Journal of KOSSE. (2019. 6) Vol. 15, No. 1 pp. 1-8
DOI: https://doi.org/10.14248/JKOSSE.2019.15.1.001
www.kosse.or.kr

ISSN (print) : 1738-480X
ISSN (online) : 2288-3592

Application of Sequence Diagrams to the Reverse Engineering Process of the Esf-ccs

Md. Mehedi Hasan, Mohamed Elakrat, Joyce Mayaka, Jae Cheon Jung*

Department of Nuclear Power Plant Engineering, KEPCO International Nuclear Graduate School (KINGS), 658-91 Haemajj-ro, Seosaeng-myeon, Ulju-gun, Ulsan 689-882, Republic of Korea

Abstract : Reverse engineering involves examining a system or component so as to comprehend its structure, functionality, and operation. Creation of a system model in reverse engineering can serve several purposes: test generation, change impact analysis, and the creation of a new or modified system. When attempting to reverse engineering a system, often the most readily accessible information is the system description, which does not readily lend itself to use in Model Based System Engineering (MBSE). Therefore, it is necessary to be able to transform this description into a diagram, which clearly depicts the behavior of the system as well as the interaction between components. This study demonstrates how sequence diagrams can be extracted from the systems description. Using MBSE software, the sequence diagrams for the Engineered Safety Features Component Control System (ESF-CCS) of the Nuclear Power Plant are created. Sequence diagrams are chosen because they are a means of representing the systems behavior and the interaction between components. In addition, from these diagrams, the system's functional requirements can be elicited. These diagrams then serve as the baseline of the reverse engineering process and multiple system views are subsequently be created from them, thus speeding up the development process. In addition, the use of MBSE ensures that any additional information obtained from auxiliary sources can then be input into the system model, ensuring data consistency.

Key Words : Model Based Systems Engineering, Reverse Engineering, Re-engineering, Sequence Diagram, APR 1400, ESF-CCS

Received: December 15, 2017 / **Revised**: July 27, 2018 / **Accepted**: July 7, 2019

* Corresponding author : Jae Cheon Jung, jcjung@kings.ac.kr

1. Introduction

Nuclear power plant (NPP) operators today are not only faced with the problem of replacing legacy instrumentation and control (I&C) systems with new ones, but they must be able to adequately maintain current systems, which are suffering from increased obsolescence. In order to achieve replacement and maintenance, the NPP operator must have a complete understanding of the current system structure and operation. While in most cases, a well-defined configuration management plan ensures that system documentation is kept up to date, occasionally there is a discrepancy between the documentation and the physical implementation of the system. In addition, the system documentation may not be in a form that readily lends itself to the use of modern tools such as MBSE. Reverse engineering of legacy systems is one way of bridging this information gap.

The aim of reverse engineering is to understand a system in detail. Therefore, it is important to have a means of representing the system that eases understanding. To this end, system representations such as UML Sequence Diagrams play an important role [1]. They help users to understand system operation and to visualize the interactions between its objects. When sequence diagrams are either absent or inconsistent, as it is the case for many legacy systems, reverse engineering can be used to extract more accurate models.

There are two main ways of generating sequence diagrams from an operational system: static analysis and dynamic analysis [2]. Static analysis involved analyzing the source code of the system. On the other hand, dynamic analysis involves analyzing the actual running system. These two methods have been researched in numerous studies concerned with the development of sequence diagrams [1][2][3][4]. However, these methods lend themselves to situations in which there is access to the source code or the ability to analyze the system as it runs. This study, however, is concerned with the case where there is no access to the source code and the primary source of system information is the system description.

This paper is structured as follows. First, the main reasons for reverse engineering of NPP I&C systems are discussed. Following this, a brief description of the use of MBSE in the reverse engineering is presented. Lastly, the use of Sequence Diagrams as an entry point to the model based reverse engineering of the Engineered Safety Features Component Control System is explored.

2. Reverse Engineering and Model Based Systems Engineering

2.1 Review of Reverse Engineering

Reverse Engineering is “the process of analyzing a system in order to identify the components of the system as well as to create representations of the system at a higher level of abstraction [5]. Reverse engineering is also called back engineering. It reveals system design and architecture from the object; similar to scientific research, the only difference being that scientific research is about a natural phenomenon. This process has three primary purposes: design recovery, supporting maintenance activities, and system re-engineering.

Design recovery involves analyzing the system in order to create up to date documentation. This process can be used in a situation where system documentation is lacking or inconsistent. While this is challenging, re-documentation gives the system owner an opportunity to make use of modern tools such as MBSE.

