

내장형 소프트웨어의 신뢰성 품질 측정 방안 연구

장윤정*, 윤청**

*충남대학교 컴퓨터공학과

e-mail:jung30233@naver.com

A Study on the Reliability Quality Measurement of Embedded Software

Yoon-Jeong Jang*, Cheong Youn**

*Dept of Computer Engineering, Chung-nam National University

요 약

오늘날 선박, 항공, 자동차산업 등의 복합화로 인하여 내장형 소프트웨어의 중요성이 크게 대두되고 있다. 중요도가 높아질수록 고장 현상 발생률이 증가하고 이에 따른 오류나 고장에 대한 영향도 급증하기 때문에, 신뢰성이 그만큼 중요하게 되었다. 신뢰성은 ISO/IEC 9126, ISO/IEC 12119 등에서 기능성, 사용성, 효율성, 유지보수성, 이식성과 함께 언급되고 있다. 신뢰성의 부특성에는 성숙성, 오류허용성, 회복성, 준수성이 있고, 이에 따라 품질평가 항목 매트릭이 제시되어 있다. 그러나 신뢰성 평가 기준은 수리적 난해함과 데이터 수집에 관한 문제에 부딪혀 적용하는 데에 어려움이 따르는 실정이다. 이와 같은 문제점을 해결하기 위해, 각 세부 평가 항목에 대한 체크리스트를 가지고 품질평가 매트릭에 적용하였다. 또한, 기존에 제시되어 있는 대표적인 신뢰성 모델 두 가지 NHPP(Non-Homogeneous Poisson Process) Model과 Jelinski-Moranda Model을 성숙성과 오류허용성의 품질평가 매트릭과 연관시켜 효율적으로 신뢰성을 측정하는 방안에 대하여 모색하였다.

1. 서론

최근 일상생활에서 내장형 소프트웨어(이하 ‘임베디드 소프트웨어’)는 없어서는 안 될 만큼 광범위하게 사용되고 있다. 내장형 소프트웨어 사용량이 증가할수록 규모와 복잡도 또한 증가하게 되며, 그 결과 시스템의 고장 발생 가능성 역시 높아질 수 있다. 이는 시스템에 대한 신뢰성이 더욱 중요하게 대두될 수 있다는 것을 의미한다. ISO/IEC 9126이나 ISO/IEC 12119 등에서는 신뢰성을 기능성, 사용성, 효율성, 유지보수성과 함께 품질 속성의 한 부분으로 언급되고 있다. 아래의 [표 1]은 신뢰성의 부특성마다 신뢰성을 평가하는 방식에 대하여 분류해 놓은 표이다.

[표 1] 소프트웨어 신뢰성 평가 방식

구분	구성
성숙성	- 문제 해결 이력 정보 제공 - 문제 해결력 - 결함 회피율 - 결함 발생 평균 시간
오류허용성	- 다운 회피율 - 고장 회피율 - 오조작 회피율
회복성	- 데이터 회복 정보 제공 - 데이터 회복율 - 이용 가능성 - 평균 복구시간 - 복구 가능성 - 복구 효과율
준수성	- 신뢰성 수준 준수율

본 논문에서는 신뢰성의 부특성 4가지 중에서 성숙성과 오류허용성의 측정 방법을 제시한다. 잘 알려진 신뢰성 모델 NHPP(Non-Homogeneous Poisson Process) Model과 Jelinski-Moranda Model을 성숙성 및 오류허용성 측정 매트릭과 연관시켰다. 위의 두 가지 모델은 고장 자료를 통해서 신뢰성을 측정하는 대표적인 모델로, 성숙성과 오류허용성을 측정하기 위한 근거 자료를 제공한다. 기존의 모델을 신뢰성 측정 매트릭에 적용함으로써 정확한 평가 결과를 얻을 수 있도록 하는 것이 본 논문의 목적이다.

