2002年日本オペレーションズ・リサーチ学会
春季研究発表会

On Relationship Between Software Safety Measurement and the Number of Debuggings

01307475 鳥取大学 *得能貢一 TOKUNO Koichi
01702425 鳥取大学 山田茂 YAMADA Shigeru

1 Introduction

This paper discusses a stochastic modeling for software safety/reliability measurement based on the number of debuggings, and provides a metrics of *software safety* [1] defined as the probability that the software-intensive system does not fall into any unsafe states as the function of the time and the number of debuggings. The unsafe state of the software system is defined as the state of the computer-intensive system that induces the hazards such as the injuries to human lives, illness, and serious financial losses due to the software's undesirable processes [2]. The attention of this paper is directed to the event that the system causes unsafe states randomly in operation. Furthermore, we consider that some of debuggings which primarily perform for software reliability growth contribute to software safety improvement as well in software safety modeling.

2 Model description

We give the following assumptions to construct the software safety/reliability model:

A1. When the software system operates safely, it falls into the unsafe state randomly, i.e., the holding time of a safe state in operation follows the exponential distribution with mean $1/\theta_n$, where $n = 0, 1, 2, \ldots$ denotes the cumulative number of corrected faults. We call θ_n the software unsafety rate. The holding time of an unsafe state is also random and follows the exponential distribution with mean $1/\eta$.

A2. A debugging activity is performed when a software failure occurs. A debugging activity is perfect with the perfect debugging rate a $(0 \leq a \leq 1)$, on the other hand, imperfect with probability $b(= 1 - a)$. One perfect debugging corrects and removes one fault from the system. Debugging activities are performed in safe states and the debugging time is not considered.

A3. Software reliability growth occurs in the case of the perfect debugging. The time-interval between software failure-occurrences follows the exponential distribution with mean $1/\lambda_n$. λ_n is a decreasing function of n.

A4. A perfect debugging activity improves software safety as well with probability p $(0 \leq p \leq 1)$, on the other hand, does not improve with probability $q(= 1 - p)$. Imperfect debugging activities have no impact on software safety.

We consider a stochastic process $\{X(t),\ t \geq 0\}$ representing the state of the software system at the time point t; this state space is $(\mathcal{W},\ \mathcal{U})$, where safe state vector $\mathcal{W} = \{W_n;\ n = 0,\ 1,\ 2,\ \ldots\}$ and unsafe state vector $\mathcal{U} = \{U_n;\ n = 0,\ 1,\ 2,\ \ldots\}$. The events $\{X(t) = W_n\}$ and $\{X(t) = U_n\}$ mean that the system is operating safely and falls into the unsafe state at the time point t, when n faults have already been corrected, respectively.

We refer to the description of the software unsafety rate θ_n. Let I_n be the binomial random variable representing the number of perfect debugging activities contributing to software safety improvement as well out of n perfect debugging activities. For describing a software safety improvement process, we assume that the software unsafety rate when $\{I_n = i\}$ can be describe as

$$\theta(i) = \alpha\beta^i$$
$$(i = 0, 1, 2, \ldots, n;\ \alpha > 0,\ 0 < \beta \leq 1)\,, \qquad (1)$$

where α and β denote the initial software unsafety rate and the decreasing ratio of the software unsafety rate, respectively. Accordingly, considering the expectation of $\theta(i)$, we give θ_n as

$$\theta_n \equiv \mathrm{E}[\theta(I_n)] = \alpha(p\beta + q)^n, \qquad (2)$$

Figure 1 illustrates the sample state transition diagram of $X(t)$.

