Qualification of safety-critical software for digital reactor safety system in nuclear power plants

KWON Kee-Choon, PARK Gee-Yong, KIM Jang-Yeol, and LEE Jang-Soo

Korea Atomic Energy Research Institute, 989-111 Daedeok-daero, Yuseong-gu, Daejeon, 305-353, Republic of Korea ({kckwon, gypark, jykim, jslee}@kaeri.re.kr)

Abstract: This paper describes the software qualification activities for the safety-critical software of the digital reactor safety system in nuclear power plants. The main activities of the software qualification processes are the preparation of software planning documentations, verification and validation (V&V) of the software requirements specifications (SRS), software design specifications (SDS) and codes, and the testing of the integrated software and integrated system. Moreover, the software safety analysis and software configuration management are involved in the software qualification processes. The V&V procedure for SRS and SDS contains a technical evaluation, licensing suitability evaluation, inspection and traceability analysis, formal verification, software safety analysis, and an evaluation of the software configuration management. The V&V processes for the code are a traceability analysis, source code inspection, test case and test procedure generation. Testing is the major V&V activity of the software integration and system integration phases. The software safety analysis employs a hazard operability method and software fault tree analysis. The software configuration management in each software life cycle is performed by the use of a nuclear software configuration management tool. Through these activities, we can achieve the functionality, performance, reliability, and safety that are the major V&V objectives of the safety-critical software in nuclear power plants.
Keyword: safety-critical software; verification and validation; digital reactor safety system; nuclear instrumentation and control

1 Introduction

Korea ranks as the sixth largest country in the world for annual electric power generation by nuclear power plants (NPPs), but did not have a domestically built instrumentation and control (I&C) system until recently. To achieve technical self-reliance in the area of nuclear I&C, a Korea instrumentation and control system (KNICS) project was conducted for seven years beginning in 2001. The final goal of the project is to apply the KNICS results to newly planned NPPs and to upgrade the operating plants[1].

The KNICS I&C architecture hierarchically consists of three layers to meet the design requirements of the APR1400 which was design certified by the Korean regulatory body a few years ago. The top level consists of a control room, an information processing computer system, and an indication system. The control and protection systems are located at the middle level. The bottom level consists of the measurement systems for various types of equipment. The networks are widely used for the intra and inter-system connections. Figure 1 shows the KNICS I&C architecture. In the KNICS project, we are developing a safety-grade programmable logic controller (PLC) as the safety-grade platform and digital reactor safety system which consists of digital reactor protection system (DRPS) and engineered safety feature-component control system (ESF-CCS).

This paper presents an independent Verification and Validation (V&V) process according to the software development life cycle for the safety-critical software in a digital reactor safety system.

2 Overview of the digital reactor safety system

The digital reactor safety system consists of a digital reactor protection system, engineered safety feature-component control system, and reactor core protection system (RCOPS) based on a safety-grade PLC platform.

The safety-grade PLC consists of various modules such as a power module, processor module, communication module, input/output module, and an engineering tool called a pSET. The processor module uses the Texas Instrument digital signal processing CPU and real-time operating system called a pCOS, which was developed based on the Micro-C real-time operating system in the KNICS project.

Received date: September 12, 2013
(Revised date: October 18, 2013)

