

COTS 하드웨어 컴포넌트 기반 임베디드 소프트웨어 신뢰성 모델링

(Embedded Software Reliability Modeling with COTS Hardware Components)

구 태 완 [†] 백 종 문 ^{**}
(Taewan Gu) (Jongmoon Baik)

요 약 최근 IT 산업은 국방, 항공, 자동차, 의료와 같은 전통 산업분야와 서로 융합하는 추세이다. 그러므로 시스템의 하드웨어를 주로 담당하는 임베디드 소프트웨어는 높은 신뢰성, 가용성, 유지보수성이 보장되어야 한다. 이를 위해 최근 COTS (Commercial Off The Shelf) 하드웨어 컴포넌트 기반 임베디드 소프트웨어를 개발하는 추세이다. 그러나 이러한 개발방법에는 일반적 소프트웨어 결함 외에 하드웨어와의 상호작용에 기인하는 결함이 추가적으로 발생할 수 있다. 이를 연동결함(Linkage Fault)라고 정의한다. 이는 발생 빈도가 낮음에도 불구하고 전체 시스템의 중단을 야기할 정도로 위험하다. 본 논문에서는 COTS 하드웨어 컴포넌트 기반 임베디드 소프트웨어 개발 시 이러한 연동결함의 발생을 고려한 신뢰성 모델을 제안한다. 또한 제안된 모델의 타당성을 분석하기 위해 베이지안 분석과 마코프 체인 몬테카를로 방법으로 계산한 베이스 요인을 이용한다. 끝으로 IT 융합 분야의 실제 데이터를 활용하여 제안된 모델의 이론적 결과를 뒷받침한다.

키워드 : 임베디드 소프트웨어, 임베디드 소프트웨어 신뢰성, 임베디드 소프트웨어 신뢰성 모델, COTS 하드웨어, 베이지안 분석

Abstract There has recently been a trend that IT industry is united with traditional industries such as military, aviation, automobile, and medical industry. Therefore, embedded software which controls hardware of the system should guarantee the high reliability, availability, and maintainability. To guarantee these properties, there are many attempts to develop the embedded software based on COTS (Commercial Off The Shelf) hardware components. However, it can cause additional faults due to software/hardware interactions beside general software faults in this methodology. We called the faults, Linkage Fault. These faults have high severity that makes overall system shutdown although their occurrence frequency is extremely low. In this paper, we propose a new software reliability model which considers those linkage faults in embedded software development with COTS hardware components. We use the Bayesian Analysis and Markov Chain Monte-Carlo method to validate the model. In addition, we analyze real linkage fault data to support the results of the theoretical model.

Key words : Embedded Software, Embedded Software Reliability, Embedded Software Reliability Model, COTS Hardware, Bayesian Analysis

· 이 논문은 2008년 정부(교육과학기술부)의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구임 (KRF-2008-313-D009321)

[†] 정 회 원 : 한국과학기술원 정보통신공학과 연구 교수

gutaewan@kaist.ac.kr

^{**} 종신회원 : 한국과학기술원 전산학과 교수

jbaik@kaist.ac.kr

논문접수 : 2009년 5월 18일

심사완료 : 2009년 6월 10일

Copyright©2009 한국정보과학회 : 개인 목적이거나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지: 소프트웨어 및 응용 제36권 제8호(2009.8)

1. 서론

임베디드 소프트웨어는 자동차, 국방, 항공, 조선, 의료, 건설 등의 전통적 산업 분야와의 IT 융합에서 하드웨어 제어 등을 담당한다. 그러므로 때로는 이들 소프트웨어 오동작은 치명적인 결과를 야기할 수 있으므로 IT 융합에서 임베디드 시스템은 높은 신뢰성, 가용성, 유지보수성 등을 보장할 수 있어야 한다. 하지만 짧은 개발 시간 및 제한된 개발 자원으로 인해 임베디드 시스템의 성능 및 신뢰성을 동시에 보장하기에 많은 어려움이 있다[1,2]. 임베디드 소프트웨어 개발에 있어 짧은 개발시

간 및 제한된 개발자원과 같은 문제들을 해결하기 위해 최근, 특히 국방 분야에서 COTS(Commercial Off The Shelf) 하드웨어 컴포넌트 기반 임베디드 시스템의 필요성이 증대되고 있으며, 그에 따른 장점으로는 개발 비용 감소, 높은 가용성 확보, 출시 시간 단축 등이 있다 [3-5]. 따라서 본 논문의 필요성(Motivation)으로 (1) 임베디드 소프트웨어의 패러다임 변화에 따른 COTS 하드웨어 컴포넌트의 특징 및 그에 따른 결함을 평가하기 위한 신뢰성 모델링에 관한 연구가 현재 거의 이루어지지 않았고, (2) 상용 제품으로서의 COTS 하드웨어 컴포넌트는 임베디드 소프트웨어 개발 프로젝트별 특성을 모두 반영할 수 없기 때문에 기존 소프트웨어 신뢰성 모델을 동일하게 적용하는 것이 무리가 있으며, 그리고 (3) COTS 하드웨어 컴포넌트 기반 임베디드 소프트웨어 개발에는 부가적인 결함이 발생할 수 있으므로, 이러한 부가적 결함을 연동결함(Linkage Fault)이라 정의하면, COTS 하드웨어 컴포넌트 기반 임베디드 소프트웨어 신뢰성 모델은 연동결함 발생빈도를 반영하여야 한다. 이때 연동결함의 예로는 일정한 시그널 전송 범위를 갖는 COTS 하드웨어 컴포넌트에 대해 다수의 소프트웨어 모듈에서 동시 발생하는 시그널을 프로그래머가 미처 인식하지 못해 혹은 협업으로 인한 의소소통의 미숙 등의 문제로 인해 시그널 전송 범위가 초과될 수 있다. 이로 인해 각 소프트웨어 모듈에서 발생하는 시그널에 대한 전송 제어의 어려움으로 전체 시스템에 장애가 발생하거나 혹은 시그널 전송 자체가 불가능하여 전체 시스템 운영을 마비시키는 문제가 발생할 수 있다. 이러한 문제들로 인해 COTS 하드웨어 컴포넌트 기반 임베디드 소프트웨어에서 연동결함을 고려한 소프트웨어 신뢰성 모델링 기법에 관한 연구가 필요하다.

기존 연구들[1,2,6-12]은 주로 결함수를 기반으로 하는 결함수 기반 모델(Fault Count Model 또는 Interval Domain Model)과 결함이 발생하기까지의 시간을 기반으로 하는 시간 도메인 모델(Time Domain Model)로 구분하여 순수 소프트웨어에 대한 신뢰성 모델을 정의하고 있다. 그러나 본 논문에서 제안하는 신뢰성 모델은 이러한 기존 연구들과 달리 COTS 하드웨어 컴포넌트를 기반으로 하는 임베디드 소프트웨어에서 발생 가능한 연동결함 발생 빈도를 추정하여 전체 임베디드 소프트웨어 신뢰성 모델에 반영한다는 면에서 기존 연구와의 차별성을 갖는다.

