods and the vessel automatically becomes whole as well. The user may also activate the color key and legend of important quantities through the tool bar. See Figure 5 for such a view of the standard SCDAP problem, scp2.i. Note that the gap between the fuel and the cladding on either side appears as a double thick line due to the staggered mesh.

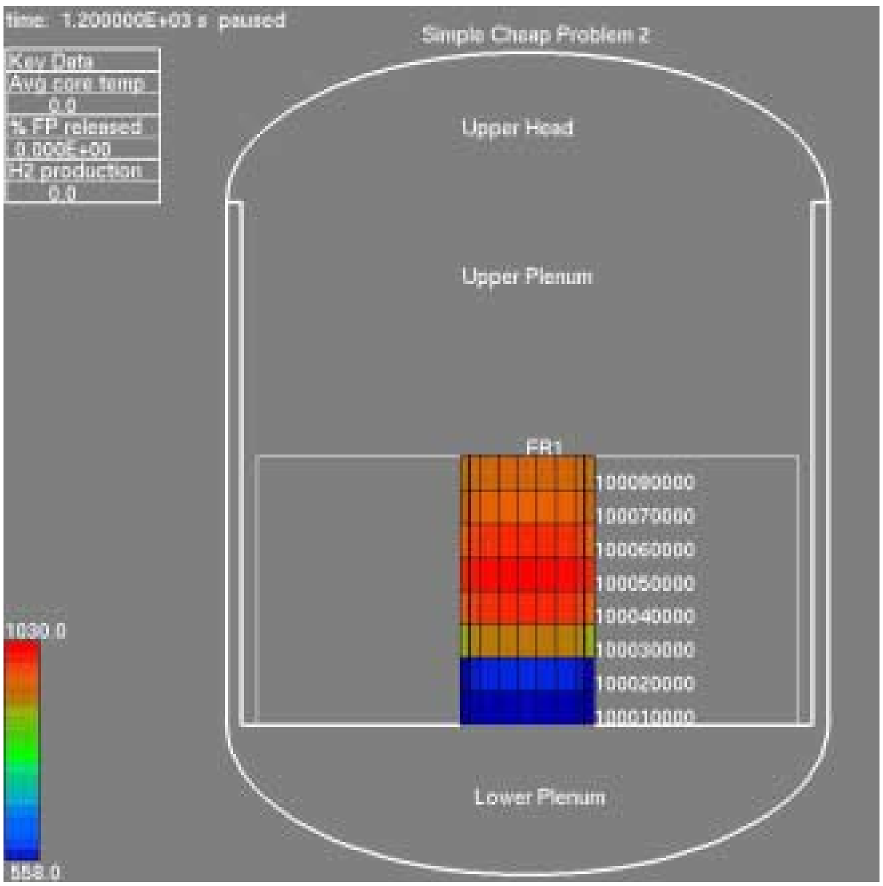

Figure 5. Reactor Vessel Screen view of Simple Cheap Problem 2 (scp2.i)

This screen is undergoing continuing development. Currently the structures remain static although the colors change; animation is scheduled for next fiscal year.

Finally for the user's convenience, *the reactor vessel image is produced automatically* solely from the information in the input file. It should be noted that some of the target display information shown in Figure 4 is difficult or impossible to obtain solely from the input file and may require additional information from the user to be included in the image.

4. Input Model Generator

An original goal for RGUI was the inclusion of an input model generator for RELAP5-3D. One option was to build one from scratch. Another option was to adapt a suitable input model generator to RELAP5-3D. A search of available generators revealed one that was better suited to our purposes than any other, the Engineering Code Pre-Processor (ECPP) from Computer Simulation Analysis (CSA). It has been completed only recently, has a very large and satisfied user base, and generates input files for RETRAN. RETRAN has a huge user base and it is an outgrowth of RELAP4; thus the plant modeling concepts are similar to those of RELAP5-3D.

Both RETRAN and ECPP are owned and maintained by the Electric Power Research Institute (EPRI). EPRI so valued RGUI, that it was willing to enter into an exchange of intellectual property with the INEEL in which EPRI received RGUI and INEEL received ECPP.

ECPP accepts any sort of pictorial representation of the plant, even scanned images. The user "sensitizes" portions of the image using the standard enclosing rectangle (click on a point, hold the mouse button, move mouse to another point while rectangle stretches to have the points at opposite corners, then release the button to set the enclosing rectangle). The rectangle is then associated with a name, such as vessel. The user can subdivide it further (for example into lower plenum, core, upper plenum, and upper head) and further until the subdivisions are equivalent to individual RETRAN or RELAP5 components.

