MTO-4.5: A Safety Case for Reactor Protection System Software Developed Using a Prescriptive Approach

Eunkyoung Jee
Korea Advanced Institute of Science and Technology, Republic of Korea
ekjee@se.kaist.ac.kr
Gee-Yong Park, Jang-Soo Lee, Kee-Choon Kwon
Korea Atomic Energy Research Institute, Republic of Korea
gypark@kaeri.re.kr, jslee@kaeri.re.kr, kckwon@kaeri.re.kr
Doo-Hwan Bae
Korea Advanced Institute of Science and Technology, Republic of Korea
bae@se.kaist.ac.kr

Abstract

A safety case is a structured argument supported by evidence. The safety case approach is considered an effective way to argue for and evaluate system safety, but it has been contrasted with prescriptive or process-based approaches, which assume that following the process prescribed in safety standards will generate evidence for safety. We create a safety case for a part of the reactor protection system software that was developed in prescriptive ways during the Korean Nuclear Instrumentation and Control System R&D Center project. Using real-world industrial project data, including a number of verification and validation artifacts, we illustrate how a safety case can be created based on existing evidence. The proposed safety case to claim safety of the target software is mainly based on the argument by satisfaction of all the desired safety requirements and the argument by safety analysis activities. The possible advantages and drawbacks of utilizing safety cases with prescriptive approaches are discussed.

1. Introduction

A safety case is a structured argument, supported by a body of evidence that provides a compelling, comprehensible, and valid case that a system is safe for a given application in a given operating environment [1]. The safety case approach is considered an effective way to argue for and evaluate system safety and it has been contrasted with prescriptive or process-based approaches, which assume that following the process prescribed in safety standards will generate evidence for safety [2]. Since the safety argument approach and the prescriptive approach each have their own merits, these two approaches could complement each other.

We create a safety case for Bistable Processor software, a part of the reactor protection system software developed using prescriptive methods under the Korean Nuclear Instrumentation and Control System R&D Center (KNICS) project, whose goal was to achieve technical self-reliance in the area of nuclear instrumentation and control. We investigate whether the safety argument approach can complement the prescriptive approach by adding values on safety assurance. In the KNICS project, conforming to important international standards and guidelines such as NUREG-0800 [3] and IEEE STD-1228 [4], over one thousand documents were generated [5].

Using real-world industrial project data, including a number of verification and validation artifacts, we illustrate how a safety case can be created based on existing evidence. We analyze and evaluate the results of applying the safety case approach to the existing target system software

developed through the use of prescriptive methods. The possible advantages and drawbacks of utilizing safety cases with prescriptive approaches are discussed.

This paper is organized as follows: Section 2 explains the target system, Section 3 presents a safety case for the target system, Section 4 analyzes and evaluates the case study, and Section 5 concludes this paper.

2. Target System: Bistable Processor (BP)

The KNICS project was carried out for seven years, starting in 2001. The safety-grade Programmable Logic Controller (PLC) and the digital safety system were developed by the KNICS project for use in newly constructed nuclear power plants (NPPs) as well as in the upgrading of existing analog-based NPPs [6] [7]. Bistable processors (BPs) are a part of the fully digitalized reactor protection system (RPS) developed in the KNICS project. A BP compares processing variables with the corresponding setpoints. All the trip-actuating functions in the BPs are implemented in the software. The trip functioning software in the KNICS RPS is classified as safety-critical since it is crucial to the safety of a nuclear power plant in that its malfunctioning may have irreversible consequences [8].

