

임베디드 소프트웨어 신뢰성 평가 모델 분석 툴 개발

서장훈¹ · 김선호^{2*}

¹중소기업진흥공단 구조고도화사업부 / ²명지대학교 산업경영공학과

Development of the Reliability Evaluation Model and the Analysis Tool for Embedded Softwares

Janghoon Seo¹ · Sunho Kim²

¹Consulting Service Office, Small and Medium Business Corp.

²Department of Industrial and Management Engineering, Myongji University

Reliability of embedded softwares, as one of factors which affect system reliability, is the probability of failure-free software operation for a specified period of time in a specified environment. and Embedded software is different from general package software because hardware and operating system are tightly coupled to each other.

Reliability evaluation models for embedded softwares currently used do not separate estimation and prediction models clearly, and even a standard model has not been proposed yet. In this respect, we choose a reliability estimation model suitable for embedded softwares among software evaluation models being used, and modified the model so as to accomodate recent software environments. In addition, based on the model, the web-based reliability prediction tool RPX is developed. Finally, an embedded software is analyzed using the tool.

Keyword: reliability evaluation, embedded software, RPX

1. 개요

임베디드 S/W는 마이크로프로세서 안에 내장되어 산업 및 군사용 제어기기, 디지털정보 가전기기, 자동센서장비 등의 기능을 다양화하고 부가가치를 높이는 핵심 S/W로서, 임베디드 시스템 S/W, 임베디드 미들웨어, 임베디드 기본 응용, 임베디드 S/W 개발 도구 등을 포함하고 있다. 이에 대한 주요 기술 분야 및 사례가 다음 <Table 1>에 나타나 있다(Graaf *et al.*, 2003).

현재, 임베디드 S/W의 수요가 증가되면서 임베디드 S/W의 신뢰성 분석은 중요한 이슈가 되고 있으며, 이는 점차 발전하여 유무선 인터넷과 웨어러블(Wearable) 컴퓨터에 의한 스마트 시티즌(Smart Citizen) 시대를 거쳐 도시 생활 환경 속의 유비쿼터스 컴퓨팅에 의한 스마트 타운(Smart Town) 등의 영역으로

그 활용 분야가 급격히 확장되어 「Embedded, Everywhere」시대로 발전할 전망이다. 그렇기 때문에, 이에 대한 연구 및 기술개발이 필요하며, S/W 신뢰성에 대한 관심은 몇몇 기관에서 S/W 인증 사업을 시작하면서 더욱 고조되었다. 특히, 의료용 S/W나 산업용 S/W, 철도·국방 관련 S/W의 경우는 신뢰성이 상당히 중요한 요소이다. 그러나 신뢰성에 대한 평가는 상당히 이론적인 접근이 쉽지 않고, S/W 시험단계에서 얻어야 할 측정 가능한 S/W 결함에 관련된 데이터 수집이 쉽지 않아서 측정에 많은 어려움이 있다. 또한, 임베디드 S/W는 탑재할 대상 플랫폼 및 정의된 기능 요구사항에 맞게 구성되고 맞춤하여 개발되므로, 그 종류가 매우 다양하고, 임베디드 S/W를 구성하는 여러 요소들이 매우 밀접하게 연결되어 있기 때문에 이를 평가한다는 것은 쉽지 않다(최현미, 2002).

*연락처 : 김선호 교수, 449-728 경기도 용인시 처안구 남동 산38-2 명지대학교 산업경영공학과, Tel : 031-330-6451,
E-mail : shk@mju.ac.kr

2007년 06월 접수, 2회 수정 후 2007년 10월 게재확정.

Table 1. 임베디드 S/W 주요 기술 분야

기술 구분	활용 사례
임베디드 기본응용	멀티미디어 재생기, MAP Viewer, 브라우저, PIMS, 게임, 모바일 슈, CNS(GPS, GIS)
임베디드 미들웨어	CORBA, COM, XML, TMO 등의 분산 미들웨어, JVM, J2ME 등 자바 미들웨어, Jini, UPnP, Havi 등 제어 미들웨어, 스트리밍 및 Codec 등 멀티미디어 미들웨어, WLAN, WPAN 관련 통신 미들웨어
임베디드 시스템 S/W	임베디드 OS, 디바이스 드라이버, 유무선 통신 프로토콜 및 멀티미디어 프로토콜 지원 라이브러리, 임베디드 DBMS
임베디드 시스템 개발 도구	설계 도구, 시험 검증 도구, IDE, 타겟시스템 재설정 도구, 각종 시뮬레이터, 크로스 컴파일러, 실시간 원격모니터, 원격 모니터, 디바이스 드라이버 툴킷, 설계 도구, 시험 검증 도구
임베디드 S/W 플랫폼	MS 닷넷 컴팩트 프레임워크, Sun ONE, Brew, WIPI

제품의 신뢰성이란 제품을 구성하는 구성품의 신뢰성을 기반으로 산출된다. 임베디드 소프트웨어도 하나의 구성품이고 임베디드 소프트웨어의 고장(오류, 에러)으로 제품은 기능을 상실하거나 H/W의 기능 수행에 치명적인 영향을 미칠 수 있다. 특히, 제품의 신뢰성 분석과 정비성 및 가용성을 분석하여 제품 개발 타당성 및 설계 개선 사항을 도출하는 활동을 RAM (Reliability, Availability, and Maintainability) 분석 활동이라고 하는데, 현재 RAM 분석의 문제점 중 하나가 임베디드 소프트웨어에 대한 분석이 미진하다는 점이다. 제품에서 임베디드 소프트웨어의 비중이 점차 증가하는 점을 감안하면 임베디드 소프트웨어의 신뢰성 분석은 신제품 개발에서 중요한 이슈이고 기계 및 전기전자 부품 수준의 신뢰성 분석이 반드시 필요하다.

