

NHPP형의 소프트웨어 신뢰도 최적 평가모델 도구

문 외 식[†] · 한 판 암^{††}

요 약

본 논문에서는 비동차 포아송 과정(NHPP)을 기초로 하는 신뢰도 성장모델들 중에서 특정의 테스트 데이터에 최적인 모델을 자동으로 선택하고 결과로서 신뢰성 평가척도를 나타내는 도구를 구현하였다. 구현된 도구로 소프트웨어 최적 신뢰성 평가척도(총기대 에러수, 에러발견율, 기대 잔존에러수, 신뢰도등)를 추정할 수 있다. 여기서 얻어진 신뢰성 평가척도에 의해 소프트웨어 개발 및 프로젝트 관리에 반영할 수 있다. 구현된 본 도구의 적합성을 검증해 보기 위해서 실측 에러 데이터를 사용하여 기존 연구 문헌과 비교, 분석 하였다.

The Optimal Evaluation Model Tool of NHPP Type Software Reliability

Wae-Sik Moon[†] · Pan-Am Han^{††}

ABSTRACT

In this paper, the optimal model for specific test data was selected automatically among software reliability growth models based on NHPP(Non Homogeneous Poisson Process), and in result the tool for the reliability estimating scales was implemented. With the implemented tool, software optimal reliability estimating scales(total expected number of errors, error detection rate, expected number of errors remaining in the software, reliability, etc) could be predicted. By the reliability estimating scales gained from this, software development and project management could be applied. In order to test the optimal of the implemented tool, the comparison with other paper and analization was done by using actual error data.

1. 서 론

소프트웨어 품질을 정량적으로 평가하는 방법으로 Jelinski와 Moranda[1]가 처음 제안한 소프트웨어 신뢰도 성장모델(SRGM: Software Reliability Growth Model)이 있다. SRGM[2]은 테스트에 의해서 발견된 총에러수와 소요된 테스트 시간과의 관계를 수학

적으로 표현한 것을 말한다. SRGM에서 도출되는 평가척도에 의해 소프트웨어 신뢰성 달성목표와 Release 되는 소프트웨어 제품의 품질 등을 추정할 수 있다.

그러나 제안된 다양한 SRGM 중에서 평가자가 모델 선택을 잘못했을 경우 평가척도의 신뢰성이 떨어지는 문제점이 발생될 수 있다.

본 논문에서는 수집된 에러 데이터에 적용 가능한 최적모델을 자동으로 선택하도록 도구를 구현함으로써 부적절한 모델에 대한 검토를 초기에 배제하도록 하였다. 또한 본 도구에 특정 데이터를 적용하여 얻어진 최적 평가척도들을 시각적으로 화면에 출력하

[†] 정희원: 창원전문대학 사무자동화과 부교수

^{††} 종신회원: 경남대학교 컴퓨터공학과 교수

논문접수: 1996년 11월 16일, 심사완료: 1997년 2월 15일

여 소프트웨어 개발시에 유용한 정보로 활용할 수 있다. 본 도구에 구현된 모델들은 NHPP를 기본가정으로 하는 지수형, 지연 S자형, 습속 S자형 신뢰도 성장 모델들에 한정하였다. 이와 같은 목적을 위해 본 논문의 2장에서는 신뢰도 관련연구로서 NHPP형 모델들의 수리적 개념을 고찰하고 3장에서는 NHPP모델의 파라미터 추정 및 신뢰성 데이터 분석과정을 정리하였다. 4장에서는 신뢰성 데이터 분석 과정을 기초로 하여 도구 설계 및 구현하였고 실측 데이터를 사용하여 기존에 연구되었던 논문들의 결과들과 비교 분석 하였으며 5장에서는 결론 및 향후 연구과제에 대해 기술하였다.

2. 관련 연구

2.1 NHPP형 SRGM

NHPP 모델은 일정한 시간 간격내에 발견되는 에러수를 비동차 포아송 과정에 의해 기술한 모델로서 다음과 같은 가정을 두고 있다.[3]

- ① 특정시각에 남아있는 에러는 소프트웨어 고장을 일으킬 수 있다.
- ② 임의의 테스트 시간 t까지 발견되는 총에러 수(에러누적수) N(t)는 NHPP를 따른다.

발견된 총에러수(N(t), t ≥ 0)는 위의 가정 ①, ②에 의해 식(1)과 같은 평균치함수 H(t)를 가지는 NHPP로 나타낼 수 있다.

$$Pr\{N(t)=n\} = \frac{\{H(t)\}^n}{n!} e^{-H(t)}, t \geq 0, (n=0, 1, 2, \dots) \tag{1}$$

식(1)에서 Pr{N(t)=n}은 테스트 시각 t까지 n개의 에러가 발견되는 에러발견 사상확률을 말하며 H(t)는 N(t)의 평균치를 나타내는 평균치 함수이다.

