제18회 한국정보처리학회 추계학술발표대회 논문집 제9권 제2호 (2002.11)

소프트웨어 신뢰도모델에서의 상관고장

최규식*, 장원석*
*건양대학교 IT학부
e-mail:che@konyang.ac.kr

Failure Correlation in Software Reliability Models

Che Gyu Shik*, Chang Won Seok*
*IT Division in Konyang University

요 약

대부분의 소프트웨어 신뢰도 모델에서 제한사항중에서 계속되는 소프트웨어의 고장중에서 통계적인 독립성의 가정이 가장 엄격할 것으로 연구해왔다. 우리는 이러한 가정이 쉽게 위배될 수 있는 실제적인 상황이 존재하는가, 그리고 소프트웨어 모델링에 관하여 발행된 대부분의 문헌들이 이러한 문제를 심각하게 다루고 있었는가에 관하여 연구하고자 하였다. 그리고, 고전적인 소프트웨어 신뢰도 이론이 고장상관관계의 가능한 시퀀스를 고려하는데까지 확장될 수 있음을 보여주고자 한다.

1. 서론

대부분의 기존 신뢰도성장모델은 테스트가 동차(homogeneous)로, 그리고 무작위적으로 수행된다고 가정한다. 즉, 테스트 데이타는 어떤 무작위 메카니즘에 의한 입력공간으로부터 선택하고, 또 소프트웨어는 이러한 데이터 가정 동차조건을 사용하여 테스트한다. 그러나, 이것은 비현실적이다. 테스트 단계 기간동안 고도의 기능성에 따라 여러 가지 테스트시나리오를 그룹으로 묶는다. 그 외에도 입력데이타는 테스트의 효율을 증가시키기 위해서 선택한다. 즉, 될 수 있는 한 많은 결함을 검출하기 위한 것이다.

성공적으로 소프트웨어를 운영하는 중에 s- 독립을 가정하는 것은 많은 경우에 있어서 적절하지 못하다.

2. 마코프 갱신공정

아래와 같이 구성된 공정을 고찰해본다.

1) 천이확률마트릭스 $P=[p_{k,l}]$를 가진 m-상태 DTMC를 취한다.

2) 상태 k에서 상태 l로 천이되는 시간들이 s-독립이 되도록 s-Cdf $F_{k,l}(t)$로 하여 연속시간 공정을 구성한다.

이는 신뢰도모델링 공정이며, 계수가능 상태공간에서 연속시간 마코프 공정 및 이산시간 마코프공정 모두에 일반적으로 적용된다.

3. 소프트웨어 신뢰도 모델링 구조

직관적인 SRP에 근거를 둔 소프트웨어 신뢰도 모델링 구조를 개발하였다. 각각의 소프트웨어 수행이 성공이냐 실패냐의 두가지 가능한 출력상태를 가지므로 소프트웨어 수행시퀀스를 관찰하는 일반적인 방법은 그것을 베르누이시행의 s-독립 시퀀스로 간주하는 것이다. 여기에서는 각각의 시행이 성공확률 p와 실패확률 $q=1-p$를 가진다. 두 개의 Cdf 즉 이항 및 기하급수는 s-독립 베르누이시행으로 연결된다.

소프트웨어 운영회수 n이 큰 값이고 q가 작은 값이므로 이러한 *pmf*에 대한 잘 알려진 제한결과는 SRM에서 가장 기본적인 기준으로 쓰는 것이 일상적인 것이다. 어떠한 운전에서건 고장이 발생되면, 근원적인 결함을 제거하여 고장확률을 변경시킬 수 있다. 우리의 목표는 고전적인 소프트웨어 신뢰도이론이 s-종속 소프트웨어 운영을 고려할 만큼 확장될 수 있다는 것을 보여주는 것이다.

연속적인 소프트웨어운영의 시퀀스(성공 또는 실패)는 포인트이벤트를 실현하여 보일 수 있다. 가장 단순한 포인트공정에서의 포아송공정은 고장포인트만을 고려한다. 즉,

1) 두 고장간의 성공적인 운영을 고려하지 않는다.

2) 그들에의해서전달되는 정보를 무시한다.