이러한 필요성에 따라, 본 연구에서는 기존 선행연구를 보완하여, 전기전자 하드웨어에 대한 임베디드 S/W의 특성을 반영한 평가 개발 모델을 연구사례로서 제안한다. 첫째, 국·내외 기존 S/W 평가 방법을 조사하였으며, 과제 개발 프로세스를 통하여 임베디드 S/W 신뢰성 산출 모델들을 비교 분석하였다. 분석 결과 신뢰성 예측에 가장 우수하다고 판단되는 RL-TR-92-52 모델을 선정하였다. 둘째, 이 모델을 기반으로 최근의 S/W 환경에 적합하도록 신뢰성 예측 모델을 보완하였다. 셋째, 이 모델을 기반으로 신뢰성 예측 평가 프로그램을 개발하였다. 마지막으로, 분석 툴을 이용하여 임베디드 S/W 사례를 통하여 논의관점을 정리하였다.

2. 임베디드 S/W 신뢰성 예측 평가 모델 선정

본 연구에서는 S/W 품질요소를 고려한 전문가 평가를 통하여, 신뢰성 평가를 위한 예측모델중에 가장 널리 선호하는 4가지를 선정하여 비교하였다. 현재에도, 임베디드 S/W 신뢰성 예측을 위한 평가모델은 국내에서는 사용하지 않고 있기 때문에, RAC(Reliability Analysis Center)에서 제시 있는 하드웨어 부분에 대한 가장 대표적인 신뢰성 예측/평가모델을 4가지를 선정하였다(RAC, 2004). 이 4가지 모델은 가장 보편화된 S/W 신뢰성 예측모델로서 historical 데이터를 이용하여 개발 초기에 예측을 위해 사용된다는 공통점은 있으나 각각의 통계적 가정과 특징

은 조금씩 차이가 있다. 이에 대한 간단한 소개는 아래와 같다.

RL-TR-92-52은 로마 실험실에서 만들어진 모델로서 이 모델의 신뢰성 산출은 KLOC(Kilo Lines Of Code)당 결합 조밀도를 나타낸다. 고유한 결합의 총 추정 수를 계산하기 위해서 결합 조밀도는 KLOC의 예측한 수로 곱해야 한다. 여기에, Function Point가 이용된다. 또한, 이 모델은 조달시간 (delivery time) 동안에 결합 조밀도 예측을 위해 매우 유용하며, 연속적인 결합 조밀도는 고유한 결합 및 결함율(Fault Rate)의 총계를 예측하기 위하여 이용되며, RL-TR-92-52 모델은 다른 모델들 보다는 수집 데이터를 이용한 예측 및 추정에 있어서 계량화가 다른 모델들보다 유용적이다(McCall, Randell, Dunham, and Lauterbach, 1992). MUSA Model은 1975년 Bell 연구소의 John Musa에 의해서 개발된 최초의 신뢰성 예측모델이다. 이 모델의 신뢰성 함수는 결함율(Fault Rate) (Failure Rate), 결합 평균 시간 (MTTF: Mean Time To Failure) 등을 도출하기 위한 하드웨어에 의존한 접근방법이다. 신뢰성은 단위시간동안의 수행 추정 확률에 의해서 결정된다(Jhon and Musa, 1984). Putnam Model은 1978년 Putnam에 의해 제안되어 동적 다 변수(dynamic multi-variable model)로서 전체 프로젝트 수행기간 중 각 분야별 (요구사항 분석 및 설계, 기능구현, 시험, 문서화 etc.) 노력(effort)분포를 추정할 수 있는 모델이며, 대형 프로젝트에 투입되는 인력분포의 특징을 알 수 있는 유용한 모델이다. 특히 인력분포 곡선이 Rayleigh 분포를 나타내는 것이 특징이다. SoftRel Prediction Models은 기본적인 Software Prediction Capabilities를 제공하며 다른 확장 모듈들의 기본이 되는 모듈이다. 100개 이상의 Parameter를 통한 Prediction Capability를 제공하는 Software Defects, Software Failure Rate와 Reliability를 예측하려는 목적뿐만 아니라 Software Reliability, Engineering과 관련해서 개선 및 의사결정에 유용하다(Rome Laboratory, 1992).

위에서 제시된 4가지 예측모델에 대하여 수정, 운영, 적용 측면에서 제시된 평가항목을 통하여 전문가들에 의해 평가되었으며, 그 평가결과가 아래 <Table 2>에 제시되어 있다. 현재, 전기전자부분의 제품들은 Life-Cycle이 짧기 때문에 이를 고려하여 임베디드 S/W 운용, 수정, 적용 부분이 가장 이슈가 되고 있으므로 이에 대한 평가지표를 분류하였다.

평가기준은 평가지표에 대한 부합성을 고려하여, 관련 부문

실무경력 5년 이상인 전공자 2명의 외부전문가(S/W 평가 전공)와 3명(신뢰성 공학, 프로그램 설계, 시스템 공학 전공)의 내부 개발자가 리커트 척도(오광근, 2003)로 평가한 후 평균점수를 기재하였다. 이에 대한 결과는 다음 <Table 2>와 같다. 여기서 제시한 평가기준에 의해서 나타난 결과, RL-TR-92-52 모델이 가장 선호도가 높은 것으로 평가되었다. 평가산정 방법은 평가 세부항목에 대하여 5명이 정량화 점수의 평균을 이용하여 계산하였으며, 항목별 가중치 설정 자체가 기존 연구나 사용된 데이터가 없기 때문에 가중치 설정은 배제하였다. 이 방법은 전문가 집단의 의견과 판단을 쉽고 빠르게 취합하기 위하여 델파이 방법(delphi technique)과 브레인스토밍 기법을 적용하였다.

<Table 2>에 제시되어 있는 예측 모델에 대한 평가표에 의해서 운영·수정·적응의 분류를 통하여 평가한 결과, RL-TR-92-52 모델을 본 연구사례로서 가장 적합할 것으로 결정되었다. 그리고 RL-TR-92-52 모델은 다른 모델에 비해 수집데이터를 이용한 예측 및 추정에 있어서 계량지표 변환이 다른 모델들 보다 유동적이라는 특징이 있다.