One of the prerequisites for the maintenance or modification of an existing system is an understanding the system structure and operation. Studies [3] have shown that a significant amount of time during the maintenance process is spent in an attempt to understand the system. Reverse engineering can, therefore, support reduce this by extracting information that facilitates easy understanding of the system operation.

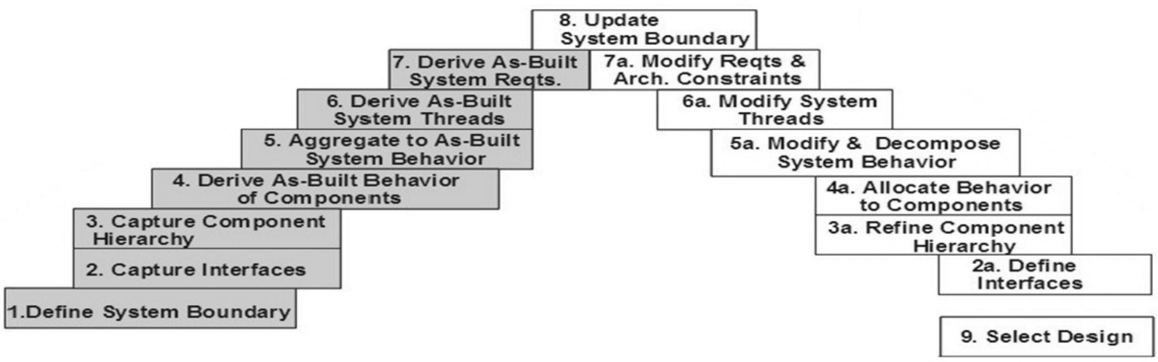
Re-engineering is the examination and alteration of a subject system to reconstitute it in a new form and the subsequent implementation of the new form. Typically it includes some degree of reverse engineering followed by forward engineering or restructuring [5]. This process is conducted when migrating a system onto a new software or hardware. Re-engineering is also performed when carrying out system modification with the aim of incorporating new requirements. In this research, the purpose of the system re-engineering is to create a system prototype on a different hardware platform.

The current ESF-CCS system is based on the PLC platform. However this platform has several disadvantages, such as rapid obsolescence and difficult verification and validation (V&V), as well as regulatory approval. In contrast FPGAs exhibit greater portability, and due to their hardware structure, easier V&V. Therefore, the change in hardware is the driving force behind this re-engineering process.

2.2 MBSE for Reverse Engineering

INCOSE defines MBSE as "the formalized application of modeling to support system requirements, design, analysis, verification and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases [6]. Most MBSE tools offer support for the reverse engineering process. The lifecycle model shown in Figure 1 depicts the principle steps in the reverse engineering process.

Reference [7] defines three models which are necessary and sufficient for the complete specification of a system: behavioral/functional model, interface model, and physical architecture



[Figure 1] Reverse Engineering Life Cycle [1] (component) model. As can be seen from Figure 1, once the system boundary has been defined, the next step is a derivation of the behavior of the system components. System behavior can be modeled using sequence dia–grams, state machine diagrams or activity diagrams. In this study sequence diagrams were chosen as the means of behavioral representation.

3. Methodology

3.1 Description of the ESF–CCS

The ESF–CCS is responsible for the actuating safety related equipment with the aim of safely shutting down an NPP and preventing the release of radioactive materials. It is composed of the electrical and mechanical devices and circuitry, from sensors to actuation device input terminals that are involved in generating those signals, which actuate the required ESF system components. This system consists of four independent channels (Channels A, B, C, and D) which are electrically and physically independent [8].

This system can be used to actuate either a whole ESF system (system level actuation), or a single component within a given ESF–System (component level actuation). The ESF–CCS receives actuation signals from the sources listed in Table 1.

On receiving an actuation signal, the ESF–CCS validates the signal, and then transmits a command to actuate the system or component. Once the component is actuated, the ESF–CCS transmits feedback to the operator as well as alarms, if present. In addition, the ESF–CCS facilitates system and component testing.

