

논문 2004-41TC-8-9

소프트웨어 컴포넌트 규모에 의한 소프트웨어 결함 밀도의 평가

(An Analysis of the Software defect density based on components size)

이 재 기*, 남 상 식**, 김 창 봉***

(Jae-Ki Lee, Sang-Sik Nam, and Chang-Bong Kim)

요 약

본 논문은 Malaiya와 denton이 제안한 모듈크기(module size)와 소프트웨어 결함밀도와와의 관계를 확률 모델화하는데 있어서 고려할 사항으로 정확한 소프트웨어 결함밀도를 추정하는데 그 목적이 있다. 구체적으로 말하면 우리가 수행한 3개 프로젝트에 대해서 지수형 분포 및 기하분포 등에 대한 소프트웨어 모듈의 크기 분포나 모듈 또는 KLOC 등 규정된 크기에서 포함하고 있는 결함수를 추정하고 실험데이터에 적합한 모델의 구성 가능성을 제안하여 시험기간 중에 관측된 데이터를 적용하여 이를 상호 비교한다.

Abstract

In this paper, we estimated the exact software defect density to build up a suitable model that is closely related to the size of module in the probability model proposed by MD (Malaiya and Denton). To put it concretely, we predict that the software defect density using some practical data sets that are the outcomes from the system test performed our three projects for the types of distribution (exponential and geometric), per a unit of module, and the size of source line that have been recommended by KLOC(kilo-line-of-code). Then, we make comparison between our proposed defect density model and those examined real data.

Keywords: 소프트웨어 신뢰도, 결함밀도, 모듈크기, 소스라인(SLOC), MD 모델, 지수분포모델, 기하분포모델

I. 서 론

고도정보사회로 불리는 現時代에서 소프트웨어 시스템의 신뢰성 평가는 매우 중요한 기반기술로 정착되고 있다. 소프트웨어 신뢰성의 정량적인 평가에 있어서는 통상 소프트웨어 신뢰도(software reliability)와 소프트웨어 결함밀도(software fault density)가 사용된다. 즉, 전자는 테스트(시험) 단계에서 규정시간 내에 결함이 발견되지 않는 경우로서 소프트웨어 장애가 규정시간 내에 발생하지 않을 확률로 정의하고 이를 확률모델에 근거하여 여러 가지 변형된 형태로 평가하고 있다.^[1] 이에 반해서 후자는 소프트웨어 결함밀도(또는 단순

결함밀도)로써 단위 코드수(통상 1,000 라인 or KLOC 기준)에 포함하는 결함수(기대치)로 정의하고 있다. 실제 사용되고 있는 평가지표의 소프트웨어 신뢰성은 소프트웨어 내에 잔존하고 있는 결함수의 크기에 의존한다는 개념하에 고 신뢰도를 갖는 소프트웨어를 개발하는데 있어서 소프트웨어 결함밀도의 통계적 성질을 정확히 파악할 필요가 있다.

한편, 소프트웨어 컴포넌트 개념에 의존한 소프트웨어공학(CBSE: component based on software engineering)의 유효성^{[2][4]}을 주장하고 있는 요즘 소프트웨어 모듈 크기(module size)와 소프트웨어 결함밀도와의 관계를 이론적으로 해석하는 것을 매우 중요하게 고려하고 있다. munson & khoshgoftaar^[5]나 khoshgoftaar & allen^[6]은 판별 분석(discriminant analysis)의 각종 수단을 이용하고 상대적으로 소프트웨어 결함을 많이 포함하고 있는 모듈을 정하는 방법을 제안했다.

또 Malaiya와 denton^[7]은 모듈크기의 확률분포를 정

* 정회원, ** 중신회원, 한국전자통신연구원 BcN시험기술팀 (BcN Test Technology Team)

*** 중신회원, 공주대학교 정보통신공학과
(Dept. of Information and Communication Engineering, Kongju University)

접수일자: 2004년2월26일, 수정완료일: 2004년8월2일

하고 모듈크기에 포함된 결함을 표시하는 결함함수를 도입하고 소프트웨어 결함밀도를 추정하는 확률모델을 고찰하였다. 특히 [7]에서 제안된 모델은 모듈 통합시험이나 시스템시험 초기단계에서 관측된 데이터수가 적은 경우에 유용할 수 있는 모델로써 판별분석에 많은 데이터 해석이 필요치 않은 이점이 있다.

반면에 Malaiya와 denton model(MD model)은 소프트웨어를 구성하는 모듈의 크기가 독립된 동일한 분포로 강한 제약이 존재하며, 아주 단순한 모델로 지적되고 있다. 이러한 MD 모델의 유효성 논란은 수천 개 또는 수만 개의 모듈을 포함하는 대규모 소프트웨어에 대해서 근사적 모듈크기의 독립성과 동일성을 보존하고 있어 MD 모델의 이론적 문제점을 포함함으로써 실제 소프트웨어 결함밀도의 정확한 추정이 곤란하다. 그래서 본 논문에서는 MD 모델의 모듈크기 확률분포 표현이나 결함밀도의 수를 표현하는 수정된 모델을 제안하는 경우의 소프트웨어 결함밀도의 간편한 추정방법에 대해 논한다.

최종적으로 데이터를 적용하여 basili & perricone^[10]이나 withdraw^[11]에 의해 발표된 소프트웨어 결함데이터를 이용한 개량모델과 MD 모델의 비교를 수행하고 소프트웨어 결함밀도의 추정도에 관한 유효성을 검증한다.