NHPP 모델에서 소프트웨어의 신뢰성 평가에 유용한 척도를 도출할 수 있다.

테스트 시각 t에서 소프트웨어 내에 잔존하는 기대 잔존 에러 수를 $\bar{N}(t)$ 라 할 때 $\bar{N}(t) = N(\infty) - N(t)$ 로 이때의 $\bar{N}(t)$ 의 평균치는 식(2)가 된다.

$$n_r(t) \equiv E[\bar{N}(t)] = E[N(\infty) - N(t)] = a - H(t) \tag{2}$$

식(2)에서 a는 테스트 시작전에 잔존하는 총기대 에러수를 나타낸다. 또한, 테스트가 시각 t까지 진행 하고 있을 때 시간구간 (t, t+x] (x ≥ 0)에서 에러가 발생하지 않을 확률은 식(3)이 되며 이를 소프트웨어 신뢰도라 한다.

$$R(x|t) = \exp[-\{H(t+x) - H(t)\}] (t \geq 0, x \geq 0) \tag{3}$$

다음은 제안된 NHPP형의 대표적 모델들이다.

2.2 지수형(exponential) SRGM

지수형 SRGM은 NHPP의 가장 기본적인 모델로서 Goel과 Okumoto[4]에 의해서 제안되었으며 단위 시간당 발견된 에러수는 그 시각의 잔존에러수에 비례하는 것으로 평균치 함수는 식(4)가 된다. 여기서 파라미터 b는 테스트시 발견되는 잔존하는 에러 발견율을 나타내는 정수이다.

$$H_e(t) = a(1 - e^{-bt}) (a, b > 0) \tag{4}$$

기대 잔존에러수와 신뢰도는 각각 (5)와 (6)이 된다.

$$n_r(t) = ae^{-bt} \tag{5}$$

$$R(x|t) = \exp[-e^{-bt} m(x)] \tag{6}$$

2.3 지연 S자형(delayed S-shaped) SRGM

지연 S자형 SRGM은 Yamada[5]에 의해 제안되었으며 고장의 발생 시각으로부터 그 고장의 원인인 에러를 제거하는 시각까지의 시간적인 지연을 고려하고 있다. 평균치 함수는 식(7)이 된다.

$$H_d(t) = a[1 - (1 + bt)e^{-bt}] (a, b > 0) \tag{7}$$

기대잔존 에러수와 신뢰도는 각각 식(8)과 식(9)가 된다.

$$n_r(t) = a(1 - bt)e^{-bt} \tag{8}$$

$$R(x|t) = \exp[-a\{(1 + bt)e^{-bt} - (1 + b(t+x))e^{-b(t+x)}\}] \tag{9}$$

2.4 습속 S자형(inflexion S-shaped) SRGM

습속 S자형 SRGM은 Ohab[6]에 의해서 제안되었

으며 테스트 팀의 소프트웨어에 대한 습속도를 고려한 모델이다. 평균치 함수는 식(10)이되며 여기서 파라미터 c 는 에러발견 능력에 대한 습속도(inflexion rate)이다.

$$H_i(t) = \frac{a(1 - e^{-bt})}{(1 + ce^{-bt})} \quad (10)$$

기대잔존 에러 수와 신뢰도는 각각 식(11)과 식(12)가 된다.

$$n_r(t) = \frac{a(1+c)e^{-bt}}{(1+ce^{-bt})} \quad (11)$$

$$R(x|t) = \exp\left[-\frac{a(1+c)e^{-bt}(1-e^{-bx})}{[1-ce^{-b(t+x)}](1+ce^{-bt})}\right] \quad (12)$$

3. 파라미터 추정 및 신뢰성 데이터 분석

3.1 파라미터 추정

NHPP 모델의 파라미터를 최우법으로 추정하며 이 때 일정한 테스트 시간간격 $(0, t_k)$ 에서 총 에러수 y_k 에 관한 n 조의 측정 데이터가 관측되는 것으로 한다. 이 데이터에 대한 평균치 함수 $H(t)$ 를 가지는 NHPP 모델의 우도함수는 식(13)이 된다.

$$L = \Pr\{N(t_1) = y_1, N(t_2) = y_2, \dots, N(t_n) = y_n\} \\ = \exp[-H(t_n)] \prod_{k=1}^n \frac{\{H(t_k) - H(t_{k-1})\}^{(y_k - y_{k-1})}}{(y_k - y_{k-1})!} \quad (13)$$

식(13)에서 $t_0 = 0, y_0 = 0$ 으로 초기 값을 주고 양변에 자연대수를 취하면 식(14)의 대수우도 함수가 된다.

$$\ln L = \sum_{k=1}^n (y_k - y_{k-1}) \ln [H(t_k) - H(t_{k-1})] \\ - H(t_n) - \sum_{k=1}^n \ln \{y_k - y_{k-1}\}! \quad (14)$$

식(14)의 $H(t)$ 에 식(4), (7), (10)의 평균치 함수를 대입하고 각 모델의 파라미터에 대해 편미분 하면 각 모델의 파라미터 최우추정치가 구해진다.[7]

3.2 신뢰성 데이터 분석

평균치 함수를 $H(t)$ 로하는 NHPP 모델을 테스트 과정에서 얻은 실측 데이터에 적용하여 최적모델 선

택 및 신뢰도 평가 척도를 추정하기 위해 다음과 같은 순서로 신뢰성 데이터를 분석한다.[8][9][10][11]

3.2.1 데이터 수집 및 입력

데이터 분석에 충분한 실측 데이터인 에러발견수를 수집하여 입력한다.

