Transactions of the Korean Nuclear Society Autumn Meeting
Busan, Korea, October 27-28, 2005

A Study on the Quantitative Assessment Method of Software Requirement Documents Using Software Engineering Measures and Bayesian Belief Networks

Heung-Seop Eom, Hyun-Gook Kang, Ki Hong Park, Kee Choon Kwon, Seung-Cheol Chang
Korea Atomic Energy Research Institute, ISA Div., P.O.Box 105, Yuseong Daejeon, ehs@kaeri.re.kr

1. Introduction

One of the major challenges in using the digital systems in a NPP is the reliability estimation of safety critical software embedded in the digital safety systems. Precise quantitative assessment of the reliability of safety critical software is nearly impossible, since many of the aspects to be considered are of qualitative nature and not directly measurable, but they have to be estimated for a practical use. Therefore an expert's judgment plays an important role in estimating the reliability of the software embedded in safety-critical systems in practice, because they can deal with all the diverse evidence relevant to the reliability and can perform an inference based on the evidence. But, in general, the experts' way of combining the diverse evidence and performing an inference is usually informal and qualitative, which is hard to discuss and will eventually lead to a debate about the conclusion.

We have been carrying out research on a quantitative assessment of the reliability of safety critical software using Bayesian Belief Networks (BBN). BBN has been proven to be a useful modeling formalism because a user can represent a complex set of events and relationships in a fashion that can easily be interpreted by others. In the previous works [1] we have assessed a software requirement specification of a reactor protection system by using our BBN-based assessment model. The BBN model mainly employed an expert's subjective probabilities as inputs. In the process of assessing the software requirement documents we found out that the BBN model was excessively dependent on experts' subjective judgments in a large part. Therefore, to overcome the weakness of our methodology we employed conventional software engineering measures into the BBN model as shown in this paper. The quantitative relationship between the conventional software measures and the reliability of software were not identified well in the past. Then recently there appeared a few researches on a ranking of some important software engineering measures with respect to their capability at predicting software reliability [2,3] and we could utilize the results of the study in improving our methodology.

2. Bayesian Belief Networks (BBN)

BBN is a network-based formalism for representing and analyzing models involving an uncertainty. Nowadays a number of efficient tools for a BBN modeling are available and BBN has become an expanding technology in many areas such as medical, military, financial, and the safety/reliability analysis of complicated systems such as digital systems. BBN consists of the following [4].

- A set of variables and a set of directed edges (arcs) between variables
- Each variable has a finite set of mutually exclusive states
- The variables together with the directed edges form a directed acyclic graph (DAC). A DAC is acyclic if there is no directed path $A_1 \rightarrow \ldots A_n$ such that $A_1 = A_n$
- To each variable A with parents $B_1 \ldots B_n$ there is attached a conditional probability table $P(A|B_1 \ldots B_n)$.

There are several things to consider when we use BBN as the modeling tool: [5]

- Computations on full joint distribution are not feasible for large problems
- Conditional independence assumptions reduce combinational explosion
- It is easier to elicit conditional probability tables from an expert than joint probabilities
- The causal structure of a BBN is easier to understand than mathematics
- Fast algorithms (junction tree based algorithm, cut-set conditioning method based) are available to compile and execute BBN
- Evidence is propagated throughout BBN by exploiting Bayes' rule
- Forecasts can be done with incomplete evidence

3. BBN-based assessment methodology for software requirement documents

In the previous works [1] we constructed a BBN model which can evaluate the software requirement specification (SRS) of a reactor protection system. The purpose of the work is to assess the reliability of the SRS in a quantitative way. The basic documents which were used to develop the model are (i) Procedures for V&V [6], (ii) V&V report [7], and (iii) Software Development Plan (SDP) [8].

