Journal of KOSSE. (2018. 12) Vol. 14, No. 2 pp. 24-32
DOI: https://doi.org/10.14248/JKOSSE.2018.14.2.024
www.kosse.or.kr

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

Development of Simplified DNBR Calculation Algorithm using Model-Based Systems Engineering Methodology

Ibrahim Fathy Awad, Jae Cheon Jung*
KEPCO International Nuclear Graduate School

Abstract : System Complexity one of the most common cause failure of the projects, it leads to a lack of understanding about the functions of the system. Hence, the model is developed for communication and furthermore modeling help analysis, design, and understanding of the system. On the other hand, the text-based specification is useful and easy to develop but is difficult to visualize the physical composition, structure, and behaviour or data exchange of the system. Therefore, it is necessary to transform system description into a diagram which clearly depicts the behaviour of the system as well as the interaction between components. According to the International Atomic Energy Agency (IAEA) Safety Glossary, The safety system is a system important to safety, provided to ensure the safe shutdown of the reactor or the residual heat removal from the reactor core, or to limit the consequences of anticipated operational occurrences and design basis accidents. Core Protection Calculator System (CPCS) in Advanced Power Reactor 1400 (APR 1400) Nuclear Power Plant is a safety critical system. CPCS was developed using systems engineering method focusing on Departure from Nuclear Boiling Ratio (DNBR) calculation. Due to the complexity of the system, many diagrams are needed to minimize the risk of ambiguities and lack of understanding. Using Model-Based Systems Engineering (MBSE) software for modeling the DNBR algorithm were used. These diagrams then serve as the baseline of the reverse engineering process and 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 : Advanced Power Reactor, Core Protection Calculator System, Departure from Nucleate Boiling Ratio, Model-based Systems Engineering, Local Power Density

Received: October 29, 2018 / **Revised**: November 28, 2018 / **Accepted**: January 7, 2019

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

1. Introduction

Modern Instrumentation and Control (I&C) systems in nuclear power plants are complex entities for which different approaches to design and qualification are necessary beyond those that were typically applied to older systems. Frequently, the functional characteristics and performance of previous generations of I&C systems were characterized by models based on physics principles and testing that validated these models [2].

Systems engineering encompasses a holistic view of the design and analysis of engineering problems by complex systems that are represented by mathematical models. In a more generic sense, system analysis has to do with planning, designing, manufacturing, operating and managing to exploit functional, technical, social, environmental, and aesthetic applications.

Model-Based Systems Engineering (MBSE) is a coherent and comprehensive means of consistently arriving at a realizable system effectively and efficiently. MBSE converges layer by layer on a system solution that successfully meets the needs behind the development process. MBSE commonly uses multi-user repository-based modeling tools which provide an environment where a precise and unambiguous worldview of the parts of the system and their behaviours and interactions can be defined and managed [6].

The present paper highlights the Systems Engineering (SE) methodology that has been used to develop a simplified Departure from Nucleate Boiling Ratio (DNBR) calculation algorithm and focuses on the design, coding and testing phases of the simplified DNBR algorithm. DNBR algorithm is part of one of the most critical system for Advanced Power Reactor 1400 (APR 1400) which is Core Protection Calculator System (CPCS). The CPCS calculates the DNBR and the Local Power Density (LPD) to protect the core of the reactor as well as continuously monitor relevant core conditions.