II. MD(Malaiya and denton) 모델의 개념

소프트웨어는 다수의 모듈로 구성되고 각 모듈의 프로그램은 실행명령을 포함한다. 소프트웨어 결함은 모듈에 관련된 결함과 프로그램 안의 실행명령에 관련된 결함으로 분류하고 각 모듈의 크기(라인수, 명령수 등)에 독립된 동일한 분포에 포함되지 않은 확률변수 $S(>0)$ 로 표현한다. 모듈에 관련된 결함은 모듈 내에 일정하게 분포된 것으로 가정하며, 이 수는

$$D_m(S) = \frac{a}{S}, a(\geq 0): \text{임의의 파라미터}$$

에 따르는 것으로 가정할 수 있다. 한편 프로그램 중에 실행명령에 관련된 결함은 모듈크기가 클수록 증가하는 것으로 고려되는데 이러한 결함수는

$$D_i(S) = b + cS, [b(\geq 0), c(\geq 0)]:$$

임의의 파라미터

로 표현된다. 크기가 S인 모듈에 포함된 소프트웨어 결

함은

$$D(S) = D_m(S) + D_i(S) = \frac{a}{S} + b + cS \tag{1}$$

로 기술된다. 이때 위의 함수 $D(S)$ 를 결함함수(defect function)라고 부른다. Malaiya와 denton은 각 소프트웨어 모듈의 크기를 근사적인 연속값으로 지수분포에 따르는 것으로 가정하고 있다. 다시 말해서 모듈크기의 확률분포를 $F_1(s)$ 라고 하면

$$F_1(s) = P_r\{S \leq s\} = 1 - \exp(-gs) \tag{2}$$

가 되고 여기서 $\frac{1}{g} (> 0)$ 은 평균 모듈크기를 나타내는 파라미터이다. 또한 모듈크기 분포의 동일성으로부터 소프트웨어에 포함된 총 결함수의 기대치는

$$E[N(S)] \approx M \int_1^{s_{max}} N(s) dF_1(s) \tag{3}$$

으로 구할 수 있다.

여기서 $M (> 1)$ 은 소프트웨어에 포함된 총 모듈수, $s_{max} (> 0)$ 는 최대 모듈크기를 나타낸다. 또 MD 모델의 소프트웨어 결함밀도 D_1 은 평균 시스템 크기 ($k_1 = M/g$)를 사용해서 구할 수가 있는데

$$D_1 = \frac{E[N(s)]}{k_1} \tag{4}$$

로 표현할 수 있다.

III. 모듈크기 분포(module size distribution)의 개선

1. 지수분포모델(exponential distribution model)

MD 모델의 문제점은 모듈크기의 확률분포 크기를 크게 표현하는 것으로 예를 들면 모듈크기분포 $F_1(s)$ 의 영향을 무시하고 식 (3)과 같이 표현된 소프트웨어에 포함된 총 기대 결함수 $E[N(s)]$ 는 엄밀한 의미의 기대치가 아니다. 이러한 문제점을 극복하는데 있어 아래와 같은 수정 지수분포

$$F_2(s) = \frac{\int_1^s g \exp(-gx) dx}{\int_1^{s_{max}} g \exp(-gx) dx} \tag{5}$$

로 가정할 수 있다. 여기서 소프트웨어 총 기대 결함수는

$$E[N(s)] = M \int_1^{s_{max}} N(s) dF_2(s) \quad (6)$$

또 소프트웨어 결함밀도는

$$D_2 = \frac{E[N(s)]}{k_2}, \quad (7)$$

$$k_2 = M \int_1^{s_{max}} s dF_1(s) / \int_1^{s_{max}} dF_1(s)$$

로 표현되고 여기서 k_2 는 수정지수분포의 평균 시스템 규모(크기)를 의미한다.

2. 기하분포 모델(geometric distribution model)

이 모델은 실시된 시험의 이력으로부터 배포되는 소프트웨어의 품질이나 신뢰성의 달성상황에 대한 진보과 악을 위해 이용되는 모델로써 시간 경과에 따른 발견된 결함수의 시간적 거동으로 시험을 수행할 때 발견된 정보나 결함수에 대한 정보로부터 소프트웨어 내에 잠재된 결함수를 정량적으로 추정하는 방법이다.

이 모델의 특징은 초기 검출되는 결함데이터가 hazard rate에 많은 영향을 주며, 지수분포모델의 일종으로 infinite failure category model 이다.^[17]

즉, 모듈 크기가 코드(라인)수의 이산치(離散值)로 측정되는바 모듈 크기는 $S = 1, 2, 3, 4, \dots, s_{max}$ 의 정수 값으로 이산형 확률변수로 표현된다.

각 모듈 크기 S 는 독립된 동일 분포로부터 기하분포 $P_r\{S = s\} = p_1(s)$ 에 따른다고 가정하면

$$p_1(s) = p(1-p)^{s-1}, s = 1, 2, 3, \dots \quad (8)$$

로 되고 여기서 $p \in [0, 1]$ 는 미지 파라미터(unknown parameter)이다.

모듈 크기가 $S \in [1, s_{max}]$ 인 경우를 고려할 때 S 에 관한 수정확률함수는

$$p_2(s) = \frac{p(1-p)^{s-1}}{1-(1-p)^{s_{max}}}, s = 1, 2, 3, \dots, s_{max} \quad (9)$$

로 표현된다. 이 모듈크기의 확률분포를 이용하여 소프트웨어에 포함된 총기대결함수 및 소프트웨어 결함밀도를 구하면 식 (10), (11)과 같이 표현된다.

$$E[N(s)] = M \sum_{s=1}^{s_{max}} N(s) p_2(s) \quad (10)$$

$$D_3 = \frac{E[N(s)]}{k_3}, k_3 = \frac{M \sum_{s=1}^{s_{max}} sp(1-p)^{s-1}}{1-(1-p)^{s_{max}}} \quad (11)$$

식 (11)에서 k_3 는 기하분포모델(幾何分布模型)의 평균 시스템 크기를 의미한다.

이와 유사한 방법의 기하분포모델을 이용해 시스템의 서비스 동작에 대한 신뢰도를 측정한 사례도 많이 발표되고 있다.^[14-15]

IV. 결함함수(defect function)의 개선

MD 모델에서 소프트웨어 결함밀도를 평가하는데 있어서 모듈 크기분포와 동일한 결함함수 $D(S)$ 가 중요한 역할을 한다.