일반적으로 순수 소프트웨어의 실패빈도는 전통적인 포아송 모델(Poisson Model)이나 음이항 모델(Negative Binomial Model)을 이용하여 설명 가능하다. 그러나 실제 수집된 데이터에 따르면 연동결함의 빈도가 매우 낮기 때문에 전통적인 포아송 모델이나 음이항 모델

만으로는 발생 빈도가 '0'인 지점을 적절하게 분석 및 설명하지 못하는 한계가 있다[13].

본 논문에서는 이러한 문제를 해결하기 위해 (1) 두 개의 영과잉 모델을 활용한다. 이들 두 모델은 우선 영(Zero)에 대한 위치 확률을 가지는 분포와 포아송 분포 또는 음이항 분포를 합성하여 영과 영이 아닌 데이터를 설명하기 위해 영과잉 포아송 모델(Zero-Inflated Poisson Model, ZIP)과 영과잉 음이항 모델(Zero-Inflated Negative Binomial Model, ZINB)이며, 이들을 이용하여 COTS 하드웨어 컴포넌트 기반 임베디드 소프트웨어 신뢰성 모델을 정의한다. 또한 (2) 두 개의 영과잉 모델 중 COTS 하드웨어 컴포넌트 기반 임베디드 소프트웨어의 연동결함의 영향을 평가하고 연동결함 발생 빈도를 보다 잘 반영하는 모델을 선택한다. 이를 위해 베이저안 분석(Bayesian Analysis)[14]과 마코프 체인 몬테카를로(Markov Chain Monte Carlo, MCMC) 방법으로 계산한 베이즈 요인(Bayes Factor)을 이용한다. 그런 다음 (3) 제안된 모델을 이용하여 실제 IT 융합 분야의 데이터를 이용하여 제안된 모델의 이론적 결과를 뒷받침 한다.

본 논문의 구성은 2장에서 기존 임베디드 시스템 신뢰성 모델 방법과 임베디드 시스템에서 발생할 수 있는 결함, 그리고 베이즈 분석에 대해 알아보고, 3장에서는 본 논문에서 제안하는 COTS 하드웨어 컴포넌트 기반 임베디드 시스템의 신뢰성 모델을 설명한다. 이때 제안하는 모델의 유도는 Jang et al.[15]의 연구와 동일한 기법을 활용한다. 4장에서는 제안된 신뢰성 모델 및 문제 해결 방안을 검증하기 위해 IT 융합 분야에서 COTS 하드웨어 컴포넌트 기반 임베디드 시스템 개발에 의해 얻어진 실제 데이터를 활용하여 검증하고 마지막으로 5장에서는 결론 및 향후 연구과제에 대해 논의한다.

2. 관련연구

2.1 연동결함 실패에 대한 영과잉 포아송 모델(ZIP) 기반 연동결함 및 영과잉 음이항 모델(ZINB) 기반 연동결함 유도함수

일반 소프트웨어에서 발생하는 결함은 포아송 모델이나 음이항 모델을 이용하여 발생빈도의 예측이 가능하다 [16]. 그러나 연동결함의 경우 소프트웨어 개발 프로세스가 잘 정립된 개발 조직에서는 그 발생 빈도가 극히 낮아진다. 그러나 낮은 빈도를 갖는 연동결함은 전통적인 포아송 모델이나 음이항 모델만으로는 적절하게 분석 및 설명하지 못하는 한계가 있다[13]. 그러므로 본 논문에서는 연동결함의 영향을 평가하기 위해 두 개의 영과잉 모델을 활용한다. 이들 두 모델은 우선 영(Zero)에 대한 위치 확률을 가지는 분포와 포아송 분포 또는 음이항 분포

를 합성하여 영과 영이 아닌 데이터를 설명하기 위해 영과 영 포아송 모델(ZIP)과 영과 영 음이항 모델(ZINB)이며, 이들을 이용하여 COTS 하드웨어 컴포넌트 기반 임베디드 소프트웨어 신뢰성 모델을 정의한다.

우선 영과 영 포아송 모델의 경우 $X = \{x_1, \dots, x_n\}$ 가 서로 독립인 영과 영 포아송 분포를 갖는 확률 변수라고 할 때, 영(Zero)은 별개의 두 상태에서 발생한다는 것을 가정한다. 첫 번째 상태는 확률 ω 를 갖고 영에서 점 질량(Point Mass)을 가지는 분포에서 발생한다는 것이고, 두 번째 상태는 $1-\omega$ 의 확률을 갖고 평균이 λ 인 포아송 분포에서 발생한다는 것이다. 일반적으로 첫 번째 상태에서 발생하는 “구조적 영”이라고 하고, 포아송 분포에서 발생하는 영을 “샘플링 영”이라고 한다. 두 상태를 혼합한 분포인 X 의 확률 질량 함수는 다음과 같다.

$$P(x_i; \lambda, \omega) = \begin{cases} \omega + (1-\omega)e^{-\lambda}, & x_i = 0, \\ (1-\omega) \frac{e^{-\lambda} \lambda^{x_i}}{x_i!}, & x_i = 1, 2, \dots \end{cases} \quad (1)$$

그리고 식 (1)에 대한 평균은 $(1-\omega)\lambda$ 이고, 분산은 $(1-\omega)\lambda + \omega(1-\omega)\lambda^2$ 가 되므로 $0 < \omega \leq 1$ 이면, 분산이 평균보다 큰 과분산(Over-Dispersion)이 발생하게 된다. 이것은 평균과 분산이 같은 일반적 포아송 모델에 비해 ZIP 모델이 연동결함 빈도의 특성에 잘 부합한다고 할 수 있다. 그러므로 ZIP 기반 연동결함 분포에서 영에 대한 우도함수는 i 개의 관측값 $Y = \{y_1, \dots, y_i\}$ 가 주어질 때 영과 영 포아송 모델의 우도함수 $L(\omega, \beta)$ 는 다음 식 (2)와 같이 정의된다. 여기서 λ 는 일반적 포아송 분포의 평균으로 정한다. 그리고 β 는 임의의 모수벡터이다. 이때 I 는 식 (1)에 따른 지시함수(Indication Function)이다.

$$L(\omega, \beta) = \prod_{i=1}^n \left[I_{(y_i=0)} \{ \omega + (1-\omega)e^{-\lambda} \} + I_{(y_i>0)} (1-\omega) \frac{e^{-\lambda} \lambda^{y_i}}{y_i!} \right] \quad (2)$$

두 번째로 영과 영 음이항 모델의 경우, X 에 대한 확률 질량 함수는 다음 식 (3)과 같다.

$$P(x_i; \lambda, \omega) = \begin{cases} \omega + (1-\omega) \left(\frac{\theta}{\theta + \lambda} \right)^\theta, & x_i = 0, \\ (1-\omega) \frac{\Gamma(\theta + x_i)}{\Gamma(\theta) \Gamma(x_i + 1)} \left(\frac{\theta}{\theta + \lambda} \right)^\theta \left(\frac{\lambda}{\theta + \lambda} \right)^{x_i}, & x_i = 1, 2, \dots \end{cases} \quad (3)$$

