Communications of the Korean Statistical Society Vol. 15, No. 6, 2008, pp. 875–881

A General Coverage-Based NHPP SRGM Framework

Joong-Yang Park[1), Gyemin Lee[2), Jae Heung Park[3)

Abstract

This paper first discusses the existing non-homogeneous Poisson process(NHPP) software reliability growth model(SRGM) frameworks with respect to capability of representing software reliability growth phenomenon. As an enhancement of representational capability a new general coverage-based NHPP SRGM framework is developed. Issues associated with application of the new framework are then considered.

Keywords: Coverage; coverage growth function; differential equation; mean value function; non-homogeneous Poisson process; software reliability growth model.

1. Introduction

Software systems are becoming important components of computer systems. Since software system failures caused by faults latent in a software system result in failures of a computer system, reliability of a computer system is dependent on the reliability of its software system. Software reliability is therefore a concern of software developers and users. Testing is a key activity for detecting and removing faults and consequently improving reliability of a software system. Reliability growth phenomena are observed while testing software systems. Since all the latent faults can not be removed within a reasonable testing time and developers should determine when to stop testing, many software reliability growth models(SRGMs) have been developed to express the reliability growth phenomena statistically.

One of successful classes of SRGMs is the class of non-homogeneous Poisson process(NHPP) SRGMs. NHPP SRGMs represent a reliability growth phenomenon by the cumulative number of software faults $\{N(t), t \geq 0\}$ discovered up to testing time t and assume that $N(t)$ follows a Poisson distribution with mean value function(MVF) $m(t)$. Therefore each NHPP SRGM is characterized by its MVF $m(t)$. Various MVFs, consequently various NHPP SRGMs, have been devised from different assumptions on the testing and debugging scenarios. Recently Pham and Zhang (1997) and Pham *et al.*

1) Professor, Department of Information and Statistics, College of Natural Science, RINS and RICI, Gyeosangsang National University, 900 Gazwa-dong, Jinju 660-701, Korea.
Email: joongyang@gnu.ac.kr

2) Associate Professor, Department of Information and Statistics, College of Natural Science, RINS and RICI, Gyeosangsang National University, 900 Gazwa-dong, Jinju 660-701, Korea.
Correspondence: gyemin@gnu.ac.kr

3) Professor, Department of Computer Science, College of Natural Science, RINS and RICI, Gyeosangsang National University, 900 Gazwa-dong, Jinju 660-701, Korea. Email: jhp@gnu.ac.kr

(1999) suggested a generalized differential equation governing MVF, which is given as

$$\frac{dm(t)}{dt} = b(t)\,[a(t) - m(t)]\,. \tag{1.1}$$

Here, $b(t)$ is the fault detection rate function and $a(t)$ is the fault content function incorporating initial faults and faults introduced by imperfect debugging. Solution of Eq. (1.1) subject to initial condition $m(t) = 0$ is obtained as

$$m(t) = e^{-B(t)} \int_0^t a(x)\,b(x)\,e^{B(x)}\,dx, \tag{1.2}$$

where $B(t) = \int_0^t b(x)\,dx$. Since Eq. (1.2) is able to produce various MVFs for various pairs of $a(t)$ and $b(t)$, differential equation Eq. (1.1) or its solution Eq. (1.2) is referred to as an NHPP SRGM framework. It was shown by Pham and Zhang (1997) and Pham *et al.* (1999) that MVFs of most of the currently available NHPP SRGMs were special cases of Eq. (1.2).

A software system can be considered as a collection of constructs, where a construct is a basic building element of a software system. Some usual contructs are statements, blocks, branches, c-uses and p-uses. Suppose now that additional testing is performed by running some test cases during $[t,\, t + dt)$. Generally the additional testing does not execute all the constructs, but only a portion of them. In such a situation only the faults latent in the constructs executed(covered) by the additional testing are exposed to fault detection activity, *i.e.*, detectable. However, the NHPP SRGM framework given by Eq. (1.1) assumes that all the faults remaining in the software system at time t are detectable during $[t,\, t + dt)$. This suggests 1) that software reliability growth phenomenon depends on construct coverage phenomenon; 2) that differential equation Eq. (1.1) does not fully reflect the reliability growth phenomenon observed in the testing phase.

This aims to resolve the above mentioned shortcomings by integrating both the proportion of contructs re-executed and the proportion of constructs newly executed by the additional testing into the framework. Since this approach requires representation of construct coverage phenomenon, Section 2 briefly reviews the coverage growth function(CGF) and the existing frameworks involving CGF. A new general NHPP SRGM framework is developed and presented in Section 3. Section 4 shows that MVFs of some existing NHPP SRGMs can be obtained from the new general NHPP SRGM framework. Issues associated with application of the proposed framework are also considered. Finally, conclusions will be given in Section 5.