3.2.2 파라미터 추정

실측 데이터에 대응하는 우도 방정식을 풀어 각 모델 파라미터를 추정한다.

3.2.3 평균치함수 $H(t)$ 추정

평균치 함수 $H(t)$ 의 최우 추정치 $\hat{H}(t)$ 에서 식(15)를 얻는다.

$$P_r\{N(t) = n\} = \frac{\{\hat{H}(t)\}^n}{n!} \exp[-\hat{H}(t)] \quad (n=0, 1, 2, \dots) \quad (15)$$

$\hat{H}(t)$ 의 확률 α 에서 추정되는 평균치 함수의 존재범위인 신뢰한계를 $\hat{H}(t) \pm K_\alpha \sqrt{\hat{H}(t)}$ 에 의해 계산해 둔다.

3.2.4 적합도 검정

평균치함수 추정이 관측된 데이터에 적합한가를 통계적으로 판정하기 위하여 적합도 검정을 한다. 본 논문에서는 콜모루그로브 스미르노(Kolmogorov Smirnov) 적합도 검정법(K-S 검정법)을 적용하였다. 여기서 에러 발견수 데이터가 관측될 때에는 식(16)의 K-S 검정 통계량을 계산한다.

$$\begin{cases} D = \max_{1 \leq i \leq n} \{D_i\} \\ D_i = \max \left\{ \left| \frac{H(t_i)}{H(t_n)} - \frac{y_i}{y_n} \right|, \left| \frac{H(t_i)}{H(t_n)} - \frac{y_{i-1}}{y_n} \right| \right\} \end{cases} \quad (16)$$

K-S 검정법에서는 데이터로부터 계산된 검정 통계량 D 와 위험율이 α (1% 또는 5%)일 때의 기각한계와 비교된다. 식(16)에 대해서 자유도가 n 일 때의 기각한계 $D_{n;\alpha}$ 와 비교하여 식(17)의 조건을 만족하면 위험률 α 에서 관측 데이터에 대해 식(15)의 NHPP 모델은 적합하다고 판정한다.

$$D < D_{n;\alpha} \quad (17)$$

3.2.5 모델 선택

적합도 판정결과 적합한 모델이 하나 이상이면 보다 나은 모델 선택을 위해서 편차자승합에 의한 모델 선정을 한다. 이때 편차자승합 값이 최소인 값을 가지는 모델이 실제 관측된 데이터에 보다 더 적합한 최적 모델로 결정된다. 편차자승합은 테스트에 의해서 발견된 총 에러수의 관측 데이터 y_k 와 그 추정 데이터 \hat{y}_k 와의 편차자승합 식(18)을 사용한다.

$$S = \sum_{k=1}^n (y - \hat{y}_k)^2 \tag{18}$$

이상의 신뢰성 데이터 분석과정에 의해 기대잔존 에러수 $n_r(t)$, 소프트웨어 신뢰도 $R(x|t)$ 등의 신뢰성 평가척도를 추정 및 예측할 수 있다.

4. 시스템 설계 및 구현

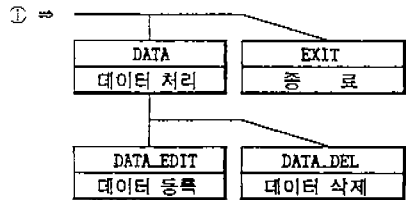
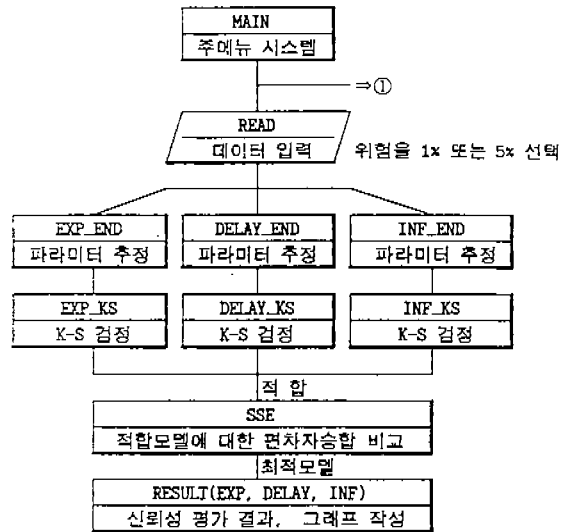
소프트웨어 신뢰성을 평가하고 개발관리를 지원하는 상업용 도구로는 일본 NEC에서 개발한 SOREM 등 몇가지가 있다[12][13]. 기존 도구들 대부분은 대형의 프로젝트에 적합하고 마이크로 컴퓨터급 이상에서 실행할 수 있도록 되어 있으며 소프트웨어 개발자가 임의로 모델을 선택하여 실행하도록 되어 있다.