이때 $0 \leq \omega \leq 1$ 이고, $\theta (0 \leq \theta < \infty)$ 는 과분산 계수(Over-Dispersion Coefficient)의 역수이다. 그러므로 식 (3)을 기반으로 영과 영 음이항 모델의 우도함수 $L(\omega, \beta, \theta)$ 은 정의하면 다음 식 (4)와 같다.

$$L(\omega, \beta, \theta) = \prod_{i=1}^n \left[I_{(y_i=0)} \left\{ \omega + (1-\omega) \left(\frac{\theta}{\theta + \lambda} \right)^\theta \right\} + I_{(y_i>0)} (1-\omega) \frac{\Gamma(\theta + y_i)}{\Gamma(\theta) \Gamma(y_i + 1)} \left(\frac{\theta}{\theta + \lambda} \right)^\theta \left(\frac{\lambda}{\theta + \lambda} \right)^{y_i} \right] \quad (4)$$

2.2 임베디드 시스템의 실패와 신뢰성 모델링

임베디드 시스템에서 발생할 수 있는 실패의 종류를 나타내는 그림 1에 따르면 임베디드 시스템의 실패는 순수 하드웨어에 의한 실패(Pure HW Failures), 순수 소프트웨어에 의한 실패(Pure SW Failures), 그리고 하드웨어-소프트웨어 상호작용에 의해 발생할 수 있는 실패(HW-SW Interaction)가 있으며, 이를 다시 영속적 실패(Permanent Failures)와 일시적 실패(Transient Failures)로 구분하고 있다[17].

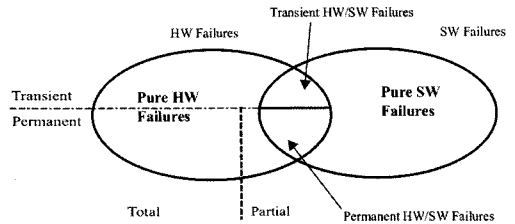


그림 1 임베디드 시스템 실패 분류

그러나 COTS 컴포넌트를 활용함에 따라 순수 하드웨어에 의한 실패 수는 많이 감소 한 반면, COTS 컴포넌트와 순수 소프트웨어 간 상호작용에 따른 추가적 결함, 즉 연동결함이 발생하게 되는 문제가 존재한다. 또한 연동결함의 빈도가 매우 낮다 할지라도 임베디드 소프트웨어에서 하드웨어와의 연동결함은 하드웨어의 제거가 어려워지거나 불가능해지고, 순수 소프트웨어 기능 수행에 장애가 될 수 있다는 문제가 있다.

지난 십여 년간 임베디드 시스템의 신뢰성을 측정하고 예측하기 위한 많은 연구들이 진행되었다[1,2,6-12]. 그러나 Schneidewind[8]의 연구와 Irene et al.[18] 등의 연구와 같이 기존 임베디드 시스템 신뢰성 관련 연구에서는 하드웨어와 소프트웨어가 각기 따로 신뢰성 모델이 연구되었기 때문에 최근의 임베디드 시스템 개발에 따른 신뢰성 측정 및 평가를 위한 모델링에 한계가 있다[10]. 이러한 한계를 극복하기 위해 최근 하드웨어와 소프트웨어를 모두 고려한 신뢰성 모델에 관한 연구도 진행되었다[10-12, 19]. 특히 Teng et al.[11]의 연구에 따르면 복잡한 임베디드 시스템의 신뢰성을 모델링함에 있어 시스템 실패(System Failure)를 순수 소프트웨어 실패와 순수 하드웨어 실패, 그리고 하드웨어-소프트웨어 상호작용에 의한 실패로 나누어 이들을 조합한 신뢰성 모델을 제안하였다. 이때 소프트웨어 실패는 NHPP 모델을 이용하여 모델링하였고[17], 하드웨어 실패는 WEIBULL 모델을 이용하여 모델링 하였다[20]. 그러나 Teng et al.[11]의 연구는 COTS 하드웨어 컴포넌트 기반 임베디드 시스템 개발의 특성을 반영하지 못

하므로, 이들 연구들만으로는 COTS 하드웨어 컴포넌트 기반 임베디드 시스템의 신뢰성을 측정하고 평가하는 신뢰성 모델링이 적합하지 않다.

2.3 COTS(Commercial Off The Shelf) 컴포넌트

COTS는 사용자가 구매하여 별도의 수정 없이 쉽게 사용할 수 있는 일반 대중에게 유용한 상용제품 또는 서비스를 말한다. COTS는 결정된 가격에 따라 효율적으로 구입할 수 있으며, 추가적인 개발 비용이나 시간 투자 없이 그대로 가용한 제품 또는 서비스이다. 이러한 COTS의 정의는 “일반 대중에게 팔고, 빌려주고, 허용되며 이를 통해 수익을 얻으려는 판매자에 의해 제공되고, 지적재산권을 소유한 판매자에 의해 유지보수 및 진화되며 소비자에 의해 내부적인 수정 없이 사용되는 제품”을 의미한다[21]. 그러므로 COTS 하드웨어 컴포넌트를 기반으로 하는 임베디드 시스템의 경우 하드웨어에서 발생할 수 있는 많은 문제들이 사전에 해결되어 순수 하드웨어 실패를 현저하게 감소시킬 수 있다. 그러나 임베디드 소프트웨어를 개발함에 있어 발견되는 결함은 일반적 소프트웨어의 결함 외에 하드웨어와 연동시 발생하는 연동결함이 발생하게 된다. 그러나 소프트웨어 개발 체계가 잘 정립된 조직에서도 연동결함이 발생할 수 있지만 그 빈도는 매우 낮음에도 불구하고 하드웨어와 연동시 발생하는 연동결함은 전체 시스템 운영의 중단을 야기 할 수 있는 높은 심각도를 갖는 결함이라 할 수 있다. 그러므로 COTS 하드웨어 컴포넌트 기반 임베디드 소프트웨어 신뢰성 모델은 이러한 연동결함 발생을 고려하여야 한다.