The proposed method relies on BBN to combine all the variables relevant to the reliability of SRS, and to propagate consistently the impact of these variables on

the probabilities of the uncertain outcomes (in this example, the reliability of SRS). The variables in our model were mostly identified from V&V procedures and SDP. A summary of the variables is:

- 14 properties of software requirement specification - Accuracy, functionality, reliability, robustness, safety, security, timing, completeness, consistency, correctness, style, traceability, unambiguity, verifiability.
- Questions belonging to the above 14 properties: Each property has a checklist and the checklist has several questions.
- The Quality of the development process, V&V process, and the complexity

A total of 160 nodes and their node probability table (NPT) were developed. All the nodes converted from the checklists have two states ("yes" and "no"). The probabilities of the NPTs in the model were assigned by a V&V expert's judgment.

An argument emerged from the previous work of assessing the SRS using the BBN model work. It was excessively dependent on experts' subjective judgments in a large part. All the probabilities of NPTs and inputs to the model were decided by experts' subjective judgments. Actually this point is strength and weakness in any kind of BBN-based model. One of the methods to supplement this problem is to use objective measures in the model. The quantitative relationship between the conventional software measures and the reliability of software were not identified well in the past. Then recently there appeared a few researches on a ranking of some important software engineering measures with respect to their capability at predicting software reliability [2,3] and we utilized the results of the study in improving our methodology. Tab1e 1 lists some important software engineering measures which we considered to improve our previous BBN model, and their relevance to the reliability of a software [2].

Table 1. Software measures and its relevance to the reliability of software (Requirement Phase)

Measures	Measure's relevance to reliability
Cause & effect graphing	0.45
Completeness	0.43
Error distribution	0.68
Fault density	0.71
Fault-days number	0.60
Feature point analysis	0.46
Function point analysis	0.51
Number of faults remaining	0.46
Requirement compliance	0.50
Requirement spec. change requests	0.70
Reviews, inspections and walkthroughs	0.61
Software capability maturity model	0.60

The assumptions of new our model are that the reliability of the software product can be measured by considering (i) the properties related to the software quality, (ii) the process used to develop it, (iii) software engineering measures. The processes can also be measured by considering their properties and engineering measures. Figure 1 shows the conceptual BBN model based on these assumptions.

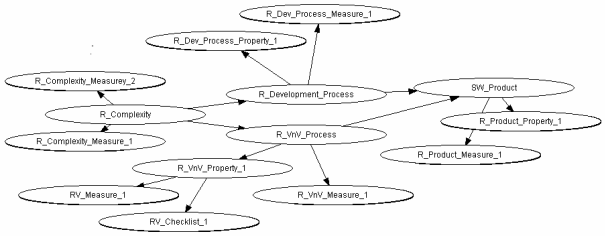

Figure 1. Conceptual BBN model for SRS assessment

4. Summary

To complement an argument which emerged from the BBN-based assessment of SRS we have surveyed some important conventional software engineering measures and employed some of them into our assessment model, which will lead to improving the objectivity of our methodology. The improved BBN-based methodology will be applied to the assessment of the reliability of the software design documents of a digitalized reactor protection system which is under development in KNICS project.

REFERENCES

[1] Heung-Seop, Eom., et al, 2004. Study on the Quantitative Reliability Estimation of Safety-Critical Software for PSA, NIPC & HMIT 2004

[2] Ming, L., et al, A Ranking of Software Engineering Measures Based on Expert Opinion, IEEE Transaction on Software Engineering, Vol. 29, N0.9, 2003.

[3] Smidts, C., et at, Preliminary Validation of a Methodology for Assessing Software Quality, NUREG/CR-6848, USNRC, 2004

[4] Jenson, F., An Introduction to Bayesian Belief Networks, Springer Verlag, New York, 1996.

[5] Yangyang, Y., et al, A BBN Approach to Certifying the Reliability of VOTS Software Systems, Reliability and Maintainability Symposium, 2003.

[6] HanSeong, S., et al, 2003. V&V Procedure for Software Requirement Specification for Reactor Protection System, KNICS-RPS-(SRS)-SVP121, KAERI KNICS, 2003.

[7] HanSeong, S., 2004. V&V Validation Reports for Software Requirement Specification for Reactor Protection System, KNICS-RPS-(SRS)-SVR121, KAERI KNICS, 2004.

[8] Du-Hwan, K., et al, 2001. Software Development Plan for Engineering Safety Features, KNICS-ESF-SDP101, KAERI KNICS, 2001.