The Transactions of the Korea Information Processing Society
TKIPS Vol.13, No.5, pp.227~235

eISSN: 3022-7011
https://doi.org/10.3745/TKIPS.2024.13.5.227

Development of Reliability Measurement Method and Tool for Nuclear Power Plant Safety Software

Lingjun Liu[†] · Wooyoung Choi[††] · Eunkyoung Jee[†††] · Duksan Ryu[††††]

ABSTRACT

Since nuclear power plants (NPPs) increasingly employ digital I&C systems, reliability evaluation for NPP software has become crucial for NPP probabilistic risk assessment. Several methods for estimating software reliability have been proposed, but there is no available tool support for those methods. To support NPP software manufacturers, we propose a reliability measurement tool for NPP software. We designed our tool to provide reliability estimation depending on available qualitative and quantitative information that users can offer. We applied the proposed tool to an industrial reactor protection system to evaluate the functionality of this tool. This tool can considerably facilitate the reliability assessment of NPP software.

Keywords : Software Reliability Measurement, Nuclear Power Plant Software, Bayesian Belief Network, Statistical Testing, Reliability Measurement Tool

원자력 안전 소프트웨어 대상 신뢰도 측정 방법 및 도구 개발

Lingjun Liu[†] · 최 우 영[††] · 지 은 경[†††] · 류 덕 산[††††]

요 약

원자력발전소에서 디지털 계측제어 시스템 비중이 높아지면서 원자력발전소에 대한 확률론적 안정성 평가 시 소프트웨어에 대한 신뢰도 평가가 중요해졌다. 원전 소프트웨어 신뢰도 추정을 위한 방법들이 몇 가지 제안 되었지만 해당 방법의 효과적 적용을 지원하는 도구 지원이 미비하였다. 본 연구에서는 소프트웨어 개발 품질 및 검증 품질과 같은 정성적 정보와 통계적 시험 결과와 같은 정량적 정보를 활용하여 원전 소프트웨어 신뢰도를 정량적으로 측정할 수 있는 자동화 도구를 설계하였고 구현하였다. 개발된 도구를 산업용 원자로 보호 시스템 사례에 적용한 결과, 개발된 도구가 원전 소프트웨어의 신뢰성 평가를 효과적으로 지원할 수 있음을 확인하였다.

키워드 : 소프트웨어 신뢰도 측정, 원자력 소프트웨어, 베이지안 빌리프 네트워크, 통계적 테스트, 신뢰도 측정 도구

1. 서 론

In nuclear power plants (NPPs), instrumentation and control (I&C) systems are implemented for monitoring the status of NPP, control, and failure response for protection. With the advances in digital technology, the nuclear power industry has begun employing digital I&C systems when building new NPPs, and the traditional analog I&C systems are superseded by digital ones [1]. As digital systems are increasingly used in NPPs, it is essential to measure the reliability of NPP software for NPP probabilistic risk assessment (PRA).

IEEE-1633 standard [2] defines software reliability as the probability of failure-free operation in a given environment over a specified period of time. In NPPs, I&C systems should perform necessary safety actions to prevent accidents from occurring. Reliability estimation for NPP software focuses on failures on demand, the accidental scenarios where the software should take safety actions. In this study, we utilize the reliability metric "probability of failure on demand (PFD)."

※ 본 연구는 원자력안전위원회의 재원으로 한국원자력안전재단의 지원을 받아 수행한 원자력안전연구사업의 연구결과입니다(No. 2105030).
※ 이 논문은 2024년 KCSE 2024의 우수논문으로 "A Reliability Evaluation Tool for Nuclear Power Plant Safety Software"의 제목으로 발표된 논문을 확장한 것임.
† 비 회 원 : 한국과학기술원 전산학부 연구원
†† 비 회 원 : 한국과학기술원 전산학부 학사과정
††† 정 회 원 : 한국과학기술원 전산학부 연구부교수
†††† 종신회원 : 전북대학교 소프트웨어공학과 부교수
Manuscript Received : March 14, 2024
First Revision : April 19, 2024
Accepted : April 22, 2024
* Corresponding Author : Eunkyoung Jee(ekjee@se.kaist.ac.kr)

Many attempts have been made to quantify the reliability of NPP software. U.S. Nuclear Regulatory Commission (NRC) investigated various quantitative software reliability methods, and they selected potential candidates, the Bayesian belief network (BBN) method and the statistical black-box testing method, to quantify the reliability of digital systems in NPP PRA [3]. Kang et al. [4] developed a BBN model that measures the number of faults considering software development life cycle (SDLC) processes and derives the PFD of NPP digital systems. Chu et al. [5] proposed a statistical black-box testing approach that validates the reliability goal by testing the software over test cases generated from operational profiles of demand scenarios. Cai et al. [6] presented a hybrid approach, combining BBN and statistical testing methods [4, 5], to fully consider the factors associated with software reliability. However, there is no available tool for NPP software manufacturers to adopt their approach.