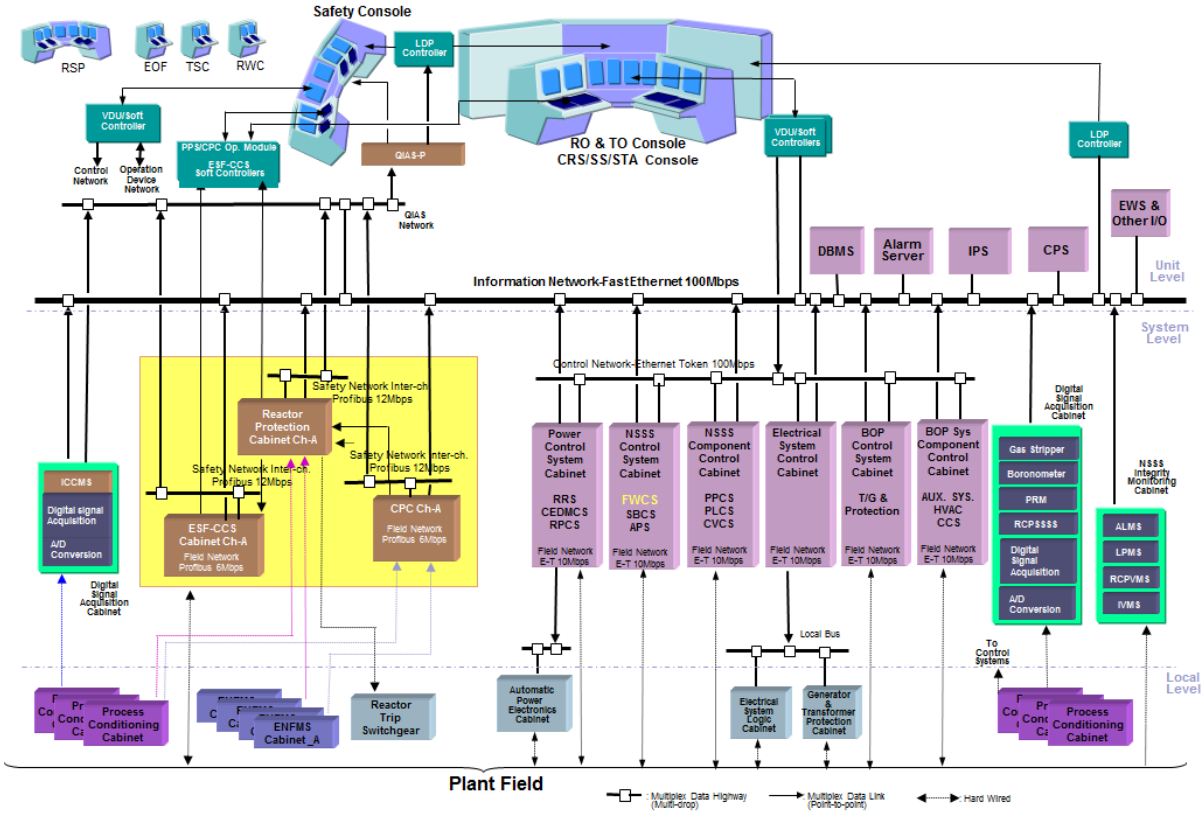


Fig. 1 KNICS I&C Architecture.

Communication modules including a profibus-fieldbus message specification module for information communication, a high reliability-safety data link for peer-to-peer communication, and high reliability-safety data network for a safety critical network. Input/output module consists of a local bus extension module, a digital input/output module, an analog input/output module, resistance temperature detector/thermocouple module, and a high-speed pulse counter module. pSET provides an engineering environment for developers to perform programming, debugging, and an application program simulation.

The DRPS has four channels located in electrically and physically isolated rooms. The DRPS generates the reactor trip and engineered safety features actuation signals automatically whenever the monitored processes reach the predefined setpoints. The DRPS has an on-line automatic periodic test capability for determining the system's operability. A bistable processor (BP), coincidence processor (CP), automatic test and interface processor (ATIP), and cabinet operator module (COM) are included in the DRPS channel.

The BP determines the trip state by comparing the measured process variables with the predefined trip setpoints. The DRPS consists of two BPs in a channel as shown in Figure 2. The first BP in channel A and second BP in channel A are reversely connected with process variables. Also the trip logic is reversely executed between the redundant BPs to provide a diversity.

The CP generates a trip signal by a two out of four voting logic. When a channel is bypassed, the trip signal is determined by a two out of three voting logic. A channel has two CPs as shown in Figure 2. The initiation logic is an analog circuit that generate trip breaker actuation signal.

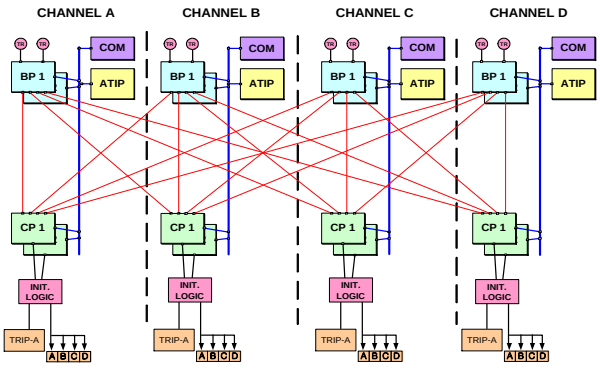


Fig. 2 Architecture of a digital reactor protection system.