2. Coverage-Based NHPP SRGM Frameworks

CGF, denoted by $\pi(t)$, is defined as the expected proportion of constructs executed (covered) up to testing time t. Unlikely to MVF, CGF has not been investigated theoretically and empirically until recent researches such as Gokhale and Mullen (2004, 2005) and Park *et al.* (2007, 2008). There are four NHPP SRGM frameworks involving CGF, which are usually called the coverage-based NHPP SRGMs. Their corresponding differential equations are given as follows:

$$\frac{dm(t)}{dt} = a\,\frac{d\pi(t)}{dt}, \tag{2.1}$$

$$\frac{dm(t)}{dt} = b(t)\,a\,\frac{d\pi(t)}{dt}, \tag{2.2}$$

$$\frac{dm(t)}{dt} = [a(t) - m(t)]\,\frac{d\pi(t)/dt}{1-\pi(t)}, \tag{2.3}$$

$$\frac{dm(t)}{dt} = b\,[a - m(t)]\,\frac{d\pi(t)}{dt}, \tag{2.4}$$

where a and b, constant versions of $a(t)$ and $b(t)$, are respectively the initial fault content and the fault detection rate. Frameworks given by Eqs. (2.1)–(2.4) were repectively proposed by Piwowarski *et al.* (1993), Gokhale *et al.* (1996), Pham and Zhang (2003) and Yamamoto *et al.* (2004). In fact, Pham and Zhang (2003) considered $d\pi(t)/dt\,[1-\pi(t)]^{-1}$ in Eq. (2.3) as the fault detection rate function. Since $d\pi(t)$, the increment of $\pi(t)$ for $[t,\, t+dt)$, is the proportion of constructs newly executed for $[t,\, t+dt)$ and only $d\pi(t)/dt$ is involved in the above frameworks, these frameworks commonly assume that only the faults in the newly executed constructs are detectable. They do not consider the faults remaining in the constructs already covered up to t.

The additional testing for $[t, t+dt)$ excutes some of the constructs not executed up to t. The expected proportion of the newly covered constructs relative to the total constructs is $d\pi(t)$. In general, the additional testing re-executes some of the covered constructs, from which faults can be detected. Frameworks given by Eqs. (2.1)–(2.4) ignore the possible detection of faults from the re-executed constructs. Realistic frameworks should take into account both the faults detected from the newly covered constructs and the faults detected from the re-executed constructs. Such a framework will be derived in the next section.

3. A General Coverage-Based NHPP SRGM Framework

Most NHPP SRGMs assume that the faults are initially distributed uniformly over the entire software. This assumption is now rephrased for the coverage-based NHPP SRGMs as that the faults are initially distributed uniformly over all the constructs. This assumption is hereafter called Assumption 1. The coverage-based NHPP SRGM frameworks discussed in the previous section are based on Assumption 1. We now introduce another assumption, hereafter referred to as Assumption 2, about the faults remaining in the executed constructs. Assumption 2 is that the faults remaining in the executed constructs are distributed uniformly over the executed constructs. Acutal distributions of the initial faults and the faults remaining in the executed constructs depend on the characteristics of the software system and the testing profile. Distributions of faults over constructs have not been studied theoretically and empirically yet. We thus adopt Assumptions 1 and 2 as approximate distributions of the initial and remaining faults.

The number of faults in the constructs not executed up to testing time t is $a\,[1-\pi(t)]$ under Assumption 1. Since the proportion of $d\pi(t)$ relative to the constructs not executed up to t is $d\pi(t)/[1-\pi(t)]$, the number of faults detected from $d\pi(t)$ is thus obtained as

$$b(t)\,a\,[1-\pi(t)]\,\frac{d\pi(t)}{1-\pi(t)} = b(t)\,a\,d\pi(t). \tag{3.1}$$

It can be easily verified that equating Eq. (3.1) to $dm(t)$ results in the coverage-based NHPP SRGM framework given by Eq. (2.2). The number of faults remaining in the constructs executed up to t is $[a\,\pi(t)-m(t)]$ due to Assumption 1. Since the proportion of constructs executed for $[t,\,t+dt)$ is $\pi'(0)\,dt$, the proportion of constructs re-executed for $[t,\,t+dt)$ relative to all the constructs executed up to t is $[\pi'(0)\,dt - d\pi(t)]/\pi(t)$. The number of faults detected from the re-executed constructs is therefore obtained as

$$b(t)\;[a\,\pi(t)-m(t)]\,\frac{[\pi'(0)\,dt - d\pi(t)]}{\pi(t)}. \tag{3.2}$$

Combining the above two quanities, we can compute the total number of faults detected for $[t,\,t+dt)$ as

$$dm(t) = b(t)\;[a\,\pi(t)-m(t)]\,\frac{[\pi'(0)\,dt - d\pi(t)]}{\pi(t)} + b(t)\,a\,d\pi(t). \tag{3.3}$$