식 (1)과 같이 결함수는 직선과 쌍곡선의 결합으로 표현되는 단순 함수로 임의의 파라미터 a, b, c는 모듈 크기의 각 개개의 모듈에 포함된 결함수를 구성하고 있는 n 개의 데이터 $(s_1, d_1), (s_2, d_2), \dots, (s_n, d_n)$ 로 최소자승법을 이용하여 추정된다.

우선 식 (1)과 같이 표현된 결함수에 대해 n 조(組)의 데이터 점을 n-1 차(次)의 Laplace 다항식의 보간법으로 함수 $D(S)$ 를 아래와 같은 다항식으로 표현하는 것을 고려할 수 있다.

$$D(S) = \sum_{i=1}^n \frac{K_i(S)}{K_i(s_i)} d_i \quad (12)$$

여기서

$$K_i(S) = (S-s_1)(S-s_2) \times \dots \times (S-s_n) \quad (13)$$

라플라스(Laplace) 다항식으로 보간(保間)된 결함 함수는 반드시 데이터 $(s_1, d_1), (s_2, d_2), \dots, (s_n, d_n)$ 를 통해서 데이터에 대한 적합성이 최고로 높아진다.

또한 결함 함수는 뉴우튼-랩슨 차분방정식(Newton-Raphson difference equation)의 근사를 고려해 볼 수도 있다.

함수 $D(S)$ 가 n개의 데이터 차분을 이용, (n-1)차의 다항식으로 표현되는 것을 가정할 때 1계차분상 $Q(s_i, s_{i+1})$ 은

$$Q(s_i, s_{i+1}) = \frac{d_{i+1} - d_i}{s_{i+1} - s_i} \quad (14)$$

r 계(係) 차분(r-order difference) 상은

$$Q(s_i, \dots, s_{i+r}) = \frac{1}{s_{i+r} - s_i} \{Q(s_{i+1}, \dots, s_{i+r}) - Q(s_i, \dots, s_{i+r-1})\} \quad (15)$$

로 표현된다.

여기서 차분상간공식에 대한 결함함수의 근사값(近似値)은

$$D(S) = d_1 + (S - s_1)Q(s_1, s_2) + (S - s_1)(S - s_2)Q(s_1, s_2, s_3) + \dots + (S - s_1)\dots(S - s_{n-1})Q(s_1, s_2, \dots, s_n) \quad (16)$$

이 되고 Newton 차분보간공식에 의해 보간(interpolation)된 함수는 반드시 데이터 점을 통과하는 극한이 없을 때 적합성을 약간 희생하는 함수의 근사를 실현하는 특징이 있다.

여기서 결함함수 D(S)를 다항식의 근사 방법 적용이 곤란한 경우에는 식 (1)과 같은 결함함수의 파라미터 표현을 직시하는 것도 고려할 수 있다. 즉, 관측(觀測) 데이터로부터 모듈크기와 결함밀도의 관계를 성립할 수 있는 경험(經驗)적인 방법을 이용하여 유용한 정보를 얻을 필요가 있다.

o 경험적인 방법(empirical method)에서 얻을 정보

- (1) 모듈크기가 s_1 이하인 경우 모듈 크기가 감소함에 따라 결함밀도가 감소하는지?
- (2) 모듈크기가 s_n 이상인 경우 모듈 크기가 증가함에 따라 결함밀도가 증가하는지?

위에 기술한 (s_1, s_n)의 값은 소프트웨어 종류나 규모에 대해 상이한 점이 존재하는지, 이러한 경험적인 식견과 고려시 대체적으로 결함수를 가정하는데 의미가 있는가를 확인해야 한다. 여기서 특히 소프트웨어 결함 밀도는

$$D(S) = a \times \exp(bS) + c \times \exp(-dS) \quad (17)$$

의 함수로 정의하고 있다. 여기서 a,b,c,d 는 임의의 파라미터로 MD 모델과 구별된 대체 결함함수라고 부른다.

V. 실측 데이터 적용 결과

실제 시스템의 모듈 크기(개별 코드수, 결함수, 사후 계측된 결함밀도[각 모듈의 결함밀도에 대한 산술평균])

에 대한 데이터를 적용하는 경우 모듈 크기의 분포 개선점이나 결함함수의 개선점에 대한 검토를 추가한다.

Malaiya와 denton의 논문^[7]에서는 식 (1)에 의한 결함함수의 파라미터 a, b, c, d를 추정하는데 있어서 초기에 c=0를 가정하는 경우 코드수(라인수)와 모듈에 포함된 결함수로부터 구성된 n개 데이터 즉,

$$(s_1, d_1), (s_2, d_2) \dots (s_n, d_n)$$

의 최소자승법에 의한 추정치 \hat{a}, \hat{b} 를 구한다. 한편 \hat{a}, \hat{b} 를 고정시 동일한 데이터에 대한 c를 추정할 수 있다. 이러한 파라미터 추정법은 문헌 [7]에 명기된 파라미터 a,b,c를 동시에 최소자승법으로 추정할 필요가 있다.

문헌 [7]의 추정결과를 MD, 3개의 파라미터를 동시에 추정한 MD 모델을 improved MD, Laplace 다항식, Newton 차분상보간 공식, 식 (17)에 의해 대체된 결함함수를 사용한 모델, 다양한 Lagrange, Newton, Alternative로 표기할 수 있다. 또 모듈크기 분포식인 식 (2)의 지수함수분포를 가정하는 경우, 수정지수분포를 가정하는 경우, 수정기하분포(mgd: modified geometric distribution)를 가정하는 경우 등으로 각각 D1, D2, D3, 로 표기한다.