3. S/W 신뢰성 평가 모델 보완

임베디드 S/W 신뢰성 예측평가 모델로 선정된 RL-TR-92-52 모델에서 FD(Fault Density)와 FR (Failure Rate)은 다음과 같이 표현된다. 특히, FD는 1에 가까울 수록, FR 값은 0에 가까울수록 신

뢰성이 높게 나타난다.

$$FD = A \times D \times S (0 \leq FD \leq 1)$$

$$FR = FD \times \text{Conversion Factor (faults/time, } 0 \leq FR)$$

여기서 A-Factor는 Application Type, D-Factor는 Development Environment, S-Factor는 Quantified Characteristics of the Software Product를 나타낸다. 그리고, Conversion Factor는 Application Type의 평균 작동 고장률을 표준 샘플 데이터의 평균고장 밀집도로 나눈 값이다. A-Factor는 임베디드 S/W 모델이 갖고 있는 상대적 복잡도를 나타내는데, 이 모델에에서는 Application Type이 일정한 상태의 같은 가중치 설정은 현실적으로 부적합하지만, 본 연구에서는 같은 임베디드 S/W에 개발 전후에 대한 신뢰성 예측 결과 비교를 통하여 이에 대한 연구관점을 제시하는 것이 목적이기 때문에 A-Factor의 가중치는 똑같이 1을 포함한다는 가정을 하고 있다. D-Factor는 개발환경에 따른 과거의 경험적이며, 상대적인 S/W 결함 밀집도를 나타내며, 일단 결함 조밀도 계산방법이 결정되면, 결함율(Fault Rate)은 설치된 결함 조밀도에 신형 유형의 과거 경험적 데이터에서 실험 데이터를 적용해서 예측될 수 있다. 그리고 S-Factor는 다음 <Figure 1>와 같이 두 개의 Sub-Main Factor(S1과 S2)로 구성된다. S1은 S/W 요구사항과 설계특성을 나타내며, S2는 S/W 수행 환경에서의 특성을 나타낸다. S1-Factor는 다시 SA, ST, SQ로 구성되며, S2-Factor는 SL, SM, SX, SR로 구성된다. 이에 대한 의미는 <Table 3>에 나타나 있다(RAC, 2004).

Table 2. 예측 모델에 대한 평가표

평가항목		예측모델			
분류	세부항목	RL-TR-92-52	MUSA Model	Putman Model	SoftRel Prediction Models
운영측면	정확성	3.5	3.2	2.3	3.2
	신뢰성	4.1	3.6	3.4	2.7
	효율성	3.8	3.7	3.8	3.4
	확장성	3.5	2.7	3.4	3.1
	사용용이성	3.7	3.4	2.8	2.9
	무결성	3.1	3.2	2.4	2.7
	유지보수성	3.6	3.4	3.2	2.6
수정측면	무결성	2.9	3.2	3.4	2.8
	이식성	2.8	2.7	2.3	2.4
적응측면	시험성	4.1	2.8	3.4	2.5
	재사용성	4.3	3.4	3.7	3.1
	상호운영성	3.8	3.6	2.9	2.7
	유지성	4.3	4.3	2.5	2.4
합계		47.5	43.2	39.5	36.5
평가 평균		3.7	3.3	3.0	2.8
순위		1	2	3	4
산정방식		평가 평균 = $\frac{\sum \text{항목별평가치}}{5}$			

주) 리커트 척도(매우 좋다 : 5점, 좋다 : 4점, 모르겠다: 3점, 안좋은 편이다 : 2점, 안좋다 : 1점).

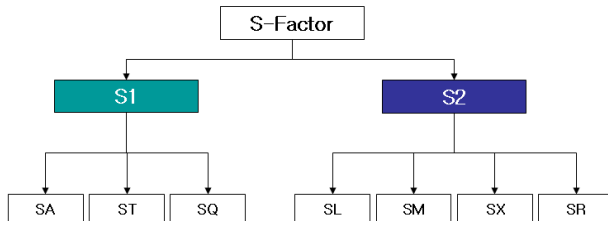


Figure 1. S-Factor 구성

현재, RL-TR-92-52 모델에서 A-Factor, D-Factor, S-Factor 중에서, 개발 환경 D-Factor IT-Governance 관점에 가장 이슈가 되고 있으며, 복잡하고 운영·수정·적응 측면에서의 요구사항이 D-Factor가 가장 많은 영향을 차지하고 있기 때문에 이에 항목 구성을 재설계 하였다. S-Factor의 영향도는 최근 개발 Tool의 고도화 기술에 힘입어 현재의 복잡한 S/W 신뢰성 평가부분을 많은 부분에서 수용하고 있으며, 적용하기가 어렵다고 판단하

Table 3. D/S-Factor 구성항목 설명

약어	명칭	설명	Tradeoff Range
SA	Software anomaly management	비정상 상황에 대한 관리	Small
ST	Software traceability	요구사항-설계-개발간의 추적성	Large
SQ	Software Quality	개발 표준 준수 여부	Small
SL	Software Language	개발 언어에 대한 가중치 부여	N/A
SM	Software modularity	모듈 사이즈	Large
SX	Software complexity	복잡도	Large
SR	Software standards review	표준 검토 수행 여부	Large

Table 4. D-Factor 구성항목 재설계

평가 범주	구성 항목		비고
조직화 (Organization)	설계와 코딩 분리	독립된 검증 및 확인	추가항목
	독립된 테스트 조직	프로그래밍 팀 조직	
	독립된 품질보증	팀원들은 평균이상의 교육 수준인가?	
	독립된 사업관리	팀원들은 평균이상의 경력 수준인가?	
	자원 분배 계획	독립된 품질통제 계획	
	독립된 형상관리 및 통제 계획	개발 프로젝트 의사소통관리	
	기술현황 분석	정보구조 정의	
방법론 (Method)	표준 정의와 실행력	탐 다운 방식과 구조화된 접근	추가항목
	고차원 언어 사용	단위 별 개발 정리	
	정형화된 검토방법	S/W 개발 라이브러리	
	자주 검토활동	진행 및 상태 보고서	
	요구사항 추출 방법		
문서화 (Documentation)	시스템 요구사항 기준	S/W 품질보증 계획	추가항목
	S/W 요구사항 기준	S/W 사업관리 계획	
	인터페이스 설계 기준	요구사항 추적 매트릭스	
	S/W 설계 기준	버전 설명 문서	
	테스계획, 진행상황 보고서	S/W 불일치	
	S/W 개발 계획(일정계획)		
	S/W 위험관리	Software 프로젝트 추적/예측	
S/W 유지보수 계획			
사용 툴 (Tools used)	요구사항에 명시된 개발언어	코드 감사	추가항목
	프로그램 설계 언어	데이터 흐름 분석기	
	프로그램 설계 도면화 기술	프로그래머 워크벤치	
	시뮬레이션/에뮬레이션	측정 툴	
	형상관리		
	개발 시스템 Storyboard	개발 시스템 Prototyping	
테스트 기술 (Test Techniques)	코드 검토	기능성 테스트	추가항목
	브랜치 테스트	오류 & 비정상적인 상태 탐지	
	무작위 테스트	구조화된 분석	
	요구사항 검증 Process		

