불완전 결함 발견과 구문 반복 실행을 고려한 커버리지 기반 신뢰성 성장 모형

박 중 양† · 박 재 흥†† · 김 영 순†††

요 약

최근 소프트웨어 신뢰성을 평가하기 위해 신뢰성 측도와 커버리지 간의 관계가 연구되고 있다. 특히 커버리지에 기반한 소프트웨어 신뢰성 성장 모델에서 평균치 함수는 소프트웨어의 신뢰성 성장을 나타내는데 매우 중요한 역할을 한다. 본 논문은 커버리지에 기반한 기존 모형들의 문제점을 평균치 함수와 그 모형이 근거하는 가정을 바탕으로 파악하고, 그 문제점을 해결하기 위한 새로운 평균치 함수를 제안한다. 제안된 새로운 평균치 함수는 불완전 결함 발견과 구문의 반복 실행이 허용되는 일반적인 테스팅 환경에서 도출된 결과이다. 마지막으로 실제 데이터에 제안된 모형을 적용하여 그 성능을 평가한다.

A Coverage-Based Software Reliability Growth Model for Imperfect Fault Detection and Repeated Construct Execution

Joong-Yang Park† · Jae-Heung Park†† · Young-Soon Kim†††

ABSTRACT

Recently relationships between reliability measures and the coverage have been developed for evaluation of software reliability. Particularly the mean value function of the coverage-based software reliability growth model is important because of its key role in representing the software reliability growth. In this paper, we first review the problems of the existing mean value functions with respect to the assumptions on which they are based. Then a new mean value function is proposed. The new mean value function is developed for a general testing environment in which imperfect fault detection and repeated construct execution are allowed. Finally performance of the proposed model is empirically evaluated by applying it to a real data set.

키워드 : 소프트웨어 신뢰성 성장 모델(Software Reliability Growth Model), 평균치 함수(Mean Value Function), 구문(Construct), 균등 테스팅(Uniform Testing), 불완전 결함 발견(Imperfect Fault Detection)

1. Introduction

Recently software is becoming an integral part of computer system. Since failures of a software system can cause severe consequences, quality of software system has become an important software product characteristic. One of quantifiable measures for software quality is software reliability. Testing is a key activity for detecting and removing faults and improving reliability of a software system. In theory, it is impossible to detect and remove all the faults within a reasonable amount of testing time. Therefore developers usually determine when to stop testing and release the software based on the estimate of reliability measures such as the initial fault content, the time to next failure and the number of remaining faults. Many software reliability growth models (SRGMs) have been proposed and applied to estimate software reliability measures.

One of most popular SRGMs is the class of Non-Homogeneous Poisson Process (NHPP) SRGMs. The NHPP SRGMs express the fault detection/removal process during testing. Let $N(t)$ denote the number of faults detected up to testing time t. Assuming that detected faults are removed immediately, $N(t)$ represents the fault detection/removal process. The NHPP SRGMs assume that $N(t)$ follows a Poisson distribution with mean value function (MVF) $m(t)$. The MVF $m(t)$ represents the relationship between the expected number of detected faults and the testing time. The NHPP

† 정 회 원 : 경상대학교 자연과학대학 정보통계학과 교수
†† 정 회 원 : 경상대학교 대학원 정보처리학과
††† 준 회 원 : 경상대학교 컴퓨터과학과 교수
논문접수 : 2004년 7월 7일, 심사완료 : 2004년 9월 9일

SRGMs are characterized by their own MVFs. Most of the existing NHPP SRGMs derive the MVF from the assumption that failure intensity is proportional to the number of faults remaining in the software. Pham and Nordmann [6] thus propose the general NHPP SRGM represented by the following differential equation :

$$\frac{dm(t)}{dt} = b(t)[a(t) - m(t)], \quad (1)$$

where $a(t)$ is the fault content function and $b(t)$ is the fault detection rate function. In this framework, various $a(t)$ and $b(t)$ reflect various assumptions on the software testing process. A constant $a(t)$ implies perfect debugging assumption ; an increasing $a(t)$ reflects imperfect debugging. A constant $b(t)$ means that the failure intensity is proportional to the number of remaining faults ; an increasing $b(t)$ implies that the fault detection rate varies due to, for example, the learning phenomenon during testing. Pham and Zhang [8], Pham, et al. [7] and Pham [5] identify $a(t)$ and $b(t)$ of the currently available NHPP SRGMs.