본 논문에서는 III, IV장에 표시한 MD 모델에 대해 모듈크기 분포와 결함함수의 2가지 관점으로부터 개선을 행한다. 원래 모듈분포와 결함함수에 대해 제안된 방법에 대한 유효성을 검토할 필요가 있으며, 데이터 set의 성질을 고려 시 종합적인 추정에 관한 성능평가에 대해 전체적인 결함밀도에 대한 추정치의 정확성에 목적을 두고 있다.

1. 적용된 실측 데이터(I)

문헌 [8]에 의거하여 보고된 소프트웨어 결함데이터(I)을 표 1에 표시하였다. 이 데이터는 음성 및 패킷데이터 전송을 위한 멀티미디어 시스템으로서 소프트웨어 개발그룹에 의해 개발된 대형 스위칭 시스템으로부터 얻어진 데이터이다. 시스템의 규모는 약 130만 라인, 138개의 모듈(기능 블록으로 호칭)로 구성되어 있다. 표 2는 본 연구에서 제안된 모델에 의해 추정된 소프트웨어 결함밀도이다.

표 1에 의하면 시스템 전체의 모듈 당 평균 결함밀도는 33.27 이며, 경험데이터(empirical data)와 비교해 볼 때 소프트웨어 결함밀도는 모듈 크기분포 모델에서 수정기하분포, 수정지수분포, MD 모델 순으로 양호한 것으로 분석되었다. 다시 말해서 결함함수 D(S)의 추정결과를 살펴보면 MD와 지수분포 등이 관측 데이터에

표 1. 소프트웨어 결함 데이터(data set I)
Table 1. Software defect data(data set I).

| 모듈 크기(size) | No. of modules | Defect density |
|-------------|----------------|----------------|
| <1,000 | 1(0.70%) | 3.0 |
| 3,000 | 26(18.8%) | 149/26=5.73 |
| 5,000 | 32(23.2%) | 293/32=9.16 |
| 10,000 | 46(33.3%) | 579/46=12.59 |
| 15,000 | 14(10.2%) | 399/14=28.5 |
| >15,000 | 19(13.8%) | 772/19=40.63 |

표 2. 소프트웨어 결함밀도 추정치(data set I)
Table 2. Estimation of the software fault density(data set I).

| Model | D ₁ | D ₂ | D ₃ |
|-----------|------------------------------------|----------------|----------------|
| MD | 1.78/KLOC | 0.87/KLOC | 1.52/KLOC |
| Newton | [28.2743/Module, 3.23/KLOC](최소자승법) | | |
| empirical | 1.2/KLOC | | |

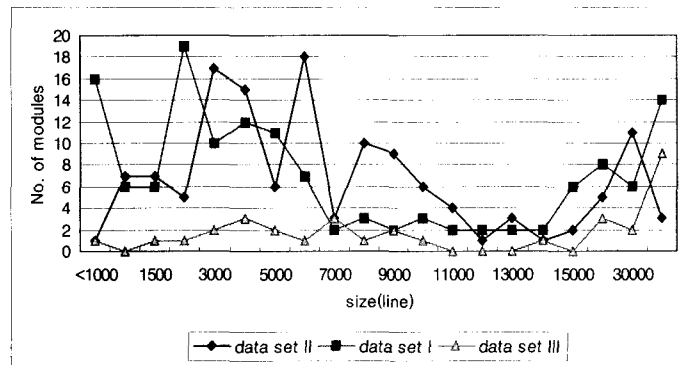


그림 1. 소프트웨어 모듈 크기 분포(data set I, II, III)
Fig. 1. Distribution of the software module size(data set I, II, III).

표 3. 모듈 크기별 소프트웨어 결함 데이터(II/III)
Table 3. Software defect data for a module size(II/III).

| 모듈 크기 (size) | data set II | | data set III | |
|-----------------|----------------|----------------|----------------|----------------|
| | No. of modules | Defect density | No. of modules | Defect density |
| < 1,000 | 16(13.2%) | 88/16=5.50 | 1(2.5%) | 6.0(1.3%) |
| 3,000 | 31(25.6%) | 345/31=11.12 | 2(5.0%) | 7.5(3.4%) |
| 5,000 | 22(18.2%) | 555/22=25.23 | 5(12.5%) | 7.0(7.8%) |
| 10,000 | 25(20.7%) | 714/25=28.56 | 10(25%) | 11.3(25.3%) |
| 15,000 | 10(8.30%) | 404/10=40.4 | 4(10%) | 15.75(14.1%) |
| >15,000 | 17(14.0%) | 911/17=53.6 | 18(45%) | 11.94(48.1%) |

대해 과대평가 또는 저평가된 상태이다.

모듈크기에 따른 각 모델별 소프트웨어 결함수의 관계는 표 2와 같이 추정되었다.

2. 적용된 실측 데이터(II, III)

문헌 [9]에 의해 보고된 소프트웨어 결함데이터를 표 3에 표시하였다. 이 데이터는 대용량 셀 처리 능력을 갖춘 ATM(Asynchronous Transfer Mode) 교환시스템의 응용프로그램 개발 과정에서 얻어진 결함데이터이다. 프로그램 개발 경험이 풍부한 소프트웨어 개발팀에서 개발한 총 136개의 모듈(본 논문에서는 121개 모듈만 적용)로 전체 시스템 규모는 240만 라인이다.

시스템 전체의 모듈 당 평균 결함밀도는 22.16으로

추정되었으며, 순수 MD 모델을 제외하고는 추정결과가 실제 관측된 데이터와 비교해 볼 때 기하분포, 지수분포 순으로 양호한 것으로 나타났다. 순수 MD모델의 결함밀도 추정결과는 저평가된 것으로 판명(判明)되었다.