여 수용되는 부분에 대하여 평가 항목을 재설계 하였다. 결과적으로, D-Factor의 재설계 항목에 대한 내용을 <Table 4>에 정리하여 제시하였다. 특히, D-Factor는 개발환경에 대해서 노력과 비용, 기능과 테스트 전단계의 요구분석을 모두 수용할 수 있도록 신뢰성 평가항목이 추가되었으며, 현재의 개발환경의 영향도는 다양하고 복잡한 환경에서 많은 영향을 받기 때문에 이를 수용하기 위해서 D-Factor 부분에 평가항목을 재구성하였다.

S-Factor는 단위 테스트(Unit testing)과 통합 테스트(Integration testing)을 고려하여 임베디드 시스템 기능 및 성능 등의 사용자 요구를 만족할 수 있도록 설계하였다. S-factor에 대하여, 각 항목에 대해 검토 결과 두 가지 부분에서 개선점을 도출하였다.

첫째, 라인수를 근거로 하는 일부 항목 병합: 개발 툴의 확대 보급으로 통합 환경의 개발 툴을 사용하는 추세인데, 이러한 개발 툴을 이용하여 개발했을 경우 정제된 소스 코드를 생성

해 주는 반면, 에러는 절감되는 효과가 있다. 결과적으로 SLOC 수가 증가하나 오류는 감소하므로 SLOC는 객관적인 신뢰성 평가 기준에서 중복되는 부분을 병합하였다. 예컨대 SLOC는 개발된 S/W 신뢰성에 일관된 세부적으로 여러 가지 라인수를 근거로 하는 항목들이 분류 되어 있으므로 이 부분에 대하여 병합할 수 있는 항목내용을 <Table 5>와 같이 정리하였다.

둘째, SL-Factor 개선: SL-Factor는 개발 언어를 기계어(Assembly Language)로 변환했을 경우, 기계어의 코드 수를 판단하여 비중을 주는 항목이다. 현실적으로 언어를 기계어로 변환한 후의 라인 수를 산정하기가 곤란한 점과 일반적으로 가중치를 부여하는 기준이 있으므로 이를 대안으로 채택하였다. 개발 언어에 대한 가중치 부여는 다음 <Table 6>에 제시되어 있는 Musa Prediction Model에서 사용하는 가중치를 적용하였다(VDC: Venture Development Corporation, 2000). 이 모델에서는 기능 점수 대비 의미 있는 소스 문장의 수, Mean Source Statements/Function

Table 5. 라인수를 근거로 하는 항목 삭제 부문

Are all applicable external inputs, checked for reasonableness reported before processing begins?
a. How many executable lines of code(LOC) in this CSCI? b. How many negative boolean and compound boolean expressions are used?(total from all units) c. Calculate 1- (b/LOC) and enter value.
a. How many iteration loops(i.e., DO/FOR loops)?(total from all units) b. In how many iteration loops are indices modified to alter fundamental processing of the loop?(total from all units) c. Calculate 1- (b/LOC) and enter value.
a. How many statement labels, excluding label for format statements? b. Calculate 1- (a/LOC) and enter value.
a. How many branches, conditional and unconditional?(total from all units) b. Calculate 1- (a/LOC) and enter value.
a. How many declaration statements?(total from all units) b. How many data manipulation statements?(total from all units) c. Calculate 1- ((a + b)/LOC) and enter value.

Table 6. 개발언어별 가중치

Programming Language	Expansion Data	Mean Source Statements/Function Point	Programming Language	Expansion Data	Mean Source Statements/Function Point
Basic Assembly	1.0	320	Modular 2	4.0	80
Macro Assembly	1.5	320	Ada 83	4.5	71
C	2.5	128	Prolog	5.0	64
Interpreted Basic	2.5	128	Lisp	5.0	64
2nd Generation language	3.0	107	Forth	5.0	64
Fortran	3.0	107	Quick Basic	5.5	58
ALGOL	3.0	107	C++	6.0	53
COBOL	3.0	107	Ada 9X	6.5	49
CMS2	3.0	107	Database Default	8.0	40
JOVIAL	3.0	107	Visual Basic	10.0	32
Pascal	3.5	91	APL	10.0	32
3rd Generation Lanaguage	4.0	80	SMALLTALK	15.0	21
PL/1	4.0	80	Generations	20.0	16

Point 항목 참조하였다. Musa Prediction Model에서의 개발 언어 사용 가중치를 참조하는 이유는 현재 정리되어 있는 개발환경을 고려한 언어에 가중치 연구 자료가 부족하며, RL-TR-92-52에서의 제시모델은 너무 오래 되어 현재의 시스템에 적용할 수 없으므로 기초데이터로서 참조되었다.

4. 신뢰성 평가 S/W 개발

본 연구에서는 개발하는 임베디드 S/W 신뢰성 예측분석 툴(RPX: Reliability Prediction eXtensible)은 시스템 개발 초기 개념 설정 및 설계단계에서 시스템의 신뢰성을 예측하고, 시스템 요구 신뢰성과 시스템 예측 신뢰성을 비교 분석함으로써 시스템 방향 설정에 활용하고자 하는 목적이 있다. RPX는 개방 환경하의 객체지향 설계 기법을 도입하여 확장성과 향후 유지보수를 고려하여 웹 기반 시스템으로 개발되었다. 개발 툴은 오라클 9i, JAVA, JSP(Java Server Page), Apache, JDK 2.0을 사용하였다.

4.1 기능 구성도

RPX는 시스템 기본 정보를 관리하는 시스템 정보 모듈, 시스템을 구성하는 구성품 간의 상하위 관계 및 상호 신뢰성 관계를 관리하는 시스템 구조 관리 모듈, 기계류/전기전자류/S/W 별로 부품의 신뢰성을 예측할 수 있는 신뢰성 예측 모듈 및 시스템 신뢰성을 산출하고, 신뢰성 보고서를 출력하는 신뢰성 결과 모듈로 구성한다. RPX에 적용된 신뢰성 예측 모델은 다음과 같으며, 기능 구성도가 <Figure 2>에 나타나 있다.

