

**MARS 1.3 System Analysis Code Coupling with CONTEMPT4/MOD5/PCCS  
Containment Analysis Code using Dynamic Link Library**

Bub Dong Chung, Jae Jun Jeong and Won Jae Lee  
Korea Atomic Energy Research Institute  
150, Dukjin-Dong, Yu-song, Taejon, Korea, 305-353

**Abstract**

*The two independent codes, MARS 1.3 and CONTEMPT4/MOD5/PCCS, have been coupled using the method of dynamic-link-library (DLL) technique. Overall configuration of the code system is designed so that MARS will be a main driver program which use CONTEMPT as associated routines. Using Digital Visual Fortran compiler, DLL was generated from the CONTEMPT source code with the interfacing routine names and arguments. Coupling of MARS with CONTEMPT was realized by calling the DLL routines at the appropriate step in the MARS code. Verification of coupling was carried out for LBLOCA transient of a typical plant design. It was found that the DLL technique is much more convenient than the UNIX process control techniques and effective for Window operating system. Since DLL can be used by more than one application and an application program can use many DLLs simultaneously, this technique would enable the existing codes to use more broadly with linking others.*

**1. Introduction**

It is essential to analyze simultaneously both the reactor coolant system and the containment system for the realistic evaluation of ECCS performance during LOCA. Martin(1993)[1] unified the codes, RELAP5 and CONTAIN, by using the parallel processing via Parallel Virtual Machine(PVM) software[2]. Park et. al.[3] also merged the RELAP5 and CONTEMPT4/MOD5 into a unified version capable of simultaneous analysis for both the systems by using concept of process control in UNIX system. The UNIX process control was proven to be feasible and valid in the development of the merged code of RELAP5 and COBRA-TF by Lee et al.(1992)[4]. Recently, the MARS1.3 code [5] for multi-dimensional system analysis was released and has being developed mainly on the PC platform with Windows system. Since the previous code coupling techniques were based on the UNIX system, a different technique is needed for codes developed on the PC platform with the Window system.

Similar concept of UNIX process control in Windows system is a multithreading [6]. A thread is a path of execution through a program and it is an executable entity that belongs to one and only one process. Typical program resources are open files, semaphores (a method of inter-thread communication), and dynamically allocated memory. A program executes when the system scheduler gives one of its threads execution control. The scheduler determines which threads should run and when they should run. Threads of lower priority might need to wait while higher priority threads complete their tasks. On multiprocessor machines, the scheduler can move individual threads to different processors to balance the CPU load.

However the applications of the multithread program make the code structure very complex for an integrated code system. It is not desirable for the code maintenance and the independence of future

developmental activities of each code. Another approach for code coupling is a method using dynamic-link-library (DLL) [6]. A dynamic-link library is a collection of source and object code in the same manner as a static library. The differences between the two libraries are that the DLL requires an interface specification and it is associated with a main program during execution, not during linking. Data and code in a dynamic-link library are loaded into the same address space as the data and code of the program that call them. However, variables and routines declared in the program and in the DLL are not shared unless one uses the special compiler directives for interfacing. These features enable the compiler and linker to map the correct portions of the address space so that the data and routines can be shared. In this study, the coupling of two independent codes, MARS 1.3 and CONTEMPT4/MOD5/PCCS [7], has been carried out using the DLL technique.

## 2. Configuration of Coupled Code System and Design of Dynamic Link Library for Containment Analysis

MARS 1.3 is a code designed to calculate the system thermal hydraulics including reactor kinetics and control functions. CONTEMPT4/MOD5/PCCS is a code for the analysis of containment thermal hydraulics. Considering code characteristics, the configuration of coupled code system is decided such that the MARS code works as a main driver program while CONTEMPT works as associated routines. Overall design of the coupled code system configuration is shown in Figure 1.