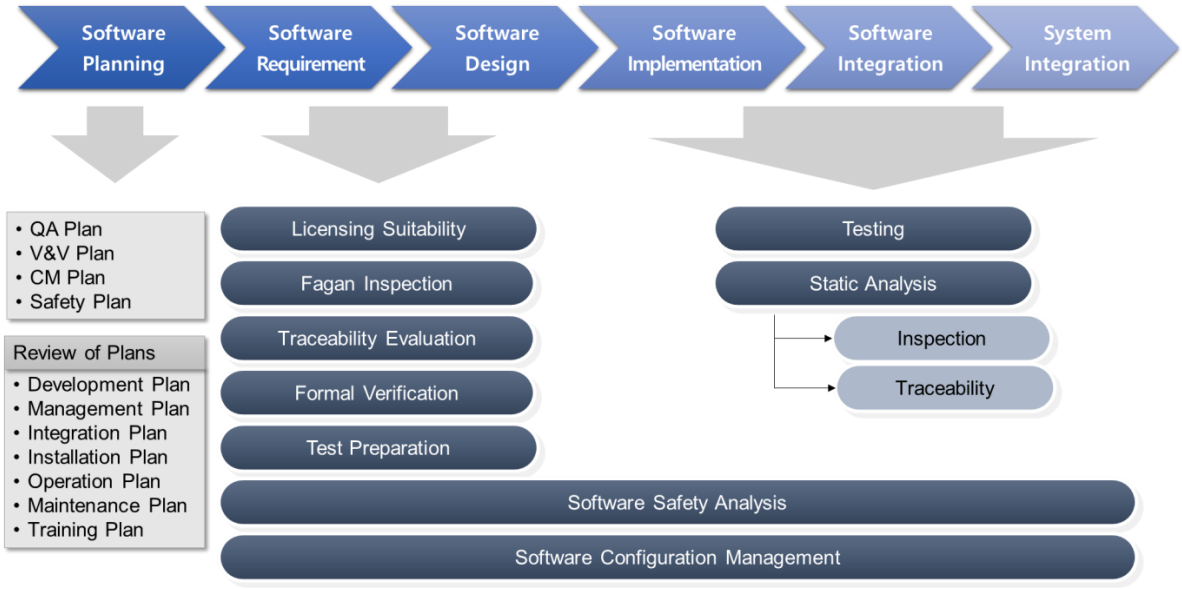

Figure 1. Software V&V activities of the KNICS RPS [8]

The software used in the KNICS RPS was developed under a rigorous procedure [6], and the verification and validation (V&V) activities were performed following the software development life cycle V&V procedure [9] [10]. Figure 1 shows the V&V activities performed by an independent V&V team for the development of the KNICS RPS software. The purpose of the V&V activities was to ensure that the KNICS software product satisfies regulatory acceptance criteria and to improve software quality by finding and resolving software defects at an early phase during software development.

After the preparation of plan documents in the software planning phase, the development of and the V&V activities for the KNICS RPS software were performed according to these plan documents. In the requirement and design phases, document evaluations such as licensing suitability evaluation, detailed inspection via the Fagan inspection procedure, and traceability

analysis were performed. Formal verification, e.g., model checking, was carried out for the formal specifications of the software requirements and the software design, respectively, through the use of automated tools [11] [12] [13] [14]. In the implementation phase, testing of the software components, the integrated software, and the integrated system was performed. Software safety analysis and software configuration management were also included in the V&V activities. For the software safety analysis in the SRS phase, software Hazard Operability (HAZOP) was performed and then software Fault Tree Analysis (FTA) was applied. The software FTA was applied to a part of a software module with some critical defects identified by the software HAZOP in the SDS phase [8] [15]. The software configuration management was performed using the in-house tool developed in the KNICS project [7].