- 전기/전자: MIL-HDBK-217F Reliability Prediction of Electronic Equipment (DOD)
- 기계: Reliability Prediction for Mechanical Parts and System (NSWC)
- S/W: 본 연구에서 정의한 임베디드 S/W 신뢰성 평가 모델 (RL-TR-92-52 ROM Laboratory Prediction Model(RAC) 변형 모델)

시스템 정보는 시스템 기본 정보 및 시스템의 주요 임무(기

능)에 대해 정의한다. 시스템 구조 관리는 시스템을 구성하는 구성품을 트리형태로 정의하며, 신뢰성 관계도에서는 신뢰성 관계 즉, 직렬, 병렬, 대기구조 등의 현재 상태를 알 수 있는 RBD (Reliability Block Diagram)을 작성한다. 신뢰성 예측은 시스템을 구성하는 구성품에 대해 기계, 전기전자, 내장형S/W 별 단위 부품의 신뢰성을 예측한다. 여기서, 결합률에 대한 단위는 시간 단위만을 사용한다. 시스템의 작동 시간에 대한 사용 비율을 정의하는 보정계수는 모두 1로 가정한다. 신뢰성 산출은 부품 간의 관계를 직렬로 가정하여 시스템 신뢰성을 산출하며 신뢰성 보고서는 신뢰성 결과를 출력한다(Sung et al. 2003).

4.2 Entity Relation Diagram

시스템 기능 구현을 위하여 데이터 모델을 다음 <Figure 3>과 같이 정의하였다.

RAM_SYS_MST는 시스템 정보를, RAM_TREE_MST는 시스템 구조 정보를 관리한다. RAM_RBD_NODE, RAM_RBD_REL, RAM_RBD_JUNC은 RBD 정보 관리를 다루고 있으며, RAM_PARAM_TREE, RAM_CODE_MST는 전기전자 및 기계에 대한 범주별 파라미터 정보 관리를 다루고 있다. RAM_PARAM_DATA는 전기전자 및 기계 부품별 파라미터 설정 값을 관리하고, SW_SL_MST, SW_TYPE_TBL, SW_ENV_TBL, SW_CHECK_TBL는 S/W 범주별 파라미터 정보 관리, S/W는 분석, 설계, 구현으로 범주를 구분한다. SW_S_DATA, SW_CHECH_DATA, SW_ENV_DATA, SW_TYPE_DATA는 S/W 구성품에 대한 파라미터 설정 값을 관리한다.

4.3 RAM 분석 기능

다음 <Figure 4>은 임베디드 S/W를 포함한 부품 신뢰성(RAM) 분석 프로세스를 나타내고 있다.

우선, RAM Tree를 통해서 시스템정보입력, 시스템구조를 정의한 다음, Reliability Prediction 부분에서는 신뢰성 예측 Parameter(모수 값) 설정 및 결과 값을 입력하고, 신뢰성 산출 및 신뢰성 보고서를 생성한다. category 선정은 RL-TR-92-52 모델에서

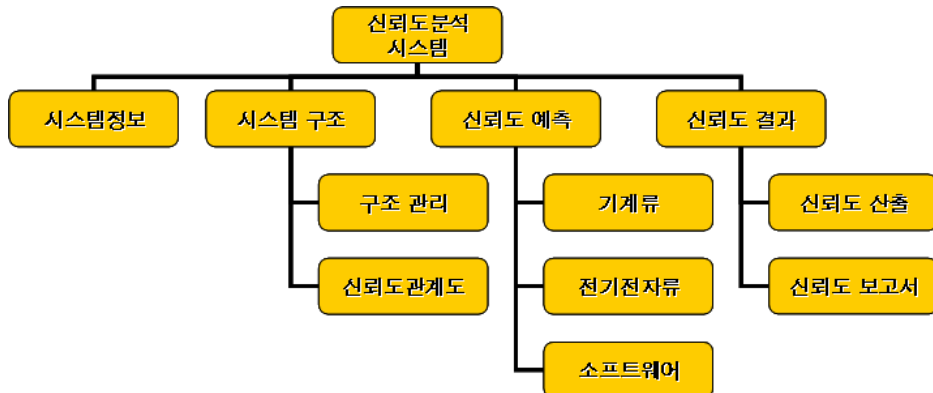


Figure 2. 기능 구성도

A-Factor, D-Factor, S-Factor의 하위 항목을 설정하는 단계이다. 일반적으로, 최종 제품의 시스템 및 임무 신뢰성은 구성품 간의 관계를 기준으로 Bottom-Up 방식으로 계산하여 산출한다.

다. 예컨대, 부품 A의 신뢰성은 $R(A) = e^{-(t/MTBF)}$ (t: 작동시간)로 정의할 때 상위 구성품의 신뢰성은 하위 구성품 간의 관계를 기준으로 다음과 같이 계산한다.