There is one ATIP in each DRPS channel. The ATIP generates the test signals for a manual test and a manual initiated automatic test. The ATIP also performs DRPS status indications, alarms, and healthiness tests to verify the operational status of the BP and CP. The ATIP is connected with the other channel's ATIPs and an engineered safety features-component control system.

The COM comprises two parts: (a) a computer based part that provides the status information regarding the overall DRPS equipment such as the BPs, CPs, and ATIP, and (b) a hardware based part that performs protection related controls such as a channel bypass and an initiation circuit reset.

The ESF-CCS initiates several emergency actuations to prevent the plant from a hazardous state during and/or after accidents. The actuations include a safety injection, a containment isolation, a main steam line isolation, an auxiliary feedwater injection, and a containment spray actuation. The ESF-CCS is designed with four redundant divisions (*i.e.*, A, B, C, and D), and implemented with the PLC platform. The principal components of an individual division are fault tolerant group controllers (GCs), loop controllers (LCs), a test and interface processor (ETIP), a cabinet operator module (COM) and a control channel gateway (CCG)[2] . The architecture of the one channel ESF-CCS is shown in Figure 3.

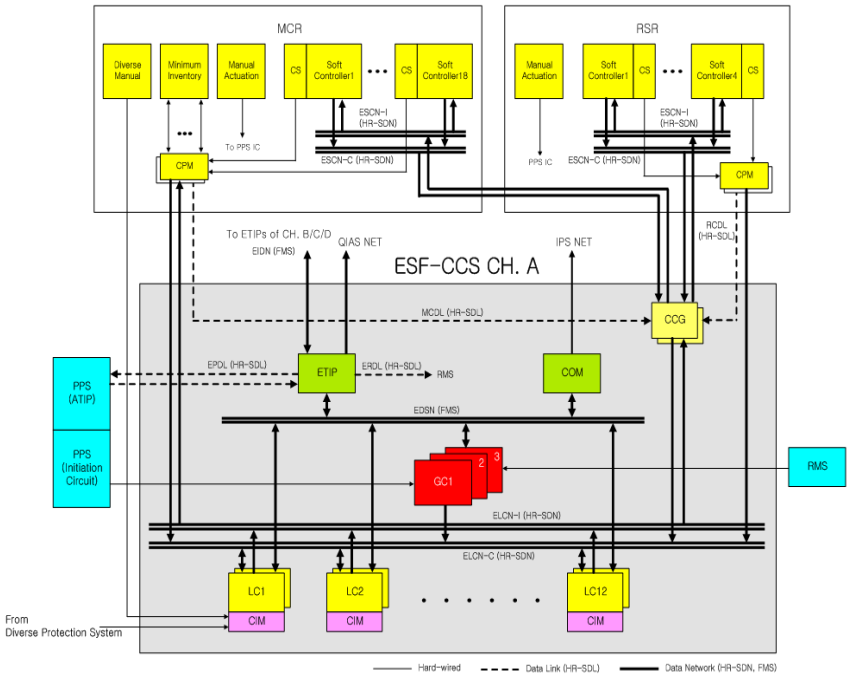


Fig. 3 Architecture of an engineered safety feature system-component control system.

3 Software qualification

Most of the software embedded in the PLC and digital reactor safety system is classified as safety-critical software whose failure can result in loss of life, significant property damage or damage to the environment. This safety-critical software shall be developed and independently qualified according to the code and standards. In the safety-critical software qualification approach, an appropriate safety analysis, V&V, and configuration management activities are conducted by following the software life cycle process.

3.1 Preparation for a verification and validation

3.1.1 Planning

According to the branch technical position (BTP) instrumentation and controls branch (HICB)-14 of the NUREG-0800 standard review plan[3], the information to be reviewed is subdivided into three areas: software life cycle process (SLCP) planning, SLCP implementation, and SLCP design outputs. Owing to the importance of planning, verifiers developed a planning documentation such as a software verification and validation plan, a software quality assurance plan, a software configuration management plan, and a software safety plan in view of the independent V&V, as well as a software management plan, a software development plan, an integration plan, and an installation plan. A software V&V plan is strongly related to a software development plan in that software V&V activity depends on which process a software development project follows. The software V&V plan addresses the issues of team organization, master schedule, software integrity level, management of the V&V,

life cycle V&V activities, V&V reporting and documentation.