2.4 베이지안 분석 (Bayesian Analysis)

베이지안 분석은 기존 분석방법에 비해 유용한 사전 정보의 사용을 가능하게 할 뿐 아니라, 표본의 크기작을 때 상대적으로 더 신뢰적인 분석을 할 수 있는 장점을 갖고 있다[22]. 베이지안 추론의 주된 목적은 관심 있는 모수의 사후 분포를 사전분포와 우도함수(Likelihood Function)로부터 구하는 것이다. 이 추론 과정은 대부분의 경우, 복잡한 적분 계산을 요구하고 있기 때문에 베이지안 분석 방법은 우도함수와 사전분포가 특정 관계가 성립하는 몇몇 경우에 대해서만 국한적으로 사용되어 왔으며, 실제 자료를 분석에는 많은 어려움이 있었다. 그러나 최근에는 사후분포의 기대값을 수치적으로 구할 수 있는 마코프 체인 몬테카를로 방법이 널리 사용됨에 따라 베이지안 분석 방법은 여러 분야에서 사용하게 되었다. MCMC 방법의 적용은 적분을 필요로 하지 않으며, 모수들의 완전 조건부 사후분포로부터 랜덤 변수를 생성하고 모수들의 추정값을 구할 수 있다. 각 모수에 대한 완전 조건부 사후분포를 구하여 각각 모수를 추정함으로써 베이지안 추론 방법을 이용하여 정확

한 추론을 할 수 있다[22]. 본 논문에서는 모수 추정을 위해 베이지안 추론에서 널리 활용되는 MCMC 기법 중 하나인 깁스 샘플러(Gibbs Sampler)를 활용하여 모수 추정을 수행한다. 깁스 샘플러는 두개 이상의 확률 변수의 결합 확률 분포로부터 일련의 표본을 생성하는 알고리즘이다. 이것은 히스토그램을 생성하기 위해 결합 분포를 근사하거나 예측치와 같은 적분값을 계산하기 위한 목적으로 사용되며, 메트로폴리스-해스팅 알고리즘(Metropolis-Hastings Algorithm)의 특별한 예이다[23].

3. COTS 하드웨어 컴포넌트 기반 임베디드 소프트웨어 신뢰성 모델

3.1 모델 정의를 위한 가정(Assumption)

COTS 하드웨어 컴포넌트 기반 임베디드 소프트웨어 신뢰성 모델을 정의하기 위해 다음과 같은 사항을 가정한다.

- A1** 소프트웨어가 하드웨어와 연동되어 발생하는 추가적 결함을 연동결함(Linkage Fault)이라고 정의한다.
- A2** COTS 하드웨어 컴포넌트 기반 임베디드 소프트웨어에서 연동결함의 수를 이산적 영과잉 데이터(Discrete Zero-Inflated Data)로 관측된다.
- A3** 순수 소프트웨어에서 결함의 발견에 따른 평균값 함수를 $m_{sf}(t)$ 라고 하며 $m_{sf}(t)$ 는 S 모양의 골곡을 갖는 비감소 함수이다.
- A4** COTS 하드웨어 컴포넌트의 결함은 미리 제거되었다고 가정하고, 이후에 발생하는 결함은 소프트웨어 결함에 독립적이다.
- A5** 임베디드 소프트웨어에서 연동결함으로 인한 실패는 전체 시스템의 실패 확률에 높은 영향을 미친다.

3.2 임베디드 소프트웨어 신뢰성 모델링

임의의 시간 t 에서 전체 임베디드 시스템의 신뢰성 R_{ES} 는 다음 식 (5)와 같다. 이때 $R_{sf}(t)$ 는 순수 소프트웨어 신뢰성, R_C 는 추후 발생할 수 있는 COTS 하드웨어 컴포넌트의 신뢰성을 의미한다. 본 논문에서는 가정 **A4**에 의해 COTS 하드웨어 컴포넌트 자체의 결함 R_C 에 대해서는 따로 고려하지 않는다.

$$R_{ES}(t) = R_{sf}(t) \times R_C(t) \quad (5)$$

이때 순수 소프트웨어 신뢰성 $R_{sf}(t)$ 는 가정 **A3**에 의해 다음 식 (6)과 같이 나타낼 수 있다.

$$R_{sf}(t) = e^{-m_{sf}(t)} \quad (6)$$

여기서 $m_{sf}(t)$ 는 다음 식 (7)과 같고, 이때 a 는 소프트웨어 초기 예상 결함수, c 는 결함당 평균 실패율, β 는 디버깅시 새로 발견될 결함 확률, p 는 결함이 성공적으

로 제거될 확률, α 는 형상 모수, 그리고 b 는 척도 모수를 나타낸다[17].

$$m_{sf}(t) = \frac{a}{p-\beta} \left(1 - \left(\frac{(1+\alpha)e^{-bt}}{1+\alpha e^{-bt}} \right)^{\frac{c}{b}(p-\beta)} \right) \quad (7)$$

그러나 COTS 하드웨어 컴포넌트 기반 임베디드 소프트웨어의 경우 발생 가능한 연동결함을 고려해야 하고, 또한 가정 A5에 따라 연동결함으로 인한 실패가 전체 시스템의 실패 확률에 높은 영향을 미치므로 COTS 하드웨어 컴포넌트 기반 임베디드 소프트웨어 신뢰성 모델은 연동결함 발생확률 $P(lf)$ 를 고려하여 다음 식 (8)과 같이 재정의 한다. 식 (8)에 따르면 연동결함 발생확률 $P(lf)$ 는 $m_{sf}(t)$ 에 영향을 미치는 요인이며, 이를 위해 $P(lf)$ 가 R_{ES} 에 대해 지수적으로 표현되어 전체 임베디드 소프트웨어 신뢰성 R_{ES} 에 영향을 미치게 된다.

$$R_{ES}(t) = (e^{-m_{sf}(t)})^{P(lf)} \quad (8)$$

그리고 식 (8)을 계산하기 위해 연동결함 발생 확률 $P(lf)$ 는 점 질량을 갖는 분포의 평균을 ϵ , 구조적 영의 비율을 τ , 포아송 분포 또는 음이항 분포의 평균을 λ , 샘플링 영의 비율을 ν 이라 하면 다음 식 (9)와 같이 된다. 이때 λ 는 베이지 요인 분석에 따라 선택되어 진다. 또한 식 (9)는 전체 연동결함 발생 확률에서 구조적 영의 비율과 샘플링 영의 비율의 합을 뺀 값으로 정의한다.

$$P(lf) = 1 - (\pi + \nu\lambda) \quad (9)$$

이러한 연동결함 발생확률 $P(lf)$ 를 계산하기 위해 두 개의 영과잉 모델을 적용하며, 가정 A1과 A2에 의해, 연동결함은 두 영과잉 모델에 적용 가능한 관측 데이터가 된다.

3.3 베이지 추론을 이용한 영과잉 모델 선택

베이지 추론은 가설이 참일 확률을 갱신하거나 새롭게 추론하기 위한 증거로 사용되는 통계적 추론 기법이다. 그러므로 본 논문에서는 두 영과잉 모델에 대해 베이지 추론을 이용한다. 이를 위해 각 영과잉 모델에 대한 사전분포를 가정하고 사후분포를 계산한다.

