

불완전수정에서 격리된 시험영역에 대한 소프트웨어 신뢰도 성장모형 연구

남경현*[†] · 김도훈*

* 경기대학교 응용정보통계학과

A Study on Software Reliability Growth Model for Isolated Testing-Domain under Imperfect Debugging

Kyung H. Nam*[†] · Do Hoon Kim*

* Department of Applied Information Statistics, Kyonggi University

Key Words : Software Reliability Growth Model, NHPP, Imperfect Testing-Domain Function, Mean Value Function, Intensity Function

Abstract

In this paper, we propose a software reliability growth model based on the testing domain in the software system, which is isolated by the executed test cases in software testing. In particular, our model assumes an imperfect debugging environment in which new faults are introduced in the fault-correction process, and is formulated as a nonhomogeneous Poisson process(NHPP). Further, it is applied to fault-detection data, the results of software reliability assessment are shown, and comparison of goodness-of-fit with the existing software reliability growth model is performed.

1. 서 론

최근 사회가 고도화, 산업화되면서 컴퓨터 시스템의 기능 및 역할은 점차 다양화되고 있으며, 현대 사회에서 필수적인 요건이 되고 있다. 이에 따라 컴퓨터 시스템을 제어하는 소프트웨어의 역할 또한 중요한 부분을 차지함에 따라 하드웨어 뿐만 아니라 소프트웨어 제품의 고품질화를 요구하고 있다. 소프트웨어 시스템을 개발하는 과정은 매우 복잡하여 개발 과정 동안 예측할 수 없는 결과가 발생할 수 있으며, 이로 인한 운영과정(operational phase)상 소프트웨어 시스템의 고장은 인간의 생명, 환경 및 경제에 심각한 영향을 초래하기도 한다. 다시 말해, 원자력발

전소, 병원시스템, 항공기 및 인공위성 등 현대의 첨단산업에서 어떤 다른 품질특성보다 소프트웨어 신뢰성이 중요한 품질요소로 취급되고 있으며, 최근 하드웨어와 소프트웨어가 결합된 임베디드 시스템(embedded system)의 급격한 성장으로 인해 소프트웨어 신뢰성은 제품 개발자 뿐만 아니라 최종 사용자 모두의 기본적인 관심사항이 되고 있다.

소프트웨어 신뢰성에 관한 연구는 지난 20여년 동안 꾸준히 수행되어 왔다. 특히 최근에 컴퓨터의 고장 없는 가동을 위한 고신뢰도의 소프트웨어 시스템에 대한 필요성에 의하여 신뢰성 모형에 관한 연구가 국내외적으로 진행되어 왔다. 이와 관련된 수리적 모형이 소프트웨어 신뢰도 성장모형(software reliability growth model : SRGM)이다(Goel and Okumoto, 1979; Yamada et al., 1983; Yamada and Osaki, 1985).

이와 관련하여 다양한 통계적 모형들이 소프트웨

[†] 교신저자 knam@kyonggi.ac.kr

* 이 논문은 2005년도 경기대학교 교내과제 연구비 지원에 의한 논문임.

어 신뢰성을 평가하기 위한 목적으로 제안되어 왔으며, 비동질적 포아송과정(nonhomo-geneous Poisson process : NHPP)에 기초한 소프트웨어 신뢰도 성장 모형은 실제적인 소프트웨어 신뢰성 공학 면에서 매우 성공적인 이론이라는 것이 증명되었다(Musa et al., 1987).

NHPP 상에서 주요 논점은 특정 시점까지의 기대 고장수를 나타내는 적절한 평균값 함수(mean value function)를 결정하는 것이다. 지금까지의 NHPP SRGM들은 모형에 맞는 다양한 가정들로 이루어져 왔다. 이러한 모형들은 결함발생시 수행되는 수정과정(debugging process)에서 모든 결함은 완전하게 수정 및 제거되고 새로운 결함이 시스템에 도입되지 않는다는 완전 수정환경(perfect debugging environment)을 가정하여 진행되어 왔다. 그러나 이러한 가정은 좀 더 정확한 소프트웨어의 결함발견 현상을 기술하는 데 있어 수정되어야 하며 실제 소프트웨어 시험과정이나 운영과정에서의 수정활동을 이해하기 위해서 불완전수정(imperfect debugging)을 갖는 소프트웨어 신뢰성 평가 모형을 개발하는 것이 필요하다(Yamada et al., 1992; Pham, 1993).