There is a lack of consensus on a reliability evaluation method for NPP software. Existing methods are not flexible enough to be applicable in situations where the availability of information is uncertain. In addition, there is a lack of reliability measurement tools for NPP software manufacturers to verify the NPP software reliability. Without a systematic tool, it is challenging for NPP software manufacturers to utilize existing methods since they need to build from scratch.

We propose a systematic reliability evaluation tool for NPP software. Inspired by Cai et al.'s approach, we developed a tool to support the flexibility for reliability estimation, which can adjust to available information, including qualitative and quantitative evaluation results. This tool can considerably ease the reliability assessment of NPP software for NPP software manufacturers and regulatory agencies.

The rest of this paper is structured as follows. Section 2 describes the background knowledge. Section 3 presents the proposed reliability measurement tool for NPP software. Section 4 evaluates the functionality by applying our tool to the industrial system. Section 5 discusses related studies. Section 6 concludes this work.

2. Background

2.1 BBN Model for SDLC Phases

Fig. 1 illustrates a procedure that applies a BBN model-based method to measure software reliability based on Kang et al.'s work [4]. For each SDLC phase, the number of remaining defects is estimated by a BBN submodel considering development quality, V&V quality, and the number of function points (FPs). The development/V&V quality is inferred based on the attribute model, which com-

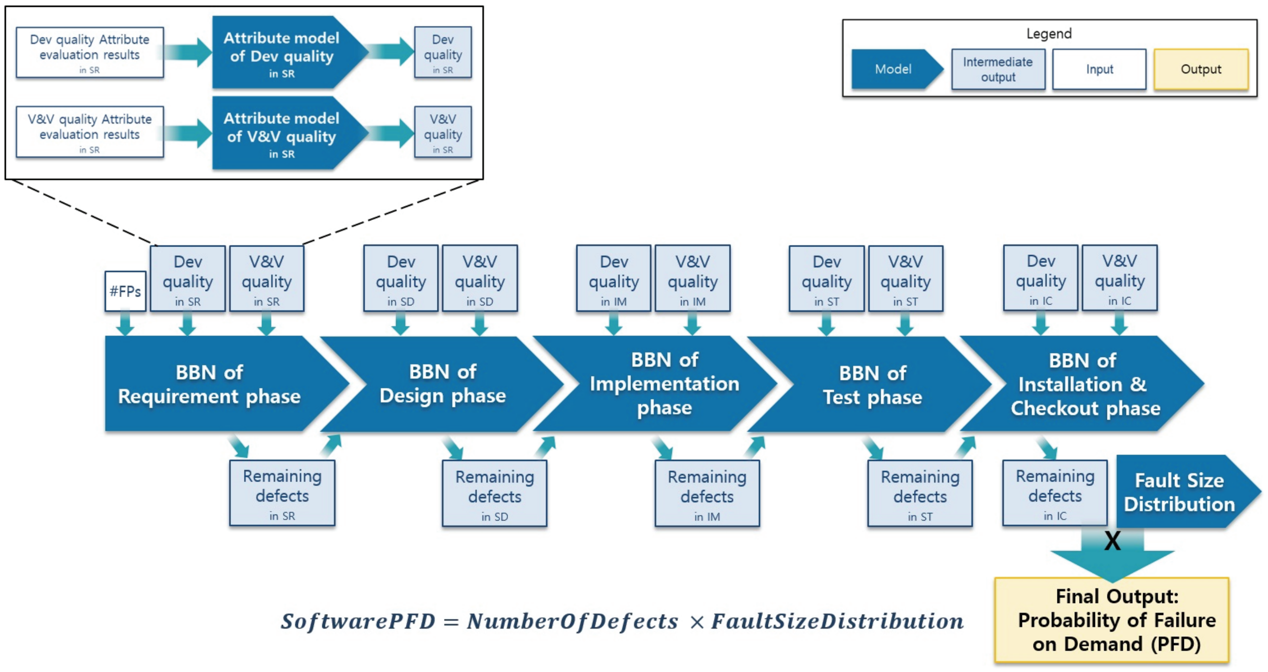


Fig. 1. High Level Structure of BBN Model

*Requirement phase: SR, Design phase: SD, Implementation phase: IM, Test phase: ST, Installation&Checkout phase: IC

