Integrated Development Environment for SPACE Problem Analysis

Chansu, Jang [a*]
[a]*KEPCO Nuclear Fuel*
[*]*Corresponding author: csjang@knfc.co.kr*

Keywords : SPACE, AESPA, Thermal hydraulic model, Transient analysis

1. Introduction

The SPACE code has been developed to simulate the transient behavior including LOCA and Non-LOCA phenomena and analyze the performance of the nuclear power plants [1]. After that, the SPACE code was adopted to perform the transient analysis of the nuclear power plant. To perform the SPACE problem analysis, the first thing to do is to develop the SPACE model corresponding to the system analyzed. On this stage, the typical text editors and plotting tools like Excel or ORIGIN have been utilized. These job procedures are really normal these days but it is very tedious and takes too much time because of the long-lasting routine jobs. Because of these reasons, the integrated development environment is required and the **A**dvanced **E**ditor for **S**PACE **P**roblem **A**nalysis (hereafter, AESPA) is developed to make the job procedures of the SPACE problem analysis fast, robust, and simple.

2. Methods and Results

In this section some of the techniques used to develop the AESPA and the AESPA functionalities are described.

2.1 Techniques used

The main computer language used to develop the AESAP is C++ [2] and the GUI framework is Qt version 5 [3]. C++ language was adopted to handle a bunch of numerical data quickly. Qt is an application programming framework that is used to develop cross-platform applications and a full development framework with tools designed to streamline the creation of applications and user interfaces for desktop, embedded, and mobile platforms. The most classes developed for the AESPA are subclassed from the Qt classes. To read and write a bunch of numerical data on a file, the parallel processing method supported by C++ and Qt is adopted so that the fast handling can be achieved.

2.2 Input Edit

One of main functionalities of the AESPA is to edit the SPACE input. This function includes editing, searching and replacing texts. Editing texts is same as typical text editors. The function of text searching and replacing is more advanced than other text editors. The AESPA searching function supplies the list of all the texts related to the user-supplied one. Through the list, the same input variables used repeatedly can be distinguished and can be edited to the right input value at the right place. To improve looks of texts on the editor, the AESPA supports the customized cursor and colorful indicators.

2.3 Input Files Comparison

The AESPA can compare and merge input files up to 3 files and directories up to 3 directories. This function can be done by featuring with WinMerge program. The WinMerge program is installed as part of the AESPA and is handled by AESPA directly. The Fig. 1 shows the sample comparison window of the input files.

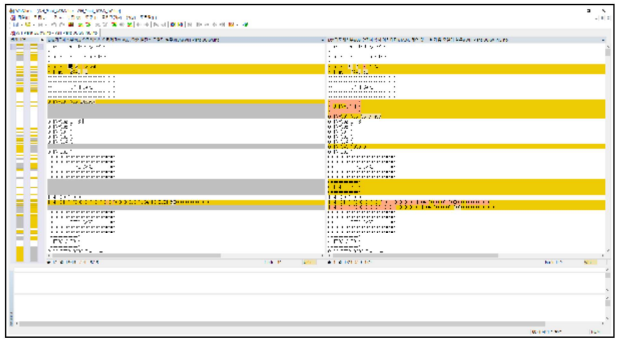

Fig. 1. Sample comparison window of the input files

2.4 Input Syntax Check

During the development of SPACE model, many silly mistakes cause wrong inputs and require user's efforts to make it right. To avoid those unnecessary efforts, the AESPA has an ability to detect syntax errors or warnings in the input values. The logic of determining right and wrong inputs is developed based on the experiences of users and the code user's manual. This logic might not to detect every errors and warnings but can prevent a loss of development

time. Fig. 2 shows the sample window of syntax check of a wrong input value.

```
307  KINE-0702 C015-01 0.54440 0.0
308  KINE-0703 C015-01 0.22780 0.0
309  KINE-0801 C001-00-01 0.22780 0.0 0.0
     Error: KINE-0801 must be connected to the heat structure. C001-00-01 is not a heat structure.
310  KINE-0802 H001-00-02 0.54440 0.0 0.0
311  KINE-0803 H001-00-03 0.22780 0.0 0.0
312  KINE-0900 9
```

Fig. 2. Sample window of syntax check

2.5 Executing SPACE code

After completion of SPACE model, the execution of SPACE code can be done on the AESPA by one click. The SPACE execution is done on another process to reduce the burden of AESPA. Previously, a ready-made batch file is prepared to run the SPACE code simply. In this way, the prompt window is required, But the AESPA dose not require that and is not showing the ugly prompt window. The percent progress of calculation is shown on the graphical item called progress bar.