3.1.2 Verification and validation procedure

The V&V process provides an objective assessment of the software products and processes throughout the software life cycle. This assessment demonstrates whether the software requirements and system requirements are correct, complete, accurate, consistent, and testable. To maintain the assessment objectives and consistency, procedures are necessary. We have developed V&V procedures for the software requirement, design, implementation, and integration phases. The V&V for the safety-critical software is performed based on the V&V procedures in each phase.

The V&V procedures provide specialized checklists for the life cycle V&V tasks, and define the V&V methods and their supporting tools, as well as inputs and outputs. For example, the procedure for the software requirement specification (SRS) V&V defines the V&V techniques and their tools for the software requirements, and provides a checklist corresponding to the acceptance criteria in the BTP HICB-14. It also provides the procedures for a formal verification. Testing procedures are also included within the V&V procedures because the regulatory position regards software testing as a V&V activity.

3.2 Verification and validation

One of the main purposes of the safety-critical software V&V is to acquire a license from the regulatory authority. Thus, it is crucial for the V&V process to meet the regulatory requirements as well as the design goals. To meet the regulatory requirements and design goals, the software V&V criteria and requirements are based on codes and standards including the BTP HICB-14 of NUREG-0800 SRP [3], Regulatory Guide 1.152[4], IEEE Std. 7-4.3.2[5], IEEE Std. 1012[6], IEEE Std. 1008[7], IEEE Std. 829[8], and IEEE Std. 1028[9].

3.2.1 Software requirement phase

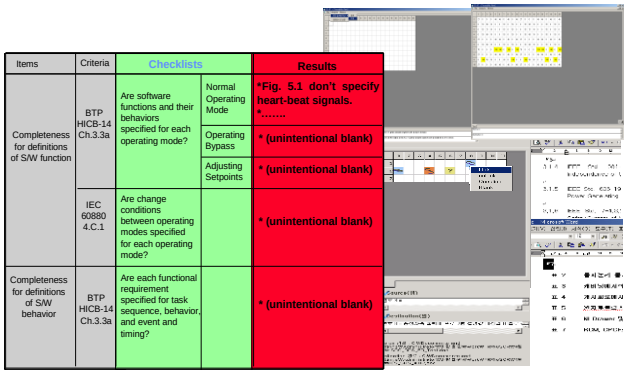
The first V&V activity in the software requirement phase is a licensing suitability evaluation. The purpose of the licensing suitability evaluation is to confirm whether or not the software requirements that coincide with the criteria of the software and the performance and safety requirements of the safety-critical software are suitable based on the code and standard. According to SRP/BTP HICB-14[3] and IEEE Std. 7-4.3.2[5], the safety-critical software requirements must satisfy all functional characteristics (accuracy, functionality, reliability, robustness, safety, security, and timing) and process characteristics (completeness, consistency, correctness, style, traceability, unambiguity, and verifiability). Through the licensing suitability review, we evaluate whether or not the contents of the software requirement specifications are reviewable from the viewpoint of the functional characteristics and process characteristics.

The second V&V activity in this phase is the Fagan inspection[10]. The Fagan inspection is a formal review process developed by Michael Fagan at IBM in the 1970s. The inspection technique is being applied for all phases of a software life cycle thanks to its applicability to a software design as well as the coding. The inspection process consists of seven steps, *i.e.*, planning, overview, preparation, inspection meeting, reworking, and a follow up. An inspection team is composed of a moderator, recorder, reader, and author, and an inspector (or tester).

The third activity for the requirement phase V&V is a traceability analysis. Throughout the software life cycle, a software requirement traceability analysis will be performed and a requirement traceability matrix will be maintained for the safety-critical software. A requirement traceability analysis traces different results (*i.e.*, deliverables) between two phases of a software development, and the requirement traceability analysis also analyzes the relationships for properties including correctness, completeness, and consistency. For example, the requirements traceability analysis for the requirements V&V traces the software requirements to the system requirements, and the system requirements to the software requirements. The software requirement traceability analysis also analyzes the identified relationships for correctness, completeness, consistency, and accuracy.

We perform an inspection and traceability analysis by focusing on the three properties of completeness, correctness, and consistency. The Fagan inspection is supported by the software inspection support-requirement traceability (SIS-RT) tool that is developed in the KNICS project. A traceability analysis between the software requirement specification and the system design specification is performed using SIS-RT. Figure 4 shows examples of a Fagan inspection and traceability analysis.