우선 ZIP 모델에 대한 2개의 모수(Parameter) ω 와 β 에 각각 독립적으로 베타분포(Beta Distribution)와 다변량 정규 분포를 가정한다. ω 에 대해 베타분포의 두 모수를 a 와 b 로 가정하고, β 에 대해서는 평균이 μ 이고 분산이 Σ 인 다변량 정규분포를 가정한다. 그러므로 ZIP 모델의 사전분포는 다음 식 (10)과 같이 가정한다.

$$\pi(\omega) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \omega^{a-1} (1-\omega)^{b-1}, \quad (10)$$

$$\pi(\beta) = \frac{1}{(2\pi\Sigma)^{\frac{1}{n}}} e^{-\frac{1}{2}(\beta-\mu)^T \Sigma^{-1}(\beta-\mu)}$$

그리고 ZIP의 사후분포는 우도함수와 사전분포의 곱

에 비례하므로 ZIP 모델의 사후분포는 다음 식 (11)과 같이 계산 가능하다.

$$\begin{aligned} \pi(\omega, \beta | y) \propto & \prod_{i=1}^n \left[I_{(y_i=0)} \left\{ \omega + (1-\omega)e^{-\lambda} \right\} \right. \\ & \left. + I_{(y_i>0)} (1-\omega) \frac{e^{-\lambda} \lambda^{y_i}}{y_i!} \right] \\ & \times \omega^{(a-1)} (1-\omega)^{(b-1)} e^{-\frac{1}{2}(\beta-\mu)^T \Sigma^{-1}(\beta-\mu)} \end{aligned} \quad (11)$$

그리고 ZINB 모델에 대해서는 3개의 모수 ω , β , 그리고 θ 에 각각 독립적으로 베타분포, ZIP 모델에 대한 2개의 모수(Parameter) ω 와 β 에 각각 독립적으로 베타분포(Beta Distribution)와 다변량 정규 분포를 가정한다. ω 에 대해 베타분포의 두 모수를 a 와 b 로 가정하고, β 에 대해서는 평균이 μ 이고 분산이 Σ 인 다변량 정규 분포를 가정한다. 다변량 정규분포, 그리고 균일분포를 가정한다. ω 와 β 에 대해서는 ZIP 모델과 같은 사전분포를 가진다고 가정한다. 따라서 ZINB 모델의 사후분포는 다음 식 (12)와 같이 계산 가능하다.

$$\begin{aligned} \pi(\omega, \beta, \theta | y) \propto & \prod_{i=1}^n \left[I_{(y_i=0)} \left\{ \omega + (1-\omega) \left(\frac{\theta}{\theta+\lambda} \right)^\theta \right\} \right. \\ & \left. + I_{(y_i>0)} (1-\omega) \frac{\Gamma(\theta+y_i)}{\Gamma(\theta)\Gamma(y_i+1)} \left(\frac{\theta}{\theta+\lambda} \right)^\theta \left(\frac{\lambda}{\theta+\lambda} \right)^{y_i} \right] \\ & \times \omega^{(a-1)} (1-\omega)^{(b-1)} e^{-\frac{1}{2}(\beta-\mu)^T \Sigma^{-1}(\beta-\mu)} \end{aligned} \quad (12)$$

임베디드 소프트웨어에서의 연동결함에 의한 실패 수를 z 라고 하고, ZIP를 $M_1(H_0)$, ZINB를 $M_2(H_1)$ 이라고 하면, COTS 하드웨어 컴포넌트 기반 연동결함 분포에서의 영의 비율을 설명하기 위해 두 가설 H_0 와 H_1 을 검정한다. 우선 다음 식 (13)은 베이지 정리를 이용하여 두 모델 M_k 의 사후 확률을 나타낸다.

$$P(M_k | z) = \frac{P(z|M_k)P(M_k)}{P(z|M_1)P(M_1) + P(z|M_2)P(M_2)}, \quad k=1,2 \quad (13)$$

여기서 $P(z|M_k)$ 는 M_k 모델이 주어졌을 때 z 에 대한 주변 확률(Marginal Probability)이 되며 θ_k 를 M_k 모델의 모수, $P(\theta_k|M_k)$ 를 모수의 사전분포, 그리고 $P(z|\theta_k, M_k)$ 를 M_k 모형과 모수 θ_k 가 주어져 있을 때 z 에 관한 확률 밀도 함수라고 하면 주변 확률 $P(z|M_k)$ 의 값은 다음 식 (14)와 같이 모수 θ_k 에 대한 적분 값을 이용해서 얻을 수 있다.

$$\int P(z|\theta_k, M_k) P(\theta_k|M_k) d\theta_k \quad (14)$$

그러므로 연동결함에 의한 실패 수 z 가 M_1 과 M_2 중 어느 것을 선호하는지는 M_1 과 M_2 의 사후 분포 비율로 측정할 수 있다. 이 비율은 사후 확률로부터 구할 수 있으며, 식 (13)에 의해 다음 식 (15)와 같이 사후 확률을

계산할 수 있다. 그리고 식 (15)의 우변의 첫번째 항은 두 모델의 예측확률이 되므로 베이즈 요인이 되며, 이러한 베이즈 요인의 해석은 다음 표 1에 따라 해석이 가능하다[22].

$$\left[\frac{P(M_2|z)}{P(M_1|z)} \right] = \left[\frac{P(z|M_2)}{P(z|M_1)} \right] \left[\frac{P(M_2)}{P(M_1)} \right] \quad (15)$$

그리고 베이즈 요인을 계산하기 위해 z에 대한 주변 확률로부터 주변 우도함수에 대해 몬테카를로 적분 추정값을 구하여 식 (15)의 계산에 이를 활용한다. 몬테카를로 적분 추정값은 우도함수 값의 조화 평균이 되므로, 사전분포의 표본의 수가 무한대에 가까워질수록 추정값은 실제값에 수렴하고, 이것은 일반적인 중심극한정리를 만족하게 된다[15].

표 1 베이즈 요인의 해석

베이즈 요인 값 (B)	M ₁ 보다 M ₂ 를 선호
1 ≤ B ≤ 3.2	선호하지만, 언급할 가치는 없다.
3.2 ≤ B ≤ 10	유의하다.
10 ≤ B ≤ 100	강력하다.
B > 100	결정적이다.

3.4 모수 추정

주어진 베이지안 추론에서 널리 활용되는 MCMC 기법중 하나인 깁스 샘플러(Gibbs Sampler)를 활용하여 모수 추정을 수행한다. 모두 θ를 임의의 모수 벡터라고 할때, θ의 초기값을 θ⁽⁰⁾ = {θ₁⁰, ..., θ_p⁰}이라 하고 현재 상태를 θ^(t) = {θ₁^t, ..., θ_p^t}라고 한다면, 깁스 샘플링 기법은 아래와 같다.

1. π₁(θ₁^t|z, θ₂^t, ..., θ_p^{t})로부터 θ₁^(t+1)를 생성한다.}
2. π₂(θ₂^t|z, θ₁^(t+1), θ₃^t, ..., θ_p^{t})로부터 θ₂^(t+1)를 생성한다.}

...