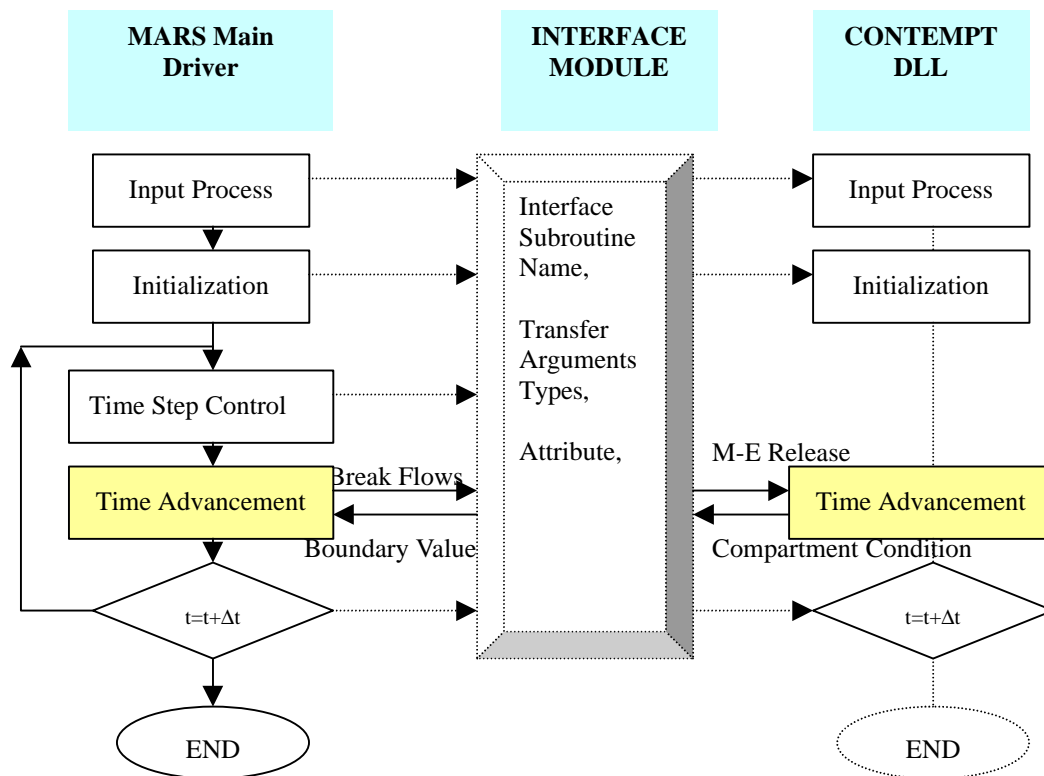


Figure 1. Configuration of Code System

Since both codes have input processing and initialization process using their own input and steam tables before the start of transient progress, CONTEMPT code should be divided into 3 major parts to synchronize with a main driver MARS 1.3. Some additional modifications of CONTEMPT are necessary

to share data with MARS by using special compiler directives for interfacing. Digital Visual Fortran compiler was used to make the source code to the dynamic-link-library with the interfacing routine names and arguments. The interfacing routines are specified using the following Digital Visual Fortran statements;

```
SUBROUTINE DTStepCont(ii,nti,dtsys,dt_fin,syslink)
!MS$ ATTRIBUTES DLLEXPORT :: DTStepCont
```

where DTStepCont routine is one of the interfacing routines in CONTEMPT DLL and the arguments ii, nti,dtsys,dt\_fin,syslink are the variables which can be shared through the interface module. According to the previous experience of RELAP5/CONTEMPT coupling [3], all interfacing routines and variables of the CONTEMPT code were identified. The CONTEMPT source code with added DLLEXPORT attribute routines is compiled into a dynamic-link-library, and all interfacing routines and variables are agglomerated into a MODULE as follows.

```
MODULE LINK_CONT
INTERFACE
SUBROUTINE InitCont
!MS$ ATTRIBUTES DLLIMPORT :: InitCont
END SUBROUTINE InitCont
SUBROUTINE TransetCont
!MS$ ATTRIBUTES DLLIMPORT :: TransetCont
END SUBROUTINE TransetCont
SUBROUTINE DTStepCont(iits,ntits,dtsys,dt_fin,syslink)
!MS$ ATTRIBUTES DLLIMPORT :: DTStepCont
INTEGER iits,ntits
REAL(8) dtsys,dt_fin
LOGICAL syslink
END SUBROUTINE DTStepCont
...
...
END INTERFACE
END MODULE LINK_CONT
```