본 논문에서는 각각의 운영 결과가 s-종속 베르누이 시행의 시퀀스로서 이전 운영의 결과에 의존할 때 s- 종속 소프트웨어 운영을 보여준다. 소프트웨어 운영에 있어서 고장에 의한 정지와 고장이 아닌 것에 의한 정지도 고려한다. 하나 이상의 포인트클래스를 가진 포인트공정을 규정하는 편리한 방법은 마코프 갱신공정이다. 이는 소프트웨어운영에서 무작위적인 두 개의 요소에 대한 직관적이고도 별도의 검토를 가능하게 한다.

3.1. 이산시간에서의 소프트웨어 신뢰도

이산시간에서의 소프트웨어 운영시퀀스를 s-종속 베르누이시행으로 보여주고 있으며, 이 시행에서 각각의 시행의 성공이냐 실패냐의 확률은 이전시행의 결과에 의존한다. 소프트웨어운영 j와 관련하여 각 특수 운영이 성공이냐 실패냐를 구분 짓는 확률변수 Z_j는 다음과 같다.

$$Z_j=\begin{cases} 0: \text{운영 } j\text{에서 성공} \\ 1: \text{운영 } j\text{에서 실패} \end{cases}$$

n개의 연속 소프트웨어운영에서 n개의 가능한 s-종속 확률변수 중 고장에 이르게 하는 운영의 수는 다음과 같다.

$$S_n=Z_1+\cdots+Z_n \tag{1}$$

어떤 운영 j가 고장에 이르렀다고 해보자. 그러면 j+1번째 운영에서는 고장확률이 q이며 성공확률이 $\bar{q}$라고할 때

$$\Pr\{Z_{j+1}=1|Z_j=1\}=q$$

$$\Pr\{Z_{j+1}=0|Z_j=1\}=\bar{q}$$

이다.

마찬가지로, 어떤 운영 j가 성공에 이르렀다고 해보자. 그러면 j+1번째 운영에서는 성공확률이 p이며 고장확률이 $\bar{p}$라고할 때

$$\Pr\{Z_{j+1}=0|Z_j=0\}=p$$

$$\Pr\{Z_{j+1}=0|Z_j=0\}=\bar{q}$$

이다.

s-종속 베르누이시행 $\{Z_j;j\geq 1\}$시퀀스는 두 가지 상태의 이산시간 마코프 체인을 정의한다. 0으로 표시된 그 중 한 상태는 성공이고 1로 정의된 것은 고장이다. 이 마코프체인의 그래프적 표현은 그림 1의 상태도로 설명한다. 천이확률 마트릭스는 다음과 같다.

$$P=\begin{bmatrix} p & \bar{p} \\ q & q \end{bmatrix},\ 0\leq p,\ q\leq 1 \tag{2}$$

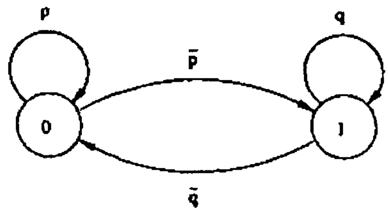

그림1 s-종속 베르누이시행에서의 마코프 설명

p와 q가 확률이므로 다음과 같은 관계가 성립한다.

$$|p+q-1|\leq 1 \tag{3}$$

우선, 마코프 체인에 대해서 좀더 자세히 고찰해보자. 확률 $p(q)$는 이전의 운영이 실패(성공)라고 했을 때 실패(고장)일 조건확률이다. 운영(j+1)에서의 고장 무조건확률은 다음과 같다.

$$\begin{aligned}
&\Pr\{Z_{j+1}=1\} \\
&=\Pr\{Z_{j+1}=1,\ Z_j=1\}+\Pr\{Z_{j+1}=1,\ Z_j=0\} \\
&=\Pr\{Z_{j+1}=1|Z_j=1\}\cdot\Pr\{Z_{j+1}=1\} \\
&\quad+\Pr\{Z_{j+1}=1|Z_j=0\}\cdot\Pr\{Z_{j+1}=0\} \\
&=\bar{p}+(p+q-1)\cdot\Pr\{Z_j=1\}
\end{aligned} \tag{4}$$

$p+q=1$이면, 마코프체인은 s-독립 베르누이 시행을 설명하며, 따라서 (4)는 다음과 같이 된다.

$$\Pr\{Z_{j+1}=1\}=\bar{p}=q$$

이는 고장확률이 이전 운영의 결과와 관계 없음을 보여준다. 그러므로, 후속 운영은 독립적으로 각각의 확률이 p와 $q=\bar{p}$인 성공이거나 실패이다. 이 경우에 (1)에서 언급한 S_n는 n개의 상호 s-독립 베르누이 확률변수이며, 이항 pmf를 가진다.