<Table 1> ESF–CCS Actuation Signal Sources

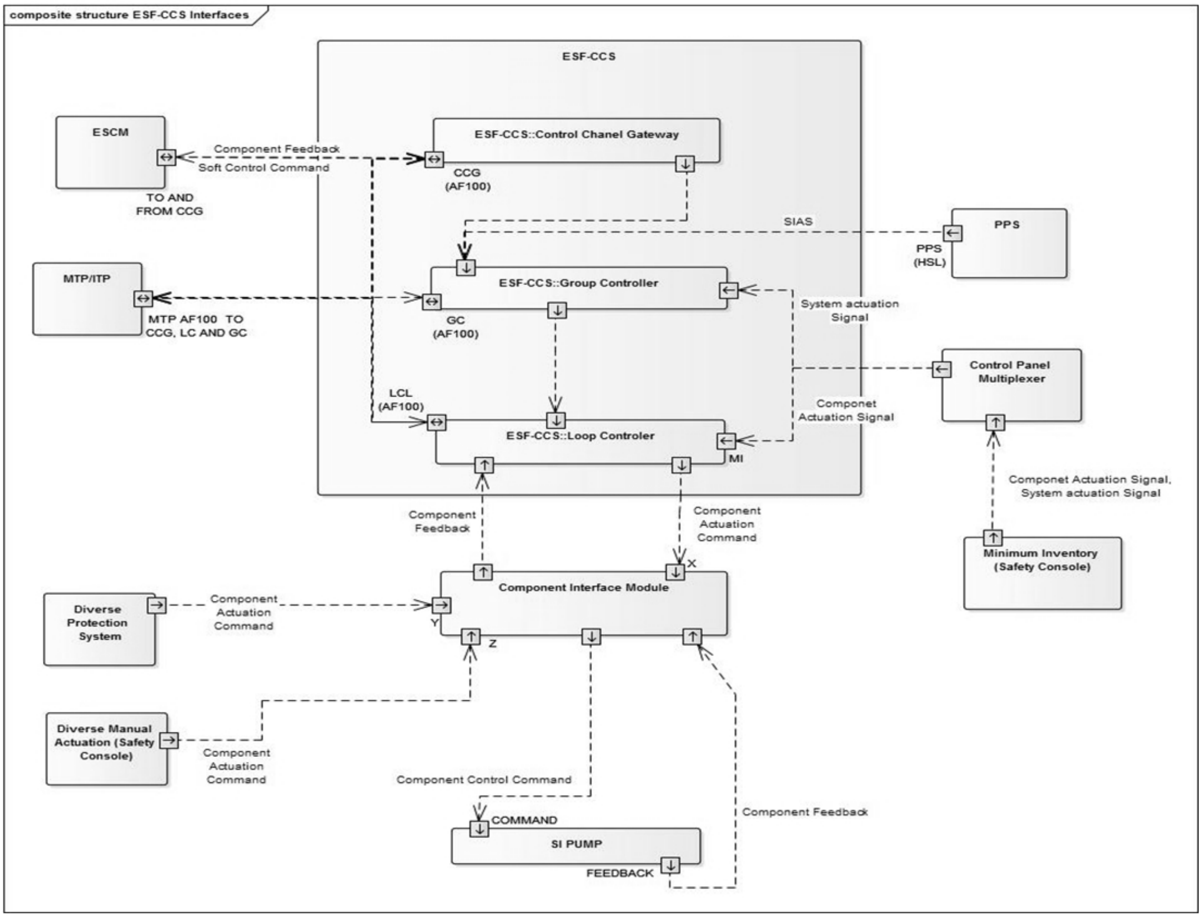
Source	Description
PPS	Automatic system level actuation
ESCM and Manual Confirm Switches	Manual individual component control
Manual System Actuation Switches	Manual system level actuation
Manual Component Control Switches	Manual individual component control for the ESF components
Diverse Manual Actuation Switches	Manual system level actuation that is not dependent upon software
Local Manual Actuation Switches	Local manual switches for channelized ESF actuation.

