Estimation of Remained defects in a Safety-Critical Software using Bayesian Belief Network of Software Development Life Cycle

Seung Jun Lee and Wondea Jung
Korea Atomic Energy Research Institute
1045 Daedeok-daero, Yuseong-gu, Daejeon, 305-353, Korea
[*]Corresponding author: sjlee@kaeri.re.kr

1. Introduction

Extensive research has been performed for decades to quantify software quality in terms of the probability of failure on demand (PFD). Although today a number of methodologies are available, there is no methodology on the methodology suitable for reliability assessment of safety critical software of nuclear power plants (NPPs). Some researchers recognized Bayesian belief network (BBN) method to be a promising method of quantifying software reliability [1,2].

Brookhaven National Laboratory (BNL) comprehensively reviewed various quantitative software reliability methods to identify the most promising methods for use in probabilistic safety assessments (PSAs) of digital systems of NPPs against a set of the most desirable characteristics developed therein [2,3]. BBNs are recognized as a promising way of quantifying software reliability and are useful for integrating many aspects of software engineering and quality assurance. The method explicitly incorporates important factors relevant to reliability, such as the quality of the developer, the development process, problem complexity, testing effort, and the operation environment [1,4].

In this work, a BBN model was developed to estimate the number of remained defects in a safety-critical software based on the quality evaluation of software development life cycle (SDLC).

2. Safety-Critical Software in NPPs

It is widely recognized that software fails due to defects (including errors made in user requirements, defects introduced during development process and deployment, and erroneous uses of software) residing in the software and the use of the software triggers these defects. Software reliability is thus a function of the manner software is used. Digital protection systems modeled in a PSA may have multiple failure modes. The scope of this work is limited to modeling software failures in performing its protection functions (represented by PDF) at an NPP. That is, the defects/faults considered in the model are those that if triggered would cause a system failure to generate a trip signal.

Presently, there is no consensus method for modeling digital systems in NPP PSAs. The possibility exists that reliability models of digital systems may include software failures representing different software failure modes at different levels of detail (e.g., the software may be modeled at a system, subsystem, or module level). The software system is a collection of software including application, operating system, and platform software implemented in a digital system consisting of multiple microprocessors. Depending on the method of reliability modeling used for digital systems in a PSA, and the associated level of detail, different methods may be needed to quantify the contribution of software failure to the digital system's failure probability or rate. It may also be necessary to separately model different types of software (e.g., application-specific software and operating system software), using different methods.

Many protection systems are designed with identical redundant channels that run the same software. As such, it is expected that these channels would fail together due to common software faults when the same input signals are encountered. Therefore, it is important to quantify the software reliability and to reflect in the PSA model.

2. BBN model for estimating the remained defects in a SW

This work develops a BBN model for estimating the number of faults remaining in a safety-related software program after it is installed and checked out at an NPP.

A BBN is a probabilistic graphical model. The model deals with Bayesian probability, which is a degree of a person's belief in the occurrence of any event based on prior and observed evidence [5]. BBNs have appeared in the literature under several different names: Bayesian Nets (BN), Belief Networks, and Causal Probabilistic Networks. Research on BBNs was initiated in 1970s and applied to the failure diagnosis of artificial intelligence, medical, information technology (IT), and machines in the 1990s. BBNs have been successfully used in non-nuclear applications.

In a typical application of BBN theory, a BBN model first is developed for a class of subjects and then subject-specific evidence is used with the BBN model to draw subject-specific conclusions. The model assumes that the quality of the activities of the software development life cycle, grouped into development and verification and validation (V&V) activities, directly impact software reliability; and the impacts of the two groups of activities can be expressed in terms of the faults that may be inserted into a software during development activities,