### 3. Making Standalone Containment Analysis Program and Verification of DLL

Standalone CONTEMPT executable code without interface with MARS can be generated by making a simple main driver which calls the interfacing routines as follows.

```
PROGRAM MAIN
USE MSFLIB ! Use Digital Visual Fortran Library
USE LINK_CONT ! Use Interfacing Module
CALL FileOpenCont (ncase,ndata,lpool)
CALL ReadInputCont (ncase,ndata,lpool,ifct,isw)
CALL InitCont
CALL TransetCont
DO WHILE(.not.err)
CALL DTStepCont(iits,ntits,dtsys,dt_fin,syslink)
CALL TimeAdvanceCont(out_ctl,t_time,n_count,print_mode,err,syslink,con_no,con_id,c_me)
ENDDO
STOP
END
```

When a standalone driver with the interface module are compiled and linked together, an executable binary file is generated. By placing the executable and dynamic link library files in the same directory, the main drivers can proceed the transients by calling the routines in a DLL file. A stand-alone containment calculation was performed using a typical plant data with mass and energy release for LBLOCA. Comparisons with results of the original CONTEMPT code showed the validity of DLL.

#### 4. Coupling with System Code, MARS 1.3

Coupling with MARS code was realized by calling the DLL routines at the appropriate calculation step. MARS code calls the containment input process and initialization routines during MARS input process and initialization process respectively. During the transient, the MARS calculation should be performed by sharing the data between CONTEMPT DLL. The previous study of merging codes using UNIX process control [3] technique showed that the numerical solution procedures of the two codes were so much different each other and the implicit coupling was not appropriate. Since the explicit coupling is much simpler than implicit coupling, it was also adopted in this study to couple the codes explicitly during a transient advancement.

In the MARS code, the containment is modeled as the pseudo time-dependent volumes, connected to each location of the breaks, while the break flows from reactor coolant system serves as source/sink in CONTEMPT DLL. The total pressure, liquid and vapor enthalpy, quality, etc. in the vapor region of the compartment adjacent to the break is calculated by the CONTEMPT DLL routine, and is transferred to MARS 1.3 through the interface module. MARS code updates the properties in the pseudo time-dependent volumes at every data-exchanging time step. On the other hand, the mass and energy discharge rates of liquid, vapor, air are calculated by the MARS, and is provided to the CONTEMPT routines through the interface module. Time advancement for containment analysis is proceeded using the mass/energy source by calling the CONTEMPT routines in MARS. Overall scheme is shown in Figure 1. The verification of code coupling was performed for the LBLOCA transients of a typical plant design. The break flows and pressures are as shown in Figures 2 and 3. The trends are reasonable and thus the strategy of code coupling is validated.