2. 관련 연구

2.1 내장형 소프트웨어

내장형 소프트웨어는 선박, 항공, 자동차, 로봇, 의료기기 등 전자장치에 내장되어 있는 소프트웨어로서, 안전한 시스템을 위해 신뢰성이 더 중요하게 된다. 하드웨어의 경우 고장 발생에 대하여 예측하고 예방할 수 있으나, 소프트웨어는 물리적인 사전 통보 없이는 고장에 대한 인지가 불가능하다. 소프트웨어 신뢰성 모델을 사용하여 신뢰성을 평가하기 위해서는 고장 자료가 필요하게 되고, 고장 발생 후 분석 절차를 걸침에 따라 어떠한 고장인지가 구분된다. 이로써 고장이 발생하기까지의 시간을 도출해낼 수 있다. 시스템이 정해진 시간 내에 정의된 작업을 수행하지 못하는 경우, 고장 자료를 잘 파악하여 소프트웨어의 신뢰도를 높이는 작업이 중요성이 대두된다.

2.2 ISO/IEC 9126

ISO/IEC 9126은 소프트웨어 품질특성과 척도에 관한 지침을 제공하고 있으며, 모델의 구성은 [표 2]와 같다.

[표 2] ISO/IEC 9126 모델 구성

구분	구성	내용
ISO 9126-1	Quality Model	- 소프트웨어 품질 특성과 Metrics
ISO 9126-2	External Metrics	- 소프트웨어 사용시의 외부적 성질 표현 - 실행 가능 소프트웨어 시험/운영으로 측정
ISO 9126-3	Internal Metrics	- 설계/코드 관련 소프트웨어 내부 속성 측정 - 설계/코딩 중인 소프트웨어 제품에 적용

ISO 9126은 품질 요구를 정의, 평가 준비, 평가하기까지의 과정으로 나뉜다. 품질 요구를 정의하는 단계에서는 품질특성 및 이용 가능한 특성들을 사용하여 품질 요구사항을 규정한다. 평가 준비 단계에서는 품질 요구사항을 정량적으로 측정할 수 있는 메트릭을 준비하는 단계이다. 평가 단계에서는 실제로 측정하고 등급을 부여하며, 수용 또는 기각 등의 판정을 내리게 된다. 이로써 선정된 메트릭을 소프트웨어 제품에 적용한다.

ISO/IEC 9126은 또한, 아래의 [그림 1]과 같이 6가지의 품질 특성으로 구성되어 있다.



[그림 1] ISO/IEC 9126의 6가지 품질 특성

2.3 ISO/IEC 9126 기반의 품질 측정 방법

ISO/IEC 9126을 기반으로 하여 정보통신산업진흥원(NIPA)에서는 소프트웨어 기술성 평가 기준 해설서로, 한국정보통신기술협회(TTA)의 웹기반소프트웨어 품질평가 지침'으로 구체적인 품질 측정 방법을 제시하였다.

2.3.1 NIPA의 소프트웨어 품질측정 방법

NIPA에서는 '소프트웨어 기술성 평가기준 해설서'에 소프트웨어 품질 측정 방법을 정의하고 있으며, 문서의 구성은 아래의 [그림 2]와 같다.

* 품질속성 (예 : 기능성, 신뢰성 등)
1. 부특성이름 (예 : 기능성 아래 상호운영성, 보안성 등)
...
- 측정유형(예 : Number, Y/N/NA 등)
- 측정요소 및 측정방법
1.1.1 측정요소 및 측정방법
...
- 계산식(예시)
결함 회피율 : 1-min(발견된 결함 수 / 단위 운용시간)
다운 회피율 : 1-소프트웨어의 재시동이 필요한 결함의 수 / 발견된 결함 수

[그림 2] 소프트웨어 기술성 평가기준 해설서

2.3.2 TTA의 소프트웨어 품질평가 지침

TTA에서는 '웹 기반 소프트웨어 품질평가 지침'을 제공하고 있다. 평가 항목은 메트릭으로 제공하며, 품질특성 - 부특성 - 세부평가항목으로 연결된다. 또한 '품질평가 모듈'로 평가항목 단위로 상세한 해설을 제공한다. 본 문서의 구성은 아래의 [그림 3]과 같다.