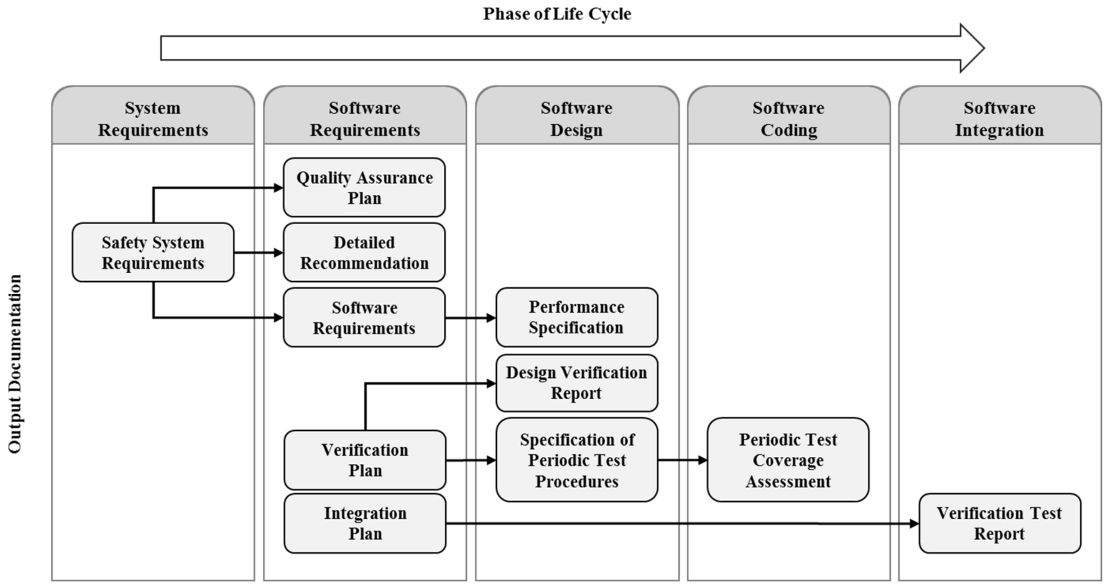
The scope of this paper is just to focus on the systems engineering and management processes that applied to develop the new simplified DNBR algorithm software without explaining many technical details about the algorithm itself.

The complete systems engineering approach that includes hardware and software integration and validation will be future work.

2. Methodology

This paper tries to add a new approach to the software development life cycle model for the safety-critical systems. This approach recommends certain types of modeling diagrams for each development phase integrated with documentation outputs. These modeling diagrams play an important role to minimize the risk of ambiguities and lack of understanding of the CPCS system. Developing the modeling diagrams for each life cycle phase does not consider only as a systems engineering process but also as management activities. These activities were included in the software configuration management plan.

In this study, the current DNBR algorithm was reverse engineered and re-engineered using MBSE approach by following the generic software development life cycle model as illustrated in Figure 1. The algorithm devel-



[Figure 1] Software Development Life Cycle Documentation Based on IEC 880

opment plan starting with an analysis of the CPCS system requirements specification. The results of this activity are a breakdown of the software requirements into functional parts and sub-modules. The design breakdown was developed for the new simplified DNBR algorithm which consists of 7 sub-modules. The software requirements were the inputs to design, coding and verification phases.

The new simplified DNBR algorithm sub-modules are:

1. Input signals collection and normalization;
2. Thermal Hydraulic coolant condition;
3. Power Distribution Synthesis (Axial and Radial);
4. Heat Flux Compensation;
5. Linear heat rate distribution;
6. Hot channel coolant condition;
7. Minimum DNBR calculation.

Applying SE when developing the simplified DNBR algorithm gives the advantages of confirming the need for each sub-module and the anticipated sub-module behaviours before proceeding with the development of the software code and presenting a clear, coherent design to those who will develop, test and deploy the algorithm, thereby maximizing productivity and minimizing error. Modeling diagrams for the algorithm were used to define the test cases and other aspects of the test program to assist in test planning and execution.

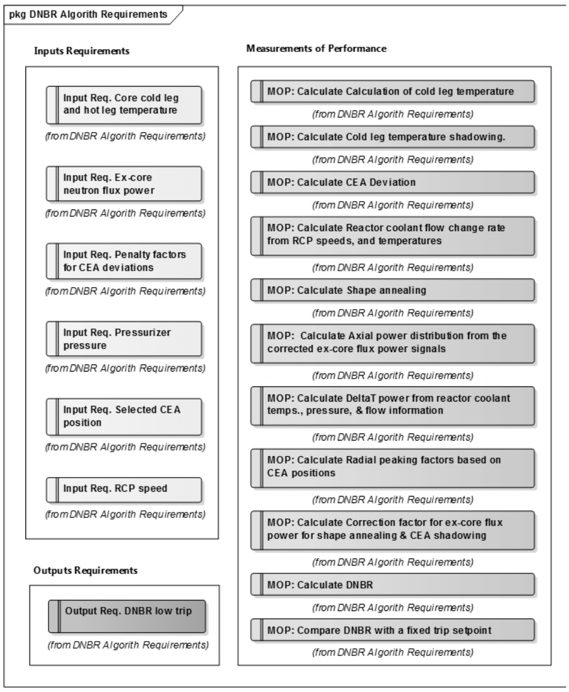
2.1 Software Requirement Phase

The Software Requirements Specifications (SRS) is the specification for the particular software product, program, or set of programs that perform certain functions in a specific environment [5].