To improve the quality of safety-critical software in the early phase of the software development process, the SRS is written using natural language and specified by a formal specification method.



Items	Criteria	Checklists		Results
Completeness for definitions of S/W function	BTP HICB-14 Ch.3.3a	Are software functions and their behaviors specified for each operating mode?	Normal Operating Mode	*Fig. 5.1 don't specify heart-beat signals. *.....
			Operating Bypass	* (unintentional blank)
			Adjusting Setpoints	* (unintentional blank)
	IEC 60880 4.C.1	Are change conditions between operating modes specified for each operating mode?		* (unintentional blank)
Completeness for definitions of S/W behavior	BTP HICB-14 Ch.3.3a	Are each functional requirement specified for task sequence, behavior, and event and timing?		* (unintentional blank)

Fig. 4 Fagan inspection and traceability analysis.

For the formal specification and verification of the SRS, the NuSRS tool was developed proprietarily for the KNICS project[11,12]. The overall configuration of the NuSRS is depicted in Figure 5.

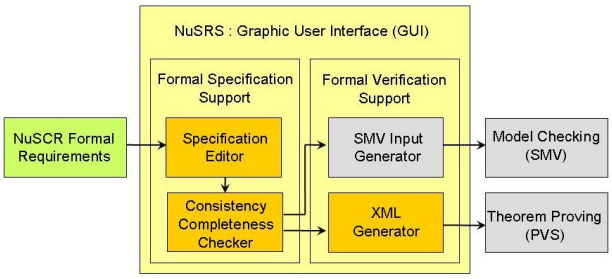


Fig. 5 Configuration of the NuSRS.

The formal specification is performed using a formal specification language called NuSCR. The NuSRS can automatically check the consistency and completeness during a formal specification. It also has an automated conversion process from the NuSCR specifications to the input format for a symbolic model verifier (SMV) model checker[11], and model checking is thereby activated conveniently within the frame of the NuSRS. Additionally, the NuSRS outputs its specifications as an extensible mark-up language (XML) file format so that other verification methods such as theorem proving can be applied to the specifications based on this file format.

In NuSCR language, three types of formal specifications are provided: a structured decision table for a function-based operation such as a simple logic operation, a finite state machine for state-based operations such as a trip hysteresis, and a timed transition system for timing-based operations such as a trip delay. In addition a functional overview diagram constructed in NuSCR, hierarchically represents the relations and operation flows of all specifications. One of the NuSCR specifications is presented in Figure 6.

The formal specifications are converted into input files for the SMV model checker where the properties are given by a computational temporal logic. All the function modules in the SRS were formally verified.

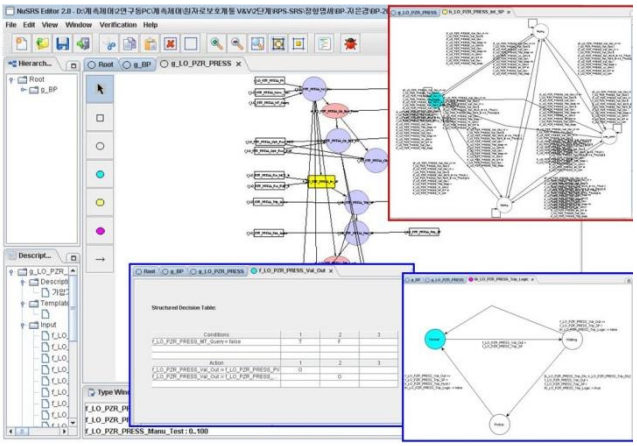


Fig. 6 Formal specifications for safety-critical software.

3.2.2 Software design phase

The V&V activities in the software design phase are almost the same as those in the software requirement phase. Formal verification is a little different because the DRPS software development process does not include NuSCR in the software design specification. Instead it adopts function block diagrams (FBD) to specify a software design. To verify the FBD specifications, Verilog, which is the most popular hardware description language is chosen as a formal verification language because the semantics of FBD are similar to those of Verilog, and thus FBD can be translated into Verilog efficiently. We used model checking techniques to verify the FBD specifications and chose Cadence SMV as a model checker to verify Verilog models generated from FBD specifications in the software design phase as shown in Figure 7[13,14].