There is another approach to the development of NHPP SRGMs. The approach takes advantage of coverage information gathered during testing. The idea behind the approach is that the more a software system is covered, the more likely reliable is this software system. A few coverage-based NHPP SRGMs have been proposed in Vouk [11], Piwowarski, et al. [9], Gokhale, et al. [1], Rivers and Vouk [10], Park, et al. [4] and Malaiya, et al. [2]. In this paper we focus on the MVF expressed in coverage, $m(c)$. MVF in coverage expresses the relationship between the expected number of detected faults and the coverage. The primary objectives are to review the existing MVFs and to propose a new MVF. We first review the existing MVFs in Section 2. For practical application we need to select a model whose underlying assumptions adequately represent the actual fault detection/removal process. Thus the underlying assumptions of each MVF are discussed in depth. It is shown that some MVFs do not comply with their underlying assumptions and that some MVFs are based on impractical assumptions. For the former case either the correct MVF or the correct set of assumptions is given; for the latter case the corresponding practical assumptions are identified. In addition, new interpretations of some assumptions are also presented. Section 3 derives a differential equation for the MVF for a general testing environment, in which imperfect fault detection and repeated construct execution are allowed. The differential equation is then implemented for the uniform testing environment. An illustrative numerical example for evaluating the practical applicability of the proposed model is presented in Section 4.

• Notation

$m(c)$	: the MVF in coverage
M	: the set of all the constructs in the software under testing
$\|\cdot\|$	: the cardinality of a set
$\overline{M_c(t)} = M - M_c(t)$	: the set of the constructs not yet covered up to testing time t
$M_c(t)$	: the set of the constructs covered up to testing time t
$c = c(t) = \frac{\|M_c(t)\|}{\|M\|}$	: the coverage at testing time t
$dM_c(t)$	: the increment of $M_c(t)$ caused by additional testing during dt
$dc = dc(t) = \frac{\|dM_c(t)\|}{\|M\|}$	: the increment of $c(t)$ caused by additional testing during dt
a	: the total number of faults in the software
$b(t)$, $b(c)$	: the fault detection rate functions in the testing time and the coverage

2. Review of the Existing MVFs in Coverage

Recently several MVFs in coverage have been proposed and applied to real test data sets. This section reviews the existing models for the MVF. Assumptions on which each model is based are specifically presented and appreciated in order to evaluate its adequacy for the fault detection phenomenon. When various model are available, we should choose the one whose underling assumptions fit best for the testing environment under consideration. If such a model does not exist for a specific testing environment, we either choose an approximate model or develop a new model for the specific testing environment. Therefore it is important to understand the underlying assumptions of each model. In addition, we provide new interpretation of some assumptions.

2.1 Vouk Model

The first coverage-based SRGM is proposed by Vouk [11]. The MVF of Vouk model is developed under the following assumptions :

- Assumption 1. (Perfect Debugging) Detected faults are removed immediately without introducing new faults.
- Assumption 2. (Imperfect Fault Detection) Coverage of a construct does not imply that the construct is fault-free.

- Assumption 3. The rate of change in the number of detected faults with respect to coverage is proportional to the number of faults remaining in the software.
- Assumption 4. The fault detection rate is proportional to the coverage.
- Assumption 5. (Resource-Constraint Non-operational Testing) Test cases are generated so as to cover as many constructs in $\overline{Mc(t)}$ as possible.
- Assumption 6. There is the minimum coverage, c_{min}, such that $c_{min} \le c(t)$.