From the sensitized image, data for each component comes from default information, menus, dialogs, and calculators. There are many auxiliary windows that each show some representation of the portion of the plant being modeled, such as the neutron cross-section display. To help the user, the ECPP system tracks what is complete and incomplete in the model. It also detects certain kinds of modeling errors as they are made and flags them. ECPP also provides Q/A for the input model developer. When the input file is generated, it is an errorless transcription of the information in the model, has uniform placement of data in fields on the input cards, and is heavily documented.

Nonetheless, there is work to be done to adapt it to RELAP5-3D. Obviously, new features must be added to handle the 3D hydrodynamic and 3D neutronic information. Although the modeling concepts are virtually the same, the actual lines of the input file are different; these must be adjusted. ECPP only works on Microsoft Windows platforms; it should be extended to Unix platforms. This can be accomplished in a variety of ways, such as by replacing the use of Microsoft Foundation Classes with Tcl/Tk or Java.

The work on adaptation has already begun. The conversion of the information written to the input file from RETRAN format to RELAP5-3D format has been completed for the basic information (100 through 115 cards), time step control (200 level cards), minor edit requests (300 level cards), trip cards and variable trip cards. Work on the hydrodynamic component cards has begun but is not complete. Along with the output conversion, the dialog boxes were converted to work on Unix as well as MS Windows platforms.

5. Other Developments and New Directions

5.1 Other New Developments

Other improvements were added to RGUI. The isometric image now has a magnifying capability that allows pinpoint control of the size of the displayed image. This feature is available in addition to the standard magnification/reduction factor of 5%. It can prove most useful when making a visual for inclusion in a report or paper.

The PYGMALION program has been refurbished and will be included in the next release of RGUI. It is an input-file post-processor that takes as input both a RELAP5 restart-plot file, that has been run to steady-state, and the corresponding input file; it then produces a new input file with the steady-state conditions replacing the original ones. This makes a convenient starting place for constructing new input files that model accident scenarios or model operational transients. For the convenience of the user, a GUI front end has been written for it and will be accessible from the RGUI 3D-Station as an item in the Tools menu.

5.2 New Directions

The three extensions to the RGUI development plan include separation of RGUI from RELAP5, adaptation to other computer programs via filters, and the PVM executive GUI.

The move to separate RGUI from RELAP5-3D is a response to the requests of several users to adapt RGUI to other versions of RELAP5 and to other computer programs. When linked directly to RELAP5-3D, RGUI can access the RELAP5-3D database and make use of all the data. It can also control the transient as it is being calculated.

In practice however, neither of these has proven to be of great benefit over simple replay. Minor edit quantities include just about everything that users want to see. The ability to control a transient calculation would be different from the ability to control a replay if the user could change the course of the calculation. Since this is not possible in the current implementation, there is no essential difference between the control of a transient and replay. Thus there currently is no great advantage in linking RGUI with RELAP5-3D.

Separating RGUI from RELAP5-3D would have no effect on the 3D-Station work area or the RELAP5 command set-up screen. The only impact would be on the isometric image, and to a much smaller extent, on the new heat structure and reactor vessel screens. The effect is that coding that directly accesses the RELAP5-3D database the data would have rewritten to access the restart plot file.

There are advantages to separating them. The replay mode actually runs through the calculation faster than when the calculations are being made; this is good for the user. Another advantage is that it will lend itself to adapting RGUI to other programs. Users have requested this for various reasons; some requests are recorded in Section 1. The means of carrying it out involves the restart-plot file.

All known versions of RELAP5 create restart plot files. These restart plot files can be post-processed into whatever form RGUI requires, then RGUI can run a replay from it.

For programs that grew out of older versions of RELAP5, such as RETRAN, THEATRE, and APROS, the post-processing will be somewhat greater. For other TH codes, such as TRAC, COBRA, and CATHARE, the work of developing a post-processor to create a usable restart-plot file is correspondingly greater, but doable if source code is available.

It is necessary to plan the format for the post-processed restart file by considering which codes will be involved and then determining what they place on their restart files. From this a common group of information will be determined. Some programs will produce extra data; that is easy to handle. Some will not produce certain data that all the others do, but will have other information from which the missing data can be extracted. An example (admittedly a stretch) might be a program that never calculates temperatures, but only works with internal energy. The post-processor will be responsible for using the information on that program's restart file to calculate temperature and then placing it on the post-processed file.

This post-processing idea is not new in the computing industry. For example, most text processors offer similar post-processing of a file when a user chooses the "Save As" option. A menu of formats for other text editors is available. The post-processing program is called a filter and the process is filtering. There is a similar pre-processing that can be done, sometimes automatically, when a file from a different text processor is opened. The text-processor determines the file's format and invokes an appropriate filter to convert the file before opening the file.

It is natural to consider the outgrowth of the post-processing work planned here. When RGUI can perform replays of numerous different programs, it will provide a consistent means of accessing and viewing data. This will be convenient for workers in the nuclear community.