2.6 Real-time plot

Generally, a SPACE calculation takes a long time to finish. During this time, users may want to know the status of their calculation and confirm if the calculation goes to a right way. Based on these requirements, the AESPA can make user-specified plots in real time while the SPACE code is running. The plots can be added before or after the start of the SPACE execution. And additional plots can be added during the SPACE running. When using the existing SPACE-owned real-time plot (GraphPlotter), the SPACE calculation becomes really much slower. To avoid those faults in the AESPA, making plots and running SPACE are performed on different processes. Therefore, the calculation time consumed is not affected by making real-time plotting. Fig. 3 shows the starting window of the SPACE code execution. On this window, variables can be selected or not for creating real-time plots.

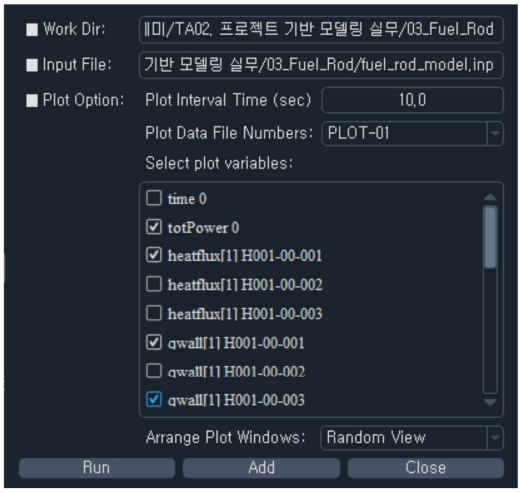

Fig. 3. Starting window for SPACE execution

2.7 Off-line plot

The AESPA provides two ways of making plots for user-specified variables after the SPACE calculation. The first is to convert and transport a SPACE data file to the Excel program. The AESPA can convert a SPACE data file to a excel file. Therefore, the Excel can be launched by AESPA. After the startup of Excel, users will work on the Excel. The second way is to use the AESPA-owned plot functionality. The AESPA has its own plot module. Using this module, users can make a beautiful and cool plot using the many graphical items such as texts, arrows, brackets, boxes and ellipses. And AESPA plot supports drag-and-drop function to make a plot and add variables in a plot. The source of numerical data can be the SPACE data file, TAPE76 and TAPE60 files based on CESEC-III code and custom data file. The plots of AESPA can perform numerical operations such as normalization of 1.0 and subtraction of null-transient time, etc. The plot's unit can be converted to other units directly. Multiple plots can be compared in the way of vertical, horizontal and grid arrangements. In this comparison, a maximum and minimum values are calculated automatically and shown following the mouse clicks. The AESPA plot can be saved and loaded as an electronic file with its own format. The plot file is saved and loaded using the JSON data format. Fig. 4 shows the sample plots of AESPA.

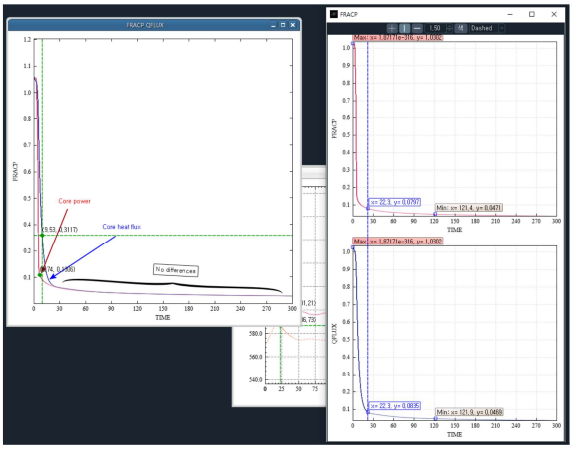

Fig. 4. Sample plots of AESPA

3. Conclusions

To construct an integrated development environment for SPACE problem analysis, the AESPA has been developed and applied to the transient analysis. Many features of the AESPA have made the job procedures developing SPACE models fast, robust and simple. In the future, the more graphical features, which help users get insights of their model, will be introduced in the AESPA like node presentations and different types of plot.

REFERENCES

[1] KHNP, "SPACE 3.0 매뉴얼 Volume 2 사용자 매뉴얼", TR-KHNP-0032.
[2] Cay Horstmann, San Jose State University, "Big C++ Late Objects", WILEY
[3] https://doc.qt.io/qt-5/qt5-intro.html