A usual modeling approach is to derive a differential equation describing the dynamic behavior of MVF. The following differential equation was derived in Vouk [11] from the above assumptions.

$$\frac{dm(c)}{dc} = \beta(c - c_{min})[a - m(c)] \quad \text{for} \quad c_{min} \le c. \quad (2)$$

Solving this differential equation with initial condition $m(c_{min})=0$, the MVF in coverage of Vouk model is obtained as

$$m(c) = a\{1 - \exp[-\beta(c - c_{min})^2]\}. \quad (3)$$

Let us first discuss the imperfect fault detection assumption and other assumptions related to it. The discussion is also applicable to the subsequent models. Consider a construct having faults at the beginning of testing. The imperfect fault detection assumption implies that the construct may contain faults even after it has been covered. The fault detection phenomenon under the imperfect fault detection assumption is usually modeled as a stochastic process in which

① faults are independent, i.e., detection of a fault is independent of detection of other faults ;
② when a construct is covered, a fault in that construct is detected with some probability.

The detection probability is usually referred to as the fault detection rate per fault or simply the fault detection rate. It generally depends on the testing time and the coverage. The time and coverage dependency of the fault detection rate reflects the learning factor, the ease of test and so forth. Assumption 4 says that the fault detection rate function $b(c)$ is a linear function of already achieved coverage, i.e., $b(c) = \beta(c - c_{min})$ where β is the proportionality parameter. If the assumption is replaced by

- Assumption 2.1. (Perfect Fault Detection) Coverage of a construct implies that the construct is fault-free,

assumptions about the fault detection rate function are not necessary. The perfect fault detection simply means that the fault detection rate is 1.

Assumption 5 is closely related to Assumption 2. The redundant execution of constructs generally occurs in a general testing environment. All the constructs executed by additional test cases do not belong to $\overline{M_c(t)}$. Some belong to $M_c(t)$ and others belong to $\overline{M_c(t)}$. That is, constructs in $M_c(t)$ may be executed over and over again. If the fault detection is perfect, the constructs in $M_c(t)$ are all fault-free. All the remaining faults are located at the constructs in $\overline{M_c(t)}$. If the fault detection is imperfect, the constructs in $M_c(t)$ may contain faults. The faults remaining in $M_c(t)$ are also exposed to fault detection activity. Under a general testing strategy the model should take into account the additional fault detection phenomenon caused by the redundant execution of constructs. Vouk model ignores this additional fault detection phenomenon by employing Assumption 5. In other words, Vouk model considers only the fault detected in $\overline{M_c(t)}$. Since Vouk model assumes the imperfect fault detection, the number of faults in $\overline{M_c(t)}$ is not $[a - m(c)]$, but $[a - m(c)]$ minus the number of faults in $M_c(t)$. Therefore Equation (2) is likely to overestimate $dm(c)/dc$. In conclusion, Vouk model does comply with its underlying assumptions. Specifically speaking, Assumption 3 cannot be adopted with Assumptions 2 and 5.

2.2 Rivers and Vouk Model

Rivers and Vouk [10] proposed a coverage-based SRGM with MVF

$$m(c) = a - (a - m_{min})\exp\left(\int_{c_{min}}^{c} \frac{b(z)}{1-z}\, dz\right), \quad (4)$$

which is obtained by solving

$$\frac{dm(c)}{dc} = \frac{b(c)}{(1-c)}[a - m(c)] \quad \text{for} \quad c_{min} \le c \quad (5)$$

with initial condition $m(c_{min}) = m_{min}$. Rivers and Vouk [10] derived Equation (5) from Assumptions 1-3, 5 and the following assumptions :

- Assumption 4.1. The fault detection rate is a function of coverage.

- Assumption 7. (Uniform Fault Distribution) Faults are uniformly distributed over all the constructs in M at the beginning of testing.
- Assumption 8. At most one fault can be located at a construct.
- Assumption 9. All the constructs in $\overline{M_c(t)}$ are equally likely to be executed by a test case.

Since Rivers and Vouk model adopts Assumptions 2, 3 and 5, it suffers the same problem with Vouk model as mentioned in the last paragraph of the previous subsection. That is, Rivers and Vouk model does not comply with its underlying assumptions.