3. Creating a Safety Case for the BP software

3.1 Structuring the safety case

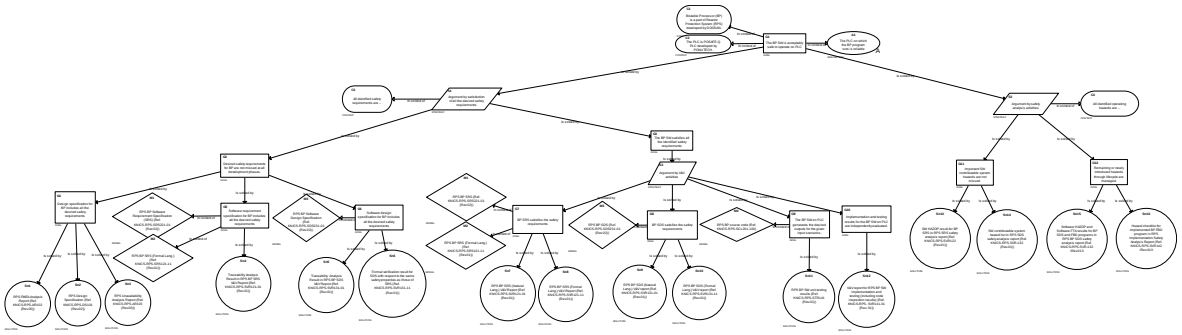

Figure 2. The BP SW safety case – Bird's eye view

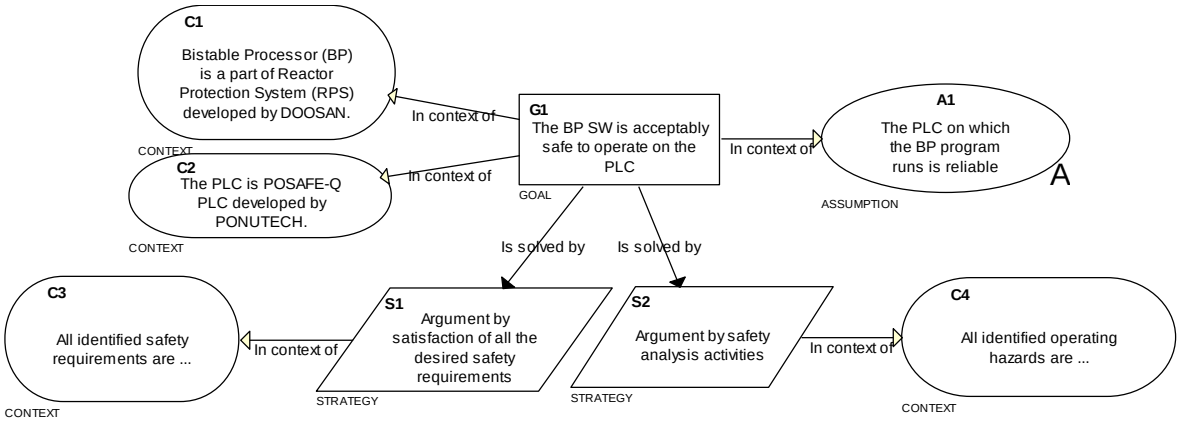

Figure 3. The BP SW safety case – The BP software is safe

Figure 2 shows an overview of the safety case for BP software. Each part of the safety case will be explained in the following sections. Figure 3 shows the top claim and the two main arguments denoted by S1 and S2. The top claim of the safety case for BP software is "The BP SW is acceptably safe to operate on the PLC." This safety case proceeds mainly based on the argument by satisfaction of all the desired safety requirements (S1) and the argument by safety analysis activities (S2). This safety case makes the assumption that the PLC on which the BP program runs

is reliable (A1). The safety of the PLC on which the BP program runs also needs to be elaborated in the safety case further, but in this paper we focus on the BP software part.

3.2 Argument by satisfaction of all the desired safety requirements

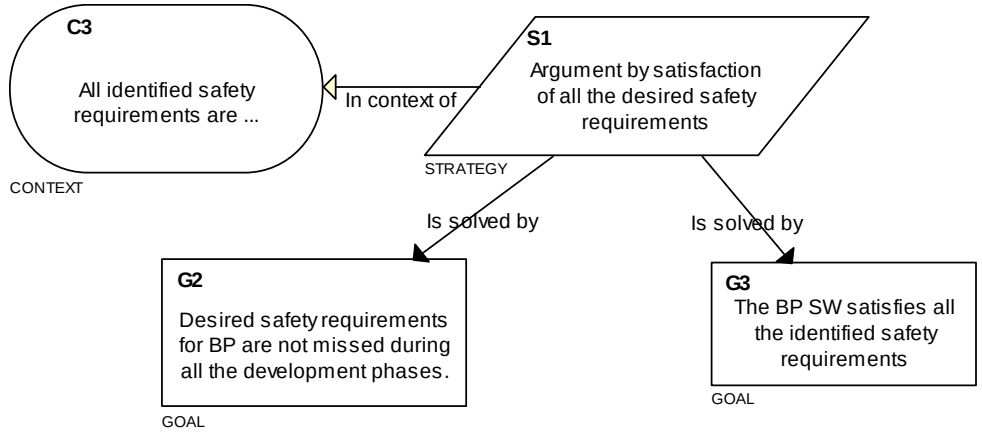

Figure 4. The BP SW safety case – Argument by satisfaction of the safety requirements

Figure 4 shows that through the argument by satisfaction of all the desired safety requirements, it can be claimed that the BP SW is acceptably safe to operate on PLC if the desired safety requirements for BP are not missed during all the development phases (G2) and the BP SW satisfies all the identified safety requirements (G3).

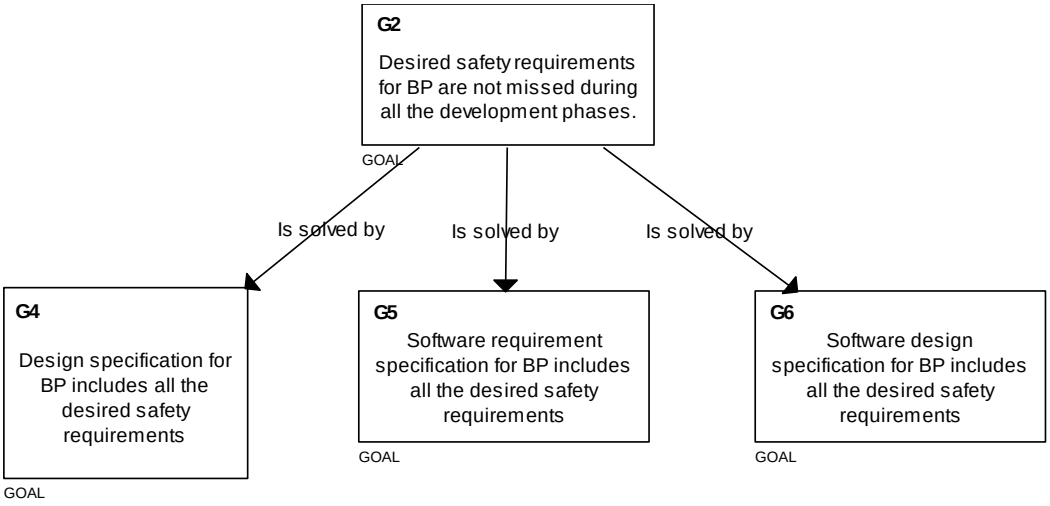

Figure 5. The BP SW safety case – Safety requirements are not missed

As shown in Figure 5, the G2 goal claiming that “the desired safety requirements for BP are not missed during all the development phases” can be split into three sub goals: the design specification for BP includes all the desired safety requirements (G4); the software requirement specification for BP includes all the desired safety requirements (G5); and the software design specification for BP includes all the desired safety requirements (G6).