The purpose of the Software Requirement Phase was to state the software functional and performance requirements from an external and synthetic view, without stating how they are programmed. In this phase, the CPCS system integrity requirements and verification criteria were defined. This criterion includes measures of performance (MOPs) and technical performance measures (TPMs), and the design constraints.

Package and requirements diagrams were used to organize the functional performance of the CPCS into packages and show how requirements relate to each other. Figure 2 shows the inputs, outputs and MOPs for the DNBR algorithm.

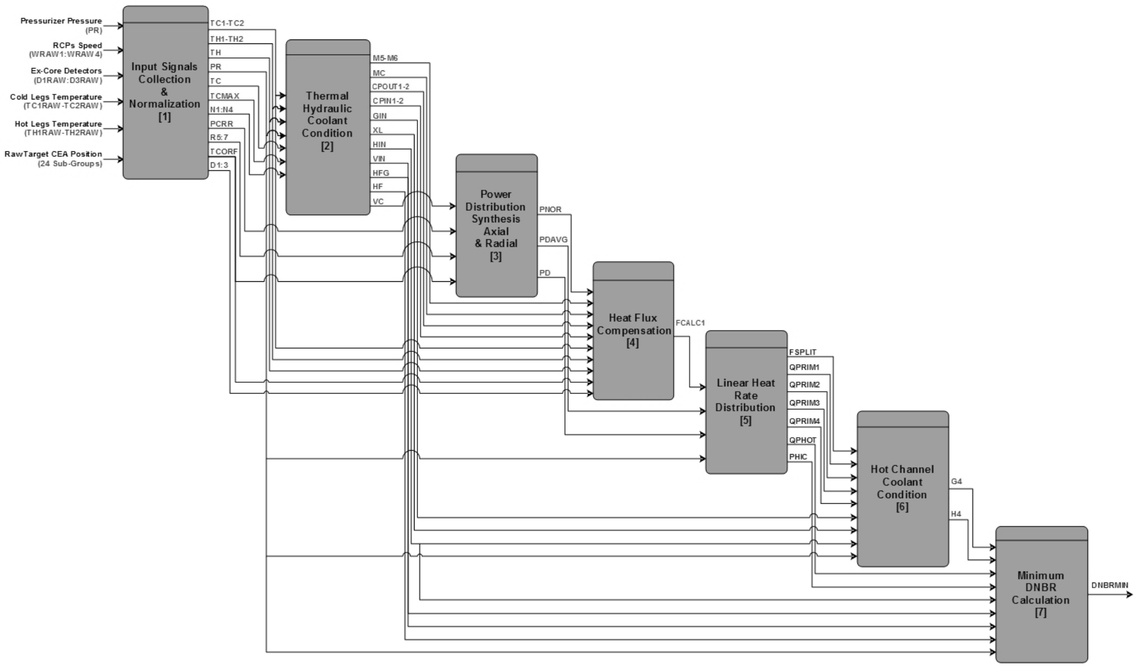
The algorithm mainly calculates the minimum value for DNBR and compare it with the predefined value to generate trip signal and sends it to the Plant Protection System (PPS). The TMP for the algorithm is the accuracy of the calculation. The deviation between the minimum DNBR value calculated by using APR 1400 simulator at three different reactor power levels 100%, 75% and 50% and the value calculated by the algorithm must be negative (The algorithm output must be less than the APR 1400 simulator).



[Figure 2] DNBR Algorithm Requirements

2.2 Software Design Phase

Software Design Document (SDD) a description of software created to facilitate analysis,



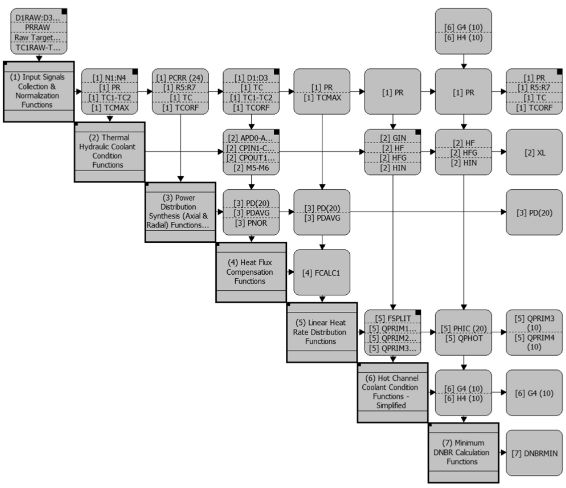
[Figure 3] DNBR Algorithm IDEF0 Diagram

planning, implementation, and decision-making. This design description is -used as a medium for communicating software design information and can be considered as a blueprint or model of the system [8].

