The MIDAS Computer Code through an Integration of the Restructured MELCOR

S.H.Park a, D.H.Kim a, Y.M.Song a , S.W.Cho b
a T/H Safety Research Division, KAERI, P.O.Box 105, Yusong, Daejon, 305-600 Korea, shpark2@kaeri.re.kr
b Korea Radiation Technology Institute Co. Kusongdong 19, Yusong, Daejon, 305-353 Korea

1. Introduction

KAERI has developed a localized severe accident code, MIDAS, based on MELCOR. MELCOR has difficulties in understanding the variables due to pointer variables for a fixed-size storage management. It deteriorates the readability, maintainability and portability of the code. To improve these difficulties, it is necessary to restructure the storage management.

As most important process for a localized severe accident analysis code, it needs a convenient method for a data handling[1]. So, it has used the new features in FORTRAN90 such as a dynamic allocation for the restructuring[2]. The restructuring of the data saving and transferring method makes it easy to understand the code. As a first step, a restructuring for each package was developed and tested. And then an integration of each restructured package was processed one by one.

In this paper, the integration includes the whole packages. Among them, some packages use their own data within a package and others use data with other packages. The verification was done through comparing the results before and after the restructuring.

2. Integration of Restructured Packages

2.1 Construction of the Module

MELCOR reserves and uses four data types for the data storage and transfer effectively. Data are saved in the most effective way within a fixed array and they are transferred through two steps. During these processes, pointer variables are applied differently according to the packages, and they point out a specific location among the database[3].

The locations that the pointer variables convey can be calculated in the subroutines within the packages. Based on these contents, a module is constructed for each package.

In MELCOR, the subroutines pass data through two steps. In these steps, the pointer variables are used to connect the data with the next level subroutines. After these steps, the actual-meaning variables are applied instead of the pointer variables. Based on these structures, the actual-meaning variables were constructed as a module for each package.

2.2 Subroutine Reorganization

In advance of the entire integration of each package (BUR, CF, CVH, DCH, EDF, etc.), a restructuring of each package was done. In the newly constructed module, all the variables were transformed into direct variables which have a meaning related to a package.

The direct variables are used instead of the local variables which are calculated with pointer variables and passed through an argument.

The execution files for each package BURMELCOR, CFMELCOR, MPMELCOR, TPMELCOR, etc., were made and verified[4].

2.3 Integration of each Package

After a reorganization of each package, it is necessary to integrate them into one name MIDAS 1.0. The first integrating step was the independent package step such as BUR, ESF, MP TF, and TP. As the next step, the CVH package was integrated which shares its own data with other packages[5]. And then the integrating step was done from simple packages to complicated packages. As a final step, RN1 package was integrated. Among each package, the data were shared in other packages. So, the variables for the shared data can be replaced with direct variables by using a module without relations in other packages.

In the integrating step, about 1200 subroutines restructured within 24 packages were integrated. The whole packages were integrated, and the execution file named MIDAS 1.0 was completed. The integrating steps are compared in Figures from 1 to 4.

3. Result and Verification

In advance, a restructuring of each package was verified by the former progress[4]. In order to verify the integrated results, ten-step processes were implemented. At first, the results which were integrated with a few independent packages were verified by comparing major variables in all the packages before and after the integration. And then the results which integrated the CVH package were verified[5]. Finally, MIDAS 1.0 which included the RN1 package was verified in the same way.

In general, the values of the major variables were the same. Though it was found that some variables were different. But it was confirmed that it was caused by a base code (MELOR) bug itself. Therefore, except for the base code bug, the integration was confirmed to be successful.

As shown in Figures 5 to 6, the integrated results were the same as the package-wised results.

4. Conclusions

To integrate restructured variables, a data management process was analyzed for the entire MELCOR code. In this procedure, the restructured variables which were spread at each restructuring step were integrated using the reconstructed modules, and the subroutines at each restructuring step were integrated.

By comparing the important variables, it was confirmed that the results were the same but differences appeared in the CPU time comparison. They are assumed to be caused by an array transformation in some subroutines. Based on the various outputs, the consulting from other expets, it has been confirmed that the integration of the restructuring was done right.

Therefore, the appropriateness of the integrating was verified. Through the integrating process, the base was constructed for an easy grasp of the variables everywhere in the code and it became easy for a code improvement and for an addition of the new models.

REFERENCES

[1] D.H.Kim, et al., Experimental and Analytical Research on Severe Accident Phenomena, KAERI/RR-2216/2001, May, 2002.

[2] A Multi-Dimensional Thermal-Hydraulic System Analysis Code, MARS 1.3.1, Vol.31, Number 3, pp.344-363, June 1999.

