Transactions of the Korean Nuclear Society Autumn Meeting
Gyeongju, Korea, October 27-28, 2011

Development of a MAAP-based Severe Accident Training Simulator using Visual System Analyzer

Jae-Seung Suh[a*], Soo-Yong Park[b], Kwang-Il Ahn[b], Kyung-Doo Kim[b]
[a]*SENTECH, 575 Yongsan, Yuseong, Daejeon, 305-500, Korea*
[b]*Korea Atomic energy Research Institute, P.O. Box 105, Yuseong, Daejon, 305-600, Korea*
[*]*Corresponding author: SenTech@eSenTech.kr*

1. Introduction

The recent environment of severe accident analysis requires high performance computers to simulate complicated reactor and containment phenomena. In parallel with this, rapid advances in computer technology now enable these codes to run in real or almost real time. The remaining limitation restricting their use on an even wider scale is that most of the existing codes are still subject to a complicated I/O structure. Even user-friendly graphical user interfaces (GUI) will be not likely to help better and efficient interpretation of the analysis results obtained from these codes as well as for their increased use. This situation has motivated the development of easy-to-use GUI tools for severe accident codes, such as ViSA [1], SNAP [2], MAAP4-GRAAP [3], SATS [4], and et al. For instance, ViSA enables the thermal-hydraulic system codes to be used like a conventional nuclear plant analyzer.

Recently, a project for a real time simulation of results obtained using MAAP4 codes under the ViSA environment has been initiated in KAERI. Such a GUI-based interactive interface can be very useful in sharing real time analysis results obtained from the MAAP code. The purpose of this paper is to introduce the current status of a MAAP-based Severe Accident Simulator being coupled with the ViSA system.

2. Methods and Results

As mentioned before, the present severe accident simulator is being currently developed as a kind of MAAP-ViSA coupling system: MAAP4 as its engine (i.e., a tool for severe accident analysis) and ViSA as its GUI tool.

2.1 ViSA

The ViSA is a user-friendly computing environment for system thermal-hydraulics codes, which was developed by KAERI [1]. Since its development the ViSA has been coupled with system thermal-hydraulics codes such as RETRAN, RELAP5 and MARS. The present coupling of ViSA with a severe accident MAAP4 code employs a concept for a plant-independent nuclear plant transient analyzer, focused on the followings:

- Plant-independent, modeled via only MAAP code inputs
- Various graphical representations of the MAAP-specific output
- On-line user-interactive control for a simulation of various operator actions

2.2 MAAP/ViSA Interface

Development of a GUI-based simulator can be broken into two steps: identification of an "engine"; and development of a MAAP/ViSA interface. A personal computer with Windows OS was used as the developmental platform. This choice was primarily dictated by the fact that the dynamic link library (DLL) for MAAP4, which was chosen as the engine for the simulator, was generated in the Windows environment. MAAP4 is selected as the engine of severe accident simulator because of its widespread use worldwide. MAAP4 input decks for most nuclear plants are already available, and can be directly or with very minor modifications used with the simulator developed here. The choice was also, at least partly, dictated by the desire to develop a means to graphically display the voluminous text-based output generated by MAAP4.

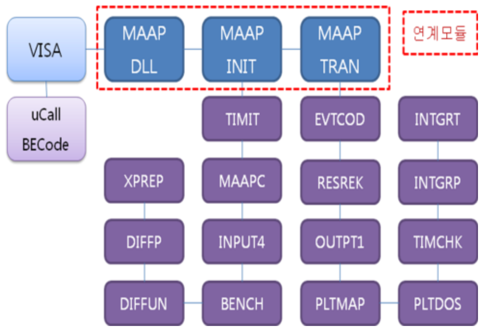

Fig. 1. MAAP/ViSA interface programs.