Dividing both sides of Eq. (3.3) by dt and simplifying, we have the following differential equation:

$$\frac{dm(t)}{dt} = b(t)\left[a\,\pi'(0) - \frac{\pi'(0)-\pi'(t)}{\pi(t)}\,m(t)\right]. \tag{3.4}$$

Its solution subject to initial condition $m(0)=0$ is obtained as

$$m(t) = e^{-\widetilde{B}(t)}\int_0^t \widetilde{a}(x)\,\widetilde{b}(x)\,e^{\widetilde{B}(x)}\,dx, \tag{3.5}$$

where

$$\widetilde{a}(t) = a\,\pi'(0)\frac{\pi(t)}{[\pi'(0)-\pi'(t)]},\quad \widetilde{b}(t) = b(t)\,\frac{[\pi'(0)-\pi'(t)]}{\pi(t)}\quad \text{and}\quad \widetilde{B}(t) = \int_0^t \widetilde{b}(x)\,dx. \tag{3.6}$$

4. Application of the Proposed NHPP SRGM Framework

A general NHPP SRGM framework was presented in the previous section. Implementation of the proposed framework requires specification of both $b(t)$ and $\pi(t)$. We can generate various MVFs by inserting different pairs of $b(t)$ and $\pi(t)$ into the framework. We first consider $\pi(t)$. Suppose that a CGF satisfies the following condition:

$$\frac{\pi'(0)-\pi'(t)}{\pi(t)} = \pi'(0). \tag{4.1}$$

It can be shown that the CGF satisfying the above condition is the exponential CGF given as

$$\pi(t) = 1 - e^{-\lambda t}, \tag{4.2}$$

Table 4.1: Currently available fault detection rate functions

$b(t)$		Reference
$b_1(t) = b,$	$b > 0$	Goel and Okumoto (1979)
$b_2(t) = \dfrac{b^2 t}{1+bt},$	$b > 0$	Yamada *et al.* (1983)
$b_3(t) = \dfrac{b_1}{1+b_2 e^{-b_1 t}},$	$b_1 > 0,\ b_2 > 0$	Ohba (1984)
$b_4(t) = b_1 b_2 t e^{-b_2 \frac{t^2}{2}},$	$b_1 > 0,\ b_2 > 0$	Yamada *et al.* (1986)
$b_5(t) = b_1 + b_2 \dfrac{m(t)}{a},$	$b_1 > 0$	Kuo *et al.* (2001)
$b_6(t) = b_1 \left[1 - \dfrac{m(t)}{a}\right],$	$b_1 > 0$	Kuo *et al.* (2001)
$b_7(t) = b_1 b_2 t^{b_2 - 1},$	$b_1 > 0,\ b_2 > 0$	Huang *et al.* (2003)
$b_8(t) = \dfrac{b_1}{b_2 + t},$	$b_1 > 0,\ b_2 > 0$	Huang *et al.* (2003)
$b_9(t) = \dfrac{b_1}{1+b_2 e^{-b_3 t}},$	$b_1 > 0,\ b_2 > 0$	Zhang *et al.* (2003)

where $\lambda = \pi'(0)$. Then the proposed differential equation for MVF is then simplified to

$$\frac{dm(t)}{dt} = \lambda\, b(t)\left[a - m(t)\right]. \tag{4.3}$$

And the corresponding MVF is obtained as

$$m(t) = a\left(1 - e^{-\lambda \int_0^t b(x)dx}\right). \tag{4.4}$$

MVFs of well-known NHPP SRGMs belong to the class of MVFs expressed by Eq. (4.3) or Eq. (4.4). MVFs of Goel-Okumoto model, delayed S-shaped model, inflection S-shaped model and Yamada Rayleigh model are obtained by substituting $b(t)$ with $b_i(t)$, for $i = 1, \ldots, 4$ in Table 4.1 and setting λ to 1 (See, Goel and Okumoto, 1979; Yamada *et al.*, (1983); Ohba, 1984; Yamada *et al.*, 1986). Setting λ to 1 is equivalent to assume that all the constructs are executed for $[t,\ t + dt)$ and that all the remaining faults are detectable for $[t,\ t + dt)$.

It was shown by Piwowarski *et al.* (1993) that Eq. (4.2) is the CGF for the uniform testing in which all the constructs are assumed to be equally likely executed. It should be noted that such case is not likely to occur in practice. Park and Fujiwara (2006) showed empirical evidence that the exponential CGF of Eq. (4.2) did not work well for real coverage growth data sets. Recently Park *et al.* (2008) suggested a class of CGFs whose element is

$$\pi(t) = \int_0^\infty \left(1 - e^{-\lambda x}\right) dG(x),$$

where $G(x)$ is some prior distribution of λ. Gokhale and Mullen (2004, 2005) and Park *et al.* (2008) examined CGFs for lognormal and gamma prior distributions and showed that they perform well for real coverage growth data sets. Therefore CGFs suggested in Gokhale and Mullen (2004, 2005) and Park *et al.* (2008) are currently more realistic candidates for practical application of the proposed NHPP SRGM framework than the exponential CGF.