일반적으로 소프트웨어 개발과정의 마지막 단계인 시험단계에서는 소프트웨어의 요구기능에 따라 제대로 된 기능을 수행하는가를 확인하기 위해 다양한 테스트 케이스를 실행하게 되며, 테스트 케이스에 의해 영향을 받는 소프트웨어 시스템 내의 모듈이나 함수에 대한 시험경로가 존재하게 된다. 이러한 시험경로의 집합을 시험영역(testing-domain)이라 한다. Ohtera, Yamada and Ohba(1990, 1992)에 의해 시험영역에 종속된 소프트웨어 신뢰도 성장모형이 처음 발표된 이후로, Ohtera, Yamada and Narihisa(1990), Yamada, Ohtera and Ohba(1992), Fujiwara and Yamada(2003) 등에 의해 꾸준한 연구가 이루어져 왔다.

본 논문의 목적은 소프트웨어 개발과정 중 소프트웨어 시험과정에서 테스트 케이스에 의해 격리된 시험영역(isolated testing-domain)에 기반을 둔 소프트웨어 신뢰도 성장모형(SRGM)을 제안하는 것이다. 이를 위해 비동질적 포아송과정(NHPP)에 기초한 SRGM을 소프트웨어 결함 탐지 현상을 기술하는데 사용한다. 특히, 본 논문에서는 결함수정과정 중 결함수정에 실패한 경우 또는 새로운 결함이 추가적으로 도입되는 불완전 수정 환경하에서 신뢰도 성장

모형을 제안하고자 한다.

끝으로, SRGM에서의 모수들을 추정하고, 실제 소프트웨어 결함 데이터를 이용하여 수치적인 방법에 의한 모수 추정치를 도출하며, 이전에 제안된 SRGM과의 적합도 비교를 수행한다.

다음은 본 논문에서 사용되는 기호를 요약한 것이다.

• 기 호

- $N(t)$ 시간 t 까지 탐지된 누적 결함수를 나타내는 계수과정(counting process)
- a 소프트웨어 내의 초기 결함수
- $a(t)$ 임의의 시점 t 에서 소프트웨어 시스템 내에 잠재되어 있는 결함수
- b 결함 발견률
- v 시험영역 성장률
- $u(t)$ 불완전수정 시험영역 함수, 임의의 시점 t 에서 격리된 시험영역 내 존재하는 탐지 가능한 총 결함수
- $\gamma(t)$ 시험영역 성장함수
- β 결함 도입률
- $m(t)$ 평균값 함수, 시간 t 까지 탐지된 기대누적결함수
- $h(t)$ 강도함수, 시간 t 에서의 순간결함탐지율

2. 시험영역함수

2.1 시험영역

모든 소프트웨어 시스템은 입출력 관계의 명세서(specification)를 갖고 있다. 즉, 명세서 S 는 소프트웨어 시스템의 기능을 기술하는 순서화된 입출력 쌍의 집합이라 할 수 있다. S 에 있는 입력 집합(input set)을 소프트웨어 시스템의 입력 영역(input domain) D 로 정의하자. 만약, 소프트웨어 시스템이 입력 i 에 대해서 출력 o 를 산출하면 즉, 어떤 $i \in D$ 에 대해서 $(i, o) \in S$ 이면, 소프트웨어 시스템은 그 입력에 대해 성공이라 하고, 그렇지 못하면 실패라고 한다. 여기서 소프트웨어 시스템이 명세서를 만족하는지 알아보기 위해 상당량의 입력을 가지고 소프트웨어 시스템을 실행시키는 것을 소프트웨어 시험(software testing)이라 한다. 즉, 소프트웨어의 개발과정 중 소프트웨어 시험은 소프트웨어 시스템

이 충분히 높은 수준의 신뢰성을 얻을 때까지 소프트웨어 시스템을 단련시켜 가는 과정이며, 소프트웨어 신뢰성 모형에 대한 근간을 제공한다. 따라서 입력영역 D 로부터 소프트웨어 시스템에 잠재된 모든 결함은 아니지만 대부분의 결함을 탐지할 수 있으며, D 를 대표할 수 있는 부분집합(subset) T 를 체계적인 방법으로 선택하여 시험한다. 이때 선택된 부분집합 T 를 격리된 시험영역(isolated testing-domain)이라 부르며 T 에 있는 각 입력을 테스트 케이스(test case)라 한다. 만약 시험과정에서 고장이 발생하면 고장의 원인이 되는 결함을 발견하고 제거하는데, 이러한 행위를 결함수정(fault-correction) 또는 디버깅(debugging)이라 한다.