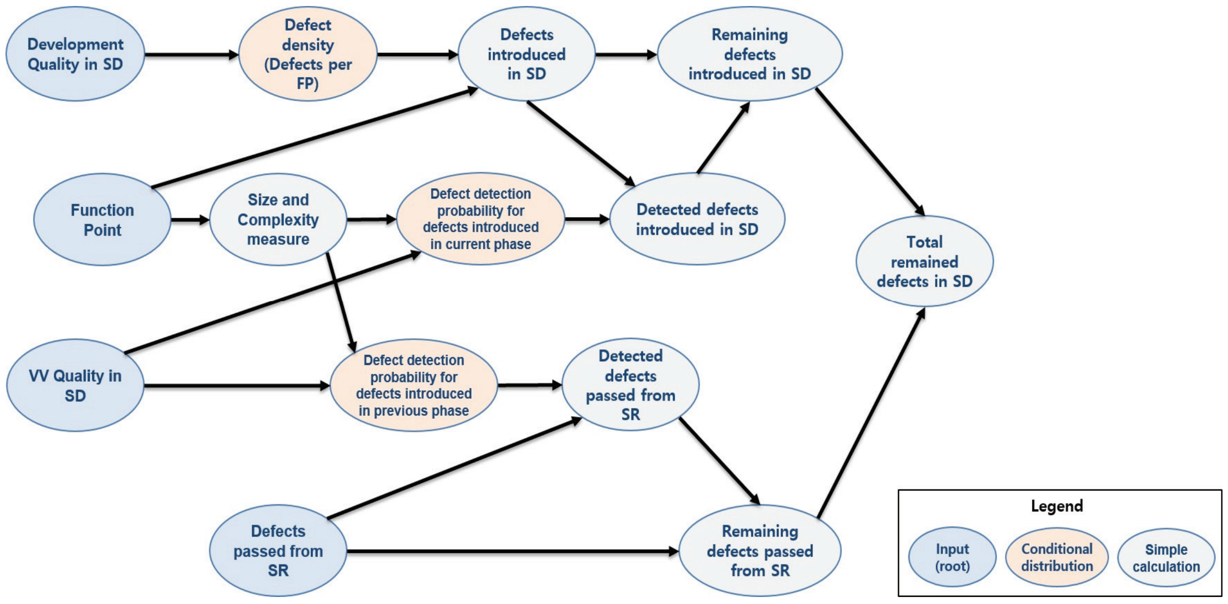

Development Quality in SD
Defect density (Defects per FP)
Defects introduced in SD
Remaining defects introduced in SD
Function Point
Size and Complexity measure
Defect detection probability for defects introduced in current phase
Detected defects introduced in SD
Total remained defects in SD
VV Quality in SD
Defect detection probability for defects introduced in previous phase
Detected defects passed from SR
Defects passed from SR
Remaining defects passed from SR
Legend
Input (root)
Conditional distribution
Simple calculation

Fig. 2. BBN Model of the Design (SD) Phase [4]

prises the development/V&V activities performed in the phase.

The number of remaining defects in the current phase is introduced to the model of the next phase to estimate the number of remaining defects in the next phase. For example, the number of remaining defects in the Requirement (SR) phase is passed to the BBN model of the Design (SD) phase to calculate the number of remaining defects in the SD phase. The PFD of NPP software is calculated by multiplying the number of remaining defects in the Installation and Checkout (IC) phase and generic fault size distribution (FSD).

Fig. 2 depicts the BBN model of the Design SD phase, which is used to quantify the number of remaining defects in the SD phase. The number of defects introduced in the current (SD) phase is estimated based on the number of FPs and the defect density (i.e., defects per FP), which is determined by the development quality. The number of detected defects is derived by the defect detection probability, which is influenced by the software complexity and the V&V quality.

2.2 Statistical Black-box Testing Method

The statistical testing method proposed by Chu et al. [5] is included in this study to incorporate quantitative testing results. Based on Bayes' theorem, Chu et al. calculate the required number of test cases to validate a given reliability target. Assuming that the prior PFD follows a Beta (a, b) distribution, the likelihood follows a binomial distribution and represents the observations from testing results (the number of test cases N and the number of failures F). The posterior distribution of PFD will follow a Beta (F+a, N-F+b) distribution. The mean of posterior PFD can be calculated based on the formula below:

$$mean of posterior PFD = \frac{a+F}{a+b+N} \tag{1}$$

The reliability target is considered to be the mean of the posterior distribution of PFD. Assuming that no failures occur (F=0), the number of demands N can be estimated with the reliability target and the prior PFD, Beta(a, b), based on the formula (1).

They assume the prior distribution of PFD to follow a Beta(1,1) distribution. The posterior distribution of PFD (PFD^{post}) can be inferred as Beta(F+1, N-F+1). The mean of PFD^{post} is (F+1)/(N+2). With no failures, the number of test cases N can be calculated based on the above equation with a reliability target.

3. Reliability Measurement Tool for NPP Software

The overall process of the proposed reliability measurement tool is shown in Fig. 3. Through our tool, target system experts can evaluate qualitative attributes based on ratings (Step 1) and input the number of FPs (Step 2). Our tool then starts PFD inference based on the BBN model using the WinBUGS tool [7] (Step 3). The parame-