We next consider the fault detection rate function. Table 4.1 shows the fault detection rate functions employed by the previous studies of NHPP SRGMs. These fault detection rate functions have been used in NHPP SRGMs not allowing for the repeated execution of constructs. Furthermore, their validity has not been investigated empirically mainly because statistical methods are not available. Therefore, before applying $b(t)$'s in Table 4.1 to the proposed framework, we should examine if the $b(t)$'s in Table 4.1 are valid models for fault detection rate.

5. Conclusions

An approach to developing an NHPP SRGM framework is to derive a differential equation for MVF. We first showed that the existing NHPP SRGM frameworks did not pay regard to the repeated execution of constructs. In order to resolve the shortcoming, we devloped a new coverage-based NHPP SRGM framework. It was shown that some well-known MVFs were derivable from the proposed framework. CGF and fault detection rate function, two essential components of the proposed framework, were also discussed for application. According to the discussion on CGF and fault detection rate function, our future research will be to investigate validity of the currently available fault detection rate functions and then to complete practical application of the proposed framework.

References

Goel, A. L. and Okumoto, K. (1979). Time-dependent error-detection rate model for software rliability and other performance measures, *IEEE Transactions on Reliability*, **28**, 206–211.

Gokhale, S. S. and Mullen, R. E. (2004). From test count to code coverage using the lognormal failture rate, In *Proceedings of the 15th International Symposium on Software Reliability Engineering*, 295–305.

Gokhale, S. S. and Mullen, R. E. (2005). Dynamic code coveage metrics: A lognormal perspective, In *Proceedings of the 11th IEEE International Software Metrics Symposium.*

Gokhale, S. S., Philip, T., Marinos, P. N. and Trivedi, K. S. (1996). Unification of finite failure non-homogeneous Poisson process models through test coverage, In *Proceedings of the 7th International Symposium on Software Reliability Engineering*, 299–307.

Huang, C. Y., Lyu, M. R. and Kuo, S. Y. (2003). A unified scheme of some nonhomogeneous Poisson process models for software reliability estimation, *IEEE Transactions of Software Engineering*, **29**, 261–269.

Kuo, S. Y., Huang, C. Y. and Lyu, M. R. (2001). Framework for modeling software reliability, using various testing-efforts and fault-detection rates, *IEEE Transactions on Reliability*, **50**, 310–320.

Ohba, M. (1984). Inflection S-shaped software reliability growth models, *Stochastic Models in Reliability Theory*, Springer, Berlin, 144–162.

Park, J. Y. and Fujiwara, T. (2006). Coverage frowth functions for software reliability modeling, *Proceedings of the 2nd AIWARM*, Busan, 435–442.

Park, J. Y., Lee, G. and Park, J. H. (2007). A class of discrete time coverage growth functions for software reliability engineering, *The Korean Communications in Statistics*, **14**, 497–506.

Park, J. Y., Lee, G. and Park, J. H. (2008). A class of coverage growth functions and its practical application, *Journal of the Korean Statistical Society*, **37**, 241–247.

Pham, H., Nordmann, L. and Zhang, X. (1999). A general imperfect software debugging model with S-shaped fault detection rate, *IEEE Transactions on Reliability*, **48**, 169–175.

Pham, H. and Zhang, X. (1997). An NHPP software reliability model and its comparison, *International Journal of Reliability, Quality and Safety Engineering*, **4**, 269–282.

Pham, H. and Zhang, X. (2003). NHPP software reliability and cost models with testing coverage, *European Journal of Operational Research*, **145**, 443–454.

Piwowarski, P., Ohba, M. and Caruso, J. (1993). Coverage measurement experience during function test, In *Proceedings of the 15th International Conference on Software Engineering*, 287–301.

Yamada, S., Ohba, M. and Osaki, S. (1983). S-shaped reliability growth modeling for software error detection, *IEEE Transactions on Reliability*, **32**, 475–484.

Yamada, S., Ohtera, H. and Narihisa, H. (1986). Software reliability growth models with testing-effort, *IEEE Transactions on Reliability*, **35**, 19–23.

Yamamoto, T., Inoue, S. and Yamada, S. (2004). A software reliability growth model with testing-coverage maturity process, In *Proceedings of the 10th ISSAT International Conference on Reliability and Quality in Design*, 299–303.

Zhang, X., Teng, X. and Pham, H. (2003). Considering fault removal efficiency in software reliability assessment, *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, **33**, 114–120.

[Received July 2008, Accepted August 2008]