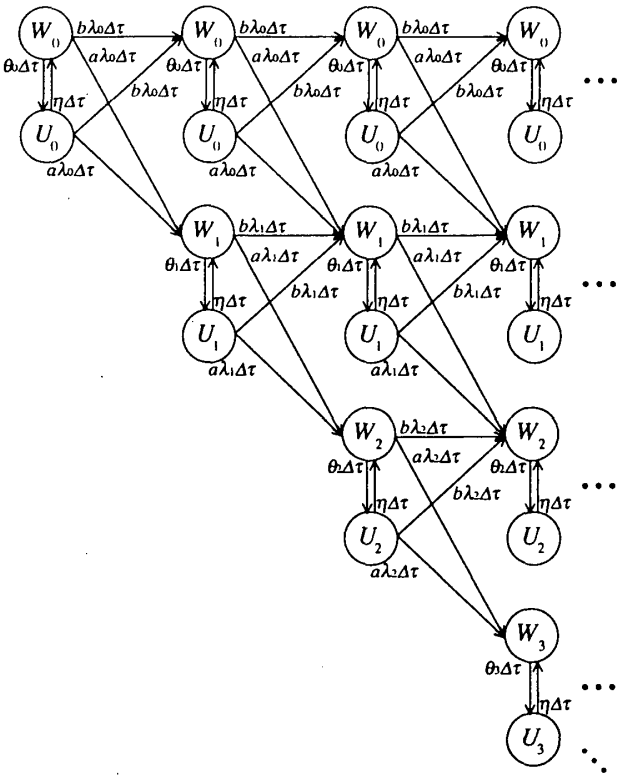

Fig.1 A sample state transition diagram of $X(t)$.

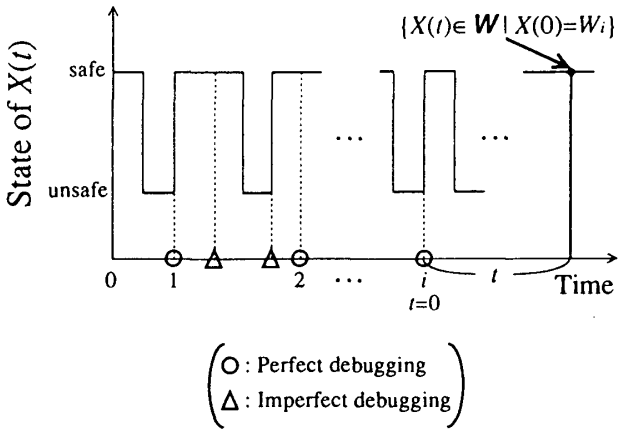

Fig.2 Sample behavior of the system and event $\{X(t) \in \boldsymbol{W} | X(0) = W_i\}$.

3 Derivation of metrics of software safety

We consider the relationship between the number of debuggings and software safety measurement. Let $l = 0, 1, 2, \ldots$ denote the number of debuggings. Furthermore,

$$S_i(t) \equiv \Pr\{X(t) \in \boldsymbol{W} | X(0) = W_i\} = \sum_{n=i}^{\infty} P_{W_i,W_n}(t), \quad (3)$$

denote the probability that the system is in a safe state at the time point t, given that it was in state W_i at time point $t = 0$ (see Fig.2), where $P_{A,B}(t) \equiv \Pr\{X(t) = B | X(0) = A\}$ $(A, B \in (\boldsymbol{W}, \boldsymbol{R}))$ is the state occupancy probability that the system is in state B at the time point t on the condition that the system was in state A at time point $t = 0$. $P_{W_i,W_n}(t)$'s can be obtained analytically.

It is noted that the cumulative number of corrected faults or perfect debuggings at the completion of the l-th debugging, C_l, is not observed explicitly and immediately since we assume imperfect debugging environment and that C_l follows the binomial distribution from assumption A2. Accordingly, the metrics of software safety after the completion of the l-th debugging is given by

$$S(t; l) = \sum_{i=0}^{l} \binom{l}{i} a^i b^{l-i} S_i(t), \quad (4)$$

which represents the probability that the system does not fall into any unsafe states at the time point t, given that the l-th debugging was complete at time point $t = 0$.

Acknowledgments

This work was supported in part by the Research Grant from the Telecommunications Advancement Foundation, the Sasakawa Scientific Research Grant from The Japan Science Society, and Grants-in-Aid from the Ministry of Education, Culture, Sports, Science and Technology of Japan under Grant Nos. 12680442 and 13780365, respectively.

References

[1] K. Tokuno and S. Yamada, "Markovian reliability modeling for software safety/availability measurement," in Recent Advances in Reliability and Quality Engineering (H. Pham ed.), pp. 181–201, World Scientific, Singapore, 2001.

[2] D. S. Herrmann, Software Safety and Reliability —Techniques, Approaches, and Standards of Key Industrial Sectors, IEEE Computer Society Press, Los Alamitos, CA, 1999.