표 3은 2개 프로젝트에 대한 소프트웨어 모듈크기에 따른 현황으로 모듈 당 결함밀도(결함수)를 나타낸 것이다. 즉, 전체 시스템에서 모듈 크기에 따른 분포 비율과 각 규모별, 모듈 당 소프트웨어 결함수를 분석한 결과로써 모듈 크기에 따라 결함수도 증가하는 추세를 보인다. 모듈의 규모는 대체로 data set II가 3,000 ~ 10,000 라인, data set III는 10,000 라인과 15,000 라인 이상인 것이 가장 많이 차지하고 있는 것으로 밝혀졌다. Data set III의 defect density 괄호 안의 데이터는

표 4. 소프트웨어 결함밀도 추정(II/III)
Table 4. Estimation of the software fault density(II/III).

| Data group | Model | D ₁ | D ₂ | D ₃ |
|------------|-----------|------------------------------------|----------------|----------------|
| II | MD | 1.14/KLOC | 1.98/KLOC | 2.11/KLOC |
| | Newton | [27.352/Module, 1.93/KLOC](최소자승법) | | |
| | empirical | 2.07/KLOC | | |
| III | Model | D ₁ | D ₂ | D ₃ |
| | MD | 2.217/KLOC | 1.285/KLOC | 0.916/KLOC |
| | Newton | [10.536/Module, 0.459/KLOC](최소자승법) | | |
| | empirical | 0.79/KLOC | | |

표 5. 소프트웨어 모듈수 및 모델 parameter 추정치
Table 5. Software modules and model parameters.

| Data set | parameter | | | M | s(Avg.) | s _{max} | s _T | N | D/LOC |
|----------|-----------|-----------|-----------------------|-----|---------|------------------|----------------|-----|-----------------------|
| | \hat{a} | \hat{b} | g | | | | | | |
| I | 1842 | 0.10157 | 1.06x10 ⁻⁴ | 138 | 8,479 | 74,325 | 1,297,510 | 405 | 3.12x10 ⁻⁴ |
| II | 940 | 0.0657 | 5.7x10 ⁻⁵ | 136 | 14,152 | 294,273 | 2,385,960 | 315 | 1.3x10 ⁻⁴ |
| III | 421 | 0.0993 | 4.35x10 ⁻⁵ | 40 | 22,944 | 110,080 | 917,818 | 191 | 2.07x10 ⁻⁴ |

s(Avg.) : 모듈의 평균 size, s_{max} : 최대 모듈 size, s_T : 전체 모듈 size, M : 모듈수, N : 결함수

D : 결함밀도(defect density), LOC : line of code, g=M/s_T, \hat{a} : 총고장수, \hat{b} : 결함발견율

전체 시스템에서 발생되고 있는 결함수에서 각 모듈 크기별로 차지하고 있는 결함수의 비율을 나타낸다.

데이터 그룹(data group) III은 FTTH(fiber to the home) 실현을 위해 광가입자망 구축의 핵심 장비인 OLT/ONU/ONT 시스템 개발에서 수집된 데이터로 시스템의 소프트웨어 규모는 약 92만 라인에 40개 소프트웨어 모듈(기능 블록)로 구성된 시스템이다.

모듈 당 크기(규모)는 평균 22,944 라인으로 데이터 그룹(data group) I, II에 비해 다소 큰 편이다. 또한 모듈 당 평균 결함수(Mean Value)는 18.28개, 1,250 라인 당 1개의 fault가 발생하는 것으로 추정되었다.

표 1, 3에 의해 관측된 모듈의 소스라인 규모(그림 1 참조)에 따른 소프트웨어 결함밀도 추세는 그림 2와 같으며, 소프트웨어 결함은 데이터 그룹(III)이 그룹(I, II)에 비해 낮은 것으로 판정되었다.

단위 소프트웨어 모듈크기는 10,000 라인 전후로 최소 수백 라인부터 30,000 라인이 넘는 경우로 파악되었으며, 이에 대한 세부 현황은 그림 1, 2와 같다.

3. 모델 parameter 추정 및 검토사항

표 5의 추정치는 문헌[7]에서 발표된 추정방법에 의거하여 5.1, 5.2항에 언급된 3개의 프로젝트에 대하여 분석한 결과이다.

3개의 parameter 추정 결과는 지수분포모델에 의한 방법을 이용하여 모듈의 크기를 추정한 결과 실제 소프트웨어 시스템 크기인 130만 라인 및 240만 라인과 거의 유사한 결과를 얻었다. 그러나 모듈크기 개선방법의 지수분포모델에 의한 결함수(N) 및 결함밀도(D) 추정치는 실제 경험데이터와는 많은 차이가 있다(낮게 평가됨). 즉, 정확한 결과를 얻기 위해서는 g-parameter에 대한 추정값과 식 (1)의 parameter c에 대한 값을 추가로 고려하여야 한다.

지수분포 모듈크기 분산인 경우를 가정하면 소프트웨어 결함밀도의 또 다른 변형식인 아래 식 (18)에 적용하여 추정하면 우선 c=0 적용시 data set(I, II)는 각각 3.17x10⁻⁴, 1.32x10⁻⁴로 추정되어 크게 차이나지 않음을 알 수 있다. 또한 문헌[7]의 모듈크기분포변수(module size distribution parameter)의 추정결과인 c=0.0097(Gnu C library : c-언어 기준)를 적용하는 경우 KLOC(kilo-line of code) 당 각각 (0.2462, 0.8298)로 추정되었다. 사용된 언어에 따른 또 다른 모델 파라미터인 c 값에 의한 비교결과는 표 6과 같다.

$$D = \int_1^{s_{max}} g^2 e^{-gs} (a/s + b + cs) \cdot 10^{-3} \cdot s \cdot ds \quad (18)$$

즉, 소프트웨어 결함밀도 추정치는 적용 모델별로 차

표 6. 소프트웨어 결함밀도(D) 비교

Table 6. Compare of the software fault density.

