Quantification of V&V results for Software Faults Estimation in Bayesian Nets

Heung-Seop Eom [a*], Gee-Yong Park [a], Hyun Gook Kang [a]
[a]*Korea Atomic Energy Research Institute, 1045 Daedeok-daero, Daejeon*
[*]*Corresponding author: ehs@kaeri.re.kr*

1. Introduction

Bayesian Nets (BNs) has been known as one of promising techniques in estimating reliability of safety-critical software [1, 2]. However there are some hard factors to consider in practical application of the BN. The most important one among them is quantification of qualitative data including experts' judgments for modeling. In this paper, we introduce a method which we used in the reliability assessment of the safety-critical software for a reactor protection system. Main quantification in a BN modeling occurs during the construction process of Node probability Tables (NPTs) in BNs. Some rules for the NPT construction are described, and examples of key NPTs are explained. Also a case study was carried out by using the proposed quantification method.

2. SW Fault Estimation in BNs

The BNs in our work was made for estimation of software faults, and was based on the V&V frame which governs the development of safety-critical software in the nuclear field [3]. The structure of the BNs developed for this purpose is depicted in Fig. 1.

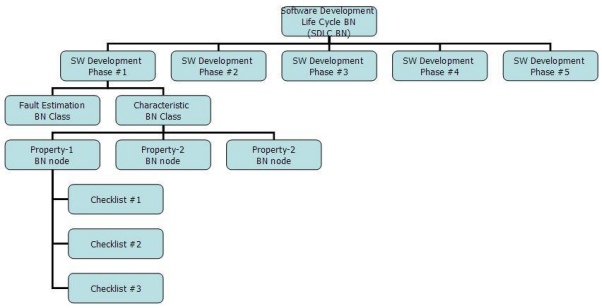

Fig. 1. BN Structure for SW faults estimation

Top level of the BN structure is the Software Development Life-Cycle BN (SDLC BN). This means that SW faults included in the final software product are estimated through the entire SW development phases, such as requirement phase, design phase, etc. Each development phase BN is composed of a "fault estimation" BN (Fig. 2.) and "characteristic" BNs. These BNs are the main part in which the quantification process occurs. The purpose of the "fault estimation" BN is to calculate the number of faults introduced in a software development phase. The "characteristic" BN is derived from the V&V frame of KNICS [4]. Its purpose is to quantify the qualitative results of checklists included in V&V reports. There are 3 instances in the "characteristic" BN class – a process characteristic BN, a function characteristic BN, and a inspection characteristic BN. The "characteristic" BN class consists of properties (nodes in the BN). Properties in the "characteristic" BN class are same as those defined in the V&V reports [4]. The sequence of the BN modeling is, ①compose "characteristic" BN by using "characteristic" BN class, ②compose "fault estimation" BN for each SW development phase by using the "fault estimation" BN class. ③construct all the development-phases BNs by using the previously composed "characteristic" BNs and a "faults estimation" BN, ④finally, connect the development-phase BNs according to the development sequence, which leads to the SDLC BN.

The SDLC BN is the final model that can estimate faults number of the target software (program). Input of the BN model is the value of properties which are calculated from the checklist in the V&V report [4].

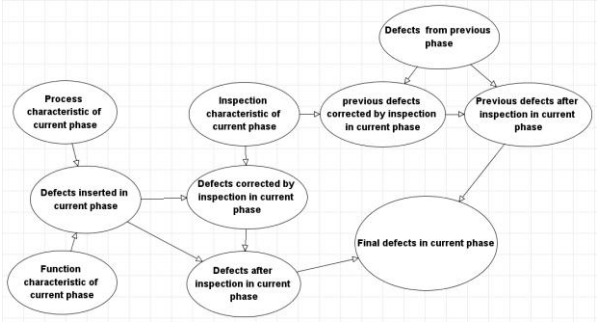

Fig. 2. BN graph for software fault estimation

3. Quantification of V&V Results

The quantification process in a BN mostly occurs in NPTs construction. The rules and methods for the NPT construction in our BN are based on the experts' knowledge and studies appeared in the BNs for software quality evaluation [2].

3.1 Some Quantification Rule for NPT

The preliminary document in a software development is the system specification. We generally assume that there are not faults in the system specification, but we also know that there is uncertainty in the assumption. We explicitly quantified this uncertainty by using normal distribution" type NPT, instead of conventional table-type NPT.