However, it is still necessary to discuss three newly employed assumptions. The discussion produces a new MVF. Suppose that Assumption 3, conflicting with Assumption 2 and 5, is eliminated from the set of assumptions listed above. Due to Assumption 5 only the faults located at $dM_c(t)$ are exposed to fault detection activity. The increment $dM(c)$ will then be proportional to the number of faults in $dM_c(t)$. Assumption 7 implies that the fault density per construct is a constant $a|M|^{-1}$ at the beginning of testing. The constructs in $dM_c(t)$ are newly covered and their fault densities are $a|M|^{-1}$ as they were at the beginning of testing. It is easily verified that the expected number of faults in $dM_c(t)$ is adc and that the expected number of faults detected from $dM_c(t)$ is $ab(c)dc$. Thus

$$\frac{dc(c)}{dc} = ab(c) \text{ and } m(c) = a\int_{c_{min}}^{c} b(\tau)\,d\tau \quad \text{for } c_{min} \le c. \tag{6}$$

It should be noted that Assumptions 8 and 9 are not used for the derivation of Equation (6).

Assumption 8 enables us to classify the constructs in $\overline{M_c(t)}$ into two categories : constructs with a fault (one-fault constructs) and fault-free constructs. The number of one-fault constructs is equal to the number of faults in $\overline{M_c(t)}$ and the number of faults in $\overline{M_c(t)}$ is $a(1-c)$ under Assumption 7. The expected number of one-fault constructs in $dM_c(t)$ is obtained under Assumption 9 as

$$a(1-c)\frac{|dM_c(t)|}{|M_c(t)|} = a(1-c)\frac{dc}{1-c} = adc.$$

Thus $dm(c)/dc$ and $m(c)$ are again obtained as Equation (6). That is, Assumptions 8 and 9 does not make any change to the model given by Equation (6).

2.3 Piwowarski, Ohba and Caruso Model

Piwowarski, Ohba and Caruso [9] developed an MVF in coverage under Assumptions 1, 2.1 and 7 and the following modified assumption :

- Assumption 9.1. (Uniform Testing Profile) All the constructs in M are equally likely to be executed by a test.

Assumption 2.1 implies that the faults in a construct are detected when the construct is executed for the first time. In other words, the fault detection rate is 1. The increment $dm(c)$ will then be the expected number of faults within $dM_c(t)$. Since the fault density per construct is $a|M|^{-1}$ due to Assumption 7 and $|dM_c(t)| = |M|dc$, we have

$$\frac{dm(c)}{dc} = a. \tag{7}$$

Solving with initial condition $m(0)=0$, the MVF in coverage is thus obtained as

$$m(c) = ac. \tag{8}$$

We can transform the above differential equation into a different form. Note that $a-m(c)=a(1-c)$ and $a=[a-m(c)](1-c)^{-1}$. Substituting these into Equation (7), we have

$$\frac{dm(c)}{dc} = \frac{a-m(c)}{1-c}. \tag{9}$$

We will now show that Equation (9) derived from a different set of assumptions. Suppose that Assumptions 1, 2.1 and

- Assumption 10. Remaining faults are uniformly distributed over all the constructs in $\overline{M_c(t)}$

are postulated. Since the fault detection is perfect, all the remaining faults are located within $\overline{M_c(t)}$. Therefore the fault density per construct in $\overline{M_c(t)}$ is computed as $[a-m(c)]\,|\,\overline{M_c(t)}\,|^{-1}$. The expected number of faults in $dM_c(t)$ is then computed as $[a-m(c)]|M_c(t)|^{-1}|dM_c(t)|$. Consequently Assumptions 1, 2.1 and 10 also produce Equation (9), which in turn is equivalent to Equation (7). Therefore, if the perfect fault detection is assumed, Assumption 7 is equivalent to Assumption 10.

Unlikely to the previous two models, Piwowarski, Ohba and Caruso model implicitly allows the redundant execution

of constructs. However, this does not influence the fault detection process under the perfect fault detection. This is because all the constructs in $M_c(t)$ are fault-free and the re-execution of constructs in $M_c(t)$ does not increase the number of detected faults. We should also note that Assumption 9.1 is not used at all for derivation of the above differential equation. Assumption 9.1 is necessary for derivation of the relationship between c and t, which is not discussed in this paper.