| Data set | Another parameter(/KLOC) | |
|----------|--------------------------|----------|
| | c=0 | c=0.0097 |
| I | 0.312 | 0.246 |
| II | 0.130 | 0.829 |
| III | 0.207 | 0.392 |

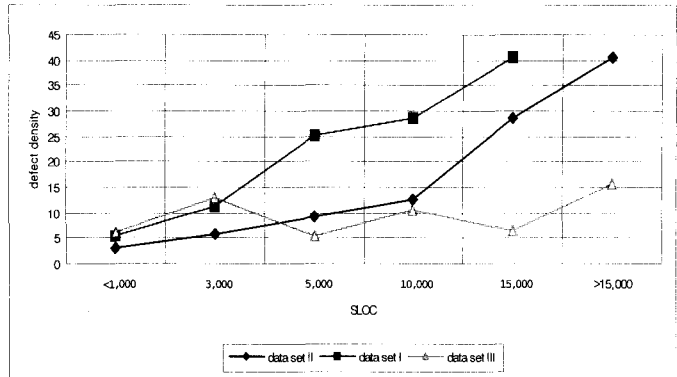


그림 2. 소스라인(SLOC) 대비 소프트웨어 결함밀도(data set I, II, III)
 Fig. 2. Source Line vs software fault density(data set I, II, III)

표 7. 데이터 그룹 III에 대한 적합도 판정 결과

Table 7. Result of goodness-of-fit data(III).

| parameter | estimation value | | |
|-------------------------|----------------------------|----------------------------------|------------------------------|
| mean | 6.4756 | | |
| variance | 8.02969 | | |
| variance MLE | 7.8472 | | |
| variance of sample mean | 0.182493 | | |
| shape report | skewness(왜도) : -1.33469 | quartile skewness : -0.155598 | kurtosis excess : 1.46224 |

이가 있으며, 다양한 모델 parameter 추정에 따라 변화된다. 실측 경험 데이터에 의존하여 개발된 소프트웨어를 정적분석(static analysis)한 결과와 비교해 볼 때 KLOC(kilo-line of code) 당 결함밀도가 2~3배 정도 차이를 보이는 경향을 보인다. 이와 같은 차이는 모듈의 크기가 적은 것(1,000 라인 이하)부터 매우 큰 모듈(수만 라인 이상)로 다양하게 분포되어 있어 적절한 소프트웨어 모듈 크기의 조정이 필요하다. 그러나 과거 경험 데이터를 분석한 결과 수행된 프로젝트의 성격상 소프트웨어 모듈의 크기 선정에는 큰 변화가 없는 것(실제 조사결과 일부 모듈을 제외하고는 module size가 커지면 결함 밀도도 약간씩 증가하는 경향을 띤다)으로 예상되며, 소프트웨어의 복잡도(complexity)에 따라 결함밀도가 좌우될 것으로 예측된다.

즉, 모듈크기에 따라서 대체적으로 결함이 증가하며, 시스템의 핵심기능과 타 기능과의 연동(interface) 부분이 많은 기능으로 소프트웨어 구현상에 복잡도가 높은 일부 분야에 많은 결함이 집중되는 결과 보이는 현상 등 이러한 연구결과는 기 발표된 바 있다.^[13,16]

이와 같은 상황을 종합해 볼 때 적용된 3개의 데이터 그룹으로부터 다음과 같은 결론을 얻을 수 있었다.

1) MD 모델의 parameter 추정법(推定法)을 개선하는

경우 결함밀도의 추정결과를 제공하는 데는 어느 정도의 한계가 있음

- 2) Laplace Newton 보간공식에 대한 근사(近似)는 결함함수의 형상을 가정하는 이점이 있는 반면에 결함밀도의 추정 정도(精度)에 관한 본질적인 개선이 필요함
- 3) 모듈크기의 분포 개선은 데이터 그룹(data set)의 추정결과에 대해 둔감하고 이런 물리적 의미로부터 절단 기하분포를 사용하여야 함
- 4) 소프트웨어 결함밀도 함수를 사용한 Project의 해석결과는 모듈크기 분포를 기준으로 할 때 기하분포, 지수분포, MD 모델 순으로 적합(실측 데이터에 근접)하다는 점

그 이외에 IV장의 경험적인 방법에 의해 얻어진 결과는 대체적으로 프로그램 크기(program size)에 따라 결함밀도가 증가하나 시스템 특성상 여러 기능과의 연동(interface)이 많은 소프트웨어 블록(high complexity or high cyclomatic)에 에러가 집중되는 현상을 보이며, 연동(interface)이 적은 경우에는 프로그램 크기에 거의 무관한 것으로 분석되었다. 다시 말해서 망 관리기능(NMF: Network Management Function)이나 HMI(human-machine interface) 기능 등 연동이 단순한 기