격리된 시험 영역은 테스트 케이스의 수와 함께 팽창하는데 불완전 수정의 가정하에서 결함수정 및 제거시 새로운 결함이 도입되기 때문에 임의의 시간 t 에서 소프트웨어 시스템 내에 남아있는 기대 총결함수는 시험 시작부터 계속 증가한다. 더욱이 시험 단계 혹은 운영 및 보전 단계에 탐지된 결함을 제거하거나 수정할 때, 시험단계의 초반부와 비교해 새롭게 도입되는 결함의 수는 상수가 아니라 지수적으로 증가하는 형태를 고려할 수 있다. 또한 소프트웨어 시스템 내에서 격리된 시험영역은 계속적으로 팽창한다고 가정하며, 이 기간 중 탐지된 결함은 많은 모듈이나 함수로 연결되어 있기 때문에 시험영역은 결함수정 활동에 의해 영향을 받는다.

2.2 시험영역 함수

소프트웨어 시스템 내에 격리된 시험영역과 시험에 의해 탐지 가능한 결함 수와의 관계를 수리적으로 나타내기 위해 다음과 같은 가정을 도입한다.

- 1) 결함수정활동에서 새로운 결함이 도입되므로 소프트웨어 시스템에 잠재되어 있는 누적 결함의 수는 증가한다.
- 2) 시험영역내에 잠재되어 있는 결함의 수는 균일하게 분포되어 있다.
- 3) 시험영역내에 존재하는 결함수의 증가율은 임의의 시간 t 에서 격리된 시험영역 밖의 소프트웨어 시스템 내 남아있는 결함의 수에 비례한다.

이러한 가정으로부터 다음과 같은 미분방정식을

유도할 수 있다.

$$\begin{aligned} \frac{du(t)}{dt} &= v[a(t) - u(t)], \quad (v > 0) \\ \frac{da(t)}{dt} &= \beta \frac{du(t)}{dt}, \quad a(0) = a, \quad u(0) = 0. \end{aligned} \quad (1)$$

여기서 v 는 시험영역의 성장율이고 $u(t)$ 는 임의의 시간 t 에서 격리된 시험영역 내 존재하는 탐지 가능한 총 결함수를 나타내며, $a(t)$ 는 소프트웨어 시스템에 잠재되어 있는 누적 결함수를 나타낸다. 또한 a 는 초기 결함수를 나타내고 β 는 결함수정 활동내 결함 도입률을 나타낸다. 결론적으로 위 미분방정식을 풀면 식 (2)와 같은 $u(t)$ 를 얻게 되며, 이를 불완전 수정 시험영역함수(Imperfect Testing-Domain Function : *Imperfect T-DF*)라 한다.

$$\begin{aligned} u(t) &= \frac{a}{1-\beta}(1 - \exp[-(1-\beta)vt]), \\ &(\beta \neq 1, \beta > 0, a > 0). \end{aligned} \quad (2)$$

여기서 $\beta=0$ 인 경우에 식 (2)는 다음과 같은 시험영역 함수로 표현된다.

$$u(t) = a(1 - \exp[-vt]) \quad (3)$$

더욱이 *Imperfect T-DF*로부터 시험영역 내에서 탐지할 수 있는 결함의 증가율인 시험영역 성장함수 $\gamma(t)$ 를 유도할 수 있다.

$$\begin{aligned} \gamma(t) &= \frac{du(t)}{dt} \\ &= av \exp[-(1-\beta)vt] \end{aligned} \quad (4)$$

3. 소프트웨어 신뢰도 성장모형

본 논문의 목적은 불완전 수정 시험영역에 기초한 NHPP SRGM을 제안하는데 있으며, 이를 위해 Fujiwara and Yamada(2003)에서 사용된 기본가정을 도입한다.

- 1) 새로운 결함은 결함수정활동을 통해 도입된다.
- 2) 발견된 결함은 항상 격리된 시험영역 내에 존재한다.
- 3) 격리된 시험영역의 비율은 시험을 실시함에 따라 증가한다.