2.4 Enhanced NHPP SRGM

Gokhale et al. [1] suggested a unified framework for finite failure NHPP SRGMs, called the Enhanced NHPP (ENHPP) SRGM. The proposed ENHPP model is based on Assumptions 1, 2, 4.1 and 7. The MVF of the ENHPP SRGM is given as

$$m(c) = a\int_0^c b(\tau)d\tau, \tag{10}$$

which is obtained by solving

$$\frac{dm(c)}{dc} = ab(c) \tag{11}$$

with initial condition $m(0)=0$. Note that Equations (10) and (11) are equal to Equation (6) except for the initial condition.

The ENHPP SRGM allows the repeated execution of constructs. Since the fault detection is imperfect, faults can be found in $M_c(t)$. This implies that the increment $dm(c)$ consists of two terms, the expected number of faults found in $dM_c(t)$ and the expected number of faults detected in the constructs re-executed. Equation (11) does not include the second term. The ENHPP model simply ignores the faults remaining in $M_c(t)$. Consequently the fault detection process under Assumption 2 is not fully reflected in the ENHPP model and the ENHPP SRGM is likely to underestimate the expected number of detected faults. The ENHPP SRGM does not comply with its underlying assumptions.

2.5 Park, Park and Park Model

Park, Park and Park [4] developed an MVF in coverage based on the Assumptions 1, 2.1, 6 and 10. The MVF is given by

$$m(c) = a\left[1-\left(\frac{1-c}{1-c_{\min}}\right)^b\right] \text{ for } c_{\min} \le c. \tag{12}$$

The corresponding differential equation is

$$\frac{dm(c)}{dc} = b\frac{a-m(c)}{1-c}, \tag{13}$$

where the fault detection rate is a constant b. Since the fault detection is assumed to be perfect, it is not reasonable to introduce the fault detection rate into the model. That is, b must be set to 1. We showed that if the fault detection is perfect, Assumption 7 is equivalent to Assumption 10. Therefore the model given in Park, Park and Park [4] is the same with Piwowarski, Ohba and Caruso model with Assumption 6 added.

3. A New Coverage-Based NHPP SRGM for Repeated Construct Execution

In the previous section we reviewed MVFs of the existing coverage-based NHPP SRGMs. Assumptions on which each model is based are specified. The differential equation representing the fault detection phenomenon subject to the assumptions are also presented and appreciated in depth. We found that some of the existing models do not comply with their underlying assumptions. One common problem of the existing models is that they do not fully reflect the repeated execution of constructs under the imperfect fault detection environment. If the fault detection activity is not perfect, $M_c(t)$ may contain faults. A test case usually executes constructs in $M_c(t)$ and $\overline{M_c(t)}$. In other words, re-execution of covered constructs occurs in the imperfect fault detection environment. Faults can be detected in both $M_c(t)$ and $\overline{M_c(t)}$. Some models simplify the fault detection phenomenon by employing a testing strategy that does not allow the redundant construct execution. Other models just ignore the fault detection phenomenon occurring in $M_c(t)$. It is therefore necessary to develop more realistic models for the fault detection phenomenon in the imperfect fault detection environment. As testing proceeds, the fault density per construct in $M_c(t)$ becomes lower than it was at the beginning of testing. Therefore the fault density per construct in $M_c(t)$ is different from the fault density per construct in $\overline{M_c(t)}$. A realistic coverage-based SRGM should take into account this fault density change. In this section we develop a new coverage-based NHPP SRGM under Assumptions 1, 2, 4.1, 6 and 7 and the following additional assumption :

- Assumption 11. The faults remaining in $M_c(t)$ are uniformly distributed over all the constructs in $M_c(t)$.