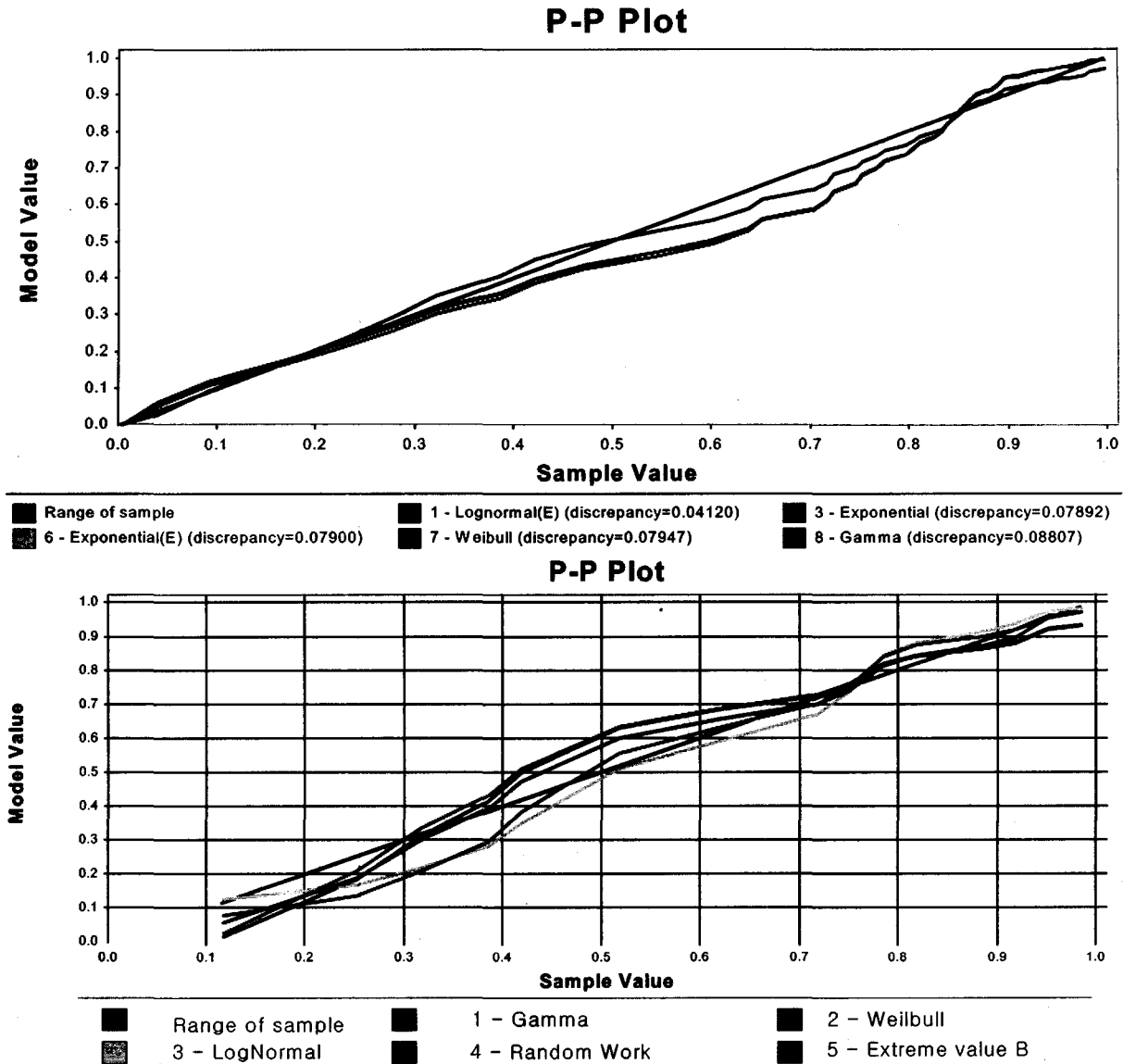


그림 3. Graphic 판별분석 방법[data set I(상), II(하)]
 Fig. 3. graphical comparisons method[data set I(upper), II(down)].

능인 경우에는 소프트웨어 컴포넌트의 프로그램 크기가 20,000 ~ 40,000 라인으로 정도인데 결함밀도는 매우 낮은 반면에 8,000 ~ 15,000 라인 전후의 시스템 운용보전 기능인 OAM(Operation and Administration, Maintenance) 기능 및 호 제어(call control & processing) 기능 관련 모듈은 소프트웨어 결함이 많이 발생하는 것으로 밝혀졌다. 가장 최근에 수행된 데이터 그룹 III은 주로 시스템 자원의 제어(system resource handling) 및 NP(Network Processor) Level의 Firmware 에러, 구성 관리(configuration management) 문제, L2 Bridge 기능 등에 많은 에러가 집중되고 있다.

본 논문에 적용, 분석된 결과를 고려해 볼 때 소프트웨어 모듈의 최적 크기(optimal size)는 8,500 line 전후

인 것으로 판정되었다.(그림 2 참조)

4 관측 데이터의 적합도 판정

그림 3은 5.1 및 5.2항에 적용된 관측데이터에 대한 적합도(goodness-of-fit) 판정을 위해서 손쉽게 적용하여 판단할 수 있는 적합도 판정中 그래픽 판별 분석법(graphical comparisons method)을 적용한 예이다. 즉, 데이터 그룹 I, II 의 관측 데이터에 대한 적합도 판별 방법으로서 기준선(x, y축의 0,1 사이의 청색 대각선)에 가깝게 데이터들이 포진하는 경우에 적합도(goodness-of-fit)가 뛰어난 것으로 판정하는 방법이다.

data 그룹 III에 대한 적합도 판정의 상세 분석 결과는 표 7과 같다. 이 데이터로 보아 모양은 대체적으로

기준선(0과 1사이의 대각선)의 아래쪽에 분포(왜도 데이터 참조)하고 첨도(kurtosis)는 약간 위쪽으로 분포하고 있음을 알 수 있다.

데이터의 편위(discrepancy)는 그룹 I(위쪽 그림)이 오른쪽, 그룹 II가 왼쪽으로 치우쳐 있으며, 그룹 II에 비해 그룹 I이 더 정확한 것으로 판정되었다. 그 이유는 관측된 데이터수의 규모가 크기 때문인 것으로 통계적 기법을 도입시 적용된 데이터가 많을수록 좀더 실측에 가까운 정확한 결과를 기대할 수 있기 때문이다.

VI. 결 론

본 논문에서는 Malaiya and denton 모델의 개선을 통한 소프트웨어 결함밀도의 정확성을 제한한 모델로 실제 시스템 개발 과정에서 관측된 결함데이터에 의한 결함밀도 추정의 정확성을 상호 비교하였다.

소프트웨어 결함밀도 추정결과는 MD 모델 중에서도 지수분포모델이 가장 실측데이터에 근접한 것으로 평가되었으며, 본 논문에서 고찰된 모델은 소프트웨어 결함밀도가 모듈크기에 의존하고 극히 단순한 모델로써 소프트웨어 규모를 단순히 코드수(혹은 라인수)로 계측하고 상이한 기술이나 환경에서 개발된 각 소프트웨어 모듈의 구조를 객관적으로 평가하는데 적용이 가능하다.