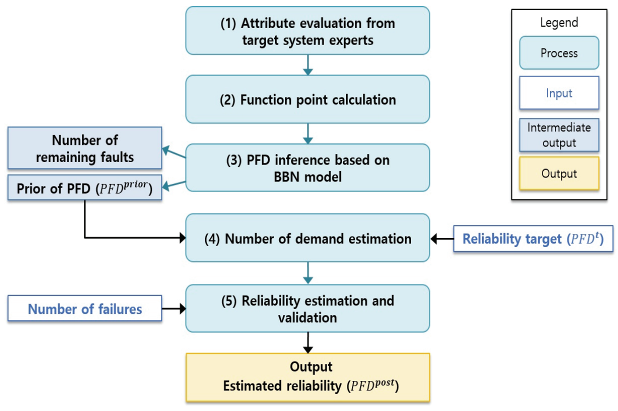

(1) Attribute evaluation from target system experts
(2) Function point calculation
(3) PFD inference based on BBN model
(4) Number of demand estimation
(5) Reliability estimation and validation
Output Estimated reliability (PFD^{post})
Number of remaining faults
Prior of PFD (PFD^{prior})
Reliability target (PFD^{t})
Number of failures
Legend
Process
Input
Intermediate output
Output

Fig. 3. Overall Process of Reliability Measurement Tool

ters of the BBN model are specialized for NPP software and recorded in the U.S.NRC technical report [8]. Users can obtain the inferred number of remaining faults in NPP software and the initial PFD before considering operational conditions.

Given a reliability target, our tool can calculate the number of demands (i.e., test cases) required for statistical testing (Step 4). After testing, our tool can update the prior PFD with the number of failures and estimate the reliability (i.e., updated PFD) to consider the development and V&V qualities during SDLC phases and software operation under demand conditions (Step 5-reliability estimation). Furthermore, our tool can validate if the reliability goal has been accomplished based on testing results (Step 5-reliability validation).

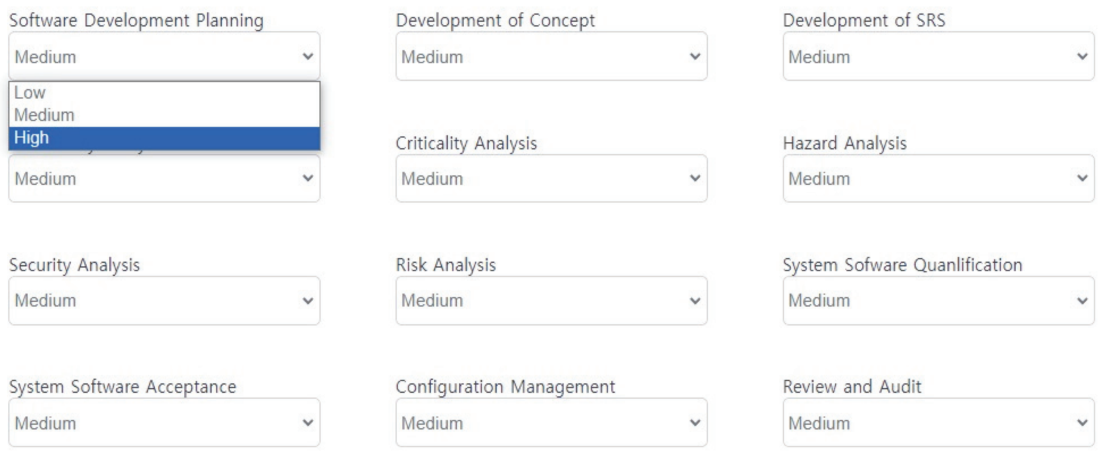
3.1 Attribute Evaluation

For each SDLC phase, development and V&V qualities are required to estimate the number of remaining defects. Development and V&V qualities of each phase depend on the qualities of performing corresponding development and V&V activities. The activities performed in the development/V&V process are considered the attributes affecting the development/V&V quality. For instance, Fig. 4 presents 12 attributes of development quality in the requirement phase. As shown in Fig. 4, there are three levels, High, Medium, and Low, for evaluating the attributes. For each attribute, a collection of tasks is required to be performed. If all the required tasks and additional tasks to improve quality are carried out, the attribute quality is scored as High; if all the required tasks are undertaken, the attribute quality is scored as Medium; if only some of the required tasks are performed, the attribute quality is scored as Low.

Quantifying the number of faults in NPP software requires a total of 10 development/V&V qualities during the SDLC process. In order to infer those qualities, there is a total of 125 attributes included in our tool. For each attribute, we set the default value to a Low level so that our tool can still calculate the reliability even if the information on attribute quality is unavailable. For example, after the implementation phase, users can evaluate all the attributes before the test phase. However, our tool can still deliver the least expected reliability (PFD) for users with the default values for the rest of the phases. During software development, our tool can support users to anticipate the reliability results and further monitor reliability progress.