P. π_p(θ_p^t|z, θ₁^(t+1), ..., θ_{p-1}^(t+1))로부터 θ_p^(t+1)를 생성한다.

이 과정을 충분히 반복해 주면 다음 식 (16)과 같이 θ_i에 대한 사후평균의 근사값을 구할 수 있다.

$$\hat{\theta}_i = \frac{1}{p} \sum_{j=1}^p \theta_i^{(j)} \quad (16)$$

그리고 깁스 샘플링 기법은 각 모수에 대한 완전 조건부 사후 분포가 필요하다. 그러므로 ZIP 모델의 모수인 ω와 β에 대한 완전 조건부 사후분포는 다음 식 (17)과 같다.

$$\pi(\beta_i; \beta_{-i}, \omega, z) \propto \prod_{i=1}^n I_{(z_i=0)} \left\{ \omega + (1-\omega) e^{-\lambda} + I_{(z_i>0)} (1-\omega) \frac{e^{-\lambda} \lambda^{z_i}}{z_i!} \right\} \times e^{-\frac{1}{2}(\beta-\mu)^T \Sigma^{-1}(\beta-\mu)} \quad (17)$$

$$\pi(\omega; \beta, z) \propto \prod_{i=1}^n I_{(z_i=0)} \left\{ \omega + (1-\omega) e^{-\lambda} + I_{(z_i>0)} (1-\omega) \frac{e^{-\lambda} \lambda^{z_i}}{z_i!} \right\} \times \omega^{\delta_0-1} (1-\omega)^{(n_0-1)}$$

여기서 β_i은 모수 β에서 i번째 요소를 나타내며, β_{-i}은 β에서 i번째 요소만 제외한 벡터를 의미한다.

그리고 ZINB 모델의 모수인 ω, β, 그리고 θ에 대한 각각의 완전 조건부 사후분포는 다음 식 (18)과 같다.

$$\pi(\beta_i; \beta_{-i}, \omega, z) \propto \prod_{i=1}^n I_{(z_i=0)} \left\{ \omega + (1-\omega) \left(\frac{\theta}{\theta+\lambda} \right)^\theta + I_{(z_i>0)} (1-\omega) A_i \right\} \times e^{-\frac{1}{2}(\beta-\mu)^T \Sigma^{-1}(\beta-\mu)} \quad (18)$$

$$\pi(\omega; \beta, \theta, z) \propto \prod_{i=1}^n I_{(z_i=0)} \left\{ \omega + (1-\omega) \left(\frac{\theta}{\theta+\lambda} \right)^\theta + I_{(z_i>0)} (1-\omega) A_i \right\} \times \omega^{\delta_0-1} (1-\omega)^{(n_0-1)}$$

$$\pi(\theta; \beta, \omega, z) \propto \prod_{i=1}^n I_{(z_i=0)} \left\{ \omega + (1-\omega) \left(\frac{\theta}{\theta+\lambda} \right)^\theta + I_{(z_i>0)} (1-\omega) A_i \right\}$$

이때 A_i는 다음 식 (19)와 같다.

$$A_i = \frac{\Gamma(\theta + z_i)}{\Gamma(\theta)\Gamma(z_i + 1)} \left(\frac{\theta}{\theta + \lambda} \right)^\theta \left(\frac{\lambda}{\theta + \lambda} \right)^{z_i} \quad (19)$$

4. 데이터 분석

4.1 데이터 분석 환경

본 논문에서 데이터 분석을 위한 실제 자료는 2008년 11월부터 2009년 4월 현재까지 약 6개월에 걸쳐 국내 국방 분야 A 업체로부터 COTS 하드웨어 컴포넌트 기반 임베디드 소프트웨어 B의 개발과정에서 발생한 실패의 횟수를 1주일 단위로 하드웨어 시스템 관련된 실패 종류별로 수집된 자료이다. 그리고 데이터 분석을 위해 하드웨어 시스템 관련 실패의 수, 즉 연동결합의 수를 종속변수로 하며, 결함 발견 준비/수행/재작업 시간, 그리고 테스트 케이스의 수를 독립변수로 사용하였다. 독립변수 역시 A 업체로부터 수집된 데이터를 활용하였다. 그림 2에서 보는 바와 같이 종속변수는 영(Zero)을 다수 포함하고 있는 영과잉 데이터이다.

또한 종속변수는 전체 임베디드 소프트웨어 결함대비 평균 14.98%를 차지한다. 이는 초기 COTS 하드웨어 컴포넌트 도입 시 발생하는 초기 연동결합의 수가 증가하였기 때문이다. 그리고 수집된 종속변수가 갖는 심각도(Severity)는 A업체가 정한 심각도 구분 기준 “상

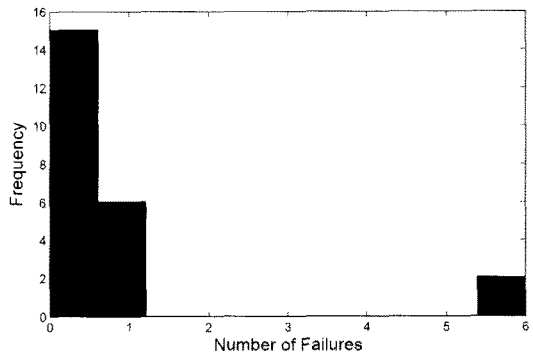


그림 2 관측 데이터(실패 횟수) 히스토그램

표 2 종속변수의 기초 통계량

관측수	23
평균	3.285714
표준편차	5.618846
중앙값	0
최빈값	0
최소값	0
최대값	6
첨도	3.661166
왜도	1.951255
평균연동결합률	14.98%

(+3), 중(+2), 하(+1)”를 따랐으나, 가정 A5에 따라 수집된 종속변수는 모두 “상(+3)”을 적용하였다.

표 2는 종속변수의 기초적 통계값들을 나타낸다. 표 2에 따르면 최빈값과 중앙값이 ‘0’이고 첨도와 왜도값이 각각 3.661166과 1.951255 이므로 종속변수는 영이 과하게 분포되어 있음을 알 수 있다.

또한 데이터 분석을 위해 MATLAB을 이용하여 베이지스 요인 계산과 모수 추정을 위한 난수 생성기로 깃스 샘플러를 구현, 활용하였다. 그리고 모수 추정값 계산 및 모수 추정 결과 분석에는 GNU의 통계 패키지인 R[24]을 이용하였다.

4.2 베이지안 분석 결과

본 논문에서는 COTS 하드웨어 컴포넌트 기반 임베디드 소프트웨어 개발에서 영(Zero)의 빈도가 많은 연동결합을 반영하기 위해 ZIP 모델과 ZINB 모델을 사용하여 신뢰성 모델을 정의 하였으며, 두 모델에 대한 베이지스 요인을 계산하여 주어진 실측 데이터에 적합한 모델을 선택하고자 하였다. 베이지안 분석을 위한 각 모델의 분석 환경은 다음과 같다.

- ZIP 모델의 모수 ω 의 사전분포는 베타 분포에서 초기 모수(a, b)에 (5,5)값을 주어 분석한다.
- ZIP 모델의 모수 β 의 사전분포는 다변량 정규분포의 평균(μ)과 공분산(Σ)에는 일반적 포아송 모델을 사용하였으며, R을 이용한 최소자승추정법(Least Square Estimation)으로 추정값을 계산하여 분석하였다.
- ZINB 모델의 모수 ω 의 사전분포는 ZIP 모델과 동일하게 설정하였다.
- ZINB 모델의 모수 β 의 사전분포는 다변량 정규분포의 평균(μ)과 공분산(Σ)에 일반적 음이항 모델을 사용하였다.

