

## Construction of TH code development and validation environment

Hyungjun Kim\*, Hee-Kyung Kim and Kyoo-Hwan Bae

Korea Atomic Energy Research Institute, Daedeok-daero 989-111, Yuseong-gu, Daejeon, 34057

\* Corresponding author : jun@kaeri.re.kr

### 1. Introduction

The development of thermal-hydraulic (TH) system code for nuclear reactor requires much time and effort, also for its validation and verification(V&V). In previous, TASS/SMR-S code (hereafter TASS) for SMART is developed by KAERI through V&V process [1, 2]. On the way of code development, the version control of source code has great importance. Also, during the V&V process, the way to reduce repeated labor- and time-consuming work of running the code before releasing new version of TH code, is required. Therefore, the integrated platform for TH code development and validation environment is constructed. Finally, Redmine, the project management and issue tracking system, is selected as platform, Mercurial (hg) for source version control and IVS (Integrated Validation System) for TASS is constructed as a prototype for automated V&V. In this paper, each component of code development and validation system, i.e. IVS and Mercurial will be introduced and Redmine, the integrated platform of IVS and Mercurial, will be explained later.

### 2. IVS

IVS (Integrated Validation System) for TH system code is an automated code validation system. With Microsoft Word/Excel, Originlab Origin and LaTeX, IVS can make simple code validation report. It consists of Origin Labtalk and Origin C script [3, 4] for data organizing and plotting (Fig. 1) and Fortran for converting data format with several DOS batch files.

For using IVS, user have to select SET (Separate Effect Test) or IET (Integral Effect Test) item for code validation. Then run the batch files to run the code with selected list. After TH code run being finished, user have to run a batch file for plotting and making document. Be careful that the list used in code run and Excel file for IVS run (Fig. 2) must be same. Then, user can get Origin project and Word document (Fig. 3) for each test cases which contains the comparison result of experiment and code.

The resultant Word document, however, shows graph only without text due to its complexity. User who has knowledge on each test and validation result can make decision to outcome but can be hard to beginner. Thus, IVS user can also generate document including simple description about test and figure caption by LaTeX. Use of 'subfiles' package in LaTeX and figures generated in

Origin, user can make or edit text on each test case also for figure caption (Fig. 4).

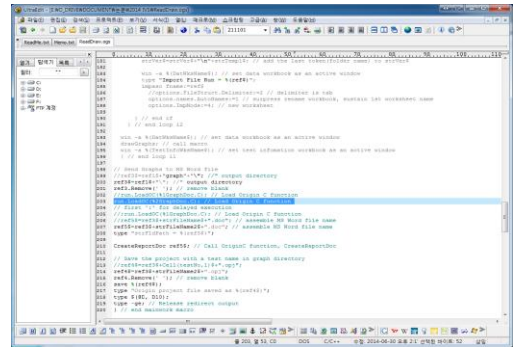


Fig. 1. Example of Origin script

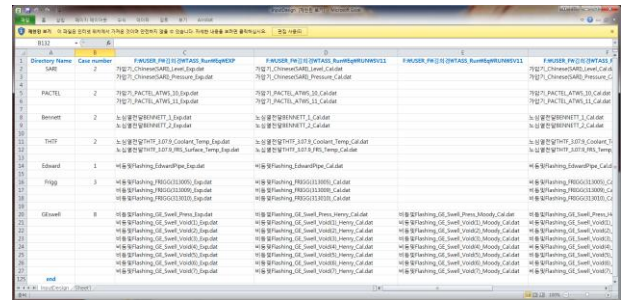


Fig. 2. Example of Excel input file

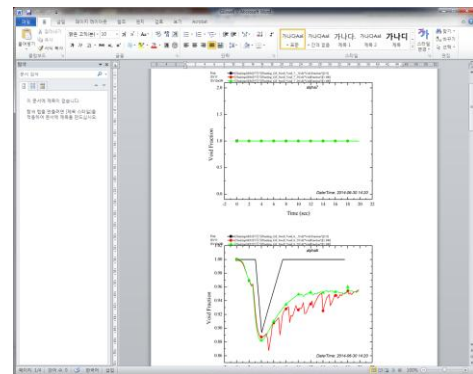


Fig. 3. Graphs file (Word)

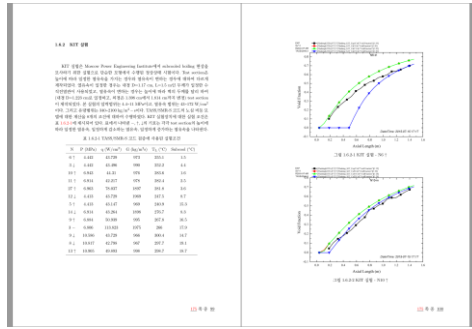


Fig. 4. Output document (PDF from LaTeX)

### 3. Mercurial

Mercurial (hg) is a free, distributed source control management tool [5]. In previous, SVN (Subversion) was used as source control of TASS code [6]. The key difference between SVN and Mercurial is the way of handling the repository as shown in Fig. 5 and Fig. 6[7]. In SVN, the repository where the source code is stored and controlled is in the central server and developer have to change code after copying the source to their local machine. After V&V, the changed code is committed to central server by administrator. However in Mercurial, there also exist central server but the individual developer can also create and control version of his own distributed repository. If a developer determines his source code is mature enough to ‘push’ to mainstream code, the changeset can be pushed to central server, just exactly same as ‘commit’ in SVN. Also, there’s lot of functions which can help users to control source code like merge/branch/push/pull/update/incoming/outgoing, etc.