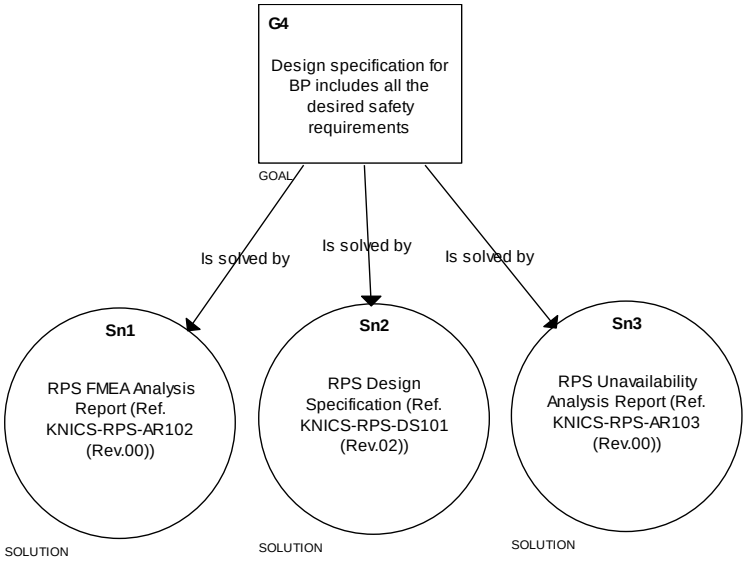

Figure 6. The BP SW safety case – Design specification includes all the safety requirements

Figure 6 shows three pieces of evidence, i.e., Sn1, Sn2 and Sn3, supporting the claim that the design specification for BP includes all the desired safety requirements. Safety requirements for the BP system were extracted from several sources such as Failure Mode and Effects Analysis (FMEA) results (Sn1) and unavailability analysis results (Sn3). The extracted safety requirements are found in the RPS design specification, indicated as Sn2 in Figure 6.

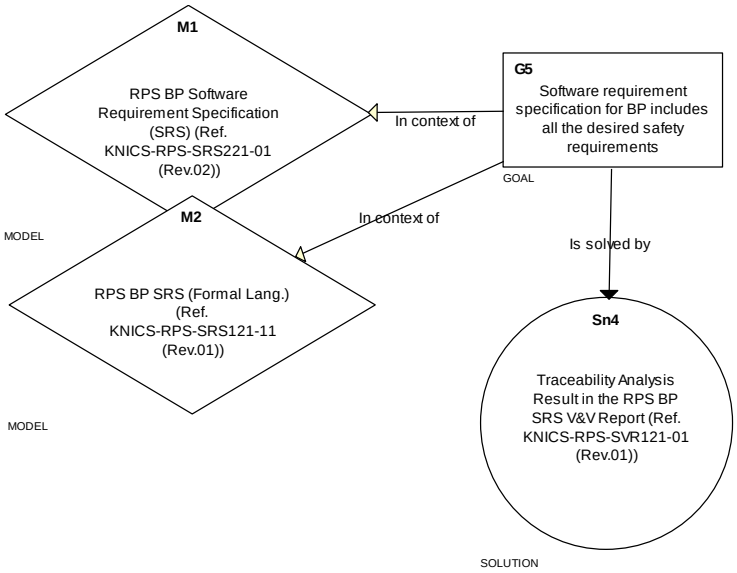

Figure 7. The BP SW safety case – SRS includes all the safety requirements

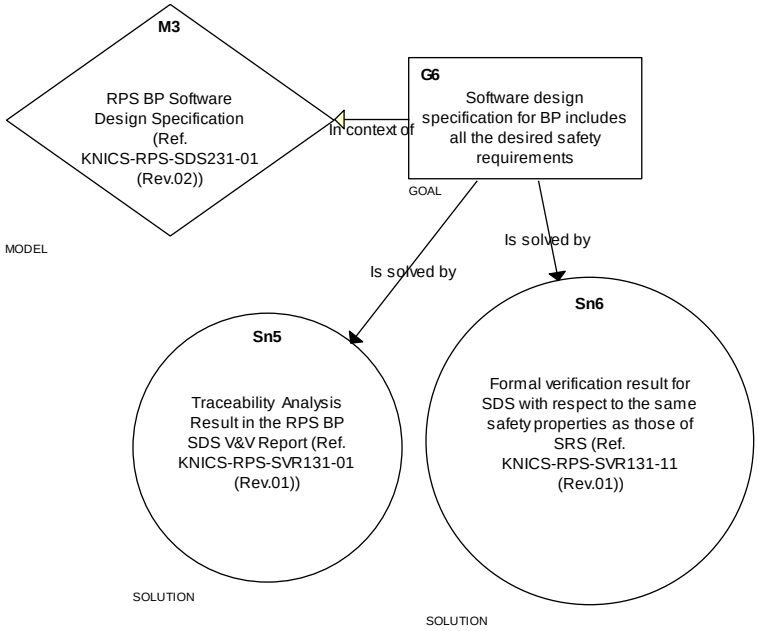

Figure 8. The BP SW safety case – SDS includes all the safety requirements

Figure 7 shows that the traceability analysis result in the BP SRS V&V report is the evidence supporting the claim that the SRS for BP includes all the desired safety requirements. The traceability analysis result links each of the safety requirements included in the design specification for BP with the corresponding safety requirement in the software requirement specification for BP.