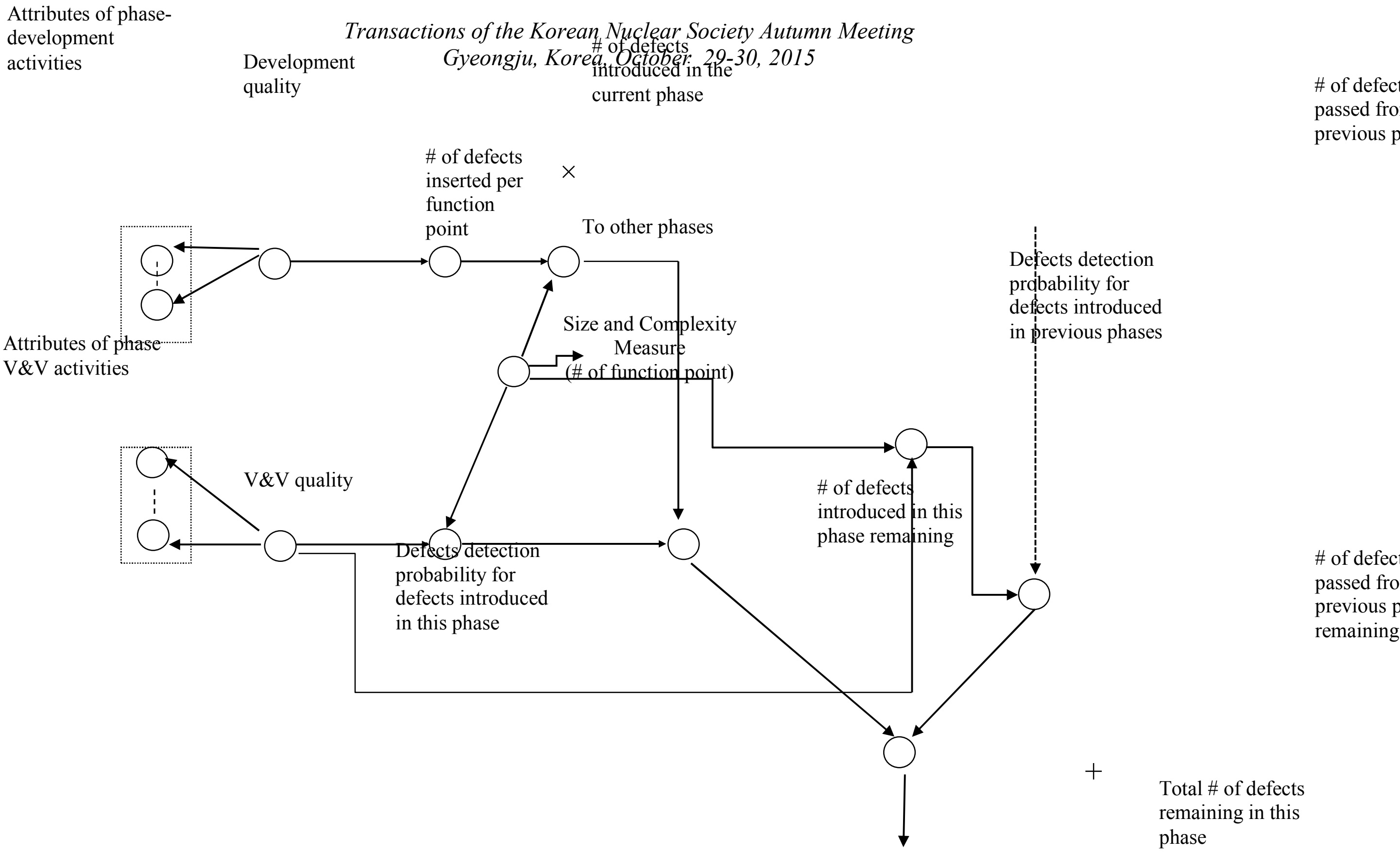
Attributes of phase-development activities
Development quality
# of defects introduced in the current phase
# of defec passed fro previous p
# of defects inserted per function point
×
To other phases
Defects detection probability for defects introduced in previous phases
Size and Complexity Measure (# of function point)
Attributes of phase V&V activities
V&V quality
# of defects introduced in this phase remaining
Defects detection probability for defects introduced in this phase
# of defec passed fro previous p remaining
+
Total # of defects remaining in this phase

Figure 1. Basic process to estimate the defects remained in a phase

and those that can be detected and removed by V&V activities, respectively. The quality in carrying out the activities is assessed by (1) developing the required activities (called attributes) of a safety-related system for each phase of software development, and (2) evaluating the software under study against these attributes. The qualities in carrying out different attributes are aggregated using the BBN model.

In this BBN model, we consider the software development life cycle consisting of five phases: requirements, design, implementation, test, and installation/checkout. For each phase, a BBN model was developed to estimate the number of faults remaining in the software at the end of the phase. Figure 1 shows the basic process to estimate the defects remained in a phase.

4. BBN model development

The BBN model for each phase of the five software development phases has two nodes that represent, respectively, the overall quality of software development, and the V&V. The development team carries out the development activities, while the V&V team undertakes an independent V&V of these activities. Each such node has a few child (attribute) nodes representing the quality in carrying out the required activities associated with this attributes. These required activities were identified by reviewing various guidance and requirement documents.

The main source used to identify these activities is the IEEE standard on V&V, i.e., IEEE Standard 1012 (the 2004 version of which is endorsed by the Regulatory Guide 1.168). Many other guidance and standards were used including IEC 60880, DO-178C, NUREG/CR-6101, and BTP-14. The activities from these standards are used to complement those defined in IEEE 1012 and references to them are provided where they are used. The latest revisions of the regulations and standards are used in developing the attributes and associated activities. Often, the development team carries out the development activities, while the V&V team performs an independent V&V of the same activities. In some cases, Informative information of a standard was used as required activities, and this is specifically pointed out. The Informative information is not a requirement, and alternative means can be used to accomplish the same objective. Additional standards, including ASME NQA-1 and DOE G414, were also reviewed but not included as they were covered by previous standards. As a result of the process described above, the identified required activities are more complete than those of individual guidance or standard.

The node probability tables (NPTs) in the BBN model were developed through expert opinion elicitations. Seven experts who have computer science and experience of NPP software development were chosen to estimate the NPTs.

5. Conclusion

Even though a number of software reliability evaluation methods exist, none of them can be applicable to the safety-critical software in an NPP because software quality in terms of PDF is required for the PSA. In fact, there is a report saying an NPP with digitalized RPS has been experienced only 10 demands for more than 10 years. Therefore, in this work, a method to estimate the number residual defects in a safety-critical software of an NPP based on the SDLC quality evaluation.

REFERENCES

[1] H.S. Eom, G.Y. Park, S.C. Jang, H.S. Son, H.G. Kang, “V&V-based remaining fault estimation model for safety-critical software of a nuclear power plant,” Annals of Nuclear Energy, Vol. 51, pp.38-49, 2013.

[2] Chu, T.L., et al., “Review of Quantitative Software Reliability Methods,” Brookhaven National Laboratory, BNL-94047-2010, September 2010.

[3] Chu, T.L., et al., “Development of Quantitative Software Reliability Models for Digital Protection Systems of Nuclear Power Plants,” U.S. Nuclear Regulatory Commission, NUREG/CR-7044, BNL-NUREG-99068-2013, October 2013.

[4] Fenton N. E. and Neil, M., Risk Assessment and Decision Analysis with Bayesian Network, CRC Press, New York 2012.

[5] Heckerman, D., “A tutorial on learning with Bayesian networks,” Technical Report MSR-TR-95-06, Microsoft Research, Microsoft Corporation, 1995.