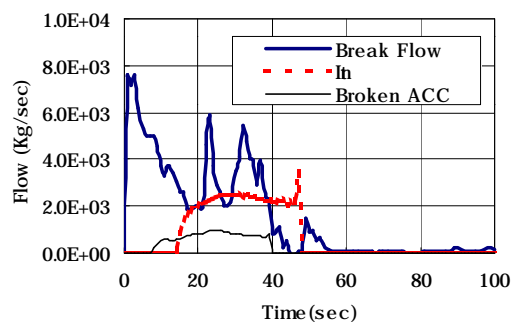


Figure 2. Break and Accumulator Flow during LBLOCA

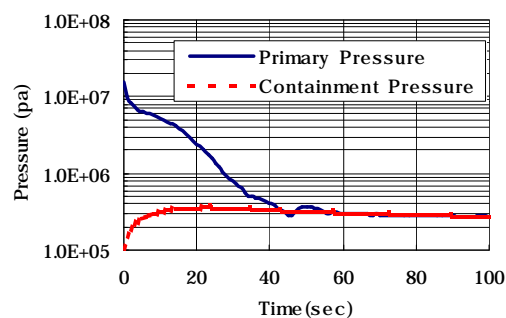
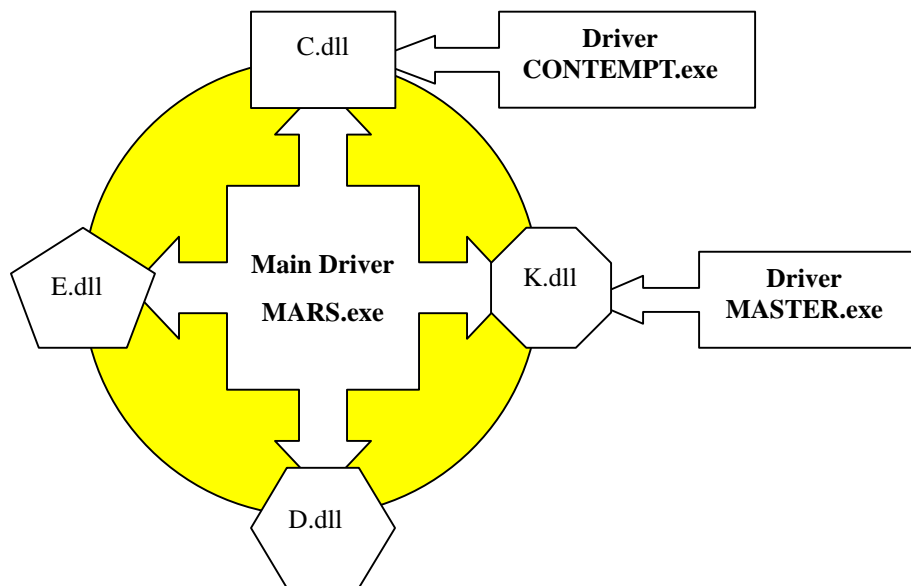


Figure 3. System and Containment Pressure

#### 5. Conclusion

Technique using dynamic-link-library is applied for the coupling the system analysis code with containment analysis code. It was found that the DLL technique is much more convenient than the UNIX process control techniques and effective for Windows operating systems. Since DLL can be used by more than one application and application program can use many DLLs simultaneously, this technique would enable the existing codes to use more broadly with linking others. An example of extending an

application program, MARS, is shown in Figure 4. Another benefit of the use of DLL technique is the enhancement in the code maintenance. Each DLLs can be independently maintained from the main driver or other DLLs until the interfacing requirements are not changed. Since the main driver can use DLLs without recompiling and linking, it is possible to consolidate the improvements on the other functional areas in the code system without any extra efforts.



C.dll : Containment Analysis Module      K.dll : 3 Dimensional Neutronics Module  
 E.dll : Environmental Analysis Module      D.dll : Dose Rate Analysis Module

Figure 4. Concept of Linking with Other Various Functional Codes

### References

1. R. P. Martin, "RELAP5/CONTAIN coupling", RELAP5 International Users Seminar, Idaho Falls(1993)
2. Al Geist, et. al., "PVM: Parallel Virtual Machine, A Users' Guide and Tutorial for Networked Parallel Computing "MIT Press (ISBN 0-262-57108-0).
3. C.E.Park, et. al., "Merged Version to CONTEMPT", App. B - Topical Report for Best Estimation of Emergency Core Cooling System (1997)
4. S. Y. Lee, J. J. Jeong, and S. H. Kim, "COBRA/RELAP5: a merged version of the COBRA-TF and RELAP5/MOD3 codes", Nuclear Technology Vol. 99, p. 177 (1992)
5. W.J.Lee, B.D.Chung, J.J.Jeong, K.S.Ha , "Development of a Multi-Dimensional Realistic Thermal-Hydraulic System Analysis Code, MARS 1.3 and Its Verification", KAERI/TR-1108/98 (1998)
6. Digital Equipment Corporation, "Digital Fortran Language Reference Manual" (1997)
7. Y.D.Hwang,B.D.Chung,B.H.Cho,M.H.Chang, Ik Chung, "PCCS Analysis Model for the Passively Cooled Steel Containment", Journal of the Korean Nuclear Society, Vol.30, No.1, pp26~39, February (1998)

### Acknowledgments

This work was sponsored by the MOST (Ministry of Science and Technology) Nuclear R&D program.