* 품질속성 (예 : 기능성, 신뢰성 등)
* 부특성 이름 (예 : 기능성 아래 상호운영성, 보안성 등)
1. 평가항목(메트릭)
1.1 개요
1.1.1 메트릭의 개념
1.1.1 측정 목적
1.1.1 메트릭의 범주(예 : 기능성 - 적합성 - 기능구현 완전성)
...
1.1 참조문서 (예 : ISO/IEC 9126-2 8.1.1 등)
1.1.1 측정 항목
...
- 측정 유형 (예 : Number, Y/N/NA 등)
1.1.1 측정 방법
1.1.1 계산식 (예시)
- 기능표준준수율 = B/A
d
- B = (Success TC / Total TC)
= 1
1.1 적용절차

[그림 3] 품질평가 모듈의 품질검사표 구성

NIPA의 문서는 품질의 단위인 부특성이지만, TTA는 이보다 더욱 상세한 평가항목(메트릭)이 단위이다.

2.4 소프트웨어 신뢰성 모델

2.4.1 NHPP(Non-Homogeneous Poisson Process) Model

NHPP Model은 비동차 포아송 과정에 일정한 시간 간격으로 발생하는 에러를 기술한 모델이다. 특정시간에 남아있는 에러는 고장을 일으킬 수 있고, 임의의 테스트 시간까지 발견되는 총 에러 수는 NHPP를 따른다는 가정에 의거하여 신뢰성을 평가한다.

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

$$n_r(t) \equiv E[N(t)] = E(N(\infty)-N(t)) = a-H(t) \dots \dots \dots (2)$$

$$R(x|t) = \exp[-\{H(t+x)-H(t)\}] (t \geq 0, (n=0, 1, 2, \dots) \dots \dots (3)$$

식(1)에서 Pr{N(t)=n}은 테스트 시각 t까지 발견되는 n개의 에러발견 사상확률, H(t)는 N(t)의 평균치를 나타내는 평균치 함수이다. 식(2)에서는 테스트시각 t에서 $\bar{N}(t) = N(\infty) - N(t)$ 중 소프트웨어 기대 잔존 에러수를 \bar{N} 라 하고, \bar{N} 의 평균치를 나타낸다. a는 테스트 시작 전에 잔존하는 총 기대 에러수를 나타낸다. 테스트가 테스트 시각 t까지 진행되는 중에 시간구간에서 에러가 발생하지 않을 확률은 식(3)이 된다.

2.4.2 Jelinski-Moranda Model

신뢰성 모델 중 가장 많이 알려진 모델인 Jelinski-Moranda Model 은, 지수함수의 성질을 이용하여 각 단계별 실패율은 동일하다고 가정하였다. 소프트웨어의 고장은 그 다음 단계 고장에 독립적이라는 가정을 하여 신뢰성을 평가하는 방식을 보인다. 기존의 지수함수는 하드웨어 신뢰성 평가에 적용되었으나, 소프트웨어적인 특성을 고려하여 모수의 범위를 만족하는 범위 내에서 신뢰성 평가가 가능하게 된다. 아래의 식은 Jelinski-Moranda Model의 기본이 되는 식이다.

$$L(t_1, t_2, \dots, t_n; N_o, \mu) = \prod_{i=1}^n \mu(N_o - i + 1) \exp\{-\mu(N_o - i + 1)t_i\} \dots (1)$$

$$\mu = \frac{1}{N_o - 1} + \dots + \frac{1}{N_o - n + 1} = \sum_{i=1}^n \frac{1}{N_o - i + 1} \dots (2)$$

식(1) 소프트웨어 신뢰성 모델에서 최우추정법을 이용하면 식(2)와 같이 모수 추정 결과가 나타난다. 식(2)에서 전체 오류 수와 실패율을 구한다. 전체 오류 수는 모수 추정 값으로서, MTTF(Mean Time To Failure)를 이용하여 소프트웨어 신뢰성 품질평가 매트릭에 적용할 수 있다. μ 은 현재 소프트웨어 실패율이 고장에 반비례한다는 것과, N_o 는 소프트웨어 결함은 알 수 없는 무한한 수라는 것을 가정한다.

