A System Structure for a VHTR-SI Process Dynamic Simulation Code

Jiwoon Chang, Youngjoon Shin, Cheung Youn*, Jihwan Kim, Kiyoung Lee, Wonjae Lee, Jonghwa Chang
Korea Atomic Energy Research Institute150 Dukjin-dong, Yuseong-gu, Daejeon, Korea 305-600
**Chungnam National University, 220 Gung-dong, Yuseong-gu, Daejeon, Korea 305-764*
E-mail;jwjang73@kaeri.re.kr, Tel; +82 42 868 8369, Fax; +82 42 868 8549

1. Introduction

The VHTR-SI process dynamic simulation code embedded in a mathematical solution engine is an application software system that simulates the dynamic behavior of the VHTR-SI process. Also, the software system supports a user friendly graphical user interface (GUI) for user input/out. [1]

Structured analysis techniques were developed in the late 1970s by Yourdon, DeMarco, Gane and Sarson for applying a systematic approach to a systems analysis. [2] It included the use of data flow diagrams and data modeling and fostered the use of an implementation-independent graphical notation for a documentation.

In this paper, we present a system structure for a VHRT-SI process dynamic simulation code by using the methodologies of structured analysis.

2. Methods and Results

System analysts use process models (i.e. data flow diagrams, DFDs) to show an information flow and processing in a system. The model usually starts with a context diagram showing a system bubble surrounded by an external environment identified by external entities. Data flows bring information to and from a system process. A process can explode to a child diagram that presents its details using data stores, data flows and sub processes. The diagram leveling process allows for complex systems to be partitioned easily into a stack of simple diagrams with a rigorous balancing of the information between levels. Information structures are defined in an associated data dictionary.

2.1 Context diagram

A system context diagram of a VHTR-SI process dynamic simulation software system is presented in Figure 1. As shown Figure 1, the context diagram shows the highest level view of a system and the system inputs and outputs from/to external factors.

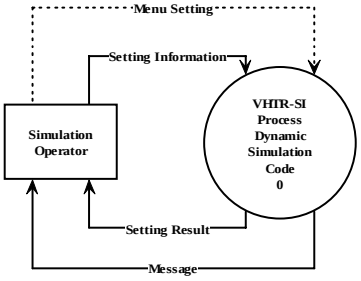

Fig. 1. Context diagram for the VHRT-SI process dynamic simulation code (Level 0).

2.2 Data flow diagram

DFDs show the flow of data from external entities into a system, how the data is moved from one process to another, as well as its logical storage.

Figure 2 presents a DFD for a VHTR-SI process dynamic simulation software system using the Gane and Sarson notation. [3] There are only four symbols:

1. Squares representing external entities, which are sources or destinations of data.
2. Circle representing process, which take data as input, do something to it, and output it.
3. Arrows representing the data flows, which can either, be electronic data or physical items.
4. Open rectangles representing data stores, including electronic stores such as databases or XML files and physical stores such as or filing cabinets or stacks of paper.

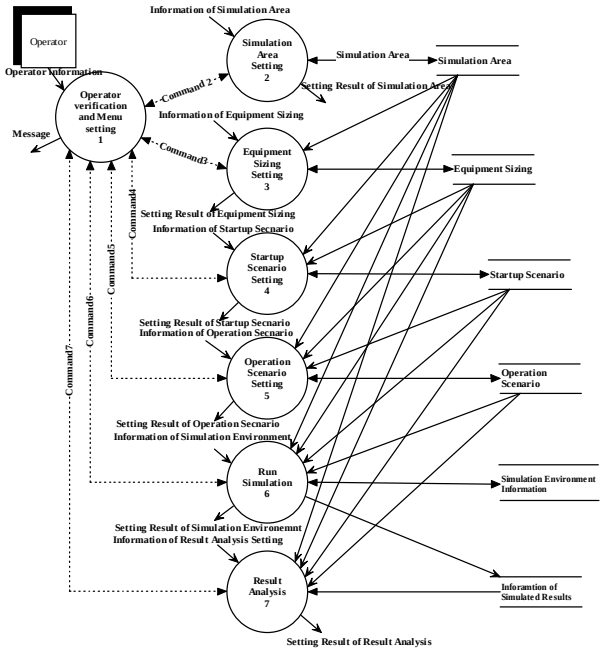

Fig. 2. Level-1 DFD for a VHRT-SI process dynamic simulation software system.