A different point is that plans already exist to calculate the transient behavior of some very complex physical systems with a combination of existing nuclear industry codes working simultaneously on individual pieces of the calculation and sharing information via PVM. It will be much easier to use one GUI to view the calculational results of every one of them than to view each with a separate GUI. Moreover, the coordination of all these programs, each doing a piece of the calculation, is to be coordinated by an executive or master program.

The work on the combined executive and combined codes has been underway for some time. The executive directs the individual programs on when to start and stop. It also receives messages from them and uses the data to make certain decisions, such as the size of time step that each program can take. It will be necessary, both to the developer and the user of such a complex system of programs, to do such things as track the message flow, recognize the executive's decisions, and locate bottlenecks in the calculation. A GUI for the executive will help by creating a visualization of these and other important functions.

The RGUI 3D-Station already performs some of the functions that an executive GUI will. It can initiate and stop jobs. It has a menu of programs that can be run as jobs. It receives and displays information from the programs it runs and from the operating system. It was, in fact, part of the original plan to develop the 3D-station into an executive that would perform many of the functions of the executive GUI described above, and more. It is not much extension of the original design plan to develop the 3D station into the GUI for the PVM executive program.

6. References

1. D. M. Snider, K. L. Wagner, W. H. Grush and K. R. Jones, "Nuclear Plant Analyzer," NUREG/CR-6291, INEL-94/0123, Vol. 3, Idaho National Engineering Laboratory, Idaho Falls, ID, January 1995.

2. The RELAP5-3D Code Development Team, "RELAP5-3D Code Manual," INEEL-EXT-98-00834, Revision 1.2a, Idaho National Engineering Laboratory, February, 2001.

3. G. L. Mesina and A. M. Austin, "New RELAP5 Graphical User Interface," Proceedings of the International RELAP5 Users Seminar, Dallas, TX, USA, March 17-21, 1996.

4. G. L. Mesina, A. M. Austin and W. H. Grush, "RELAP5 Graphical User Interface Business Plan and Functional Requirements Document," business sensitive private company document, Idaho National Engineering Laboratory, Idaho Falls, ID, January, 1996.

5. L. J. Siefken, E. W. Coryell, E. A. Harvego and J. K. Hohorst, "SCDAP/RELAP5/MOD 3.3 Code Manual," NUREG/CR-6150, INEL-96.0422, , Idaho National Engineering Laboratory, Idaho Falls, ID, January 2001.

6. J. H. McFadden, et al, "RETRAN-03 Code Manual," EPRI NP-7450, May, 1992.

7. R. P. Stewart and C. E. Peterson, "The EPRI Engineering Code Pre-Processor," 8th International RETRAN Meeting, Denver, CO, USA, October 1-4, 1995.

8. K. Carlson, P. A. Roth, and V. H. Ransom, "ATHENA Code Manual Volume I: Code Structure, System Models, and Solution Methods," EGG-RTH-7397, Idaho National Engineering Laboratory (1986).

9. C. W. Stewart, et al, "COBRA-IV: The Model and the Method," BNWL-2214, Pacific Northwest Laboratory, 1997.

10. K. W. Washington, et al, "Reference Manual for the CONTAIN 1.1 Code for Entrainment Severe Accident Analysis," NUREG/CR5715, SAND91-0835, Sandia National Laboratory, May, 1991.

11. J. Spare, et al, "TRAC-M/Fortran 90 (Version 3.0) Theory Manual," Los Alamos National Laboratory, NUREG/CR-6724, July 2001.

12. R. O. Gauntt, et al, "MELCOR Computer Code Manuals," NUREG/CR-6119, SAND2001-0979P, Sandia National Laboratory, April, 2001.

13. G. L. Mesina, "The RELAP5 3D Station, a Graphical User Interface," RELAP5 International Users Seminar, College Station, TX, USA, May 17-21, 1998.

14. G. L. Mesina and J. A. Galbraith, "RGUI 1.0, New Graphical User Interface for RELAP5-3D," 7th International Conference on Nuclear Engineering (ICONE-7), Tokyo, Japan, April 19-23, 1999.

15. G. L. Mesina and J. A. Galbraith, "RELAP5-3D Graphical User Interface, RGUI 1.1," RELAP5 International Users Seminar, Park City, UT, USA, July 25-27, 1999.

16. G. L. Mesina and J. A. Galbraith, "RELAP5/RGUI Architectural Framework," 8th International Conference on Nuclear Energy, Baltimore, MD, USA, April 2-6, 2000.

17. G. L. Mesina and J. A. Galbraith,"New Developments and Value of the RELAP5-3D Graphical User Interface (RGUI 1.2)," 2000 RELAP5 International Users Seminar, Jackson Hole, WY, USA, September 12-14, 2000.