3.1 Development of Sequence Diagrams

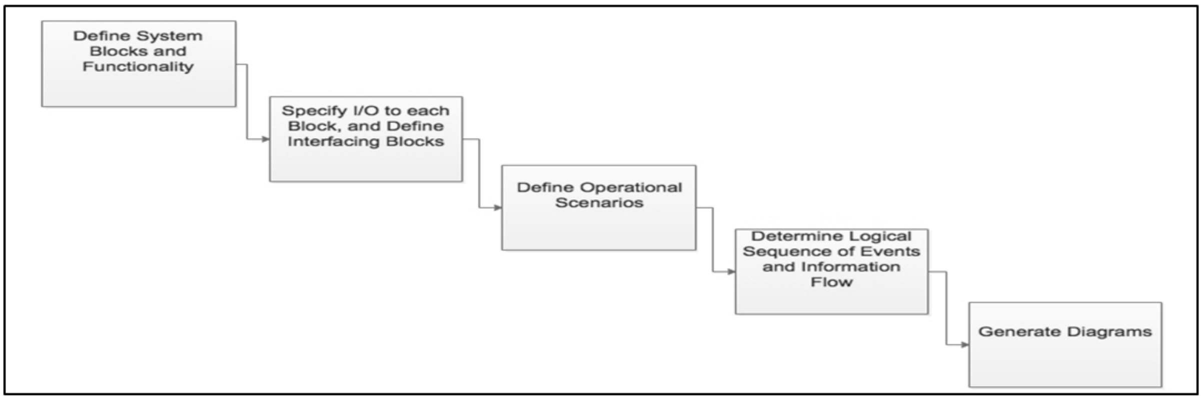
The aim of this study was to use MBSE in the reverse engineering of the ESF–CCS, specifically using sequence diagrams as the initial behavioral model. In order to achieve this the activities were guided by the life–cycle model in Figure 1. The primary source of information of this activity was the system description [8] [9].

First, to define the system boundary, ESF–CCS Interface Diagram was created as illustrated in Figure 2. This diagram was carried out to enable a clear definition of the inputs, outputs and interfaces of the system. This activity was carried out in MBSE software.

Following this, sequence diagrams were generated. In order to generate this behavioral model, the process outlined in Figure 3 was followed. As mentioned previously, the primary source of information was the system description in natural language. This description was complemented by the expert knowledge of system operators. The knowledge from system