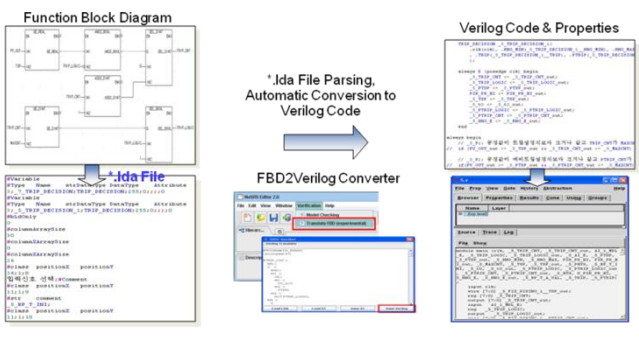


Fig. 7 FBD translation into Verilog language.

3.2.3 Implementation phase

In the implementation phase, we are perform a licensing suitability evaluation, a Fagan inspection, and a traceability analysis. However, in this phase, the main activity is testing, specifically component testing. Figure 8 shows the software test life cycle including component testing, integration testing, and system testing. Each testing follows the software test life cycle tasks (*e.g.*, test plan generation, test design generation, test case generation, test procedure generation, and a test execution).

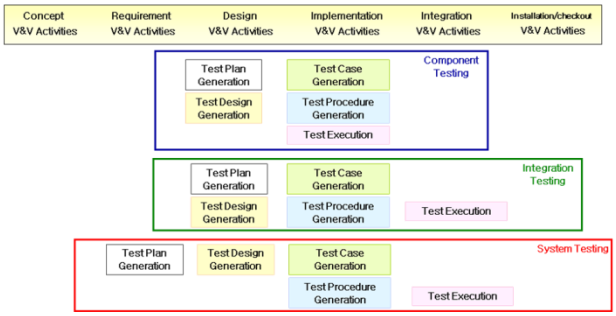


Fig. 8 Software test life cycle.

Component testing is a test conducted to verify a correct implementation of the software design and compliance with the software requirements for one software element (*i.e.*, unit and/or module) or a collection of software elements. Component testing of the safety-critical software is challengeing because the safety-critical software is implemented on a PLC and the testing techniques for the PLC software are not yet mature. We developed a component testing technique for defining the test coverage criteria in a FBD. Using this technique, we are deriving test cases with which we are performing component testing for safety-critical software.

3.2.4 Integration phase

The integration phase can be divided into a software integration phase and a system integration phase. The main V&V activity in the software integration phase is integration testing. Integration testing is an orderly progression of a testing of incremental pieces of software systems in which software elements are combined and tested until the software has been integrated to show compliance with the software design and requirements of the software system.

In the system integration phase, the main V&V activities are system testing and/or acceptance testing. System testing is the activities of testing an integrated hardware and software system to validate whether or not the system meets its original objectives. Boundary value testing and equivalence partitioning techniques are applied during the safety-critical software system testing process.

3.3 Software safety analysis

The software qualification activities are applied to improve software safety as well as software quality. Thus, we can say that they include a software safety analysis. Since the licensing criteria require a safety analysis of the product from each phase of the life cycle, a software safety analysis procedure for each phase has been developed. The procedures include hazard and operability (HAZOP) procedures for the requirement and design phases, and software fault tree analysis (FTA) procedures for the design and implementation phases.

The HAZOP method has been suggested for a safety analysis in the software requirement and design phases[15]. HAZOP is a powerful hazard analysis technique that has a long history in process industries. As the use of digital systems for nuclear power plants becomes more common, it is clear that there is a need for a HAZOP method that can be used effectively with such systems. We developed guide phrases, checklists, and a software HAZOP procedure for safety-critical software. One of the software HAZOP results for the software requirements is presented in Table 1[16].

The FTA is one of the most widely used methods in a system reliability analysis. The FTA has been used for many industrial applications, and one of the advantages of this is that safety engineers are already familiar with it. FTA was primarily used for a safety analysis of hardware systems such as electromechanical devices. Software FTA was proposed almost twenty years ago, and since that time, the technique has been refined for analyzing the safety of software designs. At the design phase, the software HAZOP was performed first, and software FTA was then applied. The software FTA was applied to some critical modules selected from the software HAZOP analysis. The software FTA can obtain some valuable results that have not been identified through a rigorous V&V procedure[17]. Figure 9 shows one of the FTA results, which is the FTA construction for an FBD-based design description for a Low DNBR Trip, as depicted in Figure 10[16] .