4) 소구간 $(t, t+\Delta t]$ 내에 발견된 결함수는 임의의 시간 t 에서 격리된 시험영역내 남아있는 결함의 수에 비례한다.

더욱이 시간 t 에서 결함발견을 기술하기 위해 $\{N(t), t \geq 0\}$ 를 시간 t 까지 발견된 누적 결함수를 나타내는 계수과정(counting process)이라 한다면 결함발견은 다음과 같은 NHPP로서 공식화 할 수 있다.

$$\Pr \{N(t) = n\} = \frac{\{m(t)\}^n}{n!} \exp[-m(t)],$$

$$(n = 0, 1, 2, \dots)$$

$$m(t) = \int_0^t h(s) ds \quad (5)$$

여기서 $m(t)$ 는 NHPP의 평균값함수(mean value function)를 말하며 $E[N(t)]$ 즉, 시간 t 까지 탐지된 기대 누적 결함수를 나타낸다. 또한 $h(t)$ 는 강도함수(intensity function)를 말하며, 시간 t 에서의 순간 결함탐지율을 나타낸다.

이러한 가정과 Imperfect T-DF $u(t)$ 로부터 소프트웨어 신뢰도 모형을 정의하려고 하며, 이를 이용하여 평균값 함수 $m(t)$ 를 유도함으로서 신뢰도 모형에 관한 특성 및 관련 모수들을 추정할 수 있다.

$$\frac{dm(t)}{dt} = b[u(t) - m(t)], \quad (0 < b < 1). \quad (6)$$

여기서 b 는 시험영역 내에 남아있는 결함당 발견률을 나타내는 비례상수이다.

초기조건 $m(0) = 0$ 하에서 미분방정식 (6)을 풀면 다음과 같은 식 (7)을 얻는다.

$$m(t) = \frac{a}{1-\beta} \{1 - \exp[-bt]\}$$

$$+ \frac{ab}{1-\beta} \frac{1}{(1-\beta)v-b} \{\exp[-(1-\beta)vt] - \exp[-bt]\}$$

$$(\beta \neq 1, \beta > 0, a > 0, 0 < b < 1) \quad (7)$$

식 (6)에서 $\beta = 0$ 이고 $v \rightarrow \infty$ 인 즉, 완전수정을 고려한 시험영역 함수인 경우에 격리된 시험영역은 새로운 결함도입 없이 시험시작 후에 즉시 소프트웨어 시스템 전체로 팽창한다. 그리고 $u(t) = a$ 이기 때문에 $m(t)$ 는 Goel-Okumoto 모형과 같게 된다. 또한 $\beta = 0$ 이고 $v = b$ 인 경우 결함은 새로운 결함도입 없

이 시험영역의 확장으로 동시에 탐지된다. 여기서 $u(t) = a(1 - e^{-bt})$ 이기 때문에 Yamada and Osaki (1985)에 의해 제안된 Delayed S-shape SRGM을 갖는다. 위 사실로부터 본 모형은 exponential SRGM과 Delayed S-shaped SRGM의 특징을 모두 갖게 된다.

식 (7)로부터 NHPP의 강도함수를 구하면 다음과 같다.

$$h(t) \equiv \frac{dm(t)}{dt}$$

$$= \frac{ab}{1-\beta} \{\exp[-bt]\}$$

$$- \frac{abv}{(1-\beta)v-b} \{\exp[-(1-\beta)vt]\}$$

$$+ \frac{ab^2}{1-\beta} \frac{1}{(1-\beta)v-b} \{\exp[-bt]\}$$

$$(\beta \neq 1) \quad (8)$$

본 논문에서 제시한 평균값 함수 $m(t)$ 에 포함된 모수 a, b, v, β 에 관한 추정을 수행한다. 모수 추정에 사용되는 자료는 개발된 소프트웨어 시스템이 시험 기간 동안에 발생한 결함의 수와 시점에 관하여 얻어진 결과로서, 이러한 자료를 이용하여 신뢰도 모형을 특징 지어주는 모수들의 값을 추정한다. 본 논문에서는 모수 추정방법으로 기존 논문에서 사용한 최우추정법을 사용한다(Musa et al., 1987). 우선 결함 데이터 $(t_k, y_k) (k = 1, 2, \dots, n; 0 < t_1 < t_2 < \dots < t_n)$ 는 소프트웨어 시험과정에서 관측될 수 있으며, y_k 는 주어진 시점 $(0, t_k]$ 사이에서 발견된 누적 결함수를 나타낸다. 따라서 평균값 함수 $m(t)$ 를 갖는 NHPP 모형에 기초한 결함확률밀도함수 즉, 우도함수는