Similarly, as shown in Figure 8, the BP SDS is claimed to include all the desired safety requirements and two pieces of evidence are presented. The traceability analysis results in the BP SDS V&V report, showing how each safety requirement in the SRS was not missed in the SDS, is one of the pieces of evidence. Since formal verification with respect to the same safety properties as those of the SRS was conducted for the BP SDS, this can also be provided as evidence.

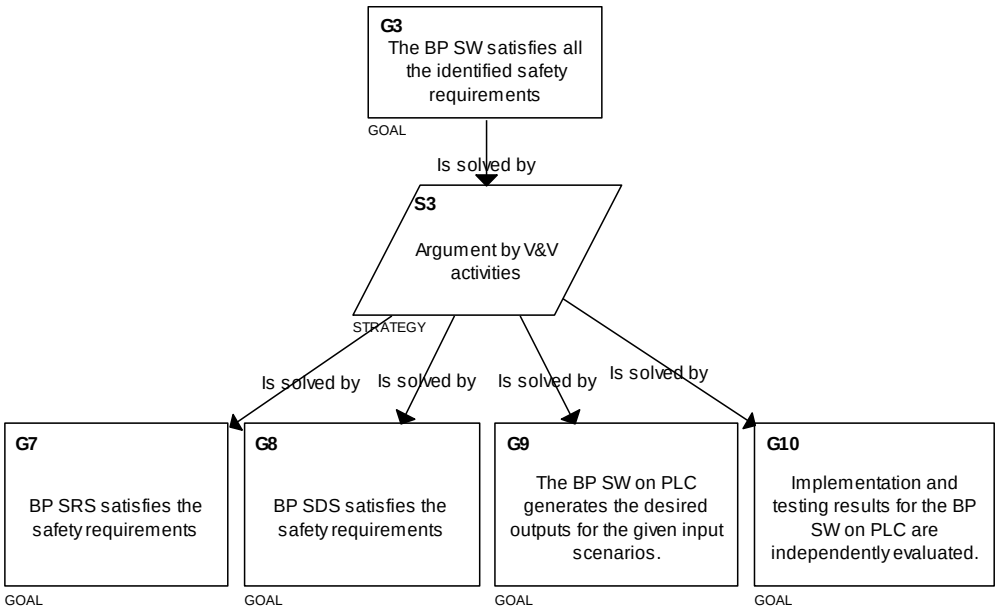

Figure 9. The BP SW safety case – Argument by V&V activities

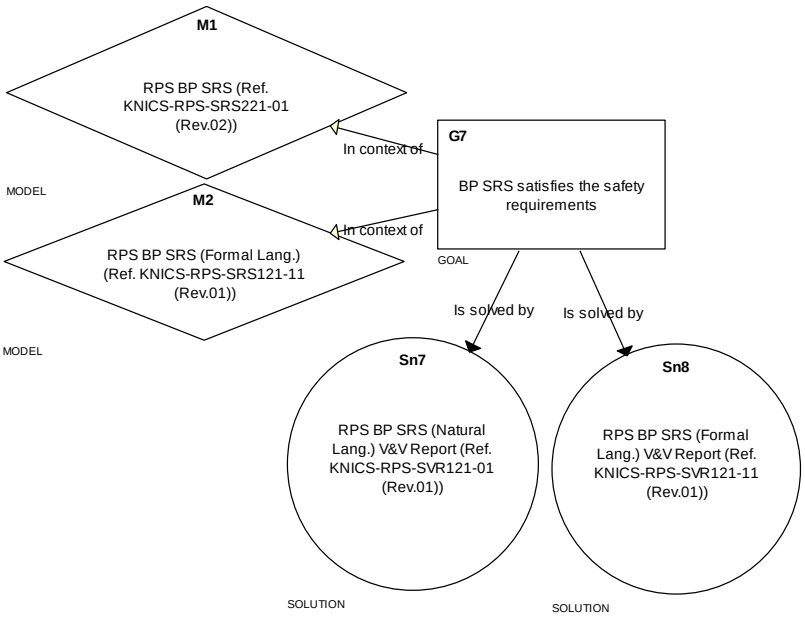

Figure 10. The BP SW safety case – BP SRS satisfies all the safety requirement

As Figure 9 shows, the BP SW is claimed to satisfy all the identified safety requirements by showing that the BP software was developed with the support of rigorous V&V activities at each development phase. The claim “BP SRS satisfies the safety requirements” is supported by the two pieces of evidence, that is, the V&V results for natural language specification and formal specification for BP software requirements, respectively. See Figure 10.