현재 소프트웨어 개발공수나 개발비용(cost), 개발계획(schedule)의 생산성 지표를 산출하는데 있어서 function point(FP) method에 기반한 시스템 규모 산출 방법이 많이 이용되고 있다. 과거에 일부 프로젝트에 대해 연구 발표한 사례[12,13]가 있으나 지속적인 연구가 필요하며, 향후 연구과제로는 FP법과 소프트웨어 결함밀도의 관계에 대한 다각적인 평가 모델의 검토가 요구된다.

본 연구에서는 프로젝트 수행시 소프트웨어의 개발 단위(unit or module) 및 관리(management)의 기본이 되는 소프트웨어 기능 블록(software functional block)을 기준으로 결함밀도를 추정하였으나 향후 소프트웨어 모듈 내의 각종 데이터(execution code 수, size, change code ratio, language 특성, complexity, defect counter 등 software metrics 정보 추출)를 분석하여 정확한 소프트웨어 결함밀도 추정에 반영하여야 한다.

참 고 문 헌

[1] 山野 茂, “ソフトウェア 信頼性モデル-基礎と應用,

日科技研, 東京, 1994.

- [2] 青山 榊雄, 中所 武司, 向山 博, “コンポーネントウェア”, 共立出版, 東京, 1998.
- [3] K. Kanoun, M. Kaaniche, and J.C. Laprie, “Qualitative and quantitative reliability assessment”, IEEE Software, vol. 14, no. 2, pp. 77-87, 1997.
- [4] J. M. Voas, “Maintaining component-based systems”, IEEE Software, vol. 15, no. 4, pp. 22-27, 1998.
- [5] J. C. Munson and T. M. Khoshgoftaar, “The detection of fault-prone programs”, IEEE Trans. Software Engineering, vol. 18, no. 5, pp. 423-433, 1992.
- [6] T. M. Khoshgoftaar and E. B. Allen, “Predicting fault prone software modules in embedded system with classification trees”, Proc. 4th IEEE Int'l High assurance system engineering symposium, pp. 105-112, 1999.
- [7] Yashwant K. Malaiya and Jason Denton, “Module size distribution and defect density”, Proc. 11th Int'l. symposium on software reliability engineering, pp. 62-71, IEEE CS press, Los Alamitos, 2000.
- [8] 유재년, 이재기, “기능 블록으로 구성된 대형 교환 소프트웨어의 신뢰도 성장”, 대한전자공학회논문지, 제35권 S편 제1호, pp. 29-38, Jan. 1998.
- [9] 이재기의 3, “대형 교환 소프트웨어의 복잡성과 고장분석 사례 연구”, 한국통신학회논문지, vol. 23, no.10c TC편, pp. 887-901, 2002.3.
- [10] V.R. Basili and B.R. Perricon, “Software errors and complexity”, Communication ACM, vol. 27, no. 1, pp. 45-52, 1984.
- [11] C. Withrow, “Error density and size in Ada software”, IEEE Software, vol. 7, no. 1, pp. 26-30, 1990.
- [12] 이재기의 3, “소프트웨어 프로젝트 평가모델을 통한 소프트웨어 Metrics 분석”, ETRI 전자통신동향분석지, 제 17권 5호, pp. 107-118, 2002.
- [13] 이재기의 3, “교환 소프트웨어 복잡도 연구”, ETRI 전자통신동향분석지, 제 17권 2호, pp. 49-60, 2002.
- [14] 宮川 治, 宍麻 喜弘, “サービス動作時におけるソフトウェアの信頼性評価”, 電子情報通信學會論文誌(日本), D-I Vol. J85-D-I No. 2, pp.202-209, 2002.2.
- [15] Philip J. Boland and Harshinder Singh, “A Birth-Process Approach to Moranda's Geometric Software-Reliability Model”, IEEE Trans. on Reliability, Vol. 52, No. 2, JUNE 2003.
- [16] 이재기의 3, “대형 교환 소프트웨어의 복잡성과 결함분석 사례 연구”, 한국통신학회논문지, Vol.

27, No. 10C, pp. 887-901, 2002.

[17] Michael R. Lyu, "Handbook of Software

Reliability Engineering", IEEE computer society press, McGraw-Hill, USA, pp. 98-99, 1996.

저 자 소 개



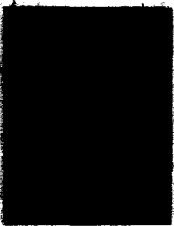
이재기(정회원)
1985년 서울산업대학교 전자공학과 졸업(공학사)
1989년 청주대학교 대학원 전자공학과 졸업(공학석사)
2004년 공주대학교 대학원 정보통신공학과 박사 수료

1983년~현재 한국전자통신연구원 책임연구원
<주관심분야: 소프트웨어 신뢰도, 시험 및 검증, 소프트웨어 품질 향상>



김창봉(정회원)
1983년 고려대학교 전자공학과 졸업(공학사)
1988년 Florida Tech 대학 전자공학과 졸업(공학석사)
1992년 Texas A&M 대학 전자공학과 졸업(공학박사)

1993년~현재 공주대학교 정보통신공학과 교수
2000년~현재 미국전기전자공학회(IEEE) Senior Member
<주관심분야: 광통신망, 광전송>



남상식(정회원)
1981년 단국대학교 전자공학과 졸업(공학사)
1983년 단국대학교 대학원 전자공학과 졸업(공학석사)
1999년 단국대학교 대학원 전자공학과 졸업(공학박사)

1985년~현재 한국전자통신연구원 책임연구원 (팀장)
<주관심분야: ATM Technology, Signal Integrity, NGN>