$p+q\neq 1$이면, DTMC(이산시간마코프체인)은 s-종속 베르누이시행 시퀀스를 설명하며, 연속 운영중의 가능한 s-종속을 수용할 수 있도록 해준다. 이 경우에, 소프트웨어운영의 결과(성공 또는 실패)는 (4)에서와 같이 이전의 운영결과에 의존한다.

$p+q>1$이면, 운영이 능동적으로 상관관계를 가지며, 소프트웨어 고장이 운영 j에서 발생되면 다음 운영에서 고장이 발생할 기회가 더 커진다. 그러므로, 이 경우에, 고장이 다발로 발생된다. (3)의 등식이 성립하는 경우 즉, $p+q=2(p=q=1)$는 경계조건이 발생한다. 이는 소프트웨어운영의 시퀀스가 고장으로부터 시작되면 모든 후속운영이 고장나고, 성공으

로 시작되면 모든 후속운영이 성공이라는 것을 의미한다. 이는 그림 2에서 보듯 마코프체인이 영원히 초기상태로 남아있음을 의미한다.

그림 2 $p=q=1$일 때의 DTMC

그 다음에 연속 소프트웨어 운영이 음성적으로 상관된 $p+q<1$이 경우를 생각해본다. 이 경우에 운영 j에서 소프트웨어 고장이 발생되면 운영 $(j+1)$에서 성공될 기회가 커진다. (3)의 등식이 성립하는 경우 즉 ($p+q=0$ 즉 $p=q=0$)가 되는 경계 조건에서는 마코프 체인은 그림 3에서 보듯이 두 가지 상태를 결정론적으로 교대한다.

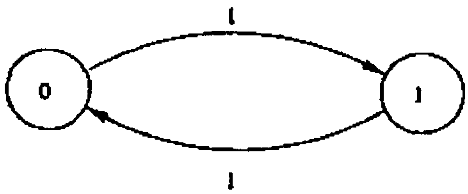

그림 3 $p=q=0$일 때의 DTMC

경계조건인 경우는 사소한 경우로서 실제적으로 관심사가 되지 못한다. 우리는 천이확률에 $0<p,\ q<1$인 조건을 부과한다.

1) $|p+q-1|<1$;

2) 그림 1의 DTMC는 단순화될 수도 없고 비동기식이다.

3.2 연속시간 소프트웨어 신뢰도 모델

모델구성의 다음 단계는 소프트웨어 운영이 실행될 수 있는 시간을 고려한 연속시간 공정을 얻는 것이다. $F_{k,l}(t)$는 그림 1의 DTMC에서 상태 k에서 상태 l로 천이하는데 쓰이는 시간의 Cdf이다. 운영실행시간이 성공운영 및 실패운영 모두에 걸쳐서 동일하게 분포되자 않는다는 가정은 합리적이라고 할 수 있다. 그러므로, $F_{k,l}(t)$는 시간간격 끝에서의 포인트 형태에만 의존한다.

$$F_{0,0}(t)=F_{1,0}(t)=F_{ex\,S}(t)$$
$$F_{0,1}(t)=F_{1,1}(t)=F_{ex\,F}(t)$$

DTMC의 천이에 $F_{k,l}(t)$을 더하여 연속시간의 SRM으로서 MRP를 얻는다. $\{N(t), t\geq 0\}$는 두 개의 s-종속 갱신 공정 $N_S(t)$와 $N_F(t)$의 중첩이다.