[3] D.H.Kim, S.H.Park, Analysis of MELCOR Code Structure, KAERI/TR-1543/00, June, 2000.

[4] S.H.Park, D.H.Kim, S.W.Cho, A Restructuring of the CAV and FDI Package for the MIDAS Computer Code, Proceedings of the Korean Nuclear Spring Meeting, May, 2006.

[5] S.H.Park, D.H.Kim, S.W.Cho, A 1^{ST} Step Integration of the Restructured MELCOR for the MIDAS Computer Code, Proceedings of the Korean Nuclear Spring Meeting, Autumn, 2006.

```
C
C   edit of the ICS  model if active
      IF (IVICPL(1).NE.0) THEN
        WRITE (NOW,304) ABS(NMIC),(ICVNUM(IVICPL(I)),I=1,NMIC)
        WRITE (NOW,504)
        WRITE (NOW,305) ICVNUM(IVICDN),ICVNUM(IVICSO),ICVNUM(IVICVN),
     *                  ELVICT,VNTICL,VNTICD,
     *                  VOLIC, PFICMX,IILTMP,IICNCN,IICDPR
```

Figure 1. Subroutine CNDEDT (Before restructuring)

```
!
       USE ESF_MDL
         . . . . . .
!!  edit of the ICS  model if active
      IF (IVICPL (1) .NE.0) THEN
        WRITE (NOW,304) ABS(ESF_NMIC),(ICVNUM(IVICPL(I)),I=1,ESF_NMIC)
        WRITE (NOW, 504)
        WRITE (NOW,305) ICVNUM(IVICDN),ICVNUM(IVICSO),ICVNUM(IVICVN),  &
                        ESF_ELVICT, ESF_VNTICN, ESF_VNTICD,            &
                        ESF_VOLIC, ESF_PFICMX, IILTMP,IICNCN,IICDPR
```

Figure 2. Subroutine CNDEDT in restructuring of ESF

```
!
       USE CVH_MDL
         . . . . . .
!   edit of the ICS  model if active
      IF (IVICPL (1) .NE.0) THEN
        WRITE (NOW,304) ABS (NMIC), (CVH_VL(IVICPL(I))%ICVNUM,I=1,NMIC)
        WRITE (NOW,504)
        WRITE (NOW,305) CVH_VL(IVICDN)%ICVNUM, CVH_VL(IVICSO)%ICVNUM,  &
                                                 CVH_VL(IVICVN)%ICVNUM,&
                        ELVICT, VNTICL, VNTICD,                        &
                        VOLIC, PFICMX, IILTMP, IICNCN, IICDPR
```

Figure 3. Subroutine CNDEDT in restructuring of CVH

```
!
       USE CVH_MDL
       USE ESF_MDL
         . . . . . . .
!!  edit of the ICS  model if active
      IF (IVICPL (1) .NE.0) THEN
        WRITE (NOW,304) ABS(ESF_NMIC),(CVH_VL(IVICPL(I))%ICVNUM,       &
                                                      I=1,ESF_NMIC)
        WRITE (NOW, 504)
        WRITE (NOW,305) CVH_VL(IVICDN)%ICVNUM, CVH_VL(IVICSO)%ICVNUM,  &
                                                 CVH_VL(IVICVN)%ICVNUM,&
                        ESF_ELVICT, ESF_VNTICN, ESF_VNTICD,            &
                        ESF_VOLIC,ESF_PFICMX, IILTMP,IICNCN,IICDPR
```

Figure 4. Subroutine CNDEDT in MIDAS (Integration)

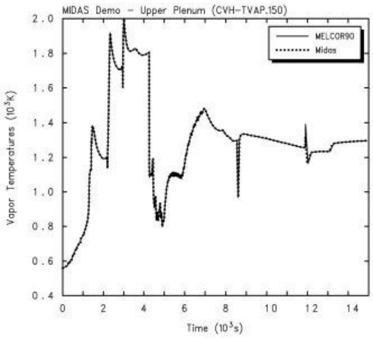

Figure 5. Temperature in Upper Plenum (CVH.150)

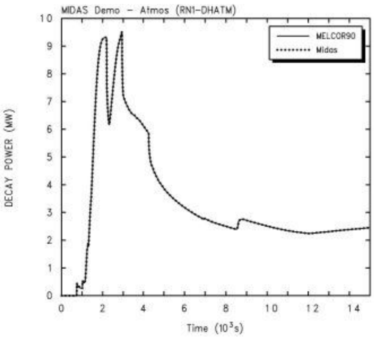

Figure 6. Decay Heat in Atmosphere