SDD was established once the CPCS requirements were addressed. SDD includes the Integration Definition for Functional Modeling (IDEF0) diagram and N Square Diagram (N^2). SDD was written to describe the algorithm and give an overall guidance to the architecture of the algorithm in order to move the software coding phase.

IDEF0 involves a preliminary or high-level design of the main sub-modules with an overall picture of the algorithm to show how the sub-modules fit together. Figure 3 shows the IDEF0 diagram for the algorithm. The diagram shows the interfaces between the seven sub-modules and the main function of each sub-module.

N^2 diagram was used to represent functional interfaces between sub-modules. It was used to systematically identify, define, tabulate, design, and analyse functional interfaces for the sub-modules as illustrated in Figure 4.



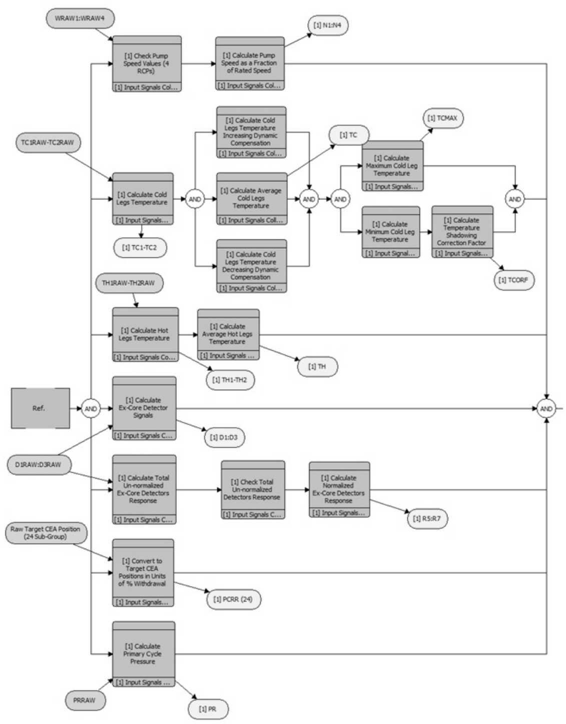
[Figure 4] N^2 Diagram for DNBR Algorithm

2.3 Software Coding Phase

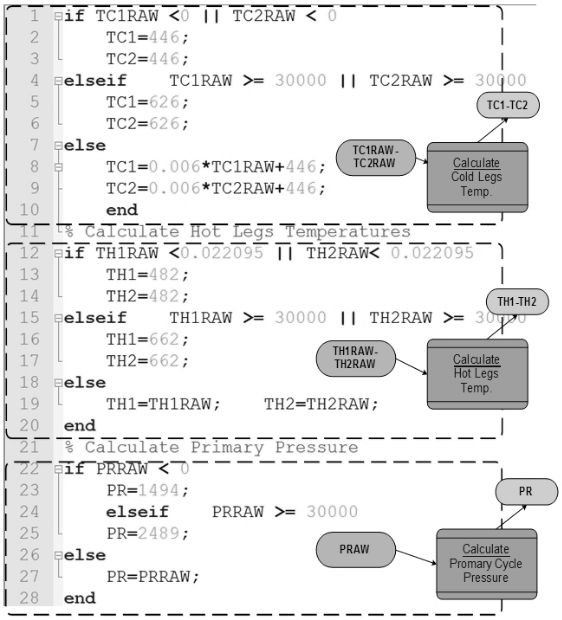
In this phase, the actual code was programmed using MATLAB software but before starting the programming of the algorithm each sub-module in the algorithm was decomposed. For each sub-module, an Enhanced Functional Flow Block Diagram (EFFBD) was developed to show the sub- module tasks identified through functional decomposition and display them in their logical, sequential relationship. EFFBDs played a key role to minimize the time required to develop the code and organize the responsibilities for the programming team.

Each box in the EFFBD diagram represents a sub-function or a task under each sub-module and it has inputs and outputs. To perform the main function of each sub-modules all the sub-functions that included in the EFFBD shall be performed.

