2000年度日本オペレーションズ・リサーチ学会
秋季研究発表会

A Software Reliability Assessment Model Based on Stochastic Differential Equations of Itô Type

02101865 鳥取大学 *木村光宏 KIMURA Mitsuhiro
01702425 鳥取大学 山田茂 YAMADA Shigeru

1 Introduction

In this study, one of our interest is to introduce the nonlinear state dependency of software debugging processes directly into the software reliability assessment model, not as a mean behavior like software reliability growth models based on an NHPP (nonhomogeneous Poisson Process). By focusing on software debugging speed, we show the importance of introducing the nonlinear behavior of the debugging speed into the software reliability modeling. The modeling needs the mathematical theory of stochastic differential equations of Itô type[1]. We derive several software reliability assessment measures based on our new model.

2 Model description

Let $M(t)$ be the number of software faults remaining in the software system at testing time t ($t \geq 0$). We consider a stochastic process $\{M(t), t \geq 0\}$, and assume $M(t)$ takes on continuous real value. In the past studies, $\{M(t), t \geq 0\}$ has been usually modeled as a counting process for software reliability assessment modeling[2]. However, we can suppose that $M(t)$ is continuous when the size of software system in the testing phase is sufficiently large[3]. The process $\{M(t),\ t \geq 0\}$ may start from a fixed value and gradually decrease with some fluctuation as the testing phase goes on. Thus we assume that $M(t)$ holds the basic equation as follows:

$$\frac{dM(t)}{dt} = -b(t)g(M(t)), \tag{1}$$

where $b(t)$ (> 0) represents a fault-detection rate per unit time, and $M(0)$ ($= m_0$ (const.)) is the number of inherent faults at the beginning of the testing phase. $g(x)$ represents a nonlinear function which is required several conditions as follows:

- $g(x)$ is non-negative and has Lipschitz continuity.
- $g(0)$=0.

Equation (1) means that the fault-removal rate at testing time t, $dM(t)/dt$, is defined as a function of the number of faults remaining at testing time t, $M(t)$. $dM(t)/dt$ decreases gradually as the testing phase goes on. This supposition has been often made in the software reliability growth modeling.

In this study, we suppose that $b(t)$ in Eq. (1) has an irregular fluctuation, i.e., we expand Eq. (1) to the following stochastic differential equation[4]:

$$\frac{dM(t)}{dt} = -\{b(t) + \xi(t)\}g(M(t)), \tag{2}$$

where $\xi(t)$ represents a noise that denotes the irregular fluctuation. Further, $\xi(t)$ is defined as:

$$\xi(t) = \sigma\gamma(t), \tag{3}$$

where $\gamma(t)$ is a standard Gaussian white noise and σ is a positive constant which means a magnitude of the irregular fluctuation. Hence we rewrite Eq. (2) as:

$$\frac{dM(t)}{dt} = -\{b(t) + \sigma\gamma(t)\}g(M(t)). \tag{4}$$

By using Wong-Zakai transformation, we obtain the following stochastic differential equation of Itô type.

$$\begin{aligned} dM(t) &= \{-b(t)g(M(t)) + \frac{1}{2}\sigma^2 g(M(t))g'(M(t))\}dt \\ &- \sigma g(M(t))dW(t), \end{aligned} \tag{5}$$

where the so-called growth condition[1] is assumed to be satisfied with respect to the function $g(x)$.

In Eq. (5), $W(t)$ represents a one-dimensional Wiener process[1], which is formally defined as an integration of the white noise $\gamma(t)$ with respect to testing time t. The Wiener process $\{W(t),\ t \geq 0\}$ is a Gaussian process, and has the following properties:

- $\Pr[W(0) = 0] = 1$,
- $\mathrm{E}[W(t)] = 0$,
- $\mathrm{E}[W(t)W(\tau)] = \min[t, \tau]$.

Under these assumptions and conditions, we can obtain a transition probability distribution function of the solution process by solving the Fokker-Planck equation[1, 4] derived from Eq. (5) as follows.

The transition probability distribution function $P(m, t|m_0)$ is obtained as:

$$\begin{aligned} P(m, t|m_0) &= \Pr[M(t) \leq m | M(0) = m_0] \\ &= \Phi\left(\frac{\int_{m_0}^{m} \frac{dm'}{g(m')} + \int_0^t b(t')dt'}{\sigma\sqrt{t}}\right), \end{aligned} \tag{6}$$

where the function $\Phi(z)$ denotes the standard normal distribution function defined as

$$\Phi(z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{z} \exp[-\frac{s^2}{2}]ds. \tag{7}$$

Let $N(t)$ be the total number of detected faults up to testing time t. Since the condition $M(t) + N(t) = m_0$ holds with probability one, we have the transition probability distribution function of the process $N(t)$ as follows:

$$\begin{aligned} P(n, t|0) &= \Pr[N(t) \leq n | N(0) = 0, M(0) = m_0] \\ &= \Phi\left(\frac{\int_0^n \frac{dn'}{g(m_0 - n')} - \int_0^t b(t')dt'}{\sigma\sqrt{t}}\right) \quad (n < m_0). \end{aligned} \tag{8}$$