본 논문에서는 자동으로 최적모델을 선택한후 신뢰성 데이터를 분석하여 평가척도를 나타내게 하여 프로젝트 관리에 반영할수 있도록 하였다. 본 도구 작성에 사용된 언어는 MS-DOS상에서 실행 가능한 Turbo-C이다.

4.1 시스템 구조

NHPP형 신뢰도 평가도구의 구조는 구조화 프로그래밍의 기법을 사용하였다. 또한 데이터 분석 과정의 각 단계를 모듈화 하였다. 그리고 각 모듈간의 변수들의 교환을 금지하여 모듈간의 부정확한 인수값 전달을 방지하였다. 모든 모듈은 반드시 주 모듈로부터 호출하도록 설계하였으며 각 모듈들을 항상 메모리에 상주시켜 실행속도를 증가시켰다. 실행결과에 대한 모든 출력은 화면에 나타내도록 하였다.

(그림 1)은 3.2의 과정을 기초로 한 도구의 전체 구조도이며 주메뉴(초기화면)를 나타내는 MAIN 모듈에서 시작한다.



(그림 1) 신뢰도평가 도구 시스템 구조도 (Fig 1) System structure of reliability evaluation tool

4.2 모듈설계 내용

(그림 1)의 시스템 구조도에 나타낸 각각의 모듈들에 대한 설계내용은 <표 1>과 같다.

<표 1> 각 모듈 설계내용 (Table 1) Design contents of each module

모 들	내 용
MAIN	초기메뉴 시스템으로 FULL-DOWN 메뉴방식으로 설계하였다. 키보드의 화살표 키 및 마우스로 선택하여 각각의 모듈로 제어를 넘긴다.
READ	디스크에 입력된 테스트 데이터 선택 모듈이다. 데이터 파일명은 *.DAT 이다. 또한 입력 데이터의 위험을 1% 또는 5%를 임의로 선택한다.
EXP_END	지수형 SRGM의 파라미터 추정(최우법) 모듈. 파라미터의 최우추정은 이분법(bisection method)을 사용하였다.

DELAY_END	지연 S자형 SRGM의 파라미터 추정(최우법) 모듈. 파라미터의 최우추정은 이분법을 사용하였다.
INF_END	습속 S자형 SRGM의 파라미터 추정(최우법) 모듈. 파라미터의 최우추정은 이분법을 사용하였다.
EXP_KS	지수형 SRGM의 적합도 검정 모듈. K-S검정표와 비교하여 1% 또는 5% 기각한계에 대한 통계량을 계산한다.
DELAY_KS	지연 S자형 SRGM의 적합도 검정 모듈. K-S검정표와 비교하여 1% 또는 5% 기각한계에 대한 통계량을 계산한다.
INF_KS	습속 S자형 SRGM의 적합도 검정 모듈. K-S검정표와 비교하여 1% 또는 5% 기각한계에 대한 통계량을 계산한다.
SSE	각모델의 K-S 검정 통계량 결과 적합한 모델에 대해 편차자승합 계산 및 비교하는 모듈.
RESULT_GRAPH	선택된 최적모델의 평균치함수, 기대잔존 에러 수 및 소프트웨어 신뢰도, K-S 검정량, 편차자승합 결과값을 출력 및 그래프로 작성하는 모듈.
DATA	테스트 데이터(에러발견수)처리 모듈.
DATA_EDIT	테스트 데이터 등록 모듈.
DATA_DELETE	테스트 데이터 삭제 모듈.
EXIT	작업 루틴 종료 DOS 상태로 전환 모듈.

4.3 최적 모델 선택 알고리즘

최적 SRGM 선택 및 신뢰도 평가척도를 나타내기 위한 각각의 주요 알고리즘은 다음과 같다.