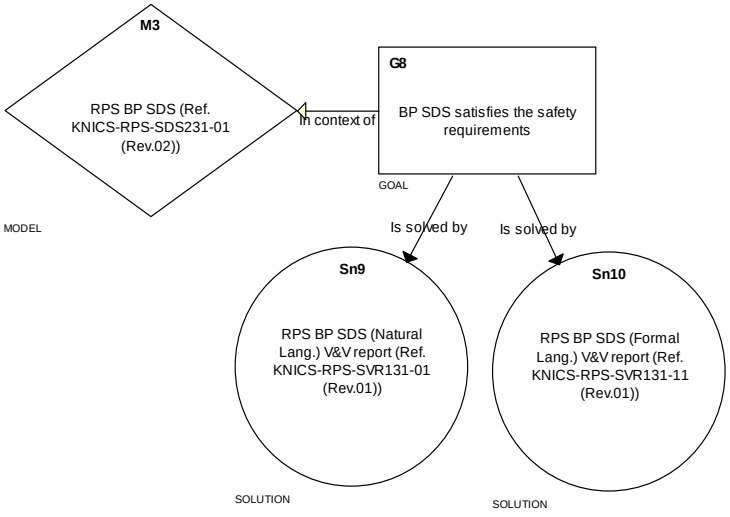

Figure 11. The BP SW safety case – BP SDS satisfies all the safety requirement

Similarly, Figure 11 shows how the satisfaction of the safety requirements of the BP SDS is supported by the evidence of V&V results for the BP SDS.

As to the implementation phase, the claim that the BP SW on PLC generates the desired outputs for the given input scenarios is supported by the evidence of BP SW unit testing results. Additionally, the claim that implementation and testing results for the BP SW on PLC are independently evaluated is supported by the V&V report for BP SW implementation and testing

which includes code inspection results. Figure 12 presents all the arguments and the solutions indicating that the BP SW satisfies all the identified safety requirements.

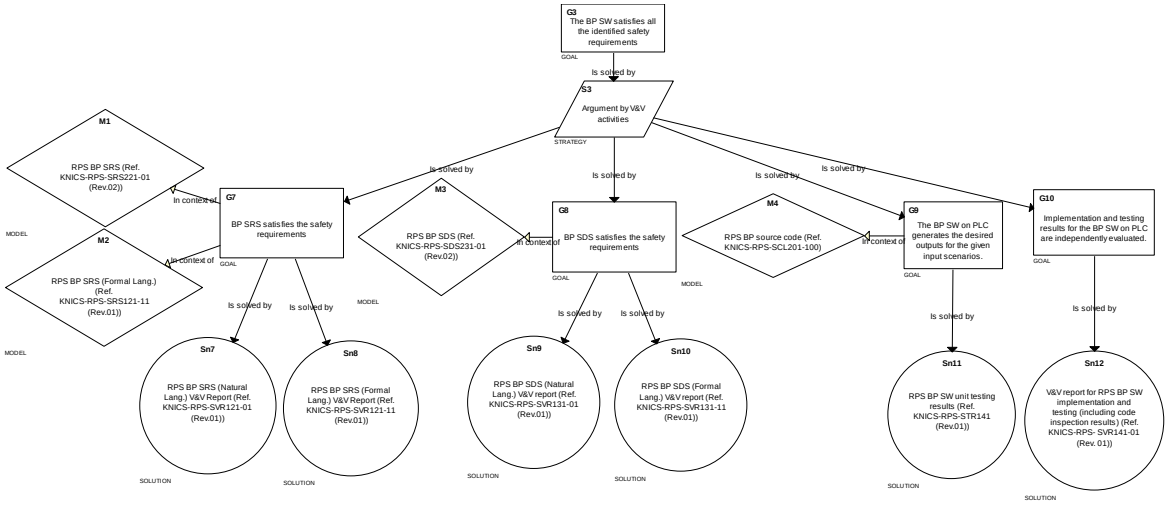

Figure 12. The BP SW safety case – BP SW satisfies all the safety requirement

3.3 Argument by safety analysis activities

This subsection describes how the top claim (the safety of the BP SW) can be argued by safety analysis activities, in addition to satisfaction of safety requirements. As illustrated in Figure 13, if important SW contributable system hazards are not missed (G11) and the remaining or newly introduced hazards through lifecycle are managed (G12), the BP SW can be claimed to be acceptably safe to operate on the PLC.