그러므로 베이지스 요인을 계산하기 위해 M_1 모델 ZIP 모델, M_2 모델을 ZINB 모델로 가정하여 두 모형의 주변우도함수를 계산하면 ZINB의 경우 3.420×10^{-9} 의 값이 나왔고, ZIP의 경우 1.2138×10^{-23} 의 값이 나왔다.

그러므로 식 (15)의 우변 베이지스 요인 B 를 계산하면

$$B = \frac{P(z|M_2)}{P(z|M_1)} = \frac{3.420 \times 10^{-9}}{1.2138 \times 10^{-23}} = 2.8111 \times 10^{14} \text{ 이 된다.}$$

이 값은 표 1에 따르면 수집된 연동결합 데이터가 M_1 보다는 M_2 를 더 선호한다는 증거가 되므로, ZINB 모델 보다 ZIP 모델이 연동결합 데이터를 더 잘 설명하고 있다는 의미이다.

또한 표 3은 깃스 샘플러를 이용하여 총 11,000번의 랜덤 변수를 생성하고, 안정화를 위해 초기 1,000개를 제거한 뒤 ZIP 모델에 대한 모수 추정 결과를 나타낸다. 표 3은 모수들의 추정값과 표준오차, 모수에 대한 95% HPD(Highest Posterior Density) 구간을 나타낸다.

표 3 모수 추정 결과

모수	평균	표준오차	95% HPD 구간
ω	0.4396	0.00069	(0.4330, 0.4462)
β_1	0.1534	0.2615	(-0.3960, 0.7029)
β_2	0.3428	0.1986	(-0.0745, 0.7601)
β_3	0.5785	0.1827	(0.1946, 0.9624)
β_4	-1.3292	0.5514	(-2.4877, -0.1707)

표 3에 따르면 ω 의 평균이 0.4396이다. 이는 연동결합의 개수가 0일 때, 이것은 구조적 영의 비율, 즉 점 질량(Point Mass)을 갖는 분포에서 발생할 확률을 나타낸다. 그러므로 실제 분석 데이터에서의 연동결합 수에 대한 0의 비율 64.22% 중 43.96%가 구조적 영의 비율, 다시 말해 점 질량을 갖는 분포에서 발생했다고 추정할 수 있고, 나머지 20.26%는 샘플링 영의 비율, 즉 포아송 분포에서 0이 발생했다고 추정할 수 있다.

4.3 COTS 하드웨어 컴포넌트 기반 임베디드 소프트웨어 신뢰성 추정

COTS 하드웨어 컴포넌트 기반 임베디드 소프트웨어의 신뢰성을 추정하기 위해 우선 순수 소프트웨어 신뢰성 R_{sf} 를 계산하기 위해 식 (7)을 이용한다. Pham의 연구와 마찬가지로 식 (7)에서 $p=0.95$ 라고 가정하고 최대우도추정 기법(Maximum Likelihood Estimation)을 이용하면 $\hat{a}=17.97, \hat{b}=0.000001, \hat{c}=0.000001, \hat{\alpha}=0, \hat{\beta}=0$ 이 된다[17].

그리고 표 3에 따르면 $\tau=0.4396, v=0.2026$ 이 되고, 주어진 데이터를 기반으로 ϵ 과 λ 를 계산하면 각각 0.9525, 0.1398이 되므로, 연동결합 발생확률 $P(t_f) = 0.5529$ 가 된다. 그러므로 COTS 하드웨어 컴포넌트 기반 임베디드 소프트웨어의 신뢰성 함수는 다음 그림 3과 같다. 그림 3에 따르면 연동결합을 고려하지 않은 신뢰성 함수에 비해, 추가적으로 발생 가능한 연동결합을

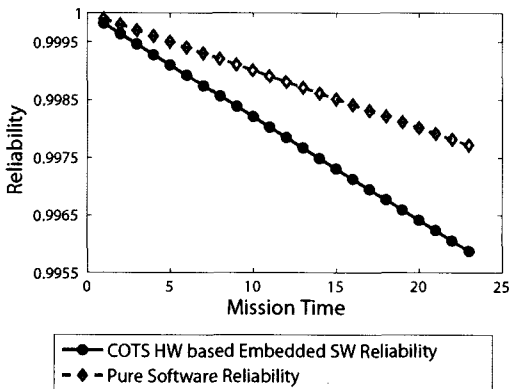


그림 3 COTS 하드웨어 컴포넌트 기반 임베디드 소프트웨어 신뢰성 함수

고려하는 경우 신뢰성 함수가 변경, 즉 신뢰성 함수의 비율이 감소하고 있음을 알 수 있다.

본 논문에서 제안하는 신뢰성 모델은 결함수(Failure Count) 기반 신뢰성 추정 모델관점에서 COTS 하드웨어 컴포넌트 기반 임베디드 소프트웨어의 연동결함의 발생을 고려한 신뢰성 모델이다. 연동결함은 순수 소프트웨어 결함 발생이외에 추가적으로 발생하기 때문에 기존 결함수 기반 소프트웨어 신뢰성 함수의 기울기에 영향을 미치게 된다. 이러한 신뢰성 함수 기울기를 계산하기 위해 연동결함의 발생빈도를 예측하고, 연동결함의 특징을 반영한 통계적 모델을 선택하였다. 이렇게 선택된 모델은 ZIP 모델이며 ZIP 모델을 이용하여 발생확률 $P(f)$ 를 계산하여 기존 결함수 기반 소프트웨어 신뢰성 함수의 기울기를 결정하였다. 이때 연동결함 발생확률 $P(f)$ 가 0.5529 이므로 전체 신뢰성 함수의 기울기는 약 0.4% 정도의 영향을 미치고 있음을 의미한다. 그러나 전체 신뢰성 함수의 기울기에 약 0.4% 정도의 영향이 매우 낮다 하더라도, 실제 연동결함의 심각도가 높다는 가정 A5에 의해 전체 임베디드 소프트웨어 신뢰성에 미치는 영향은 매우 높다 할 수 있으며, 특히 본 논문에서 사용된 분석 데이터 수집기간이 매우 짧고 적은수의 데이터만 수집되었다는 한계를 고려한다면, 시간이 지날수록 전체 임베디드 소프트웨어 신뢰성 함수에서 연동결함의 발생확률 예측값은 매우 큰 영향을 미친다고 볼 수 있다.

5. 결론 및 향후 연구과제

COTS 하드웨어 컴포넌트 기반 임베디드 소프트웨어 개발에서 발생 가능한 연동결함은 일반적 신뢰성 모델로는 설명하기 어려우므로 본 논문에서는 이러한 연동결함을 고려한 임베디드 소프트웨어 모델을 제안하였다.