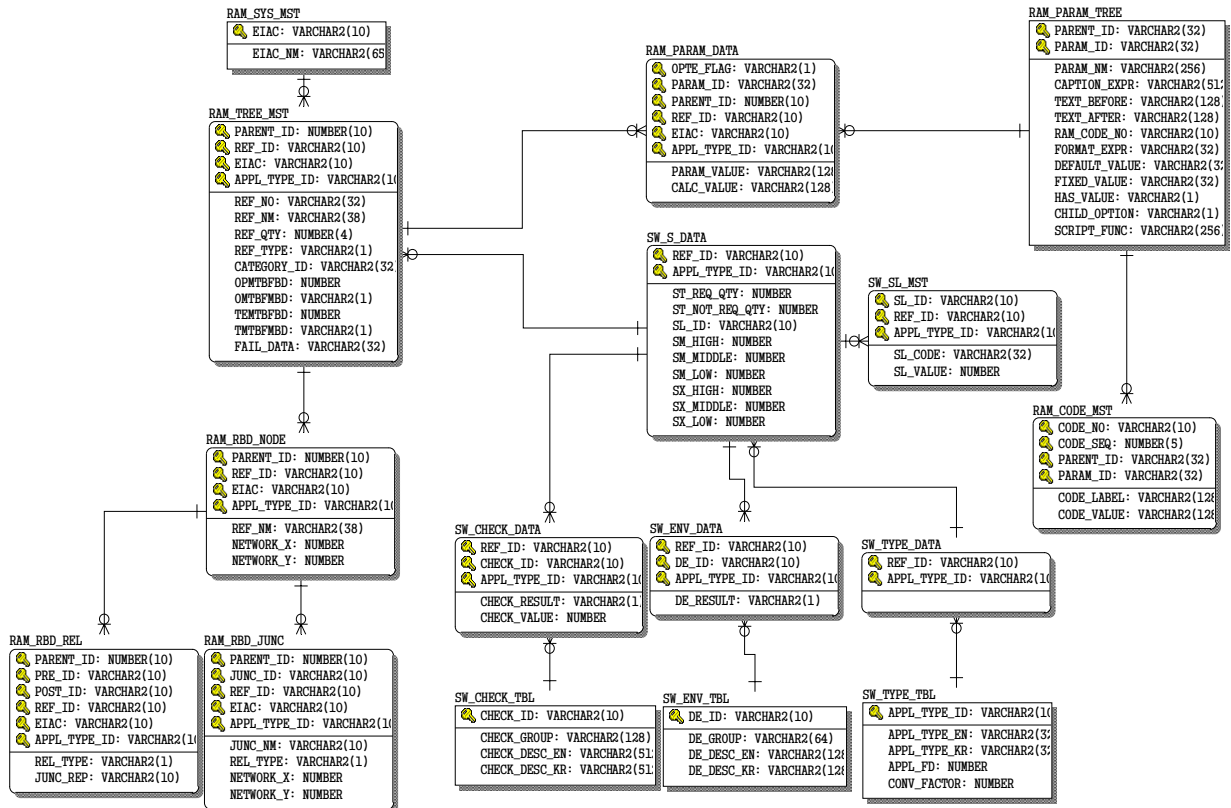


Figure 3. 데이터 모델

RAM Tree

Reliability Prediction

Category 선정

신뢰도 분석결과 예시

시스템명(번호)	Part Name	Part No	Q/Y	중복회기(중복)	교정률(중기)
항공기, 시험용	0000001-01	1	508.982349705187242		
엔진 지시계통	0177000	1	1000042		
INTEGRATED ENGINE INSTRUMENT SYSTEM	0177000	1	1000042		
수력 지시 계통	01771000	1	1000042		
엔진 회전수 계기를 발전기 장번호	0178101-01-6	2	1000042		
엔진기 발전 회전수 계기용	0178101-01	2	1000042		
회기 계통	01780000	1	213809164.5187056812		
노출 계통	01781000	1			
해기스터브 장번호, 영웅 11	0178101-01	1	588.285284116474762		
신뢰도 계통	FE511001	1	5000262		
해기스터브 조립체, 우측	01781102-01	1	1000042		
해기스터브 조립체, 좌측	01781101-01	1	1000042		
전스도	00000000	1	1398.168470907018412		

부품 신뢰도 예측

- . Category 별 Parameter 값을 이용하여 부품 신뢰도 예측
- . 부품 신뢰도를 기준으로 시스템 신뢰도 계산

▶ 전기/전자 부품 : MIL-STD-217F
 ▶ 기계 부품 : NSWC (naval Surface Warfare Center) Mechanical Part Prediction
 ▶ 소프트웨어 : RL-TR 92-52

Figure 4. S/W를 포함한 부품 신뢰성 예측 프로세스

직렬 관계 : $R(p) = R(A) \times R(B)$

병렬 관계 : $R(p) = R(A) + R(B) - R(A) \times R(B)$

대기 관계 : $R(p) = R(A) + (\frac{\lambda_A}{\lambda_B - \lambda_A})[R(A) - R(B)]$
 $= \text{EXP}(-\lambda_A t) + (\frac{\lambda_A}{\lambda_B - \lambda_A})[\text{EXP}(-\lambda_A t) - \text{EXP}(-\lambda_B t)]$

RL-TR-92-52 모델의 신뢰성을 나타내는 결함 밀집도 FD (Fault Density) 계산방법은 아래와 같이 직렬구조로 되어있다.

$$FD(\text{Fault Density}) = A \times D \times S$$

$$= A \times D \times S1 \times S2$$

$$= A \times D \times SA \times ST \times SQ \times SL \times SX \times SM \times SR$$

초기 고장 수 = $N = FD \times SLOC$

(N: 과거의 경험적 데이터에 근거한 시스템 크기와 결함밀집도(Fault Density) 예측 수)

$$\text{Failure Rate} = FD \times C$$

(Faults/Times, C: Conversion Factor)

4.4 신뢰성 예측 구현 화면

본 연구에서 제시된 신뢰성 예측 구현화면은 다음 <Figure 5>와 <Figure 6>과 같다. <Figure 5>는 시스템 기본 정보 입력, RAM 정보, 부품 신뢰성 예측, 부품에 대한 임베디드 S/W 평가