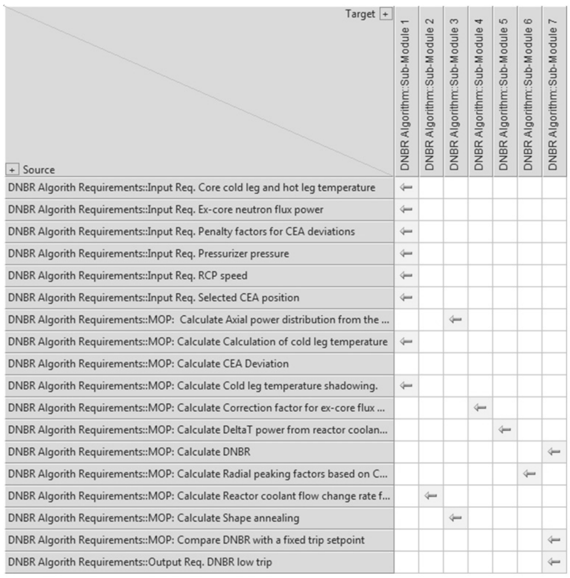
As shown in Figure 5 the main function of



[Figure 5] EFFBD Diagram for Sub-module 1



[Figure 6] MATLAB Code in conjunction with EFFBD for Sub-module 1



Source \ Target	DNBR Algorithm::Sub-Module 1	DNBR Algorithm::Sub-Module 2	DNBR Algorithm::Sub-Module 3	DNBR Algorithm::Sub-Module 4	DNBR Algorithm::Sub-Module 5	DNBR Algorithm::Sub-Module 6	DNBR Algorithm::Sub-Module 7
DNBR Algorith Requirements::Input Req. Core cold leg and hot leg temperature	⇐						
DNBR Algorith Requirements::Input Req. Ex-core neutron flux power	⇐						
DNBR Algorith Requirements::Input Req. Penalty factors for CEA deviations	⇐						
DNBR Algorith Requirements::Input Req. Pressurizer pressure	⇐						
DNBR Algorith Requirements::Input Req. RCP speed	⇐						
DNBR Algorith Requirements::Input Req. Selected CEA position	⇐						
DNBR Algorith Requirements::MOP: Calculate Axial power distribution from the ...			⇐				
DNBR Algorith Requirements::MOP: Calculate Calculation of cold leg temperature	⇐						
DNBR Algorith Requirements::MOP: Calculate CEA Deviation							
DNBR Algorith Requirements::MOP: Calculate Cold leg temperature shadowing.	⇐						
DNBR Algorith Requirements::MOP: Calculate Correction factor for ex-core flux ...				⇐			
DNBR Algorith Requirements::MOP: Calculate DeltaT power from reactor coolan...					⇐		
DNBR Algorith Requirements::MOP: Calculate DNBR							⇐
DNBR Algorith Requirements::MOP: Calculate Radial peaking factors based on C...						⇐	
DNBR Algorith Requirements::MOP: Calculate Reactor coolant flow change rate f...		⇐					
DNBR Algorith Requirements::MOP: Calculate Shape annealing			⇐				
DNBR Algorith Requirements::MOP: Compare DNBR with a fixed trip setpoint							⇐
DNBR Algorith Requirements::Output Req. DNBR low trip							⇐

[Figure 7] MOP Traceability Matrix

sub-module 1 was decomposed into 17 sub functions and the sub-module required inputs and the calculated outputs were defined.

From the programming point of view to perform each sub-function, there were many mathematical equations. These equations were programmed using MATLAB M file code. Using the EFFBDs were important during testing the code because it made the verification process easier to find the error and solve it. Figure 6 shows how the code and EFFBD were in conjunction with each other during the coding phase.