3.2 Function Point Calculation

The function point (FP) is referred to as a unit to meas-



Software Development Planning: Medium	Development of Concept: Medium	Development of SRS: Medium
Medium	Criticality Analysis: Medium	Hazard Analysis: Medium
Security Analysis: Medium	Risk Analysis: Medium	System Sofware Quanlification: Medium
System Software Acceptance: Medium	Configuration Management: Medium	Review and Audit: Medium

Fig. 4. Qualitative Attributes of Development Quality in the Requirement Phase

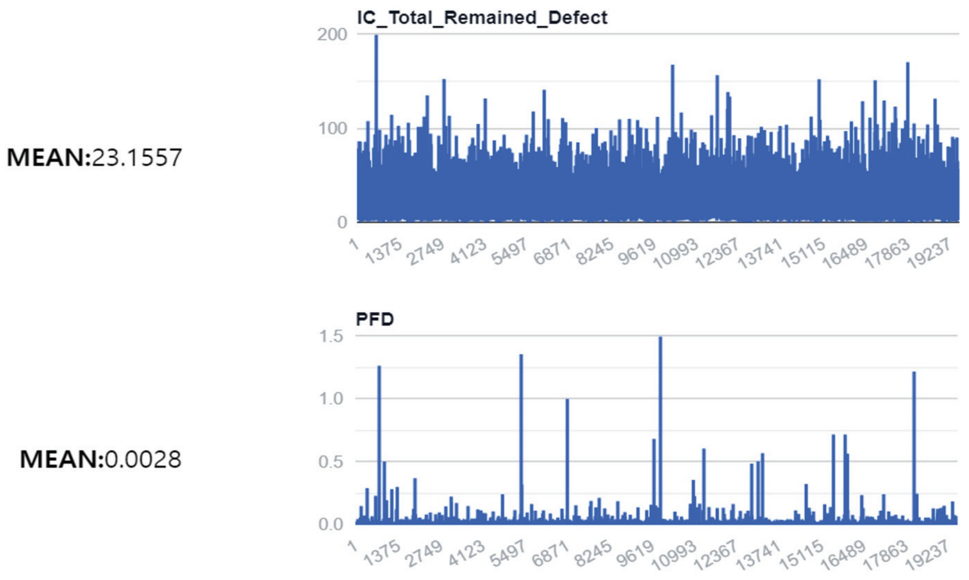


Fig. 5. Reliability measurement results

ure the software size based on the software's functionality view [9]. The number of FPs was used to calculate the number of introduced faults and an indicator of software complexity. The number of FPs is assumed to be provided by users. Users can utilize the function point calculation method tailored to their software, e.g., the function point analysis method from IFPUG [10] and function point conversion methods from LOC data [11].

3.3 PFD Inference based on BBN Model

After collecting available attribute qualities in each SDLC phase and the number of function points, our tool uses the WinBUGS tool to quantify the number of remaining faults and the prior PFD. Fig. 5 shows the reliability results view of our tool. The reliability results view shows the mean values and simulation traces of the number of remaining defects in the IC phase and prior PFD inferred by the BBN model. A simulation trace contains 19,500 simulated values, and a mean value is calculated from each simulation trace.

3.4 Number of Demand Estimation

Given a reliability target, our tool can calculate the required number of demands or test cases with a prior distribution of PFD using formula (1). The reliability target is considered to be achieved if the software is tested over the number of required test cases and no failures occur. The prior PFD based on the BBN model can be utilized to estimate the number of demands. If the qualitative information for each SDLC phase is unavailable, the uninformative uniform distribution can be used to estimate the number of demands.

3.5 Reliability Estimation and Validation

We assumed test generation and execution have already been performed by users since our tool focuses on reliability estimation instead of testing. With the number of test cases and failures, our tool can estimate the reliability (PFD) based on the prior PFD. The prior PFD can be either the informative distribution inferred from the qualitative information of SDLC or the uninformative uniform distribution. The reliability target can be validated if no failure occurs among the required number of tests. Our tool can adapt to the uncertainty in the availability of qualitative information and various reliability test results.

4. Case Study and Evaluation

4.1 Target System

To assess the tool functionality, we chose the same target system in previous BBN-based reliability quantification work [4], i.e., the Integrated Digital Protection System-Reactor Protection System (IDiPS-RPS) from the Korea Nuclear Instrumentation and Control System (KNICS) project [12]. The IDiPS-RPS monitors process variables (e.g., measured pressure and temperature) and generates trip

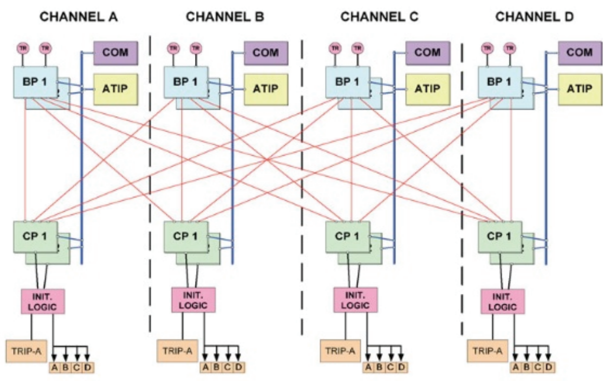


Fig. 6. Configuration of IDiPS-RPS [13]

signals to shut down the nuclear reactor when process variables attain their respective trip setpoints. Fig. 6 presents the configuration of the IDiPS-RPS. The IDiPS-RPS is composed of four redundant channels, and each channel comprises four main processors: the bistable processor (BP), the coincidence processor (CP), the automatic test and interface processor (ATIP), and the cabinet operator module (COM). The trip function is implemented in the BP, and the related voting function is implemented in the CP.