Suppose that additional testing is performed at testing time t during dt. In general, the additional testing will execute some constructs in $M_c(t)$ and/or some constructs in $\overline{M_c(t)}$. Execution of some constructs in $\overline{M_c(t)}$ expands $M_c(t)$ and consequently increases $c(t)$. That is, the set of constructs in $\overline{M_c(t)}$ executed during dt is the increment $dM_c(t)$. The constructs in $M_c(t)$ re-executed during dt increase neither $M_c(t)$ nor $c(t)$. However, it may contribute to the number of detected faults. The set of constructs in $M_c(t)$ re-executed during dt is denoted by $RM_c(t)$. Therefore, the increment of the number of detected faults, $dm(c)$, is the sum of the numbers of faults detected in $RM_c(t)$ and $dM_c(t)$. The number of detected faults in $RM_c(t)(dM_c(t))$ is the product of the fault density of $RM_c(t)(dM_c(t))$ and the fault detection rate. Due to Assumption 7, the fault density of a construct in $dM_c(t)$ is a constant $a|M|^{-1}$ and the number of faults in $dM_c(t)$ is adc. Similarly the total number of (detected and undetected) faults in $M_c(t)$ is obtained as ac. The number of faults remaining (undetected) in $M_c(t)$ is therefore $ac-m(c)$. Assumption 11 implies that the fault density of a construct in $M_c(t)$ is

$$\frac{ac-m(c)}{|M_c(t)|}=\frac{ac-m(c)}{|M|c}. \tag{14}$$

Consequently the increment $dm(c)$ is expressed as

$$dm(c)=b(c)\frac{ac-m(c)}{|M|c}|RM_c(t)|+b(c)\,adc. \tag{15}$$

We derived a general expression for $dm(c)$. It is not yet applicable because $|RM_c(t)|$ is not represented in terms of c and dc. The relationship between $|RM_c(t)|$ and c depends on the testing strategy. Specifically it depends on the strategy of selecting test cases. For example, the repeated execution of constructs in $M_c(t)$ does not occur under Assumption 5. That is, $|RM_c(t)|=0$. Then the general expression results in the following differential equation :

$$\frac{dm(c)}{dc}=ab(c), \tag{16}$$

which is identical to that of the ENHPP model.

Now we derive the general expression for $dm(c)$ under Assumption 9.1. One of important reliability measures is the number of faults remaining in the software. When the primary objective of testing is to detect and remove as many faults as possible and the managemental decision on when to stop testing is made based on the number of remaining faults, all the faults in the software should be considered equally important. It is then reasonable to adopt the testing profile distributing equal frequencies over all the constructs. Even when an operational profile is not available, the uniform testing profile may be used as an alternative. Suppose that p constructs are executed during dt. The p constructs are partitioned into $RM_c(t)$ and $dM_c(t)$, where $|RM_c(t)|$ and $|dM_c(t)|$ are respectively expected to be pc and $p(1-c)$. Therefore, since $|dM_c(t)|$ is $|M|dc$, $|RM_c(t)|$ is expected to be $|M|c(1-c)^{-1}dc$. Substituting this into Equation (15) and dividing both sides by dc, we have

$$\frac{dm(c)}{dc}=b(c)\frac{a-m(c)}{1-c}. \tag{17}$$

Solving with initial condition $m(c_{\min})=0$, the mean value function for the uniform testing profile is obtained as

$$m(c)=a\left[1-\exp\left(-\int_{c_{\min}}^{c}\frac{b(z)}{1-z}\,dz\right)\right]. \tag{18}$$

If we set $c_{\min}=0$ and $m_{\min}=0$ in Equation (4), then Equations (4) and (18) become identical. However, we should note that the two models are based on different sets of assumptions and that Equation (4) cannot be obtained from its underlying assumptions as explained in Subsection 2.2.

The coverage-based NHPP SRGM with MVF of Equation (18) is actually a class of NHPP SRGMs, whose members are defined by the corresponding fault detection rate functions. The simplest is the case where fault detection rate is a constant b. The constant fault detection rate reflects that there is no learning during testing. The MVF for the constant fault detection rate is obtained as

$$m(c)=a\left[1-\left(\frac{1-c}{1-c_{\min}}\right)^{b}\right]. \tag{19}$$

This is equal to Equation (12) of Park, Park and Park model. It is to be remembered that Equations (12) and (19) are derived from different sets of assumptions and that Park, Park and Park model does not comply with its underlying assumptions. If the learning phenomenon seems to occur during testing, an appropriate fault detection rate function should be derived and applied to Equation (18). This will not be discussed further in this paper.