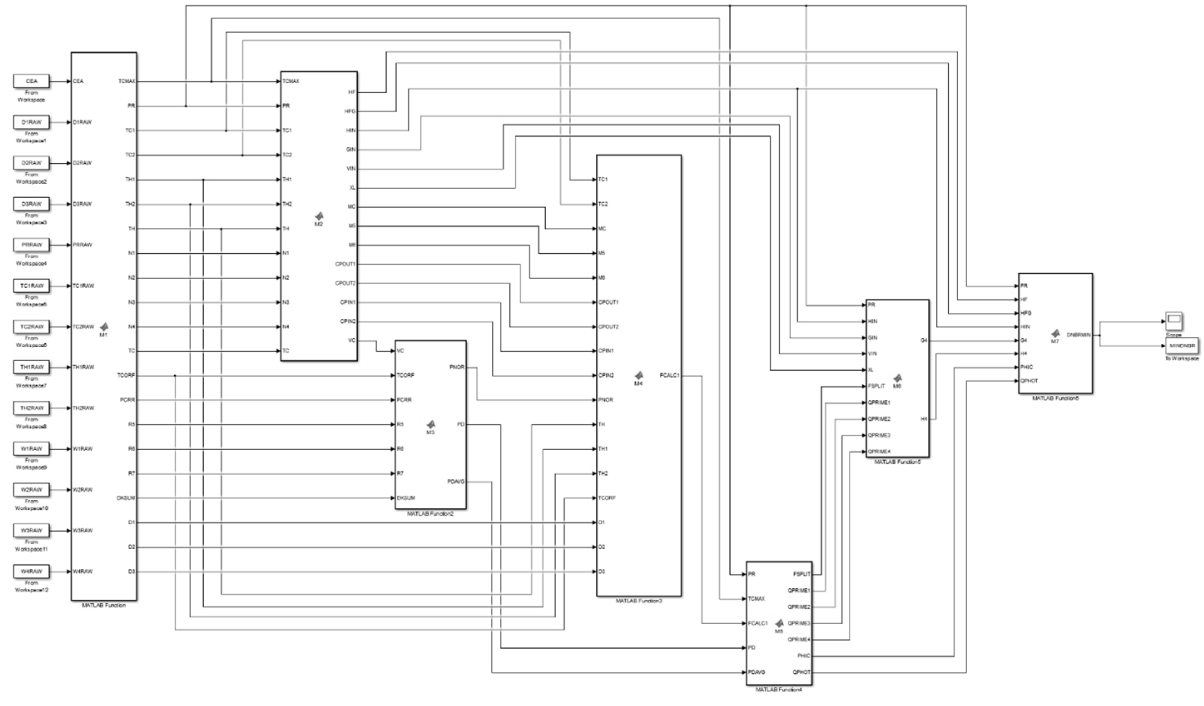
3. Validation and Verification

The purpose of this phase is to validate and verify the algorithm with the inputs, output and MOP requirements. A MOP traceability matrix between the DNBR algorithm requirements and the developed algorithm was developed using Enterprise Architect software. Figure 7 shows the MOP traceability matrix and defines the relation between the sub-modules and the algorithm requirements.

According to the assumptions which are listed during design phase about considering that there is no deviation between the Control Element Assembly (CEA), so CEA deviation calculation function is not covered in the algorithm.

In this phase also MATLAB Simulink was used to verify the new simplified DNBR algorithm as shown in Figure 8. Three test cases were used to verify the minimum DNBR value. Input signals for three test cases 100%, 75% and 50% of reactor power were collected from APR 1400 simulator. The inputs signals were inserted to the Simulink model to calculate the minimum DNBR value.

The algorithm verifies the TPM regarding to the accuracy of the calculation as shown in Table 1.



[Figure 8] MATLAB Simulink Model for DNBR Algorithm

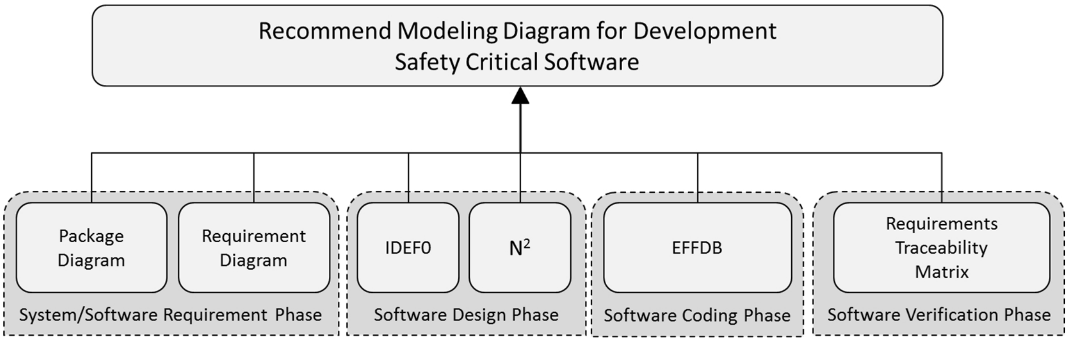
<Table 1> The Algorithm TPM

Minimum DNBR	100% Power	75% Power	50% Power
APR 1400 Simulator	1.6551	2.4344	3.800
Simplified DNBR Algorithm	1.6219	2.4183	3.7946
Percentage Change	−2.01%	−0.66%	−0.14%