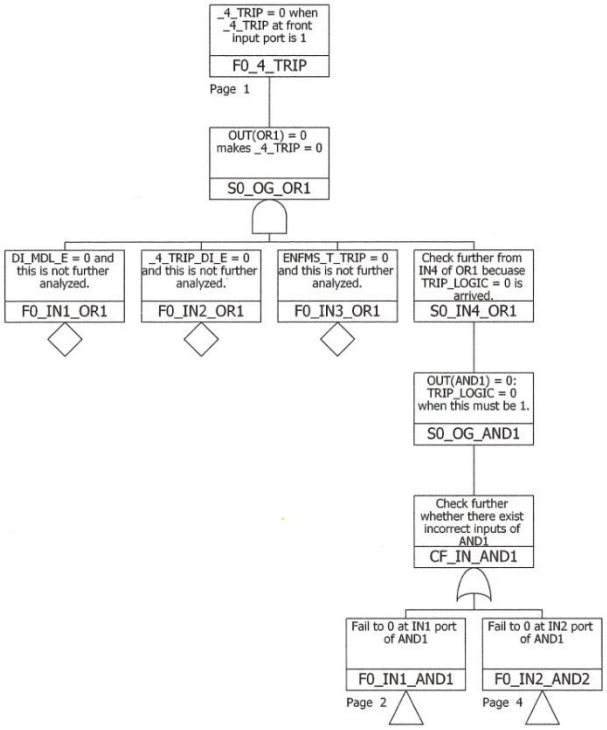


Fig. 9. FTA for Low DNBR Trip Logic

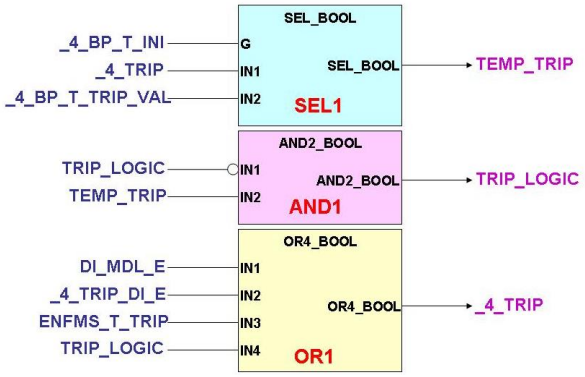


Fig. 10. FBD Module for DNBR_Lo (Low DNBR) Trip.

3.4 Configuration management

The software configuration management is an activity that configures the form of a system (documents, source codes, and hardware) and systematically manages and controls the modifications used to compile plans, developments, and operations resulting from software development and maintenance. This process improves the quality of the software and is highly correlated with the development of reliable software. A software configuration management tool, NuSCM, has been developed for software life cycle V&V management of the safety-critical software.

Table 1. Software HAZOP analysis for trip logic with rate-limited rising setpoint.

Attribute	Deviation Checklist	HAZOP Analysis
Accuracy	What is the consequence if the sensor value is below its minimum range?	Trip Decision Block (SRS 6.3.13.2) handles this fact. → No hazard state.
Accuracy	What is the consequence if the sensor value is above its maximum range?	
Accuracy	What is the consequence if wrong variable type is used?	Variable type is not specified in SRS.
Accuracy	What is the consequence if an equation is wrong?	At 4)th logic, by the condition of "(_1_PV_T < _1_PV_T_INIT)", _1_PV_T is immediately set to _1_PV_T_INIT. Then, the logic goes to "else" part → unnecessary trip occurs. At (4) rate-limited variable setpoint logic (SRS pp.74), the setpoint is always set to its maximum value → Trip signal cannot be generated under the trip state. At (5) rate-limited rising comparison logic, there is no change in conditional logic (> or <) → Trip signal always occurs when trip condition is not satisfied.
Capacity	What is the consequence if there is an error in the ICN data?	There is no specification handling an error occurrence at ICN.
Capacity	What is the consequence if there is an error in the SDL data?	There is no specification using an error notification of SDL in trip logic.
Functionality	What is the consequence if a software module has a defect or cannot perform the intended behaviour?	At rate-limited rising setpoint logic block (4) (pp.74), setpoint is recalculated regardless of the trip state → cancelling trip signal at trip state and trip signal cannot be generated.