4.3.1 파라미터 값 추정 알고리즘

NHPP형 SRGM의 파라미터값 a, b를 추정하는 알고리즘은 다음과 같다. 에러 발견을(b)이 하한값일 때 (b₁)와 상한값(b₂)일때와 중간값($\frac{b_1+b_2}{2}$)일때의 우도방정식 차이값을 에러발견을 상대오차가 10⁻⁶이 될때까지 이분법을 수행한후 exp_end()를 실행하여 최종 파라미터값 \hat{a}, \hat{b} 를 구한다.

```
int parameter(int n, M) /* n: 에러 발견수 데이터 개수, M[n]: 에러 발견수 데이터 */
{
    double b1, b2;
    double b3;
    double rel_b1, rel_b2, rel_b3;
```

```
b1 = 0.001;
b2 = 0.9999;
for(g = 1; g < 81; g++)
{
    b = b1;
    rel_b1 = b_function(b); /* 에러발견을 b1일때의 우도방정식값 */
    b = b2;
    rel_b2 = b_function(b); /* 에러발견을 b2일때의 우도방정식값 */
    b3 = (b1 + b2) / 2;
    rel_b3 = b_function(b); /* 에러발견을 b3일때의 우도방정식값 */
    if(fabs(b1 - b2) / fabs(b2) < 0.000001)
        exp_end(b3); /* 에러발견을 상대오차 10-6이 2분법 종료 및 최종파라미터 추정 */
    if((rel_b1 * rel_b3) < 0)
        b2 = b3;
    else
        b1 = b3;
}
}
```

```
double b_function(double b_imsy)
{
    double r_over, r_under, l_side;
    double first, diff;
    int k;
    first = m[1] * t[1] * exp(-b_imsy * t[1]) / (1 - exp(-b_imsy * t[1]));
    diff = 0;
    for(k = 2; k <= n; k++)
    {
        r_over = (m[k] - m[k-1]) * (t[k] * exp(-b_imsy * t[k]) - t[k-1] * exp(-b_imsy * t[k-1]));
        r_under = exp(-b_imsy * t[k-1]) - exp(-b_imsy * t[k]);
        diff = diff + r_over / r_under;
    } /* 우도방정식 우측값 */
    l_side = m[n] * t[n] * exp(-b_imsy * t[n]) / (1 - exp(-b_imsy * t[n])); /* 우도방정식 좌측값 */
    diff = first + diff - l_side;
    return(diff);
}
```

```

}
/* 파라미터 a, b값 계산 */
double exp_end(double b_imsy2)
{
    double ass_a, ass_b;
    ass_b = b_imsy2;
    ass_a = m[n]/(1-exp(-ass_b*t[n]));
}

```

4.3.2 적합도 검정 알고리즘

각모델의 위험을 1% 또는 5% 때의 적합도를 검정하기 위한 K-S검정 통계량 계산 알고리즘은 다음과 같다. 앞에서 구한 파라미터 추정값 a, b를 받아서 평균치함수를 구한후 통계량식에 의해 $D_i(1 \leq i \leq n)$ 값을 구하며 D_i 값중 최대값을 검정 통계량 DMAX로 하며 자유도가 n일 때 기각한계 $D_{\alpha, n}$ 와 비교하여 작으면 적합하다.

```

for(g=1;g<max-1;g++){
    d[g]=max(fabs(h[g]/h[max-2]-m[g]/m[max-2]),
             fabs(h[g]/h[max-2]-m[g-1]/m[max-2]));
}
for(g=1;g<max;g++){
    if(dmax<d[g]){
        dmax=d[g];
    }
}
/* 위험을 1%일 때의 기각한계값 */
if(damage==1){
    fff2=fopen("ks_d1.dat", "r");
    for(g=1;g<max-1;g++){
        fscanf(fff2, "%f", &ks_d[g-1]);
    }
}
/* 위험을 5%일 때의 기각한계값 */
else {
    damage=5;
    fff2=fopen("ks_d5.dat", "r");
    for(g=1;g<max-1;g++){
        fscanf(fff2, "%f", &ks_d[g-1]);
    }
}

```

```

}
if(dmax <= ks_d[max-3])
/* 이 모델은 유의수준 1%(5%)에서 데이터에 적합하다. */
else
/* 이 모델은 유의수준 1%(5%)에서 데이터에 적합하지 않다. */

```

4.3.3 최적 모델 선택 알고리즘

최적모델 선택을 위한 알고리즘은 다음과 같다. 각 모델에 대한 적합도 검정 결과 적합한 모델에 한해서 편차자승합을 계산하고 비교한 후 최소값을 가지는 모델을 최적모델로 선택하고 이 모델에 대한 파라미터값을 출력한다.

```

/* 편차자승합 계산 */
for(i=0;i<max-1;i++){
    sse_exp = sse_exp + pow(m[i]-expl_a*(1-exp(-expl_b*i)), 2);
}
sse = sse_exp; /* 편차자승합 초기값 설정 */
model = model1; /* 모델명 초기값 설정 */
if(sse > sse_del) /* 각 모델간 편차자승합 비교 */
    sse = sse_del; /* 최소값을 가지는 모델의 편차자승합과 모델명 저장 */
    model = model2;
else if(sse > sse_inf) {
    sse = sse_inf;
    model = model3;
}
/* 선택된 모델 출력 */
if(sse == sse_exp) /* 지수형 SRGM일 때의 파라미터값을 그래프 작성 파라미터에 할당 */
    graph_a = expl_a;
    graph_b = expl_b;
    graph_c = 0;
    graph_sse = sse_exp;
    graph_dmax = dmax_exp;
} else if(sse == sse_del) /* 지연S자형 SRGM일 때의 파라미터값을 그래프 작성 파라미터에 할당 */