The sub-programs added for the DLL programming are located in a separate directory and maintained as an independent static library. The MAAP4 files that have been modified to implement the interactive control feature are also saved in a directory different from the one where these sub-programs (or subroutines) are saved in a standard version of MAAP4. The number of variables for the interface between MAAP4 dynamic link library (DLL) and ViSA is minimized to simplify the interface. Standard input decks of MAAP4 are used as input for this simulator. Since interactive control feature has been added, control parameters for this new feature however must be provided along with the user input with MAAP4. The interactive, GUI-based user interface is developed using the feature in Delphi. Delphi is a powerful and flexible graphic programming language. It provides a platform to efficiently develop user interfaces and to display data. Moreover, with the click of a mouse it provides a GUI-based running environment. For Delphi programming, an OOP (object oriented programming) approach was adopted. Each component is encapsulated with well-defined interfaces. The component, such as the tool bar, list box, etc., are then simply assembled together to create the complete simulator. This OOP approach reduces the programming effort and the complexity of the design. Because each component is independent and self-contained with well-defined interfaces, it can be repeatedly used, saving time and effort in future development work. The simulator developed here taps into the data directly from the MAAP4 arrays for the variables to be plotted. This allows the display of data while simulation is still continuing. ViSA is coupled with MAAP4 as dynamic link library (DLL). MAAP4 main program is changed into a subprogram, and is exported to ViSA. The changes made to accomplish this are shown in Fig. 1. The same figure also shows a new sub-program, set files, added to set up the I/O files. Also, as an example, the part of MAAP4 plot input and the related part of a new sub-program to graphically show the output of MAAP4 in the ViSA are shown and described in Fig. 2.

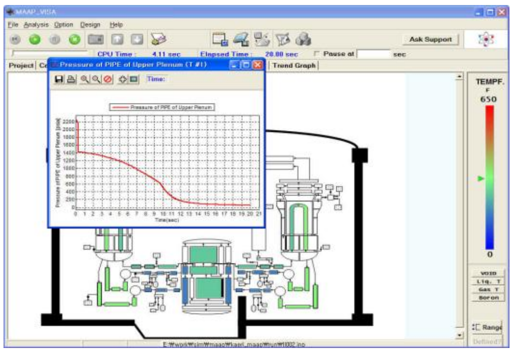

Fig. 2. Nodalization windows with trend graph.

Fig. 1 shows a schematic diagram of the GUI-based severe accident simulator. The user provides the input and output file names for MAAP4 via the main control module by either entering the path and name of the file, or by simply browsing through the storage media on the computer. These paths and file names are transferred to MAAP4 DLL when user clicks on the “run” button in the main control module. The simulator runs using these files as input decks. Local or remote master user, who has the control of the simulator, can simulate operator’s actions through the main control page, and this control action is also passed to MAAP4 DLL through MAAP/ViSA interface. During the transient calculation, the MAAP4 DLL transfers the results to ViSA, and the ViSA show the data in graphical form.

2.3 Data Visualization

MAAP4 produces a large amount of text-based output in a transient simulation. GUI-based nuclear simulator is designed to provide graphical displays of the results during or after a transient simulation so that the users can easily follow plant dynamics. There are three data visualization components (nodalization, real-time graph, mimic) that can be accessed to display a wide range of data.

Since MAAP4, a code that calculates fairly detailed spatial distributions, is the engine behind the simulator developed here, it is possible to display the evolution of spatial distribution of quantities of interest such as temperature or pressure, etc. The nodalization window is designed to show temperature in primary nuclear system. Color is used to show the level of temperature in each cell. Fig. 2 shows the change in temperature distribution during a large loss of coolant accident.

3. Conclusions

A GUI-based severe accident simulator based on a MAAP4-ViSA coupling is being developed in KAERI. As its engine, the MAAP4 computer code provides a flexible, efficient, integrated tool for evaluating the in-plant effects of a wide range of postulated accidents and for examining the impact of operator actions on severe accident progressions. Within the present framework of the simulator, the simulator engine MAAP4 can be switched with any other system analysis code since the interface between the simulator engine and graphical user interface program is well-defined and the two are coupled by dynamic link libraries. Additional user interactive features for operator actions are currently being incorporated into the present simulator.

REFERENCES

[1] K.D. Kim, S.W. Lee, M. Hwang, B.D. Chung and S.J. Cho, “Development of a Visual System Analyzer based on reactor system analysis codes,” Progress in Nuclear Energy, 49, p.452-462, 2007.
[2] Jones, K.R., Symbolic nuclear analysis package. In: Proceedings of the 2000 ANS/ENS International Mtg, Embedded Topical Mtg. #2, “Best Estimate” Methods in Nuclear Installation Safety Analysis, Washington, DC, November 12-16, 2000.
[3] P.J.T. Bakker, “Use of MAAP-GRAAP for Training of Borssel NPP Plant Operators”, SAMOA-2 Meeting, Lyon, France, Sept. 9-10, 1997.
[4] K.R. Kim, S.Y. Park, Y.M. Song and K.I. Ahn, Development Of Desktop Severe Accident Training Simulator, Nuclear Engineering And Technology, Vol.42 No.2, p.151-162, 2010.