$$L \equiv \Pr \{N(t_1) = y_1, N(t_2) = y_2, \dots, N(t_n) = y_n\}$$

$$= \prod_{k=1}^n \frac{\{m(t_k) - m(t_{k-1})\}^{y_k - y_{k-1}}}{(y_k - y_{k-1})!}$$

$$\times \exp\{-[m(t_k) - m(t_{k-1})]\} \quad (9)$$

이며, $t_0 \equiv 0, y_0 \equiv 0$ 이다.

모수 a, b, v, β 에 대한 추정치는 식 (9)에서 양변에 자연로그를 취한 후, 로그우도함수를 모수 각각에 대하여 편미분하여 0을 대입하여 풀면 얻을 수 있다.

$$\frac{\partial \ln L}{\partial a} = \frac{\partial \ln L}{\partial b} = \frac{\partial \ln L}{\partial v} = \frac{\partial \ln L}{\partial \beta} = 0 \quad (10)$$

4. 수치예제

4.1 모수추정

본 논문에서 제안된 모형의 모수를 추정하기 위해 Misra(1983)에 의해 제시된 소프트웨어 결함 데이터를 이용하였으며, 이는 <표 1>과 같다. 데이터의 속성은 시험시간에 따라 탐지된 결함의 수를 측정한 것이다.

<표 1> 소프트웨어 결함 데이터(Misra, 1983)

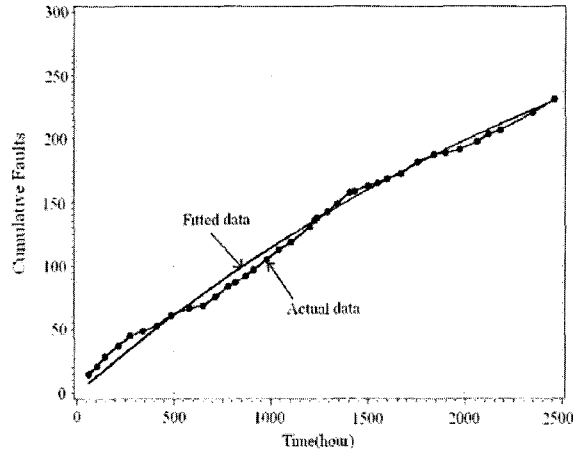
Test		Number	Test		Number
Week	Hour	of faults	Week	Hour	of faults
1	62.5	15	20	25	5
2	44	6	21	12	2
3	40	8	22	55	5
4	68	8	23	49	6
5	62	8	24	64	9
6	66	4	25	26	1
7	73	4	26	66	4
8	73.5	8	27	49	2
9	92	6	28	52	4
10	71.4	2	29	70	4
11	64.5	7	30	84.5	9
12	64.7	8	31	83	6
13	36	3	32	60	1
14	54	5	33	72.5	3
15	39.5	5	34	90	6
16	68	8	35	58	6
17	61	8	36	60	3
18	62.6	6	37	168	14
19	98.7	12	38	111.5	10

식 (10)으로부터 Misra(1983) 데이터에 대한 평균값 함수 $m(t)$ 의 모수인 a, b, v, β 를 수치적인 방법에 의해 구하였으며, 이에 대한 결과는 <표 2>와 같다.

<표 2> Misra(1983)데이터에 대한 모수 추정치

구 분	\hat{a}	\hat{b}	\hat{v}	$\hat{\beta}$
모수 추정치	372	0.000305	0.3679	0.1588

<그림 1>은 Misra(1983) 데이터를 이용하여 추정된 모수 추정치를 평균값함수 $m(t)$ 에 대입하여 적합시킨 그림이다.