```

```

graph_a = del_a;
graph_b = del_b;
graph_c = 0;
graph_sse = sse_del;
graph_dmax = dmax_del;
} else { /* 습속S자형 SRGM일 때의 파라미터값을
그래프 작성 파라미터에 할당 */
graph_a = inf_a;
graph_b = inf_b;
graph_c = inf_c;
graph_sse = sse_inf;
graph_dmax = dmax_inf;
/* 그래프 작성 루틴 호출 */
graph(graph_a, graph_b, graph_c, graph_sse, graph_
dmax, nSel, ret, model);

```

4.4 신뢰성 데이터 사례에 의한 도구실행

구현된 본 도구를 이용하여 <표 2>의 데이터로 작업시작에서 종료까지의 결과를 나타내고 문헌[14]에서 얻은 결과와 비교 및 분석한다.

<표 2> 에러발견수 데이터
<Table 2> Number of detected error data

테스트시각 t_k (일별)	발견된 총에러개수	테스트시각 t_k (일별)	발견된 총에러개수
1	0	15	90
2	7	16	95
3	8	17	96
4	9	18	102
5	15	19	111
6	34	20	117
7	40	21	119
8	48	22	125
9	51	23	135
10	59	24	137
11	64	25	138
12	70	26	141
13	78	27	142
14	83		

<표 2>에 나타난 테스트 데이터는 참고문헌[14]에서 사용된 데이터로 외국증권 회사의 온라인 프로그램 수와 배치 프로그램 수가 각각 50 및 150인 대규모 시스템으로 이 시스템의 통합 테스트에서 관측된 27조의 에러발견수 데이터이다.

4.4.1 시스템 초기화면

도구를 실행시키면 주메뉴 시스템인 MAIN 모듈의 실행으로 다음과 같은 초기화면이 나타난다. 초기화면은 도움말, 데이터 처리, 종료의 3가지 메뉴로 구성되어 있다.



(그림 2) 소프트웨어 신뢰성 평가도구 초기화면
(Fig 2) Initial Display of reliability evaluation tool

4.4.2 데이터 입력

초기화면 상태에서 커서 또는 마우스로 데이터처리 메뉴의 서브메뉴인 데이터 입력을 선택하면 (그림 3)과 같은 테스트 데이터 선택메뉴가 나타난다. 이때 화면에 나타나는 테스트 데이터들은 이미 디스크에 저장된 실측 에러발견수 데이터들이다.



(그림 3) 테스트 데이터 선택메뉴
(Fig 3) Option menu of test data

(그림 3)의 화면상태에서 커서 및 마우스로 특정 데이터를 선택한다. 본 논문에서는 <표 2>의 테스트 데이터인 MHS27.DAT를 선택한다. 이후 화면에는 선택한 파일명이 맞는지 확인하는 메시지를 출력하며 맞으면 <ENTER>키를 누른다.

4.4.3 위험율 선택

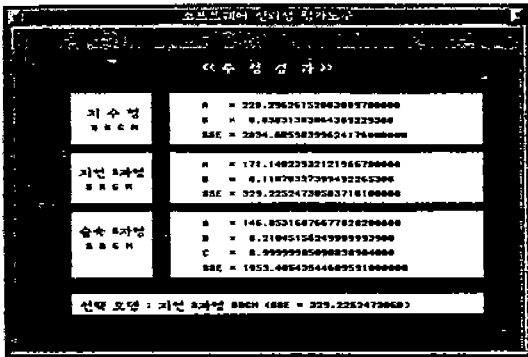
테스트 데이터 선택과정이 끝나면 추정모델의 적합도 검정 계산을 위한 위험율 1% 또는 5%를 선택한다. 본 논문에서는 5%를 선택하였다.



(그림 4) 위험율 선택 메뉴
(Fig 4) Option menu of hazard rate

4.4.4 최적모델 선택을 위한 계산 및 결과

테스트 데이터 입력 및 위험을 선택이 끝나면 NHPP 형 각 모델에 대한 추정 파라미터값 계산, 적합도검정 계산, 그리고 적합한 모델에 대한 편차자승합을 구한 후 최적모델을 출력한다. (그림 5)는 계산된 지수형, 지연S자형, 습속S자형 모델들 각각의 파라미터값, 편차자승합값을 나타내고, 그리고 입력된 테스트 데이터(MHS27.DAT)에 대한 최적모델을 나타내고 있다. 이때 최적모델은 편차자승합이 최소인 지연 S 자형 모델임을 알 수 있다.