3. 신뢰성 측정을 위한 품질평가 매트릭

품질평가 매트릭을 적용하여 소프트웨어 신뢰성을 측정하기 위해서는, 신뢰성 모델에 근거한 예상잠재 고장 밀도 등의 예측 데이터를 도출하여야 한다.

기존의 품질평가 매트릭과 신뢰성 모델은 별도의 내용으로 연구되어 왔으나, 신뢰성 측정에 있어서 이 두 가지 개념은 밀접하게 연관될 수 있다. 신뢰성의 부특성 중 성숙성과 오류허용성에 대한 측정 매트릭을 소개하고, 이를 고장 자료를 이용한 신뢰성 평가 모델로 가장 많이 알려져 있는 NHPP(Non-Homogeneous Poisson Process) Model, Jelinski-Moranda Model과 연관시켰다. 두 신뢰성 모델을 근거로 한 신뢰도 평가 수행과정에서 성숙성과 오류허용성의 품질평가 매트릭은 반드시 있어야 할 요소이다. 예를 들어, 시스템에 대해 시험 단계에서 날짜별로 오류 수를 추정한다는 상황이라고 가정한다. 발견된 오류는 즉시 제거되고 새로운 에러를 만들어내지 않는다고 한다면, 오류가 발생하는 시간은 독립적이어야 한다. 포아송 분포의 평균값

으로 단위 시간당 발생한 오류 수를 기준으로 고장시간을 구한다. 구해진 고장시간의 값으로 총 오류 수와 고장 강도를 예측할 수 있다. 측정되지 않은 값은 신뢰도 함수에 대입하여 신뢰도를 측정하고, 남아 있는 오류 수에 대하여 그 다음에 발생할 평균오류시간을 구한다. 각 결과를 계속해서 모델에 적용하면 시간이 지날수록 고장 수가 감소하는 현상을 볼 수 있다.

3.1 성숙성

성숙성은 소프트웨어 내의 결함으로 인한 고장을 피해가는 능력이다. 예로써, 전체적인 자동차 사용 결함 체크리스트에서 사용자가 자동차를 사용하는 도중 생긴 결함의 수가 어느 정도인지를 평가하는 방법이 있을 수 있다. 아래의 식은 체크리스트 기준으로 하여 결함발생밀도를 나타내는 계산식이다.

$$\text{결함발생밀도} = 1 - \text{결함 수} / \text{총 평가 수}$$

신뢰성의 부특성 중 성숙성 품질평가 매트릭에서 결함발생밀도를 구하기 위한 결함 수와 총 평가 수는 Jelinski-Moranda Model을 통해서 알아낼 수 있다.

2.4.2절에서 소개한 수식을 토대로 하여, ISO/IEC 9126에서 제시하고 있는 소프트웨어 신뢰성 품질평가 매트릭으로 예상잠재 고장 밀도를 측정한 후 신뢰성의 부특성 중 하나인 성숙성을 평가할 수 있으며, 테스트 기간 내에 평가되어 밝혀진 발생 고장에 대하여 소프트웨어 품질을 평가할 수 있다.

3.2 오류허용성

오류허용성은 소프트웨어 결함이 발생했을 때 명세된 성능 수준을 유지할 수 있는 능력이다. 시스템의 재시동이 필요한 시점에서의 회피율이라든지, 사용자의 오조작으로 인해 생기는 오류를 미연에 방지하는 회피율의 결함 발생 수를 낮추는 것과 같은 작업을 일컫는다.

$$\text{회피율} = 1 - \text{발견 결함 수} / \text{총 결함 수}$$

신뢰성의 부특성 중 오류허용성의 품질평가 매트릭에서 회피율을 구하기 위한 발견 결함 수와 총 결함 수는 NHPP(Non-Homogeneous Poisson Process) Model을 통해서 알아낼 수 있다.

일례로, ISO/IEC 9126에 제시된 신뢰성의 부특성 중 하나인 오류허용성에서 발견 결함수를 파악하고 그것을 회피하기 위한 척도로 중요하게 활용될 수 있다.

2.4.1절에서 소개한 수식을 토대로 하여, 테스트 시각 t내에 발견되는 n개의 에러로써 테스트한다는 전체 하에 기대 잔존치 을 이용해 테스트 시각 t내 발생 결함을 파악하고 고장을 평가할 수 있다.