As shown Figure 2, there is one external entity, seven processes and six data stores in the level-1 DFD for the VHTR-SI process dynamic simulation software.

Figure 3 presents a Level-2 DFD for the operator verification and the menu setting. As shown in Figure 3, the process of the menu setting is receiving data information for the menu setting and data information for the six optional commands.

The optional commands are a simulation area setting, an equipment sizing setting, a startup scenario setting, an operation scenario setting, a run simulation and a result analysis.

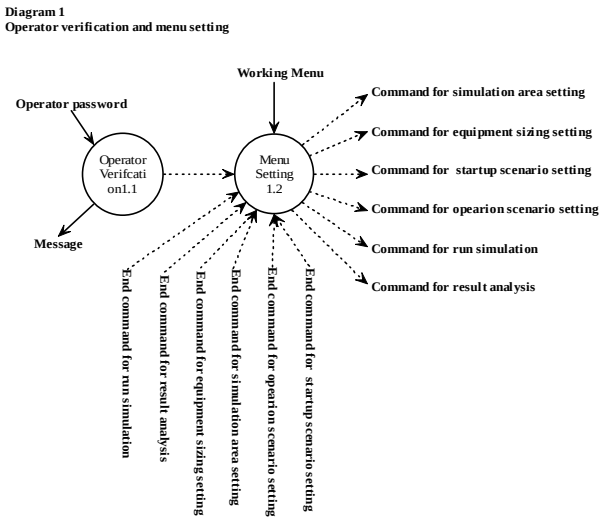

Fig. 3. Level-2 DFD for the process 1 of operator verification and menu setting.

2.3 Process logic specification

The lowest level of processing is called the functional primitive level, and this primitive level has been traditionally used as the starting point for a systems design.

Process logic specification (sometimes called a mini-spec) is a description of all the functional primitive processes. A process logic specification can be expressed as structured English, decision tress, or any of the many other techniques used to describe how data flows are being changed. Figure 4 presents an example of a process logic specification for the primitive process 1.2 in the Level-2 DFD.

Process logic specification

Name
1.2
Title
Menu setting
Input/Output
Menu
Body
- Receive data for the menu setting from operator
- Call a proper program according to the menu setting

Fig. 4. Process logic specification for the level 1.2 DFD of menu setting process.

2.4 Graphic User Interface

The GUI for a VHTR-SI process dynamic simulation software system has been established by using a functional decomposition based on structured analysis methodologies.

Figure 5 presents a GUI for a VHTR-SI process dynamic simulation software system. As shown Figure 5, the system supports a user friendly graphical user interface (GUI) for user input/out.

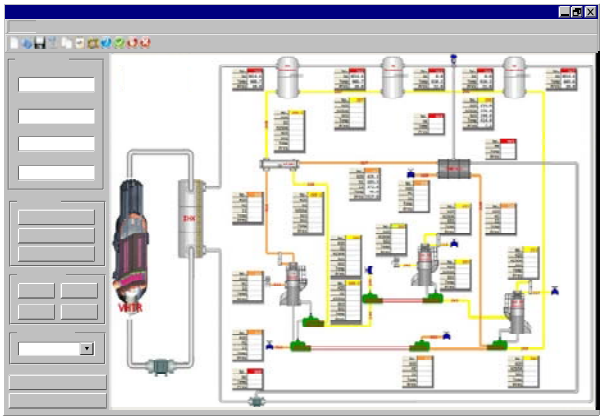

Fig. 5. GUI for a VHTR-SI process dynamic simulation code.

3. Conclusion

The analysis of a system structure for a VHTR-SI process dynamic simulation software system has been done. A structured analysis methodology was used to apply a systematic approach to the software system analysis

For the analysis of the system structure for a VHTR-SI process dynamic simulation software system, various modeling notations were used such as a context diagram, a data flow diagram and a process logic specification.

Acknowledgments

This study has been done under mid-and-long-term Nuclear R&D Projects by the Ministry of Science and Technology, Republic of Korea.

REFERENCES

[1]Jiwoon Chang, Jihwan Kim, Heesung Shin Youngjoon Shin, Cheung Youn, Kiyoung Lee, Jonghwa Chang, Development of an integration program for the dynamic analysis of a VHTR-SI process(I), Korea Nuclear Society, 2007.
[2]Cheung Youn, Software Engineering, saeng-nung publishing company, pp. 400-410, 2006.
[3]http://www.smartdraw.com/tutorials/software/dfd/tutorial_01.htm