3.3. MRP에 근거한 소프트웨어 신뢰도 모델

테스트단계에서 소프트웨어는 운영시퀀스에 매이게되며, 고장이 없을 때는 아무런 변화가 없다. 어떠한 운영에서 고장이 발생되면 다음 운영상에서 조건성공확률 및 실패확률을 변화시키도록 하는 중요결함을 수정하려는 시도가 있게 된다. 그러므로,

$$P_i=\begin{bmatrix} p_i & \bar{p}_i \\ \bar{q}_i & q_i \end{bmatrix} \tag{5}$$

가 다음 고장의 발생$(i+1)$까지 고장 i 발생을 따르는 테스트운영에 대해서 $p_i,\ q_i$를 정의한다. 이산시간의 SRGM은 계단 종속 확률을 가지는 s-종속 베르누이 시행시퀀스로 설명할 수 있다. 중요한 통계적 공정은 비동차 DTMC이다

그림 4에서 운영실행시간 분포를 DTMC의 천이에 할당하여 연속시간 SRGM을 얻는다. 다순화하기 위해서 결과에관계없이 동일한시간분포를선택하였다.

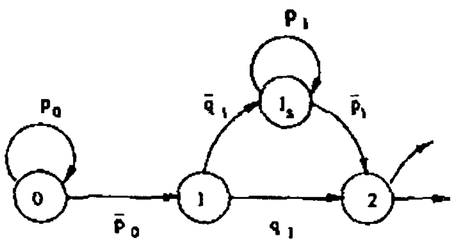

그림 4 SRMG에 대한 비동차 DTMC

$F_{ex}(t)=F_{exS}(t)=F_{exF}(t)$이므로 각 소프트웨어 운영의 T_{ex}는 Cdf $T_{ex}(t)=\Pr\{T_{ex}\leq t\}$를 가진다.

성공운영과 실패운영의 소프트웨어 실행시간이 동일하게 분포되지 않은 상황을 고려하여

곧바로 $F_{exS}(t)\neq F_{exF}(t)$를 얻는다.

그림 4로부터 확률변수 $X_{i+1}(i\geq 1)$는 다음과 같은 pmf를 가진다.

$$\Pr\{X_{i+1}=k\}=\begin{cases} q_i & \text{if } k=1 \\ \overline{q}_i \cdot p_i^{k-2} \cdot \overline{p}_i & \text{if } k\geq 2 \end{cases} \tag{6}$$

(0, t]의 고장수를 기록하는 포인트 공정 $N_F(t)$의 간격분포인 연속확률변수 X_{i+1} 의 분포를 유도한다.
식(6)으로부터 시스템이 I개의 고장을 가졌다고 했을 때 고장시간 (i+1)의 조건 Cdf는 다음과 같다.

$$F_{i+1}(t)=q_i \cdot F_{ex}(t)+\sum_{k=2}^{\infty}\overline{q}_i \cdot p_i^{k-2} \cdot \overline{p}_i \cdot F_{ex}^{k}(t) \tag{7}$$

$F_{ex}(s)$의 LST는 s≥0에서

$$\overline{F}_{ex}(s)=\int_0^{\infty}\exp(-s\cdot t)dF_{ex}(t)=E[\exp(-s\cdot t)]$$

로 정의된다.
$F_{i+1}(t)$의 LST는

$$\overline{F}_{i+1}(s)=q_i\cdot \overline{F}_{ex}(s)+\sum_{k=2}^{\infty}\overline{q}_i\cdot p_i^{k-2}\cdot \overline{p}_i\cdot \overline{F}_{ex}^{\,k}(s)$$

$$=\frac{q_i\cdot \overline{F}_{ex}(s)+(1-p_i-q_i)\cdot \overline{F}_{ex^2}(s)}{1-p_i\cdot F_{ex}(s)} \tag{8}$$

로 된다. 그리고, MTBF는

$$F[T_{i+1}]=-\frac{dF_{i+1}(s)}{dt}\,|\,s=0-\frac{\overline{p}_i+\overline{q}_i}{\overline{p}_i}\cdot\frac{d\overline{F}_{ex}(s)}{ds}|s=0$$

$$=-\frac{\overline{p}_i+\overline{q}_i}{\overline{p}_i}\cdot E[T_{ex}] \tag{9}$$

이다.

감사의 글
본 연구는 한국과학재단 목적기초연구 (R01-2000-00273) 지원으로 수행되었음

참고문헌

[1] S. Bittanti, P. Bolzem, R.Scatolini, "An introduction to software reliability modeling", vol.29, pp43-66

[2Katerina Goseva-Popstrojanova, Kishor S. Trivedi, "Failure Correlation in Software Reliability Models", IEEE Tran. on Reliability, 2000, 3, vol.49, pp37-48