4.2 Application to Target System

1) Estimating the Prior PFD based on BBN Model

With the attribute qualities collected from Kang et al.'s work, we applied our tool to obtain the reliability results for the IDiPS-RPS. The attributes were evaluated before the installation phase, so all the attribute qualities for the installation phase were set to a default Medium level. The number of function points was set to 56, the mean of the number of function points estimated in Kang et al.'s work. Table 1 outlines the estimated number of remaining faults in each SDLC phase. With the number of remaining faults in the installation phase and generic FSD, the distribution of PFD for the IDiPS-RPS is inferred and presented in Table 2.

2) Calculating the Number of Demands

Based on nuclear safety software used in related studies [5, 6], we assumed IDiPS-RPS to have a reliability target of 10E-4. Thus, the mean of posterior PFD is expected to achieve 10E-4. With the prior PFD from the BBN model and the reliability target, we calculated the number of demands using the formula (1). Since we assume the prior PFD follows a Beta(a, b) distribution, we applied the distribution fitting method to find the best beta distribution that fits the distribution of the prior PFD obtained from

Table 1. The Number of Remaining Faults in Each SDLC Phase

Phase	Mean	Standard deviation	5%	Median	95%
Requirement	7.937	8.935	0.4593	4.9	25.9
Design	26.33	18.57	5.462	21.79	62.57
Implementation	40.8	24.3	11.28	35.82	87.4
Test	28.3	16.07	8.882	25.06	59.16
Installation	11.27	9.107	1.714	8.829	28.97

Table 2. The PFD for the IDiPS-RPS

Mean	Standard deviation	5%	Median	95%
1.142E-3	1.054E-2	9.927E-7	5.789E-5	3.491E-3

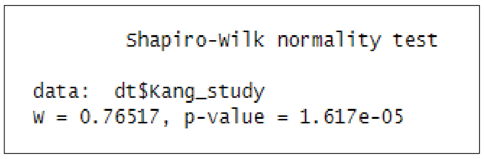
the BBN model. We utilized the distribution fitting method provided by the beta.fit() function in the Python library SciPy [14]. The distribution fitting results indicated that the parameters a and b are 0.2382 and 148.9146, respectively. Therefore, based on formula (1), IDiPS-RPS is required to pass 2233 (step 1: substituting a: 0.2382 and b: 148.9146 into the formula (1) $\frac{0.2382+0}{0.2382+148.9146+N}=10^{-4}$, step 2: calculating N $N=\left(\frac{0.2382+0}{10^{-4}}\right)-(0.2382+148.9146)\approx 2233$) test cases to reach the reliability target 10E-4.

4.3 Evaluation

1) Evaluation Method

To evaluate the functionality of the BBN model in our tool, we performed hypothesis testing to check if the results obtained from our tool were significantly different from those presented in Kang et al.'s work. We collected a total of 30 pairs of data points, including mean, standard deviation, 5%, median, and 95% values from PFD and numbers of remaining faults for SDLC phases. A total of 60 data points were gathered from Table 1 and 2 and Table 9 and 14 in [4].

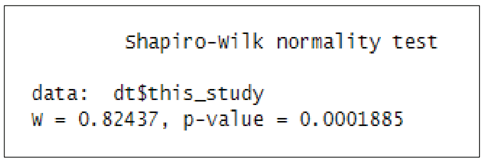
To select a hypothesis test tailored to our data, we applied the Shapiro-Wilk test to check the normality of the distributions of data points and differences between two data sets. Fig. 7, 8, and 9 present the results of the Shapiro-Wilk test for the data set from Kang et al.'s work, the data set obtained from our tool, and differences between two data sets using R studio [15]. The p-values of three test results are less than 0.05, indicating that we can



```
        Shapiro-Wilk normality test

data:  dt$Kang_study
W = 0.76517, p-value = 1.617e-05
```

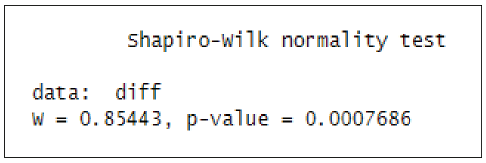
Fig. 7. Shapiro-wilk Test Results of the Data Set from Kang et al.'s Work



```
        Shapiro-Wilk normality test

data:  dt$this_study
W = 0.82437, p-value = 0.0001885
```

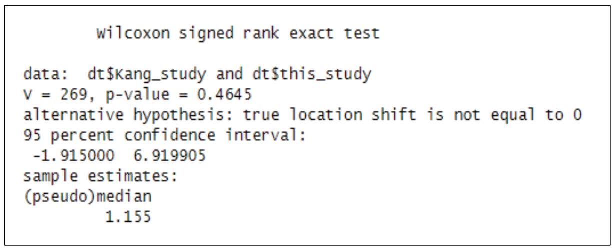
Fig. 8. Shapiro-wilk Test Results of the Data Set Obtained from Proposed Tool



```
        Shapiro-Wilk normality test

data:  diff
W = 0.85443, p-value = 0.0007686
```