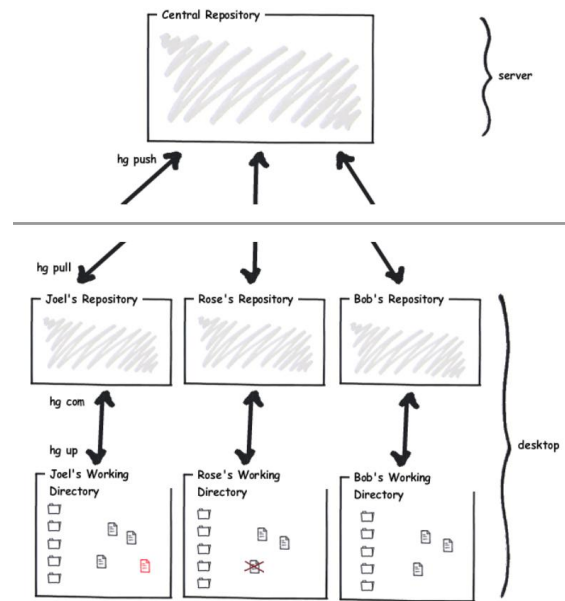


Fig. 6. Structure of Mercurial

### 4. Redmine

Redmine is a flexible project management web application which containing issue tracking system, wiki and many other features with useful plugins [8]. There are lot of issue or bug tracking system like Jira, Mantis, Bugzilla, etc. but Redmine is selected because of its intuitive user interface and simplicity. Natively, Redmine can easily link to source version control tools like SVN, Mercurial, Git, etc. by back and forth. Also, Redmine has many useful plugins which can support code development, also to project management.

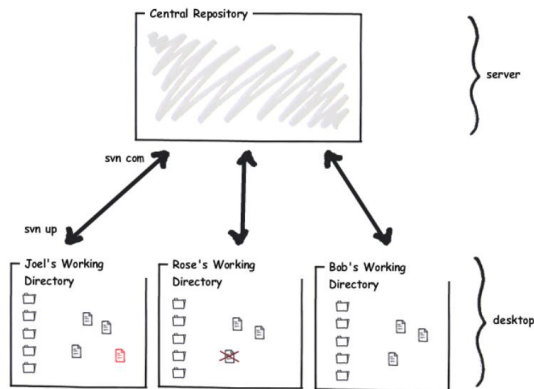


Fig. 5. Structure of SVN

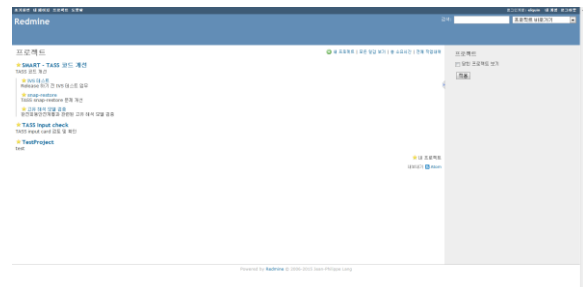


Fig. 7. Redmine (Project page)

### 5. Application to V&V process

First of all, the IVS was developed for automated V&V process. Through simple text editing, IVS can run hundreds of SET/IET cases and after that, the resultant figures are used for documentation by Word or LaTeX. However, the way to respond of developer group from review of IVS document or user report to TH code was not efficiently organized. Hence, Redmine is adopted as a communication tool between user and developer, also for effective process of TH code development.

Now, the improved code validation process is shown below;

- I. Before new code released, select SET or IET set to test the new code
- II. Run the IVS with given input and code
- III. Make IVS document and notify to reviewer
- IV. After reviewing IVS result, user post issue about bug/error or request new feature in Redmine
- V. Modify or improve code from issues and release new

## **6. Conclusion**

Integrated TH code validation system, IVS and code development & management environment are constructed. The code validation could be achieved by a comparison of results with corresponding experiments. IVS is useful before release a new code version. The code developer can validate code result easily using IVS. Even during code development, IVS could be used for validation of code modification. Using Redmine and Mercurial, users and developers can use IVS result more effectively. With Mercurial, developers can make changes on TH code freer than SVN. And Redmine can provide an integrated development and management platform to manager and code developer, also for users.

## **Acknowledgments**

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No. 2012M2A8A4025980).

## **REFERENCES**

- [1] S.H. Kim, TASS/SMR-S code technical report, vol. 1: Code Structure, Models and Numerical Analysis Method, KAERI, 2011.
- [2] Y.-J. Chung, TASS/SMR-S code technical report , vol. 3: Code Validation and Verification, KAERI, 2011.
- [3] Origin On-line manuals (LabTalk, Origin C programming).
- [4] <http://ocwiki.originlab.com>.
- [5] <http://mercurial.selenic.com>.
- [6] <http://tortoisesvn.net>.
- [7] <http://hginit.com/00.html>.
- [8] <http://www.redmine.org>.