3 Software Debugging Speed

In order to apply the model constructed in the former section to the software fault-detection data, we should determine the function $g(x)$ previously. First we investigate two data sets (denoted as DS1 and DS2) which have been collected in the actual testing phase of software development, to illustrate the behavior of their debugging speed. Each data set forms (t_j, n_j) $(j = 1, 2, \ldots, K;\ 0 < t_1 < t_2 < \cdots < t_K)$, where t_j represents testing time and n_j is the total number of detected faults up to the testing time t_j[5]. By evaluating numerically, we obtain the debugging speed $dN(t)/dt$ from these data sets. Figures 1 and 2 illustrate the behavior of the respective debugging speed versus the cumulative number of detected faults $N(t)$. Figure 1 shows a linear relation between the debugging speed and the degree of fault-detection approximately. Therefore we can give $g(x)$ as:

$$g(x) = x, \tag{9}$$

for such a data set. On the other hand, Fig. 2 illustrates there exists a nonlinear relation between them. Thus we expand Eq. (9) and assume the function as follows:

$$g(x) = \begin{cases} x^r & (0 \le x \le x_c,\ r > 1) \\ r{x_c}^{r-1}x + (1-r){x_c}^r & (x > x_c,\ r > 1) \end{cases}, \tag{10}$$

which is a power function that is partially revised in $x > x_c$ so as to satisfy the growth condition. In Eq. (10), r denotes a shape parameter and x_c is a positive constant which should be needed to ensure the existence of the solution process $M(t)$ in Eq. (5). The parameter x_c can be given arbitrarily. We assume $x_c = m_0$ in this study.

4 Software Reliability Measures

4.1 Expected number of remaining faults and its variance

In order to assess software reliability quantitatively, information on the current number of remaining faults in the software system is useful to estimate the situation of the progress on the software testing phase.

The expected number of remaining faults and its variance can be respectively evaluated by:

$$\mathrm{E}[M(t)] = \int_0^\infty m\, d\Pr[M(t) \le m | M(0) = m_0], \tag{11}$$

$$\mathrm{Var}[M(t)] = \mathrm{E}[M(t)^2] - \mathrm{E}[M(t)]^2. \tag{12}$$

If we assume $g(x) = x$ as a special case, we have

$$\mathrm{E}[M(t)] = m_0 \exp[-(b - \frac{1}{2}\sigma^2)t], \tag{13}$$

$$\mathrm{Var}[M(t)] = {m_0}^2 \exp[-(2b - \sigma^2)t]\{\exp[\sigma^2 t] - 1\}. \tag{14}$$

4.2 Cumulative MTBF

In the fault-detection process $\{N(t), t \ge 0\}$, average fault-detection time-interval per one fault up to time t is denoted by $t/N(t)$. Hence the cumulative MTBF, $MTBF_c(t)$, is approximately given by:

$$MTBF_c(t) = \mathrm{E}[\frac{t}{N(t)}] = \mathrm{E}[\frac{t}{m_0 - M(t)}] \approx \frac{t}{m_0 - \mathrm{E}[M(t)]}. \tag{15}$$

Acknowledgment

The authors wish to thank Professor Shunji Osaki of Hiroshima University, Dr. Hiroaki Tanaka of Kyoto University, and Mr. Hiroshi Ohnishi for their useful suggestions and accomplishment of the fundamental work of this study.

References

[1] L. Arnold, *Stochastic Differential Equations: Theory and Applications*, John-Wiley & Sons (1974).

[2] S. Yamada, "Software quality/reliability measurement and assessment: Software reliability growth models and data analysis", *J. Information Processing*, Vol. 14 (1991).

[3] S. Yamada, M. Kimura, H. Tanaka, and S. Osaki, "Software Reliability Measurement and Assessment with Stochastic Differential Equations", *IEICE Trans. Fundamentals*, Vol. E77-A, No. 1, pp. 109–116 (1994).

[4] S. Karlin and H. M. Taylor, *A Second Course in Stochastic Processes*, Academic Press (1981).

[5] S. Yamada, H. Ohtera, and H. Narihisa, "Software reliability growth models with testing-effort", *IEEE Trans. Reliability*, Vol. R-35, No. 1, pp. 19-23 (1986).

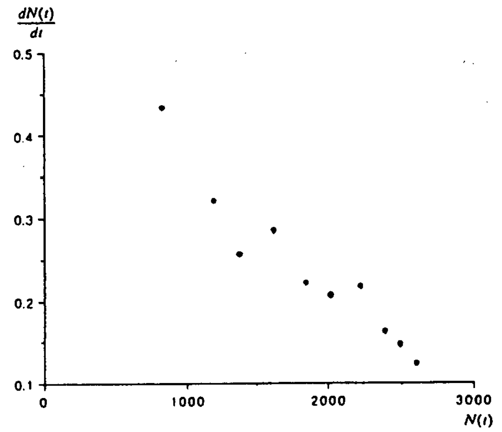

Figure 1: Software debugging speed (DS1).

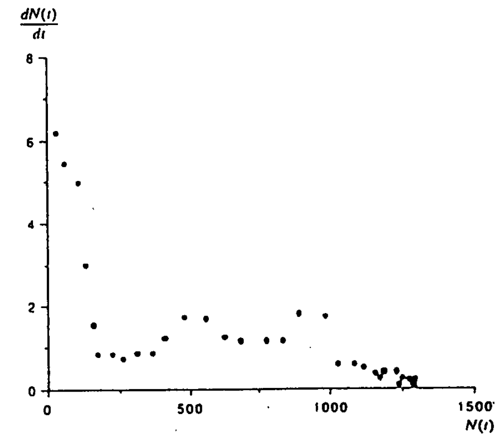

Figure 2: Software debugging speed (DS2).