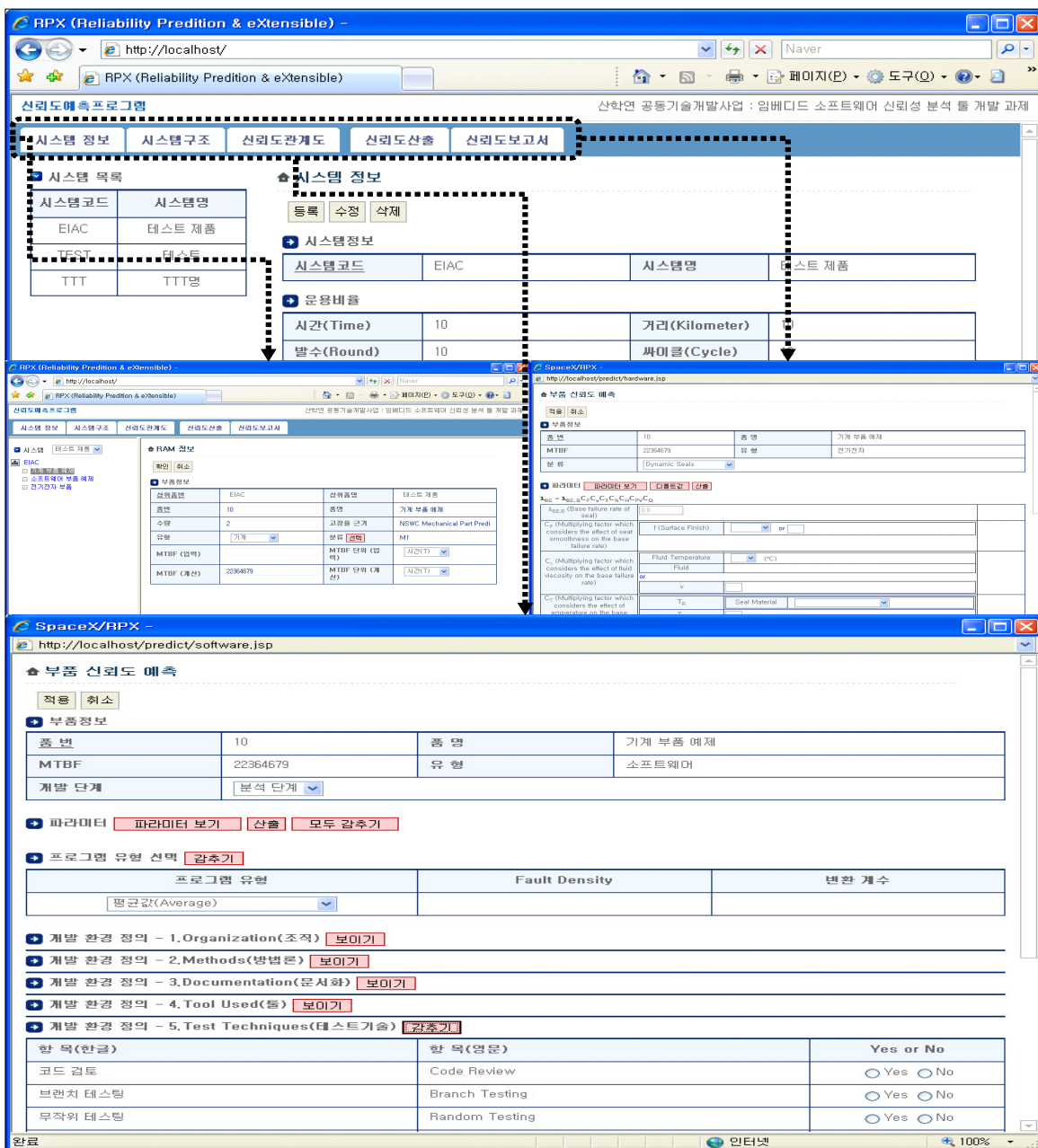


Figure 5. 신뢰성 예측 구현 화면



Figure 6. 신뢰성 예측 보고서

화면을 나타내고 있다. 그리고 <Figure 6>은 신뢰성 예측결과 보고서 사례를 나타내고 있다.

5. 평가모델 비교분석

평가 사례로서, A사에서 국방 무기체계 개발사업의 일환으로 개발한 차기전차의 부속품인 내장형 IETM S/W를 대상으로 평가모델을 비교하였다. 개발된 S/W 신뢰성 평가 모델과 이 모델의 기반이 되는 RL-TR 92-52 모델을 <Table 6>과 같이 비교 분석하였다.

A-Factor의 0.014 결과 값은 0.014로 Application의 가중치를 의미하며, D-Factor는 재설계된 개발환경 영향도에 따라 산출된

값이다. 일부, D-Factor에 대해 제시모델에서 변경이 있었으나 동일한 결과 값이 도출되었다. 괄호 안의 값(-0.448)은 D-Factor 계산 방식을 통해 산출한 값이다. D-Factor의 계산방식은 각각의 평가항목에 대해서 “yes” or “No”로 평가된 결과로 “No”의 항목을 RL-TR-92-52 모델에서 제시한 D-Factor Value = 7.79* Proportion No-2.86 계산방식이 적용되었으며, 이에 대한 신뢰성은 0.5값이 도출되었다. S-Factor는 설계된 평가항목들에 대해서 충족항목 건수(total number of design requirements : NR)와 불충족건수(not fulfilled determine: DR)에 대하여 계산방식 (NR-DR)/NR을 이용하여, RL-TR-92-52 모델에서 제시하는 S-Factor 가중치를 계산한 값이다. 특히, SL항목에 대해서는 기존 모델을 기준으로 하는 값을 도출하기가 현실적으로 불가능하여 가중치 1를 부여하여 계산하였다. 결과적으로, 계산 로직에 따라

Table 6. RL-TR 92-52과 개선된 모델의 신뢰성 분석 결과 비교

항목	측정 내용	RL-TR-92-52	제시 모델
A	Difficulty in developing various application types	0.014	0.014
D	Development org, methods, Tools, Techniques, documental	0.5488095	0.568901
SA	Indication of fault-tolerant design	0.7	0.9
ST	Traceability of design and code to requirement	1.1	1.1
SQ	Adherence to coding standards	1	1
SL	Normalizes faults density by language type	1	1
SX	Unit complexity	1.09	1.09
SM	Unit size	0.901817424	0.909847434
SR	Compliance with design rules	1.5	1.5
Conversion Factor		132.6	132.6
FR* (faults/time)		1.156698798	1.55355908
MTBF		0.864529298	0.642939661

서, 대상 S/W에 대한 결함율(Fault Rate)은 RL-TR 92-52 모델은 1.157 값을 제시 모델은 1.555 값이 계산되었다. 즉, 기준 모델은 0.865 단위시간 마다 한 번씩 오류가 있음을 제시 모델에서는 0.642 단위시간마다 한 번씩 오류가 있음을 의미한다. 제시 모델이 보다 밀도 높은 결과 값을 도출하였다. 참고로 위 계산 결과는 구현 단계에서의 historical 데이터를 이용한 결과이고, 이를 바탕으로 한 시험을 통해 요구되는 Estimation 신뢰성 값을 도출하기 위해서는 지속적으로 더 많은 샘플 데이터를 개발과정에서 추출할 수 있어야 할 것이다.