Functionality	**What is the consequence if a function is executed in incorrect operational mode?**	**DEADBAND is executed in test scan mode → affect running resource adversely.**
Reliability	**What is the consequence if software fails untimely?**	**CP detects BP failure by monitoring BP HB signal → CP generates trip signal. (No hazard)**

NuSCM is based on a systematic management of the software design documents and source codes based on projects and activities. Since NuSCM is designed based on the foundation of the Internet, it provides the user with a high accessibility, thus maximizing the efficiency in carrying out tasks.

4 Conclusion

This paper discussed the KNICS approach to the life cycle V&V for the safety-critical software in nuclear power plants. Through the KNICS projects, the V&V process for a nuclear power plant safety-critical software is being established. The KNICS approach involves structured checklists, V&V procedures, specialized V&V techniques and their tools. Representative V&V techniques include a licensing suitability evaluation, Fagan inspection, traceability analysis, model checking, theorem proving, and various testing. The main features of the KNICS software V&V process include a strict compliance with the related codes and standards, a configuration of various V&V activities in each software development phase, a combined approach between the informal and formal verifications, and an incorporation of newly and self-developed V&V and testing techniques. Through these activities, we believe that we can achieve functionality, performance, reliability, and safety which are the V&V objectives of a nuclear safety-critical software. The safety-grade PLC platform and digital reactor safety systems have obtained licensing approval from the Korean regulatory body through a topical report by the successful V&V activities of safety-critical software. The safety evaluation reports were issued in February 2009. Furthermore, a Korean utility and vendor company is determined to apply the KNICS results to the Shin-Hanul unit 1&2 nuclear power plant in March 2009.

References

[1] KWON, K.C., and LEE, M.: Technical Review on the Localized Digital Instrumentation and Control Systems, Nuclear Engineering and Technology, 2009, 41(4): 447-454.

[2] KWON, K.C., and LEE, D.Y.: Technical Self-Reliance of Digital Safety Systems, Proceedings of the 24th KAIF/KNS Annual Conference, Seoul(Korea) April 8-10, 2009.

[3] USNRC: Guidance on Software Reviews for Digital Computer-Based Instrumentation and Control Systems, Branch Technical Position HICB-14, NUREG-0800, Standard Review Plan, Chapter 7, 1997.

[4] USNRC: Criteria for Use of Computers in Safety Systems of Nuclear Power Plants, Regulatory Guide 1.152 Rev.2, 2006.

[5] IEEE: Std. 7-4.3.2, IEEE Standard for Digital Computers in Safety Systems of Nuclear Power Generating Stations, 2003.

[6] IEEE: Std. 1012, Standard for Software Verification and Validation Plans, 2004.

[7] IEEE: Std. 1008, Standard for Software Unit Testing, 1987.

[8] IEEE: Std. 829, Standard for Software Test Documentation, 1998.

[9] IEEE: Std. 1028, Standard for Software Reviews and Audits, 1997.

[10] FAGAN, M.E.: Design and Code Inspections to Reduce Errors in Program Development. IBM Systems Journal, 1976, 15(3).

[11] YOO, J., KIM, T., CHA, S., LEE, J.S., and SON, H.S.: A Formal Software Requirements Specification Method for Digital Nuclear Plant Protection Systems, Journal of Systems and Software, 2005, 74(1): 73-83.

[12] YOO, J., JEE, E., and CHA, S.: Formal Modeling and Verification of Safety-Critical Software, IEEE Software, 2009, 26(3): 42-29.

[13] KOH, K.Y., *et al.*: A Formal Verification Method of Function Block Diagrams with Tool Supporting Practical Experiences, Annals of DAAAM for 2008 & Proceedings of the 19th International DAAAM Symposium, 2008.

[14] YOO, J., CHA, S., and JEE, E.: A Verification Framework for FBD based Software in Nuclear Power Plants, Proceedings of the 15th Asia-Pacific Software Engineering Conference, Beijing(China) Dec. 3-5, 2008.

[15] USNRC: Software Safety Hazard Analysis, NUREG/CR-6430, February 1996.

[16] PARK, G.Y., CHEON, S.W., KWON, K.C., KOH, K.Y., SEONG, P.H., JEE, E., and CHA, S.: Software Qualification Activities for Safety Critical Software, NPIC&HMIT 2009, Knoxville(USA), April 5-9, 2009.

[17] PARK, G.Y., *et al.*: Fault Tree Analysis of KNICS Software, Nuclear Engineering and Technology, 2008, 40(5): 397-408.