(그림 5) 각 모델에 대한 추정 결과 및 최적모델 (Fig 5) Result of prediction about each models and optimal model

4.4.5 신뢰도 평가결과 출력

최적 모델이 선택되면 적용된 테스트 데이터에 대한 신뢰도 평가척도 및 그래프 표시를 위해 (그림 6) 처럼 평가결과 선택메뉴를 나타낸다.

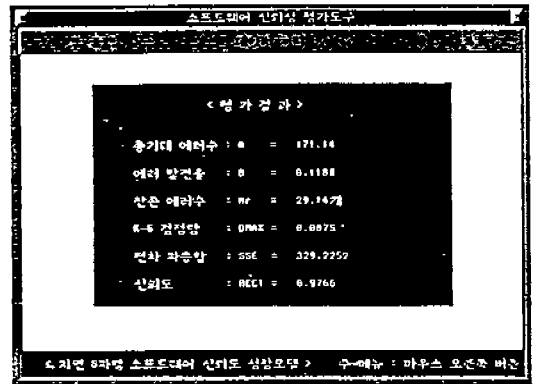


(그림 6) 평가결과 출력메뉴 (Fig 6) Output menu of evaluation result

(그림 7)은 (그림 6)의 메뉴에서 서브메뉴인 결과를 선택한 경우에 출력되는 평가결과의 내용 들이다.

4.4.6 평가결과를 그래프로 출력

(그림 6) 메뉴에서 그래프 메뉴를 선택하면 (그림



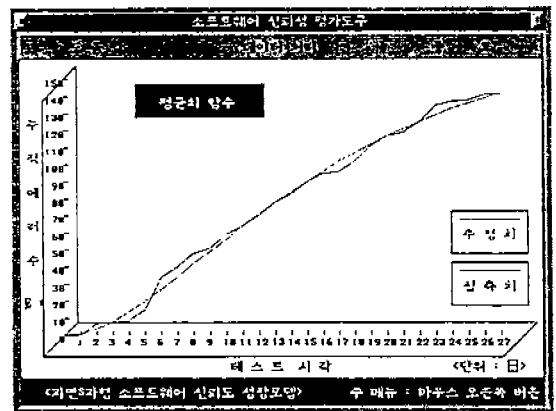
(그림 7) 신뢰성 평가결과 (Fig 7) Result of reliability evaluation

8)과 같은 평가결과에 대한 평균치함수, 잔존예러수, 신뢰도를 그래프로 나타낼 수 있는 서브메뉴가 나타나며 이때 하나를 선택할 수 있다.



(그림 8) 평가척도 그래프작성 메뉴 (Fig 8) Graph menu of evaluation scale

(그림 9)는 위의 서브메뉴에서 평균치함수 메뉴를 선택했을 때의 그래프이다.



(그림 9) 평균치함수 그래프 출력 (Fig 9) Graph output of mean value function

4.4.7 신뢰성 데이터 평가 분석 및 비교

신뢰도 평가도구를 <표 2>의 테스트 데이터를 적용하여 분석한 결과 3가지의 NHPP형 모델 중에서 편차자승합이 가장 적은 지연 S자형 SRGM이 최적모델로 판정이 났다. 기존의 연구 및 상업용 평가도구에서는 이러한 최적모델을 선정하기 위해서는 숙련된 평가자가 신뢰성 데이터 분석과정을 정확히 파악해야 하는 번거로움이 뒤따랐다. 27일간의 테스트 데이터를 선택된 최적모델에 적용한 후의 신뢰성 평가 결과를 보면 테스트 시작 전에 잠재하는 총기 대 에러수가 171개이고 잔존 에러 발견율은 0.1188이며, 이때 테스트에 의해서 발견되지 않는 기대잔존 에러수는 29개임을 추정할 수 있다. 또한, 신뢰도는 테스트 시각 $t=28$ 일째 0.9766으로 약 98%로 추정된다. 이와 같은 결과는 문헌[14]에서 발표한 연구와 일치함을 알 수 있으며 일부 평가척도들의 값들은 보다 더 정밀하게 나타났다. 또한 <표 2>의 데이터 이외에도 문헌 [15]에서 인용한 실측 데이터(10일간의 테스트후 발견된 에러 데이터)를 본 평가도구에 적용한 결과 문헌 [15]에서 연구 발표한 결과와도 일치 했다. 특히 문헌 [11]에서 제시된 59주간의 비교적 많은 실측데이터를 본도구에 적용한 결과에서는 59주째의 실측데이터에 대한 실제 누적에러수와 추정 누적에러수가 일치 (5186개)하는 정밀성을 나타내어 문헌[11]에서 나타낸 결과보다 우수함을 알 수 있었다.