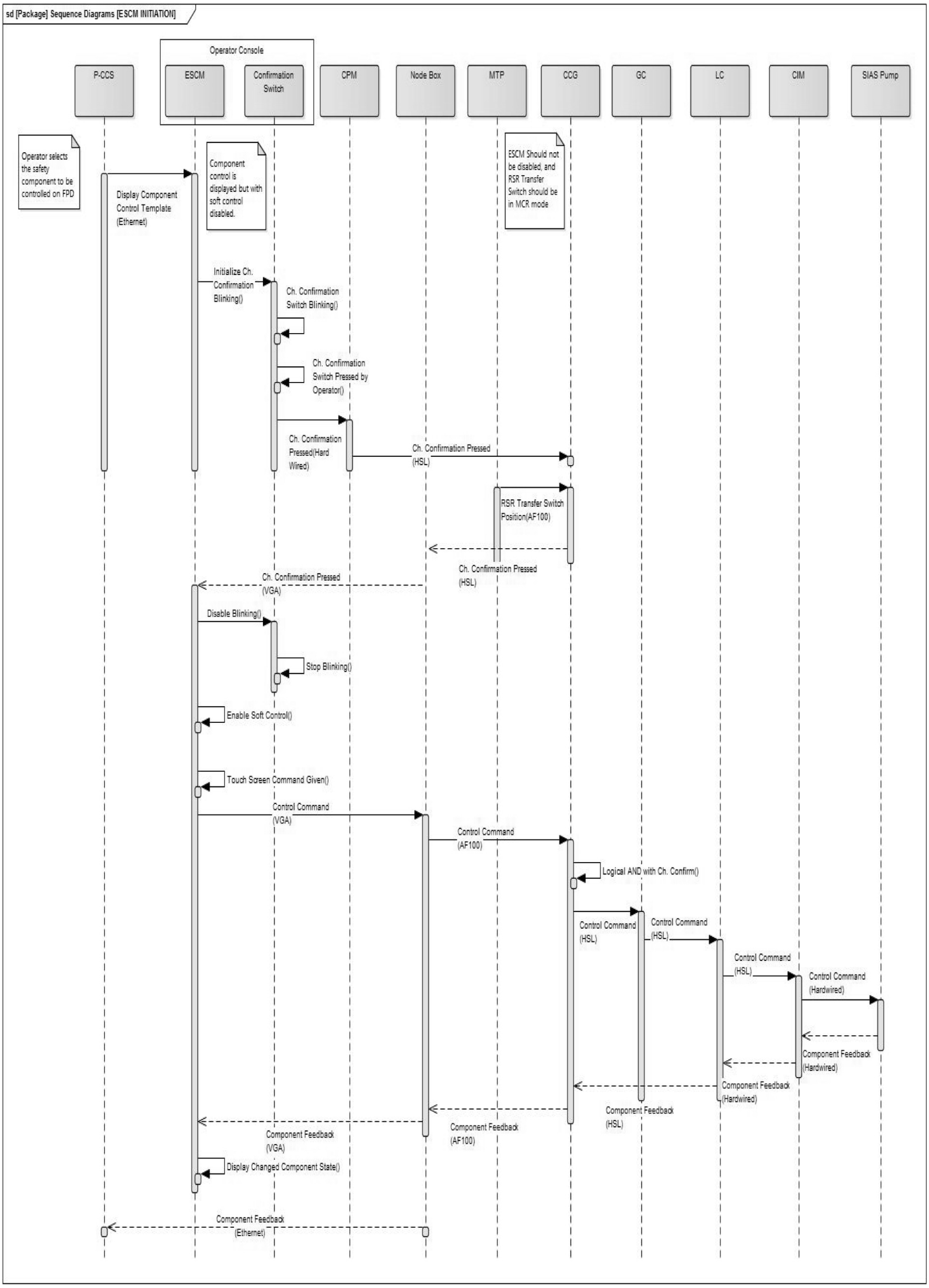
[Figure 2] ESF-CCS Interface Diagram



[Figure 3] Sequence Diagram Generation

experts was of particular importance when defining the different operational scenarios and the associated behavior.

Following the illustrated process, the sequence models were derived, providing a clear under-standing of the system. One of the diagrams created is shown in Figure 4. This diagram depicts the scenario in which the NPP operator



[Figure 4] ESCM Initiation Sequence Diagram actuates a system component (Safety Injection Pump), using the Engineered Safety Features Component Control System Soft Control Module (ESCM). Similar diagrams were created for different operational scenarios.

Due to the fact that these diagrams were created in the MBSE environment, subsequent creation of different system models was greatly facilitated. Furthermore, it ensured consistency in the system models when new information was received and the model was updated.

4. Conclusion

In this study, sequence diagrams were generated from the systems description. Using MBSE software, the sequence diagrams for the ESF-CCS of the Nuclear Power Plant were created. Sequence diagrams were chosen because they are a means of representing the systems behavior and the interaction between components. Baseline of the reverse engineering process and multiple system views were subsequently be created from them, thus speeding up the reverse engineering process.

Acknowledgement

This work was supported by the 2017 Research fund of the KEPCO International Nuclear Graduate School (KINGS), Republic of Korea.

References

1. C. Bennett,D. Myers, M. A. storey, D. M. German, D. Ouellet, M. Salois, and P. Charland, “A survey and evaluation of tool features for understanding reverse-engineered sequence diagrams,” Journal of Software Maintenance and Evolution Research and Practice, vol. 20, pp. 291-315, 2008.
2. Tewfik Ziadi, Marcos Aurelio Almeida da Silva, Lom Messan Hillah, and Mikal Ziane, “A Fully Dynamic Approach to the Reverse Engineering of UML Sequence Diagrams,” in *16th IEEE International Conference on Engineering of Complex Computer Systems (ICECCS)*, 2011.
3. B. Cornelissen, A. Zaidman, and A. Deursen, “A Controlled Experiment for Program Comprehension through Trace Visualization,” in *IEEE Transaction on Software Engineering*, 2010.
4. L. C. Briand, Y. Labiche, and J. Leduc “Towards the Reverse Engineering of UML Sequence Diagrams for Distributed Java Software,” in *IEEE Transactions on Software Engineering*, vol. 32, 2006, pp. 642-663.
5. E. J. Chukofsky and J. H. Cross II, “Reverse Engineering and Design Recovery: A taxonomy,” *IEEE Software*, vol. 7, pp. 13-17, January 1990.
6. INCOSE, “Systems Engineering Vision 2020,” INCOSE-TP-2004-004-02, 2007.
7. Jeff A. Estefan, “Survey of Model-Based Systems Engineering (MBSE) Methodologies,” INCOSE MBSE Focus Group, 2007.
8. KEPCO and KHNP. (2013, February) Safety I&C System for the APR1400. [Online]. https://www.nrc.gov/docs/ML1309/ML13092A005.pd
9. KHNP, APR 1400 Design Control Document Chapter 15, 2014.
10. R. Delamare, B. Baudry, and Y.L Traon, “Reverse engineering of UML 2.0 Sequence

Diagrams from Execution Traces," in *Work-shop on Object-Oriented Reengineering at ECOOP,* Nantes, France, 2006.

11. Tewfik Ziadi, Marcos Aurelio Almeida da Silva, Lom Messan Hillah, and Mikal Ziane, "A Fully Dynamic Approach to the Reverse Engineering of UML Sequence Diagrams," in *Engineering of Complex Computer Systems (ICECCS), 2011 16th IEEE International Conference on*, 2011.