4. Application to A Real Data Set

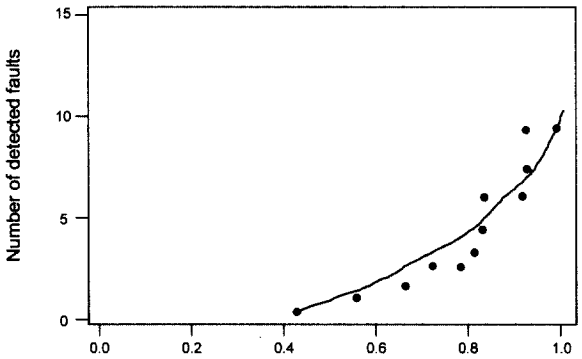
In this section we empirically evaluate performance of the coverage-based NHPP SRGM proposed in the Section 3. Especially the simple model with MVF of Equation (19) is applied to a real data set reported in Vouk [11]. <Table 1> presents the data set gathered from a NASA supported project implementing sensor management in the inertial navigation system. There are three separate implementations of the sensor management system. The data set in <Table 1> is from the first implementation. The data set is also reproduced in Park et al. [4]. The cumulative number of detected faults is plotted in (Figure 1) and (Figure 2). It is evident that the MVF is not a straight line through origin. Therefore Piwowarski, Ohba and Caruso model and the ENHPP with constant fault detection rate are not applicable to this data set. Now the model with MVF of Equation (19) is applied to both block coverage and branch coverage. Maximum likelihood estimates of a, c_{min}, and b and the corresponding performance measures are computed and presented in <Table 2>. The fitted MVFs are also plotted in (Figure 1) and (Figure 2). The proposed coverage-based NHPP SRGM with constant fault detection rate works reasonably well for both coverages. We can say that at most 2 faults remains in the software at the end of testing.

<Table 1> Sensor management system data

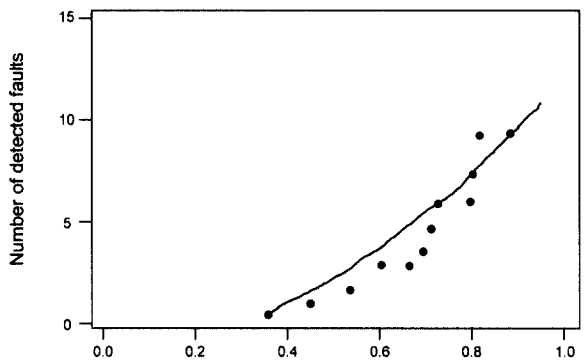
Cumulative number of detected faults	Block coverage	Branch Coverage
0	0.4574	0.3702
1	0.5597	0.4748
2	0.6517	0.5795
3	0.7050	0.6380
3	0.7628	0.6921
4	0.7780	0.7150
5	0.7920	0.7280
6	0.8000	0.7380
6	0.8682	0.8169
7	0.8720	0.8250
9	0.8853	0.8410
9	0.9597	0.9376

<Table 2> Maximum likelihood estimates and performance measures

Parameter	Block coverage	Branch coverage
a	10.5160	10.4660
c_{min}	0.4574	0.3702
b	0.7450	0.8501
SSE	10.7992	10.8000
AIC	43.3655	41.8902



(Figure 1) Plot of the number of detected faults and the fitted MVF for block coverage.



(Figure 2) Plot of the number of detected faults and the fitted MVF for branch coverage.

5. Concluding Remarks

In this paper we first review MVFs of the analytical coverage-based NHPP SRGMs. Thus the empirical model suggested in Malaiya et al. [2] is excluded from this paper. The review shows that some models do not comply with their underlying assumptions and that some models do not appropriately represent the fault detection phenomenon. Specifically all the models do not reflect the repeated execution of constructs in the imperfect fault detection environment. Thus a new model explicitly taking account of the repeated execution of constructs is developed. For practicalapplication of the proposed model we need to provide a fault detection rate function. The proposed model is implemented for the case where the fault detection rate is constant and applied to a real data set. The constant fault detection rate implies that there is no learning during testing. Therefore, fault detection rate functions reflecting the learning phenomenon need to be developed. Future research will be directed to the development of the fault detection rate function.