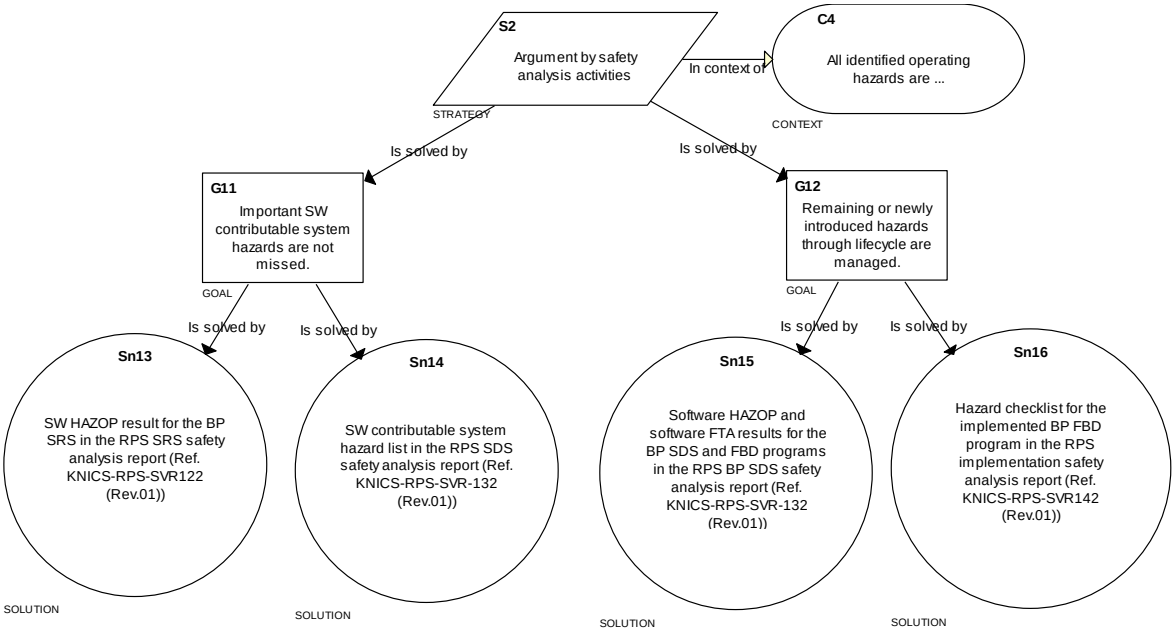

Figure 13. The BP SW safety case – Argument by safety analysis activities

The software HAZOP was performed in the software safety analysis during the requirements phase of the BP development, and software HAZOP [15] and software FTA techniques were used in the

design and implementation phases [8]. Thus, the SW HAZOP result for the BP SRS in the RPS SRS safety analysis report (Sn13) and the SW contributable system hazard list in the RPS SDS safety analysis report (Sn14) can both serve as evidence supporting claim G11.

In order to confirm that the remaining or newly introduced hazards through the lifecycle are managed, the remaining or newly introduced hazards during each development phase need to be traced. In the KNICS BP development, safety analysis was done during each development phase. Therefore, the software HAZOP and software FTA results for the BP SDS and FBD programs included in the BP SDS safety analysis report (Sn15) and the hazard checklist for the implemented BP FBD program in the RPS implementation safety analysis report (Sn16) can serve as evidence supporting claim G12.

4. Discussion

The main motivation for this case study is to investigate whether the safety argument approach can complement the prescriptive approach. We were able to identify possible advantages and drawbacks regarding the use of the safety case approach with the prescriptive approach.

First, we found that the way BP software safety issues had been addressed while following the prescriptive approach could be explicitly presented by creating a safety case. In the case of KNICS RPS, over 500 documents were generated and submitted to the regulators. Reviewing those artifacts from the beginning to the end took significant effort and time. The relevance of each artifact to system safety varies, and figuring out whether a specific part of those documents is more or less important in the aspect of system safety is not an easy task. If safety cases can be submitted to the regulator with the artifacts produced by following prescriptive approaches, clearer and more efficient communication focusing on safety between the developers and the regulators can proceed in the review process for certification.

While following the prescriptive approaches, i.e., conforming to safety-related standards generally requires producing many artifacts such as V&V results and documents, creating additional safety cases entails extra efforts and costs. Several studies on safety case patterns exist [16] [17] [18] [19] [20], but still a significant portion of safety case creation and management relies on manual work. Efforts to develop proper guidelines and tools for creating and managing safety cases should be continued. Combining prescriptive approaches and safety case approaches in an effective and efficient way needs to be studied further.

When applying the safety case approach to the target system developed by conforming to standards, it was found that the safety case approaches were not enough to cover all the requirements since not all the requirements are safety requirements. Requirement specifications can include other aspects of requirements, e.g., security, performance, etc., as well as safety requirements. Generally, the prescriptive approaches consider not only safety requirements, but also other quality attributes of the system. Therefore, the safety case approaches may not be able to replace the prescriptive approaches.

It should be noted that since the presented safety case was created with existing artifacts of an already developed system, we could not evaluate through this case study how the prescriptive approach and the safety case approach can complement each other during the development phase.

5. Conclusion

We created a safety case for a part of the RPS software developed using a prescriptive approach under the KNICS project. Satisfaction of all the desired safety requirements and safety analysis activities were used as the main arguments. A number of existing artifacts including V&V documents and safety analysis documents were provided as evidence to support claims for the safety of the BP software. We investigated whether and how the safety case approach could complement the prescriptive approach. We found that the way BP software safety issues had been addressed while following the prescriptive approach could be explicitly presented by creating a safety case; this merit could contribute to making the review and certification process of the target system clearer and more efficient.