<그림 1> 추정된 평균값 함수 $\hat{m}(t)$

4.2 적합도 비교 평가

본 절에서는 선행모형과 본 제안모형과의 적합도를 평가한다. 이를 위하여 평균제곱오차(mean square error : MSE)를 모형의 적합도를 평가하기 위한 기준으로 사용하고자 한다. 우선 평균제곱오차는

$$MSE = \frac{1}{n} \sum_{k=1}^n [y_k - \hat{m}(t_k)]^2 \quad (11)$$

으로 계산되며, 이는 제안된 신뢰도 성장모형이 실제 결함 데이터와 일치정도를 평가할 수 있는 척도로 사용된다. <표 3>은 기존의 완전수정을 고려한 시험영역 SRGM과 새로운 모형의 MSE를 비교한 결과를 나타낸다.

<표 3> MSE에 기초한 모형 비교

구 분	제안 모형	기존의 시험영역 SRGM
MSE	4.398	4.697

결과를 살펴보면 불완전 수정을 고려한 모형에서의 MSE 값이 기존 모형에서의 MSE 값보다 작은 것을 알 수 있다. 따라서 새로운 모형이 실제 결함 데이터를 잘 적합시키는 것을 확인할 수 있다.

5. 결 론

본 논문에서는 소프트웨어 개발과정의 시험단계에서 시간에 의존하는 격리된 시험영역을 고려하여

NHPP에 기초한 소프트웨어 신뢰도 성장모형을 제안하였다. 특히, 결합수정 활동에서 새로운 결합이 추가적으로 도입되는 불완전수정 환경을 고려한 시험영역함수를 유도하였으며, 이를 통해 평균값 함수를 구하였다. 본 제안 모형은 exponential SRGM과 Delayed S-shaped SRGM의 특징을 모두 갖고 있음을 알 수 있다.

실제 결합 데이터를 이용하여 본 모형의 적합성을 평가한 결과 기존의 완전수정을 고려한 시험영역 SRGM보다 실제 데이터를 잘 적합시키고 있음을 알 수 있었다.

참 고 문 헌

- [1] Fujiwara, T. and Yamada, S.(2003), "A testing-domain-dependent software reliability growth model for imperfect debugging environment and its evaluation of goodness-of-fit", *Electronics and Communications in Japan*, Part 3, Vol. 86, No. 1, pp. 11-18.
- [2] Goel, A. L. and Okumoto, K.(1979), "Time-dependent error-detection rate model for software reliability and other performance measures", *IEEE Trans. on Reliability*, Vol. R-28, No. 3, pp. 206-211.
- [3] Misra, P. N.(1983), "Software reliability analysis", *IBM Systems Journal*, Vol. 22, pp. 262-270.
- [4] Musa, J. D., Iannino, A., and Okumoto, K. (1987), *Software reliability: Measurement, prediction, application*, McGraw-Hill, New York.
- [5] Ohtera, H., Yamada, S., and Ohba, M.(1990), "Software reliability growth model with testing-domain and comparison of goodness-of-fit", *Intern. Symp. Reliability and Maintainability*, pp. 289-294.
- [6] Ohtera, H., Yamada, S., and Narihisa, H. (1990), "Software reliability growth model for testing-domain", *Trans. IEICE*, J73-D-I, pp. 170-174.
- [7] Pham, H.(1993), "Software reliability assessment : Imperfect debugging and multiple failure types in software development", EG&G-RAMM-10737, Idaho National Engineering Laboratory.
- [8] Yamada, S., Ohba, M., and Osaki, S.(1983), "S-shaped reliability growth modeling for software error detection", *IEEE Trans. on Reliability*, Vol. R-32, No. 5, pp. 475-478, 484.
- [9] Yamada, S., Ohtera, H., and Narihisa, H. (1986), "Software reliability growth models with testing-effort", *IEEE Trans. on Reliability*, Vol. R-35, No. 1, pp. 19-23.
- [10] Yamada, S., Ohtera, H., and Ohba, M. (1992), "Testing-domain dependent software reliability growth models", *Computers & Mathematics with Applications*, Vol. 24, No. 1/2, pp. 79-86.
- [11] Yamada, S. and Osaki, S.(1985), "Software reliability growth modeling : model and application", *IEEE Trans. Software Engineering*, Vol. 11, pp. 1431-1437.
- [12] Yamada, S., Tokuno, K., and Osaki, S.(1992), "Imperfect debugging models with fault introduction rate for software reliability assessment", *International Journal of Systems Science*, Vol. 23, pp. 2241-2252.