References

[1] S. S. Gokhale, T. Philip, P. N. Marinos and K. S. Trivedi, "Unification of Finite Failure Non-Homogeneous Poisson Process Models Through Test Coverage," Proceedings of IEEE International Symposium on Software Reliability Engineering, pp.299-307, 1996.

[2] Y. K. Malaiya, N. Li, J. Bieman and R. Karcich, "Software Reliability Growth with Test Coverage," IEEE Transactions on Reliability, Vol.51, pp.420-426, 2002.

[3] J. D. Musa and K. Okumoto, "A Logarithmic Poisson Execution Time Model for Software Reliability Measurement," Proceedings of the 7th International Conference on Software Engineering, pp.230-238, Orlando, 1984.

[4] J. Y. Park, J. H. Park and S. J. Park, "A Study on Test Coverage for Software Reliability Evaluation," Transactions of Korean Information Processing Society, Vol.8, No.4, pp.409-420, 2001.

[5] H. Pham, "Software Reliability and Cost Models : Perspectives, Comparison, and Practice," European Journal of Operational Research, pp.475-489, 2003.

[6] H. Pham and L. Nordmann, "A Generalized NHPP Software Reliability Model," in Proceeding 3rd Conference On Reliability and Quality in Design, Anaheim, March, 1997.

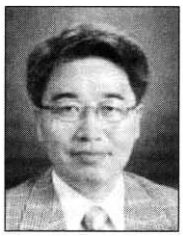
[7] H. Pham, L. Nordmann and X. Zhang, "A General Imperfect-Software-Debugging Model with S-Shaped Fault-Detection Rate," IEEE Transactions on Reliability, Vol.48, No.2, pp.169-175, 1999.

[8] H. Pham and X. Zhang, "An NHPP Software Reliability Models and its Comparison," International Journal of Reliability, Quality and Safety Engineering, 4, pp.269-282, 1997.

[9] P. Piwowarski, M. Ohba and J. Caruso, "Coverage Measurement Experience During Function Test," Proceedings of the 15th International Conference on Software Engineering, pp.287-300, Baltimore, MD, May, 1993.

[10] A. T. Rivers and M. A. Vouk, "Resource-Constrained Non-Operational Testing of Software," Proc. 9th International Symposium on Software Reliability Engineering, pp.154-163, 1998.

[11] M. A. Vouk, "Using Reliability Models During Testing with Non-operational Profiles," Proceedings of the 2nd Bellcore/Purdue Workshop on Issues in Software Reliability Estimation, pp.103-111, Oct., 1992.



박 중 양

e-mail : joongyang@nongae.gsnu.ac.kr

1982년 연세대학교 응용통계학과 졸업 (학사)

1984년 한국과학기술원 산업공학과 (공학석사)

1994년 한국과학기술원 산업공학과 (공학박사)

1984년~1989년 경상대학교 전산통계학과 교수

1989년~현재 경상대학교 정보통계학과 교수

관심분야 : 소프트웨어 신뢰성, 신경망, 선형 통계 모형, 실험계획법 등

박 재 흥

e-mail : pjh@nongae.gsnu.ac.kr

1973년~1978년 충북대학교 수학과(학사)

1978년~1980년 중앙대학교 대학원 전산학과(석사)

1985년~1988년 중앙대학교 대학원 전산학과(박사)

1983년~현재 경상대학교 컴퓨터과학과 교수

관심분야 : 소프트웨어 신뢰성, 시험도구 자동화, 시스템 분석 및 설계, 신경망

김 영 순

e-mail : coaskim@hanmail.net

1994년 경상대학교 통계학과 졸업(학사)

1998년 경상대학교 통계학과(이학석사)

2000년~현재 경상대학교 정보처리학과 (박사과정)

관심분야 : 소프트웨어 신뢰성, 신경망, 선형 통계 모형, 실험계획법 등