As a feasibility study, the safety case presented in this paper is not complete and has several parts that need to be improved and concretized. We have a plan to revise the safety case and study how the prescriptive approach and the safety case approach can be effectively harmonized for assuring system safety.

6. Acknowledgement

This work was supported by the Nuclear Safety Research Program through the Korea Foundation of Nuclear Safety (KOFONS), granted financial resource from the Nuclear Safety and Security Commission (NSSC), Republic of Korea (No.1403008).

7. References

[1] MoD, Defence Standard 00-56 Issue 4 (Part 1): Safety Management Requirements for Defence Systems, UK Ministry of Defence (MoD).

[2] R. Hawkins, I. Habli, T. Kelly and J. McDermid, "Assurance Cases and Prescriptive Software Safety Certification: A Comparative Study," *Safety Science,* vol. 59, p. 55–71, 2013.

[3] USNRC, NUREG-0800, Rev.04, Standard Review Plan: BTP HICB–14, Guidance on Software Reviews for Digital Computer-Based Instrumentation and Control Systems, U.S. Nuclear Regulatory Commission, 1997.

[4] IEEE Std-1228, ″Software Safety Plan,″ Institute of Electrical and Electronic Engineers, 1994.

[5] K.-C. Kwon and M. Lee, "Technical Review on the Localized Digital Instrumentation and Control Systems," *Nuclear Engineering and Technology,* vol. 41, no. 4, pp. 447-454, 2009.

[6] J. H. Park, D. Y. Lee and C. H. Kim, "Development of KNICS RPS Prototype," in *Proceedings ISOFIC 2005, Session 6*, Tongyeong, 2005.

[7] K.-C. Kwon, D.-Y. Lee and J.-G. Choi, "Software Qualification for Digital Safety System in KNICS Project," in *3rd International Conference on Nuclear Power Plant Life Management for Long Term Operations*, Salt Lake City, USA, 2012.

[8] G.-Y. Park, K. Y. Koh, E. Jee, P. H. Seong and K.-C. Kwon, "Fault Tree Analysis of KNICS RPS Software," *Nuclear Engineering and Technology,* vol. 40, no. 5, p. 397–408, 2008.

[9] K. H. Cha, H. S. Sohn, J.-S. Lee, J. Y. Kim, S. W. Cheon, Y. J. Lee, I. K. Hwang and K. C. Kwon, "The KNICS Approach for Verification and Validation of Safety Software," in *Proceedings of the Korean Nuclear Spring Meeting*, Gyeong ju, Korea, 2003.

[10] K.-C. Kwon and G.-Y. Park, "Formal Verification and Validation of the Safety-Critical Software in Digital Reactor Protection System," in *NPIC & HMIT 2006*, Albuquerque, NM, USA, 2006.

[11] J. Yoo, T. Kim, S. Cha, J.-S. Lee and H. S. Son, "A Formal Software Requirements Specification Method for Digital Nuclear Plants Protection Systems," *Journal of Systems and Software,* vol. 74, no. 1, pp. 73-83, 2005.

[12] G. Y. Park, S. W. Cheon and K. C. Kwon, "Software Requirements V&V Works in KNICS Reactor Protection System," in *Transactions of the Korean Nuclear Society Spring Meeting*, Chuncheon, Korea, 2006.

[13] J. Yoo, S. Cha and E. Jee, "A Verification Framework for FBD based Software in Nuclear Power Plants," in *15th Asia-Pacific Software Engineering Conference*, 2008.

[14] E. Jee, S. Jeon, C. Sungdeok, K. Kwangyong, J. Yoo, G. Park and P. Seong, "FBDVerifier: Interactive and Visual Analysis of Counterexample in Formal Verification of Function Block Diagram," *Journal of Research and Practice in Information Technology,* vol. 42, no. 3, pp. 255-272, 2010.

[15] G.-Y. Park, K.-C. Kwon, E. Jee, K. Y. Koh and P. H. Seong, "Safety Analysis of Safety-Critical Software for Nuclear Digital Protection SystemSafety Analysis of Safety-Critical Software for Nuclear Digital Protection System," in *Proceedings of the 26th International Conference on Computer Safety, Reliability and Security (SAFECOMP), LNCS 4680*, Nuremberg, Germany, 2007.

[16] T. Kelly, ″Arguing Safety: A Systematic Approach to Managing Safety Cases,″ *Ph.D Dissertation*, University of York, 1998.

[17] R. Weaver, ″The Safety of Software – Constructing and Assuring arguments,″ *Ph.D. dissertation*, University of York, 2003.

[18] R. Alexander, T. Kelly, Z. Kurd and J. McDermid, "Safety Cases for Advanced Control Software: Safety Case Patterns," in *Final Report for NASA Contract FA8655-07-1-3025*, 2007.

[19] A. Ayoub, B. Kim, I. Lee and O. Sokolsky, "A Safety Case Pattern for Model-Based Development Approach," in *Proceedings of the NASA Formal Methods Symposium (NFM)*, 2012.

[20] E. W. Denney and G. J. Pai, "Safety Case Patterns: Theory and Applications," in *NASA/TM–2015–218492*, 2015.