4. 연구 평가

내장형 소프트웨어의 실행 과정에서 발생한 고장 자료는 신뢰성 평가에 있어 중요한 근거를 제공하고, 많은 도움을 준다. 소프트웨어 신뢰성 모델은 고장 자료의 결과 형태로 고장 현상을 기술하는데, 확률론적 신뢰성 모델인 NHPP(Non-Homogeneous Poisson Process) Model과 Jelinski-Moranda Model을 이용하면 내장형 소프트웨어의 신뢰성 모델링이 가능하다.

기존의 연구는 신뢰성 품질평가 메트릭과 신뢰성 모델의 개념이 따로 적용되어 연관성을 찾아볼 수 없었지만, 본 논문에서는 연구를 통하여 두 가지에 대한 연관성을 찾고 신뢰성을 측정하는 방안에 대해 소개하였다. 연구 결과, 신뢰성 품질평가 메트릭과 신뢰성 모델은 밀접한 관계가 있다는 것이 밝혀졌다. 신뢰성 모델에서 근거하고 있는 내용이 신뢰성 품질평가 메트릭에 적용되어, 신뢰성 품질평가 메트릭은 신뢰성 모델의 예측 데이터로 신뢰성을 측정할 수 있다.

이처럼 신뢰성 품질평가 메트릭과 신뢰성 모델 두 가지를 연관시킴으로써 보다 나은 신뢰성 평가 척도 방안에 이바지할 수 있게 되었다.

5. 결론 및 향후 연구과제

급변하는 사회에서 내장형 시스템의 중요성은 나날이 증대되고 있고, 그만큼 신뢰성 또한 중요성이 대두되고 있는 실정이다. 특히나 위험요소가 가미되어 있는 부분에서는 안전성과 직결되어 있기 때문에 신뢰성에 대한 관심도가 점점 높아지고 있는 추세이다.

본 연구에서는 궁극적으로 신뢰성을 향상시키기 위해 효율적인 신뢰성 측정 방안에 대하여 모색하였다. 신뢰성의 부특성 4가지 중 성숙성과 오류허용성에 대한 품질평가 메트릭에 대해서 기존의 대표적인 신뢰성 모델을 적용할 수 있음을 설명하였고, 종속성이 없던 품질평가 메트릭과 신뢰성 모델을 연관시켜 종속관계가 되도록 하였다.

향후 연구에서는 실제 내장형 시스템을 모델링 및 구현하고, 시스템에 대하여 본 논문에서 다뤘던 신뢰성 측정 메트릭을 적용해보고자 한다. 신뢰성 평가를 진행하기 위해 실제 시스템을 분석하여 더욱 정확한 데이터로 성숙성, 오류허용성의 품질평가 메트릭과 연관시킨 신뢰성 모델 두 가지를 심도 있게 연구할 계획이다. 아울러, 적용 결과를 통해 연관성의 정확한 근거를 제시할 것이다. 이를 실질적으로 적용한다고 했을 때, 실제 시스템에서 신뢰성을 높이는 작업에 적극 활용할 수 있도록 해야 할 것이다.

참 고 문 헌

- [1] ISO/IEC 9126 "Information Technology - software Quality Characteristics and Metrics - Part 1,2,3
- [2] ISO/IEC 12119 "Software Packages - Quality requirement and Testing"

- [3] ISO/IEC 14598, "Information Technology Software Product Evaluation-Part 1,2,3,4,5,6.
- [4] Jelinski, Z. and Moranda, P.B, "Software reliability research", in Statistical Computer Performance Evaluation, Freiberger, W.(ed.), pp. 465-484, Academic Press, 1972.
- [5] NIPA, "Software Technique Evaluation Standards(MKE 2010-53ed), 2010.05.
- [6] The laws and regulations and formulas, <http://www.nipa.kr>
- [7] TTAS, "Web-based Software Quality Evaluation Guideline", TTAS.KO-11.0059.
- [8] TTA, <http://www.tta.or.kr/index.jsp>.