The range of fault numbers which might be introduced in software development phases (requirement, design, implement, and integration phase, etc.) is hard to estimate, especially for the safety-critical software. To estimate the range, we averaged the highest frequency of anomaly reports (ANR) cases issued in development phases and V&V experts' estimation for the maximum fault number.

There are always uncertainties in converting the qualitative evaluation data of checklists in a V&V report into quantitative probabilities for the BNs. To handle these uncertainties "normal distribution" type NPT was adopted for nodes belong to "characteristic" BN class. "Normal distribution" type NPT has a mean value and a variance value. By utilizing this variance, uncertainty of data can be explicitly handled. For example, if a checklist's evaluation data has high uncertainty, we can set the variance of the corresponding NPT high.

3.2 NPT for Characteristic BN Class

All the nodes in the characteristic BN class have 5 states (Very-Low, Low, Medium, High, Very-High). They also have 2 types of NPT structure. One is table type for parent nodes, and another is "normal distribution' type for child nodes. Table I is the table type NPT structure. The "normal distribution' type uses discrete normal distribution structure in the NPT which has a mean value and a variance.

Table I: NPT of parent a node in the characteristic BN class

Very Low	Low	Medium	High	Very High
0.2	0.2	0.2	0.2	0.2

3.3 NPT for Fault Estimation BN class

There are 4 types of NPT in the "fault estimation" BN class. They are arithmetic, "normal distribution", table, and "binomial distribution" type which has parameters such as "probability of success" and "number of trials." The structures of "normal distribution' type and table type are similar to those in the "characteristic" BN class. The arithmetic type is for arithmetic calculation among related nodes, and "binomial distribution" type is for calculation of eliminated faults number by inspection activities which is expressed as "inspection characteristic" node in BNs.

4. Case Study: Reactor Protection System Software

Evaluation results in the V&V reports were all qualitative form such as "good, bad, normal, or abnormal." These qualitative evaluation data were converted into real number (probability) between 0~1 by V&V experts. For example, 1 is excellent, 0.5 is medium, and 0 is very poor. Then these numbers were mapped to 5 status of a NPT as shown in Table II. The input data used in our BNs were not from final the V&V report, but from the progress report in the middle of the spiral-type development life-cycle. Thus the calculation results appeared in this paper do not mean the current status of programs of the case study.

Table II: Mapping of a checklist's value to a node' value

Values of a checklist	Values (states) of a node
1 ~ 0.95 over	Very High
0.95 less ~ 0.90 over	High
0.90 less ~ 0.85 over	Medium
0.85 less ~ 0.80 over	Low
0.80 less ~	Very Low

Table III shows fault numbers estimated by ANR, and the calculation result of the BN. The probability of finding faults in the previous development phase was set to 0.5. This means that perfect inspection activity can find and fix 50% of faults left in the previous development phase.

Table III: Estimated faults from ANR and BN model

ANR cases/evaluation points/estimated number	Calculated number by the BN
4/2.5/3.75	3.69

5. Summary and future works

The insight from the case study is that the most critical and hardest factors affecting the fault number of the target software are the faults in the system specification, maximum number of faults introduced in a development phase, ratio between process/function characteristic and faults number introduced in a development phase, uncertainty sizing (variance size in the NPT), and faults elimination rate by inspection activities. There are not yet sufficient experimental or analytical data related to these factors, especially for safety-critical software such as a reactor protection system. Though the BN is the most promising technique for the reliability assessment of safety-critical software, acquiring these data is very important for BN based reliability assessment methods to be a practical and credible methodology in the nuclear field.

REFERENCES

[1] Validation of Ultrahigh Dependability for Software-Based Systems, B. Littlewood, Communication of the ACM

[2] Fenton NE, Neil M, Hearty P, Marsh W, Marquez D, Krause P, Mishra R, "Predicting Software Defects in Varying Development Lifecycles using Bayesian Nets", Information & Software Technology, Vol. 49, pp 32-43. (2007)

[3] H.S. Eom, S.C. Jang, General Method of Using Bayesian Nets for a Software Reliability Assessment in Varying SW Development Lifecycle, KNS, 2008.5

[4] G. Y. Park and K. C. Kwon, Software Verification & Validation for Digital Reactor Protection System, Information and Control Symposium, pp. 190-192, (2005).