Fig. 9. Shapiro-wilk Test Results of Differences Between Two Data Sets



```
        Wilcoxon signed rank exact test

data:  dt$Kang_study and dt$this_study
V = 269, p-value = 0.4645
alternative hypothesis: true location shift is not equal to 0
95 percent confidence interval:
 -1.915000  6.919905
sample estimates:
(pseudo)median
         1.155
```

Fig. 10. Wilcoxon Matched-pairs Signed-rank Test Results

reject the null hypothesis that data sets follow a normal distribution. Since the distributions of data points and differences between two datasets were not in accordance with a normal distribution, we conducted the Wilcoxon matched-pairs signed-rank test, a non-parametric test, to investigate whether a significant difference exists between two data sets or not.

2) Results and Analysis

Fig. 10 shows the Wilcoxon matched-pairs signed-rank test result in R studio. The p-value of the Wilcoxon matched-pairs signed-rank test is 0.4645, which is higher than 0.05. Thus, we failed to reject the null hypothesis that there is no significant difference between the results from our tool and Kang et al.'s work. Based on the hypothesis testing results, the developed BBN model in our tool conforms to the functionality of Kang et al.'s approach.

3) Discussion for Demand Estimation

We compared the estimated number of demands (2233) with the number of demands estimated in the statistical testing study [5]. In the statistical testing study, the reliability target was set to 10E-4, and Chu et al. estimated that their subject software required to pass 10,000 tests to achieve the reliability target with a prior distribution following a uniform distribution. The mean of uniform distribution is 0.5, which is substantially high for safety software [6]. Moreover, Kang et al. quantified the generic PFD based on operation experiences of NPP safety software worldwide, and the distribution of the generic PFD assessed has a mean value of 5.53E-4. Based on our estimation results, the mean of the prior PFD estimated from the BBN model is 1.142E-3, which has a considerable difference from the mean of uninformative uniform distribution. Therefore, the estimated number of demands in our tool is more tailored to the quality of the development process than the statistical testing study.

5. Related Work

Research has been ongoing to develop reliability estimation methods for non-nuclear software based on software reliability growth models (SRGMs). Nevertheless, NPP software has substantial differences from non-nuclear software in view of the number of defects due to the rigorous V&V activities during SDLC phases [13]. Kim et al. [16] applied time-to-failure models, the Jelinski-Moranda model and Goel-Okumoto's NHPP model, to safety-critical software. Son et al. [17] applied a failure count model, the delayed s-shaped growth model, for NPP software. Kim et al. and Son et al. point out failure data sufficiency as a prerequisite for applying SRGMs. However, the availability of failure data is uncertain. Park et al. [13] employed a failure count model, NHPP-based model, to NPP software and showed that SRGMs tend to overestimate the reliability of NPP software owing to the scarcity of failure data in NPP software.

Kang et al. [4] proposed a BBN-based method to estimate the number of remaining defects and PFD for NPP software considering qualitative information during SDLC phases. The considered faults may include the faults causing false alarms, making the estimation of PFD too conservative.

Chu et al. [5] proposed a statistical testing method based on the PRA context. They used the PRA context and operational profile to generate scenarios of demand conditions, such as abnormal events in the reactor. The thermal-hydraulic simulation model is utilized to simulate the test environment of the reactor. To apply this approach, they assume that a PRA model and an appropriate thermal-hydraulic model have been developed. They used the uniform distribution (the Beta(1,1) distribution) as the prior distribution to estimate PFD. This uninformative assumption of prior distribution brought uncertainty in reliability quantification.

Cai et al. [6] combined Kang et al.'s and Chu et al.'s approaches to estimate reliability considering qualitative information during SDLC and operational conditions. The studies discussed above do not provide tools to support their methods and the flexibility to adapt to the uncertainty in data availability.

6. Conclusion

This paper proposed a systematic reliability estimation tool for NPP software. Our tool provides the flexibility of available qualitative information during SDLC and quantitative testing results. We evaluated our tool functionality by applying it to the IDiPS-RPS. The reliability results generated from our tool have no significant difference from Kang et al.'s results. The number of demands calculated by our tool is more informative to reflect the quality of the SDLC process compared to Chu et al.'s work. The parameters of the BBN model can be further extended for safety-critical software in other domains once expert knowledge is available. In future work, we plan to develop a reliability test generation tool for NPP software.

References

[1] International Atomic Energy Agency (IAEA), "Instrumentation and Control (I&C) Systems in Nuclear Power Plants: A Time of Transition," *Nuclear Technology Review*, pp. 83-94, 2008.