6. 결론

본 연구에서 제시한 RPX는 시스템 기본 정보를 관리하는 시스템 정보 모듈, 시스템을 구성하는 구성 품간의 상하위 관계 및 상호 신뢰성 관계를 관리하는 시스템 구조 관리 모듈, 기계류/전기전자류/S/W 별로 부품의 신뢰성을 예측할 수 있는 신뢰성 예측 모듈 및 시스템 신뢰성을 산출하고, 신뢰성 보고서를 출력하는 결과 모듈로 구성하였다. 특히, 본 과제에서 임베디드 S/W 신뢰성 모델을 재설계한 RL-TR-92-52 Rome Laboratory Prediction Model(RAC) 변형 모델은 S/W 유형, 개발 환경, 예외관리, 요구추적, 품질, 개발언어, 표준, 모듈 및 복잡성을 고려하여 사례분석을 실시하였다.

본 연구에서 제시한 RPX 평가 모델은 임베디드 S/W의 특성을 반영하여, 예측에 대한 신뢰성을 평가하기 위해 개발되었다. 임베디드 S/W는 탑재할 대상 플랫폼 및 정의된 기능 요구 사항에 따라 매우 다양한 형태로 존재하기 때문에, 주로 특정 제품에 한정된 기술에 대한 연구개발이 대부분이다. 또한, 임베디드 S/W에 대한 시스템 테스트를 주로 수행하는 경우, 오류가 발생하여도 오류가 발생한 모듈을 찾는 것이 어렵다는 문제점이 있다는 것을 알 수 있었다. 이는 임베디드 S/W의 컴포넌트가 매우 밀접한 구성으로 이루어져 있고, 또한, 직접 실행될 때에 문제가 발생하기도 하기 때문인 것으로 판단된다.

결과적으로, 제안한 평가 모델을 사용하여 임베디드 S/W를 평가하여 신뢰성 분석을 시행할 경우, 대상 임베디드 S/W에서 발생 가능한 다양한 구현상의 오류, 시간 지연으로 인한 오류, 스택 오버플로우(overflow)와 같은 임베디드 S/W에 대한 오류를 발견할 수 있으며, 검출된 오류에 대한 보완점을 제시할 수 있었으며, 이 연구를 통하여 H/W 제한적인 신뢰성 분석 틀을 S/W로 확대할 수 있다는 가능성 확인하였다. 앞으로 이 분석 틀을 제조업계에 적용할 경우 제품의 신뢰성(품질)을 향상시킬 수 있을 것으로 예상된다. 또한, 국제 표준에 따른 유연성과

특정 목적을 가진 임베디드 S/W 신뢰성 평가가 가능하다는 것을 알 수 있었다.

향후, 제안한 예측 모델이 실질적인 임베디드 S/W 테스트에 적용될 수 있도록 평가 항목별로 구체적인 테스트 데이터 선정 항목에 대한 연구개발을 수행할 예정이며, 또한 기능 평가 모델 뿐만 아니라, 성능, 실시간성, 신뢰성, 효율적 메모리 사용, 전력 소모, 재사용성과 같은 비기능적 요소에 대한 평가모델을 제안할 예정이다. 그리고 임베디드 S/W 시장에서의 수요를 고려할 때, 직접적인 임베디드 S/W 개발 및 판매에 따른 수익성과 마찬가지로, 분석 틀을 도입함으로써 얻어지는 제품의 신뢰성 및 임베디드 S/W 신뢰성 평가 시장 확보가 매우 중요하기 때문에 본 연구의 확장 연구는 반드시 필요할 것으로 판단된다.

참고문헌

- 가진호 (2005), 임베디드 시스템 관리 프레임워크의 구현, 한국정보과학회 학술발표논문집, 29(1), 424-426.
- 양해술 (2002), 임베디드 S/W의 품질평가 모델 개발 연구, 한국정보통신기술협회 위탁과제 최종보고서.
- 오광근 (2003), 임베디드 시스템 S/W 측정을 위한 품질특성 연구, 한국정보과학회 학술발표논문집, 30(2), 385-387.
- 정혜정 (2003), S/W 신뢰성 평가 기술 및 품질관리 적용방안, 산업자원부 기술표준원.
- 최현미 (2002), 임베디드 S/W를 위한 기능 중심의 평가 모델, S/W 및 응용, 32(12), 1192-1204.
- Capers, J. (1995), Backfiring or Converting Lines of Code Metrics Into Function Points, Software Productivity Research.
- Department of Defense (1994), Military Standard Software Development and Documentation, MIL-STD-498.
- Graaf, B., Lormans, M., and Toetenel, H. (2003), Embedded Software Engineering : The State of the Practice, *IEEE Software*, 61-69.
- Hailpern, B. and Santhanam, P. (2002), Software Debugging, Testing, and Verification, *IBM Systems Journal*, 41(1), 4-12.
- Jhon, D. and Musa, Kazuhira Okumoto (1984), A comparison of time domains for software reliability models, *Journal of Systems and Software*, 4(4), 277-287.
- McCall, J., Randell, W., Dunham, J., and Lauterbach, L. (1992), Software reliability measurement and testing guidebook, Technical Report RL-TR-92-52, Rome Laboratory USAF, 201-203.
- RAC (2004), Introduction to software reliability, RAC(Reliability Analysis Center), 93-108.
- Rome Laboratory (1992), Methodology for Software Reliability Estimation and Assessment, Technical Report RL-TR-92-52, 1-2.
- Sung, A. and Choi, B. (2003), Interaction Testing in an Embedded System using Hardware Fault Injection and Program Mutation, *LNCS*, 2931, 192-204.
- VDC(Venture Development Corporation)(2000), The 2000 Embedded Software Strategic Market Intelligence Program : Internet Enabling Applications and Middleware in Embedded Systems.



서장훈

명지대 산업공학 석사
아주대 경영학 석사(MBA)
명지대 산업공학 박사
KMAC 생산성본부 컨설턴트 및 심사원
(주)썬더 기술연구소 소장
현재: 중소기업진흥공단 구조고도화사업부
관심분야: 6시그마, S/W 평가, IT-Governance,
BPM/PM, 품질공학



김선호

명지대학교 산업경영공학과 교수
Pennsylvania State University 산업공학과 석·
박사
서울대학교 산업공학과 학사
연구분야: Data Quality, 전자거래 표준화,
BPM,