본 논문에서 제안하는 신뢰성 모델은 결함수(Failure Count) 기반 신뢰성 추정 모델관점에서 COT 하드웨어 컴포넌트 기반 임베디드 소프트웨어의 연동결함의 발생을 고려한 신뢰성 모델이다. 이를 위해 우선 연동결함 발생의 특징을 반영한 두개의 모델(영과잉 포아송 모델, 영과잉 음이항 모델)에 대해, 실제 IT 융합 분야에서 수집한 데이터를 이용하여 베イズ 요인분석을 통한 적합한 신뢰성 모델을 선정하였으며, 이를 바탕으로 마코프 체인 몬테카를로 방법에서 대표적 샘플링 방법인 깃스 샘플러를 이용하여 선정된 모델에 대한 모수 추정을 통해 연동결함의 발생의 특징을 설명할 수 있는 모수를 추정하였다. 그리고 이를 기반으로 연동결함 발생 확률을 계산하여 기존 소프트웨어 신뢰성 모델에 적용한 새로운 형태의 임베디드 소프트웨어 신뢰성 모델을 정의하였고, 이를 실제 IT 융합 분야 데이터를 활용하여 모델의 적합성을 분석하였다. 향후 연구과제로는 현재 정의된 신뢰성 모델을 통계적 기법이 아닌 추후 실제 활용 데이터를 가지고 검증하는 과정이 필요하며, 이를 위해 향후 지속적인 데이터 수집 활동을 수행하여야 한다. 또한 연동결함에 대한 가중치를 결정하여 이를 고려한 신뢰성 모델을 새롭게 정의하도록 한다.

참고 문헌

- [1] Alfredo Benso and Paolo Prinetto, "Fault Injection Techniques and Tools for Embedded Systems Reliability Evaluation," Springer, 2003.
- [2] Vijaykrishnan Narayanan and Yuan Xie, "Reliability Concerns in Embedded Systems Designs," *IEEE Computer*, vol.39, no.1, pp.118-120, 2006.
- [3] Steve Manuel and Dan Lincoln, "COTS-Based SoC Technology Benefits Defense and Aerospace Electronics Applications," <http://www.ecnmag.com>, June, 2005.
- [4] LinuxElectron, "Embedded Military COTS Market to Reach \$3.25 Billion By 2008," <http://www.linux-electronics.com/news/embedded/embedded-military-cots-market-reach-3-25-billion-2008>, 2004.
- [5] VDC Research Group, "Embedded COTS systems in Military, Aerospace, and Defense Applications - Third Edition A Market Demand Analysis Volume I: North American & Volume II: European," 2006.
- [6] Ravishankar K. Iyer and Paola Velardi, "Hardware-Related Software Errors: Measurement and Analysis," *IEEE Transactions on Software Engineering*, vol.SE-11, no.2, pp.223-232, Feb. 1985.
- [7] Kapsu Kim and Chisu Wu, "A Software Reliability Model in the Embedded System," *Proc. of the First International Conference on Software Testing, Reliability and Quality Assurance*, 1994.
- [8] Norman F. Schneidewind, "Reliability Modeling for Safety-Critical Software," *IEEE Transactions on*

- Reliability*, vol.46, no.1, pp.88-98, 1997.
- [9] Naruemon Wattanapongsakorn, "Integrating Dependability Analysis and Optimization into the Distributed Embedded System Design Process," *Ph.D., School of Electrical Engineering*, University of Pittsburgh, 2000.
- [10] Naruemon Wattanapongsakorn and Steven P. Levitan, "Reliability Optimization Models for Embedded Systems With Multiple Applications," *IEEE Transactions on Reliability*, vol.53, no.3, pp.406-417, Sep. 2004.
- [11] Xiaolin Teng and Hoang Pham and Daniel R. Jeske, "Reliability Modeling of Hardware and Software Interactions, and Its Applications," *IEEE Transactions on Reliability*, vol.55, no.4, pp.571-578, Dec. 2006.
- [12] Javier Cano and David Rios, "Reliability Forecasting in Complex Hardware/Software Systems," *Proc of the First International Conference on Availability, Reliability and Security (ARES'06)*, 2006.
- [13] Shankar, V. and Milton, J. C. and Mannering, F. L., "Modeling accident frequencies as zeroaltered probability process: An empirical inquiry," *Accident Analysis and Prevention*, vol.29, no.6, 1997.
- [14] Newton, M. A. and A. E. Raftery, "Approximate Bayesian Inference by the Weight Likelihood Bootstrap (with Discussion)," *Journal of the Royal Statistical Society, Series B*, vol.56, pp.3-48, 1994.
- [15] Jang H. J. and Kang Y. H. and Lee S. and Kim S. W., "Bayesian Analysis for the Zero-Inflated Regression Models," *The Korean Journal of Applied Statistics*, vol.21, no.4, pp.603-613, 2008.
- [16] Jansakul, N. and Hinde, J. P., "Score Tests for Zero-Inflated Poisson Models," *Computational Statistics and Data Analysis*, vol.40, pp.75-96, 2002.
- [17] Hoang Pham, "System Software Reliability," Springer, 2006.
- [18] Irene Eusgeld and Bernhard Fechner and Felix Salfner and Max Walter and Philipp Limbourg and Lijun Zhang, "Hardware Reliability," *Lecture Notes in Computer Science*, Springer, 2008.
- [19] Neil Steiner and Peter Athanas, "Hardware-Software Interaction: Preliminary Observations," *Proc. of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS'05)*, 2005.
- [20] Lee Bain and Max Englehardt, "Statistical Analysis of Reliability and Life-Testing Models: Theory and Methods, 2nd Edition," Chemical Rubber Company (CRC) Press, 1991.
- [21] B. Craig Meyers and Patricia Oberndorf, "Managing Software Acquisition: Open Systems and COTS Products," SEI, 1996.
- [22] Peter Congdon, "Bayesian Statistical Modeling," John Wiley and Sons, Ltd., 2006.
- [23] S. Geman and Donald Geman, "Stochastic Relaxa-

tion, Gibbs Distributions, and the Bayesian Restoration of Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.6, pp.721-741, 1984.

- [24] The Comprehensive R Archive Network, <http://cran.r-project.org/>



구 태 완

2000년 한림대학교 컴퓨터공학과, 수학과 학사. 2002년 한림대학교 컴퓨터공학과 석사. 2007년 한림대학교 컴퓨터공학과 박사. 2007년 3월~2008년 5월 대한상공회의소 강원인력개발원 정보기술과 교사. 2008년 6월~2009년 5월 한국정보통신대학교 공학부 Post-Doc. 2009년 6월~현재 한국과학기술원 정보통신공학과 연구조교수. 관심분야는 임베디드 소프트웨어 신뢰성 모델링, 소프트웨어 테스트, 소프트웨어 품질보증



백 중 문

1993년 조선대학교 컴퓨터과학 및 통계학과 학사. 1996년 미국 University of Southern California Computer Science 석사. 2000년 미국 University of Southern California Computer Science 박사. 2001년~2005년 미국 모토로라 SSERL (Software and System Engineering Research Lab.) 수석 연구원. 2005년~2009년 2월 한국정보통신대학교 공학부 부교수. 2009년 3월~현재 한국과학기술원 전산학과 부교수. 관심분야는 소프트웨어 신뢰성, 소프트웨어 매트릭스, 소프트웨어 비용추정, 소프트웨어 동적 모델링, 소프트웨어 프로세스 개선, 소프트웨어 품질보증, 소프트웨어 식스 시그마