[2] A. M. Neufelder, "IEEE Recommended Practice on Software Reliability," IEEE Standard, 1633-2016.

[3] T. L. Chu, "Development of quantitative software reliability models for digital protection systems of nuclear power plant," United States Nuclear Regulatory Commission, Office of Nuclear Regulatory Research, 2013.

[4] H. G. Kang et al., "Development of a Bayesian belief network model for software reliability quantification of digital protection systems in nuclear power plants," *Annals of Nuclear Energy*, Vol.120, pp.62-73, 2018.

[5] T. L. Chu, "Development of a statistical testing approach for quantifying safety-related digital system on demand failure probability," United States Nuclear Regulatory Commission, Office of Nuclear Regulatory Research, 2017.

[6] Y. Cai, Y. Wu, J. Zhou, M. Liu, and Q. Zhang, "Quantitative software reliability assessment methodology based on Bayesian belief networks and statistical testing for safety-critical software," *Annals of Nuclear Energy*, Vol. 145, pp.107593, 2020.

[7] D. Spiegelhalter, A. Thomas, N. Best, & D. Lunn, "WinBUGS user manual," 2003.

[8] T. L. Chu, "Developing a Bayesian belief network model for quantifying the probability of software failure of a protection system," United States Nuclear Regulatory Commission, Office of Nuclear Regulatory Research, 2018.

[9] A. J. Albrecht, "Measuring application development productivity," *Proc. Joint Share, Guide, and Ibm Application Development Symposium*, 1979.

[10] International Function Point Users Group (IFPUG), "Function Point Counting Practices Manual," 2000.

[11] C. Jones, "Applied Software Measurement: Global Analysis of Productivity and Quality," McGraw Hill, 2008.

[12] J. H. Park, D. Y. Lee, and C. H. Kim, "Development of KNICS RPS Prototype," *Proceedings of ISOFIC (International Symposium on Future I&C) 2005*, pp.160-161, 2005.

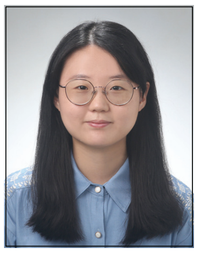
[13] G. Y. Park and S. C. Jang, "A software reliability estimation method to nuclear safety software," *Nuclear Engineering and Technology*, Vol.46, No.1, pp.55-62, 2014.

[14] P. Virtanen et al., "SciPy 1.0: fundamental algorithms for scientific computing in Python," *Nature Methods*, Vol.17, No.3, pp.261-272, 2020.

[15] J. S. Rachine, "RStudio: a platform-independent IDE for R and Sweave," *Journal of Applied Econometrics*, Vol.27, No.1, pp.167-172, 2012.

[16] M. C. Kim, S. C. Jang, and J. J. Ha, "Possibilities and limitations of applying software reliability growth models to safety-critical software," *Nuclear Engineering and Technology*, Vol.39, No.2, pp.129-132, 2007.

[17] H. S. Son, H. G. Kang, and S. C. Chang, "Procedure for application of software reliability growth models to NPP PSA," *Nuclear Engineering and Technology*, Vol.41, No.8, pp.1065-1072, 2009.



Lingjun Liu

https://orcid.org/0000-0001-6802-4090

e-mail : riensha@se.kaist.ac.kr

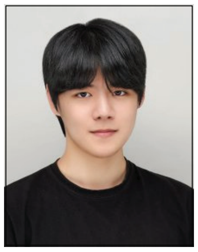
She is a full-time researcher in School of Computing at Korea Advanced Institute of Science and Technology (KAIST). She received her B.S. degree in Computer Science from National Tsing Hua University in 2019 and her M.S. degree in School of Computing from KAIST in 2021. Her research interests include software testing, software reliability, safety-critical systems, and system- of-systems engineering.

Eunkyoung Jee

https://orcid.org/0000-0003-0358-5369

e-mail : ekjee@se.kaist.ac.kr

She is a Research Associate Professor in School of Computing at KAIST. She was a Postdoctoral Researcher in the Computer and Information Science Department at the University of Pennsylvania. She received her B.S., M.S., and Ph.D. degrees in Computer Science from KAIST. Her research interests include safety-critical software, software testing, formal verification, and safety analysis.



Wooyoung Choi

https://orcid.org/0009-0009-8935-3032

e-mail : uy0417@kaist.ac.kr

He is currently pursuing the B.S. degree in School of Computing at KAIST. His research interests are in the areas of timed automata, model verification, and blockchain.

Duksan Ryu

https://orcid.org/0000-0002-9556-0873

e-mail : duksan.ryu@jbnu.ac.kr

He is currently an Associate Professor with the Software Engineering Department, Jeonbuk National University. He received the bachelor's degree in computer science from Hanyang University, in 1999, the dual master’s degree in software engineering from KAIST and Carnegie Mellon University, in 2012, and the Ph.D. degree from the School of Computing, KAIST, in 2016. His research interests include software analytics based on AI, software defect prediction, mining software repositories, and software reliability engineering.