5. 결론 및 향후연구과제

국내에서 중·소 규모의 프로젝트 신뢰성 평가 및 연구에 사용되는 소프트웨어 신뢰도 평가 도구가 제대로 없는 실정에서 구현된 본 시스템은 PC급에서 중·소 규모의 프로젝트에 쉽게 적용할 수 있다. 또한 복잡한 실측 데이터 분석과정을 상세히 파악하지 않더라도 도구에 간단히 데이터를 입력하여 실측 데이터에 대한 최적 신뢰성 평가를 화면에 문자, 그래프 등 시각적인 출력을 하도록 구현하였다. 구현된 도구의 검증에 위해 사용된 실측 데이터는 데이터 수집의 어려움으로 기존의 연구논문에서 인용한 몇가지의 데이터를 이용하여 적용·검증하였다. 본 논문에서 분석한 결과로서 특정 테스트 데이터에 대한 최적모델을 정확히 선택해 주고 신뢰성 있는 평가 결과를 나

타내 주는 우수함을 알 수 있었다. 특히 데이터량이 어느정도 많을수록 본 도구의 신뢰성 평가 결과가 높은 것을 알 수 있었다. 본 도구의 신뢰성을 계속하여 검증해 보기 위해서 많은 실측 데이터 수집이 필요하다. 본 논문에서는 제안된 NHPP형 모델중 대표적인 3가지 모델에 대해서만 한정시켜 구현했다. 향후 연구과제로는 제안된 나머지 모델 및 통계적 데이터 해석모델도 함께 추가 구현이 필요하다.

참 고 문 헌

- [1] Jelinski, Z. and Moranda, P.B, "Software reliability research", in Statistical Computer Performance Evaluation, Freiberger, W.(ed.), pp. 465-484, Academic Press, 1972.
- [2] Capers Johnes, "Applied Software Measurement: Assuring Productivity Quality", pp.279~338., 1991.
- [3] Goel, A. L, "Software reliability Models: Assumption, limitations, and applicability", IEEE Trans. Software Engineering, Vol.SE-11, No.12, pp.1411-1423, 1985.
- [4] Goel, A.L. and Okumoto, K, "Time dependent error detection rate model for Software Reliability and other performance measures", IEEE Trans. Reliability, Vol.R-28, No.3, pp.206-211, 1979.
- [5] Yamada, S., Ohba, M.and Osaki, S, "S shaped reliability growth modeling for Software error detection", IEEE Trans. Reliability, Vol.R-32, No.5, pp.475-484, 1983.
- [6] 山田茂, 大寺浩志, "ソフトウェアの信頼性~理論と實踐的應用~, ソフト.リサーチ・センター, 1990.
- [7] 高宗雄, "ソフトウェア信頼度成長モデルに基づく定量的品質評價法", 九州大學學位論文, 1989.
- [8] Xie, M, "Software Reliability Modeling", World Scientific, 1991.
- [9] Musa, J. D., Iannino, A.and Okumoto, K, "Software Reliability: Measurement, Prediction, Application", Mc graw Hill, 1987.
- [10] 山田茂, 大寺浩志, "ソフトウェアの信頼性~理

論と實踐的應用~”, ソフト・リサーチ・センター, 1990.

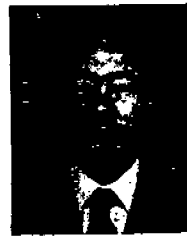
- [11] 山田茂, “ソフトウェア信頼性評価技術”, HBJ出版局, 1989.
- [12] 大村茂和, “ホスト適用業務プログラムにおけるソフトウェア信頼性評価”, ソフトウェア・シンポジウム '91 論文集, pp. B-26-B-34, 1991.
- [13] 三角武: “ソフトウェアの品質評価法”, 日科技連出版社, 東京 1981.
- [14] M. Ohba, “Software quality=Test coverage × Test accuracy” Proc. 6th Int. Conf. Software Engineering, pp. 287-293., 1982.
- [15] Ohba, M., “Inflection S shaped software Reliability growth model in stochastic models in reliability theory, Osaki, S and Hatayama, Y”, pp.144-162, Springer-Verlag, Berl in, 1984.



문 의 식

- 1980년 울산대학교 공과대학 전자계산학과 졸업(공학사)
- 1985년 부산대학교 산업대학원 전자계산학과 졸업(공학석사)
- 1995년 경남대학교 컴퓨터공학과(박사과정 수료)

1980년~1984년 한국전력공사 전자계산소근무
1984년~현재 창원전문대학 사무자동화과 부교수
관심분야: 소프트웨어 공학(특히 소프트웨어 품질관리 및 신뢰성 평가)



한 판 암

- 1969년 동국대학교 졸업
- 1975년 동국대학교 경영대학원 졸업(경영학석사)
- 1989년 명지대학교 대학원 졸업(공학석사)
- 1992년 인천대학교 대학원 졸업(경영학박사)

1980년~현재 경남대학교 공과대학 컴퓨터공학과 교수
관심분야: 소프트웨어 품질관리 및 신뢰성, 소프트웨어 개발환경, 정보공학