*Transactions of the Korean Nuclear Society Spring Meeting*
*Gwangju, Korea, May 30-31, 2013*

# Development of a Wrapper Object for MARS TH Systems Code and Its Applications in Object Oriented Programs

Sun Byung Park, Young Jin Lee, Hyong Chol Kim, Sam Hee Han, Hyun Jik Kim
*Nuclear Safety Evaluation(NSE), #508 Convergence Technology Research Commercialization Center,*
*218 Gajeong-ro, Yuseong-gu, 305-700, Daejeon, Korea*
**Corresponding author: yjlee@nse.re.kr*

## 1. Introduction

A wrapper object class, TMars, has been developed. TMars is an object class that wraps the functionalities of the MARS-KS[1] thermal hydraulic system analysis code. TMars is designed to be used when developing an applications using object oriented program techniques[2]. TMars is written for the object pascal program language, and "wraps" the Dynamic Link Library (DLL) manifestation of the MARS-KS code written in Fortran 90. TMars behaves as a true object and it can be instantiated, inherited, and its methods overloaded. The functionality of TMars was verified and demonstrated using two programs built under object oriented program environment. One is a text based program for reviewing the data interface of TMars, and the other is a graphic intensive prototype NPA program for testing the overall performance of TMars. The prototype NPA was also used to assess the real-time capability of TMars. The demonstration programs show that application of TMars is straight forward and that its functions facilitate easy application developments.

## 2. Development of TMars Wrapper Object Class

### *2.1 Development of Mars-KS Dynamic Link Library*

MARS code is written in Fortran language but modern object oriented program languages provide better environment for developing large scale graphic intensive programs such as the Nuclear Plant Analyzers (NPA) or the simulators. Thus, a mixed language programming is one of the better solutions when the MARS code is to be used as the Thermal-Hydraulic engine of a program to be written in more modern languages. The mixed language programs can be greatly simplified by using dynamic link library (DLL).

Hence, a DLL version of the MARS-KS code was developed in a similar fashion to developing the Relap5 DLL[3]. However, the recent re-structuring of the MARS-KS code and the use of data modules presented two significant difficulties. First, the restructuring of MARS-KS caused important major subroutines to be not accessible readily. Second, the simple and elegant data transfer scheme used for the RELAP5.DLL is unsuitable due to the modularized data structure. Thus, different approaches were developed to solve these problems. The first problem was solved by introducing a parameter that is passed from the calling program to Mars DLL to control the calculation flow. The second problem of data transfer was solved by mimicking the large 'fa' array of Relap. A large array 'dl_fa' is defined and then groups of equivalence data are defined within. The data are stored into the array in the defined equivalence sequence. Then the address of the first element of dl_fa is passed from TMars to calling program. And, then all data in dl_fa can be interpreted in the calling program.

### *2.2 Development of Object Class TMars*

TMars is a wrapper for the Mars-KS.dll written in Delphi (Object Pascal)[4]. TMars is a true object class and it can be inherited, polymorphed and overridden as needed, and multiple instances of TMars can be created with careful input/output/rstplt file maintenance.

Major public properties of TMars include:

- time step advancement count
- hydro time
- indices for accessing major variables of components, volumes, trips, heat structures, etc.

Major public methods include:

- method to create an instance of TMars
- method to destroy an instance of TMars
- method to read input and carry out initialization
- method to run specified number of time steps
- method to run to the end of problem

If more functionality is required, it is a simple matter of adding more methods or properties to the TMars. In addition, because TMars is a true object of OOP, it is possible to make new object classes through inheritance and polymorphism where the variables and methods can be added, encapsulated, overloaded or overridden.

The detachment of the thermal-hydraulic calculation from other modules such as GUI, through the use of TMars makes it simpler to modify the thermal hydraulic routines, as the modification would not, in most case, affect the functions of other modules.

## 3. Verification and Application of TMars

### *3.1 Verification Program*

A GUI (Graphic User Interface) based program was written to test, demonstrate and verify the TMars. The program was written in Delphi and consists of TMars and GUI program modules. The program reads in the input deck, creates an instance of TMars, and controls the time step advancement (run/stop). The program was used to examine the TH data calculated by TMars. The

data examinations can be carried out interactively using the GUI.

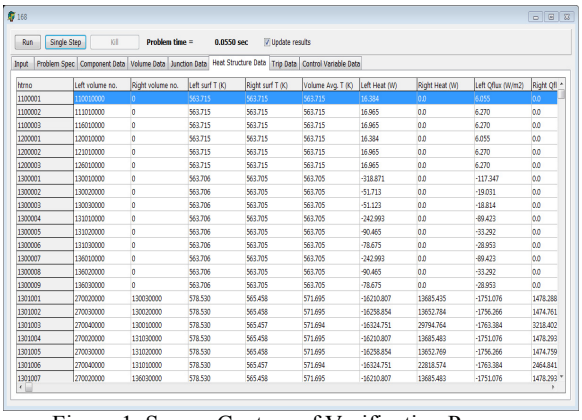


Figure 1. Screen Capture of Verification Program

Figure 1 shows the screen captures obtained during the execution of the demonstration program. Using TMars, the write up of the thermal hydraulic part of the program was a trivial effort. Most of the efforts in producing the demonstration program were in the coding of the GUI for the interactive program.