4. Conclusion

This paper has attempted to show how a systems modeling language diagrams help to improve the DNBR algorithm development through the software life cycle especially in designing and coding the algorithm. The generic software development life cycle can be updated to include the recommended modeling language diagrams as following in Figure 9. During the software requirements process, package diagram will be helpful to organize the requirements by its type or category. Also, requirement diagrams are important to define the relationship between all requirements. For software design phase IDEF0 and N^2 diagrams were used to track the interdependencies of various functions, inputs, and outputs within the algorithm and its sub−modules. Also, these diagrams can provide interface management between sub−modules. At the coding process EFFBD diagram is a tool in functional analysis to define the functional level and the sequences of activities. EFFBD enables the developer to break down the code into a small parts to be easier for programming. EFFBD is a powerful tool for testing the code from the perspective of functional and structural testing that guarantee 100% of code coverage test that is required for safety critical software.

For validation and verification phase, the requirements traceability matrix is used to map and trace the algorithm requirements to



[Figure 9] Software Development Life Cycle included recommended Modeling Diagrams

be sure that all requirements are fulfilled. MATLAB Simulink is a powerful tool to model and test the algorithm.

Systems engineering software tool – that is used to develop modeling diagrams – enhances and improves the configuration management by storing the diagrams defined with its version number and the date update to track any changes, approving changes, ensuring versions are combined correctly, releasing design documents and software for use, and establishing and maintaining a chronological record. Two different systems engineering tools were used to apply this methodology first is Enterprise Architect and the second is Core 9.

The algorithm development process is a part from the overall Systems Engineering Management Plan (SEMP). This SEMP defines how the entire DNBR algorithm will be integrated with the hardware solution to develop the system.

References

1. Functional Design Requirements for a Core Protection Calculator System, 2006, KNF-KSNGEN-03026, Rev.03.
2. Design of Instrumentation and Control Systems for Nuclear Power Plants, 2016, IAEA, Specific Safety Guide, SSG-39.
3. Verification and Validation of Software Related to Nuclear Power Plant Instrumentation and Control, 1990, IAEA, Technical Reports Series, No. 384.
4. Verification and Validation of Software Related to Nuclear Power Plant Instrumentation and Control, 1990, IAEA, Technical Reports Series, No. 384.
5. Recommended Practice for Software Requirements Specifications, 1993, IEEE, 830.
6. Systems Engineering Handbook, Guide for system Life Cycle Process and Activities, 4'th edition, 2015, INCOSE-TP-2003-002-04.
7. NASA Systems Engineering Handbook, Rev1, National Aeronautics and Space Administration, NASA/SP-2007-6105.
8. Systems Design, Software Design Descriptions, 2009, IEEE, 1016.
9. Nuclear power plants - Instrumentation and control systems important to safety - Software aspects for computer-based systems performing category a functions, 2006, IEC, 60880.
10. Nuclear power plants -Enhanced functional analysis system technique for managing complex engineering projects, 2007, Sofia

Tan, Missouri University of science and Technology, 4572.
11. Nuclear power plants -A prime for Model-Based Systems Engineering, Second Edition 2011, David Long and Zane Scott, Vitech, ISBN 978-1-105-58810-5.
12. Reference document for the organization and conduct of Engineering Safety Review Services (ESRS) on software important to safety in nuclear power plants, 2000, IAEA, Services Series No. 6.
13. Business Modelling: UML vs. IDEF, 2000, Ovidiu S.Niran, ttp://www.cit.gu.edu.au/~noran.
14. IAEA SAFETY GLOSSARY, Terminology Used in Nuclear Safety and Radiation Protection, 2016.
15. Dependability Assessment of Software for Safety Instrumentation and Control Systems at Nuclear Power Plants, 2018, IAEA SERIES No. NP-T-3.27.
16. Verification, Validation, Reviews, and Audits for Digital Computer Software Used In Safety Systems of Nuclear Power Plants, 2004, REGULATORY GUIDE 1.168, U.S. Nuclear Regulatory Commission (NRC).