TMars performed in a stable manner with little degradation in execution time. The results of the calculations were identical to those calculated using the original Mars-KS.

### *3.2 NPA demonstration and the assessment of real time capability*

A prototype NPA (proto-NPA) program was written to verify the workings of essential major functions of an NPA. TMars object class was used as the TH engine and GLScene[5] package was used to graphically display the results in 3D graphics. A screen capture of the program is shown in Figure 2.

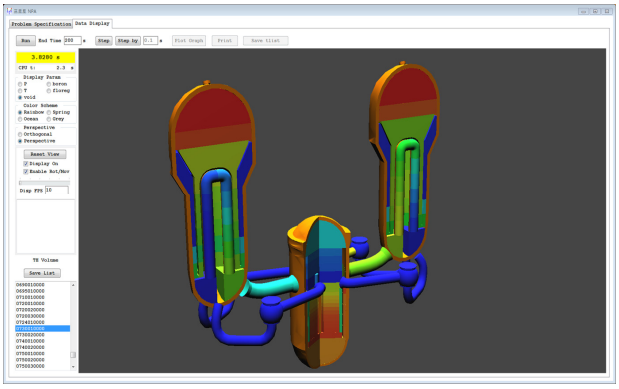


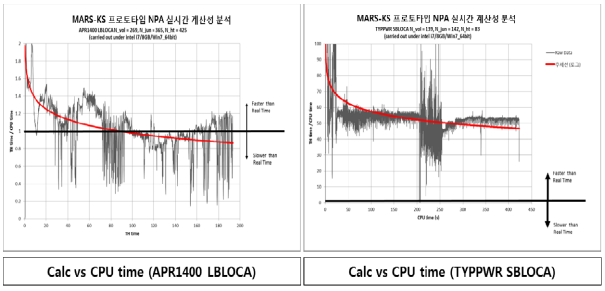
Figure 2. Screen Capture of Proto-NPA

The proto-NPA program was used to assess the real time capability of the TMars object. For this, the LBLOCA of APR1400 and the typpwr SBLOCA were selected for assessment. The calculations were carried out using a PC with intel i7 processor and 8GB main memory working under Windows 7 64bit OS. Summary of the results is shown in Table 1 and Figure 3 shows the graphs of calculated time vs. clock time.

Table 1. Results of Realtime Assessment Calculation

| | Volume 개수 | Heat struc. 개수 | Run Option | Graphic Display | 모사시간 (초) | Nearly Implicit 해법 | | Semi Implicit 해법 | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | | CPU 시간 (초) | 실시간 비 | CPU 시간 (초) | 실시간 비 |
| APR1400 정상상태 | 273 | 425 | stdy-st | ON | 300 | 17.5 | 5.83% | 45.1 | 15.03% |
| | | | | OFF | | 16.9 | 5.63% | 43.7 | 14.57% |
| | | | trnsnt | ON | 300 | 16.5 | 5.50% | 45.5 | 15.17% |
| | | | | OFF | | 16.9 | 5.63% | 44 | 14.67% |
| APR1400 200% 저온관 파단사고 | 269 | 425 | trnsnt | ON | 200 | 계산실패 ( 4.85 초) | | 188.2 | 94.10% |
| | | | | OFF | | | | 183.4 | 91.70% |
| TYPPWR SBLOCA | 139 | 83 | trnsnt | ON | 2500 | 거산 지연 | | 30.8 | 1.23% |
| | | | | OFF | | | | 30 | 1.20% |

Real time Capable




Calc vs CPU time (APR1400 LBLOCA) | Calc vs CPU time (TYPPWR SBLOCA)

Figure 3. Calculated vs. Real Time

The figures show that real time calculation is achievable with TMars and graphic modules for such complex calculations as LBLOCA with a large number of volumes and junctions.

## 4. Conclusions

TMars, a wrapper object class encapsulating the calculation functions of Mars-KS code, was successfully developed and verification of its functionality was carried out using custom made programs. The verification results show that TMars is capable of providing reliable TH calculation results and sufficient performance to realize real time calculations.

## REFERENCES

[1] "MARS Code Manual, Volume IV: Developmental Assessment Report," KAERI/TR-3042/2005, Korea Atomic Energy Research Institute, December 2009.
[2] "Object Oriented Software Construction", Bertrand Meyer, Prentice Hall, 1997.
[3] "Development of a Wrapper Object, TRelap, for RELAP5 Code for Use in Object Oriented Programs", Y. J. Lee, Transactions of the Korean Nuclear Society Autumn Meeting, PyeongChang, Korea, October 30-31, 2008.
[4] "Borland Software Corporation, Delphi 6 for Windows Developer's Guide", 1998.
[5] "GLScene : OpenGL Solution for Delphi", http://glscene.sourceforge.net/wikka/HomePage.