

한국모바일학회논문지

Journal of Society of Mobile Technology
VOLUME 4, NUMBER 1, JUNE 30, 2007

- 타원 곡선 매트릭스를 이용한 RFID 상호 인증 프로토콜 ————— 김성진, 김영수, 박석천
- 모바일 네트워크에서의 효율적인 인증 프로토콜 ————— 김기원
- 임베디드 소프트웨어 신뢰성에 대한 조사 ————— 김기두, 김동호, 김영철, 김장한
- 이동단말의 효과적 전력 사용을 위한 적응적 페이징 방법 ————— 서봉수
- 휴대용 단말기의 광원을 이용한 무선 결제 시스템 모델 ————— 황호영, 허문행
- 센서 네트워크에 적용한 p-persistent CSMA-CA의 시뮬레이션에 관한 연구 ————— 김호석, 손수국, 조성관
- ZigBee와 PLC 시스템간의 인터페이스에 관한 연구 ————— 김호석, 손수국, 조성관
- 계층적 센서 네트워크를 위한 강화된 키 관리 기법 ————— 이원진, 전일수
- WPI 기반의 단말기 관리 에이전트 설계 ————— 박주건, 박기현, 장대진, 장명숙, 우종정
- SVG 기반 무선지도 생성 시스템의 개발 ————— 부기동, 안해순
- 웹2.0 기반 지역 공간 커뮤니티 프레임워크 ————— 김운용

SMT

Society of Mobile Technology

사단법인 한국모바일학회

<http://www.smt.or.kr>

임베디드 소프트웨어 신뢰성에 대한 조사*

김기두*, 김동호**, 김영철**, 김장한**
한국정보통신기술협회*
홍익대학교 컴퓨터정보통신**,
e-mail : kdkim@tta.or.kr* ray@selab.hongik.ac.kr**

A Survey on Embedded Software Reliability

Ki Du Kim*, Dong Ho Kim**, R. Y Kim**, Jang Han Kim**
Telecommunication Technology Association*
Dept. of Computer Information & Communication, Hongik University**
e-mail : kdkim@tta.or.kr* ray@selab.hongik.ac.kr**

요약문

현재 임베디드 시스템에 대한 임베디드 소프트웨어 신뢰성 연구가 진행되고 있다. 본 논문의 목적은 임베디드 소프트웨어에서의 품질 요소 중 신뢰성 측정에 관한 것을 알아보고, 이를 위해 기존 소프트웨어 품질 평가 및 개선요소를 살펴본다. 이를 기반으로 임베디드 소프트웨어 신뢰성 평가를 위한 임베디드 소프트웨어 신뢰성 매트릭스를 제안한다.

Abstract

Recently there are going to research on the embedded software reliability. This paper mentions one of embedded software quality attributes, reliability measurement, and survey on the existing software quality assessment and improvement attributes of embedded software reliability. Based on this research, we propose the metrics for assessing the embedded software reliability.

Key Words: Embedded Software, Reliability, Quality Assessment

* 본 연구는 2006년도 홍익대학교 학술연구진흥비에 의해 지원되었음.
접수일자 : 2007년 4월 17일 / 게재확인일자 : 2007년 5월 20일
한국정보통신기술협회*, 홍익대학교 컴퓨터정보통신**

I. 연구배경

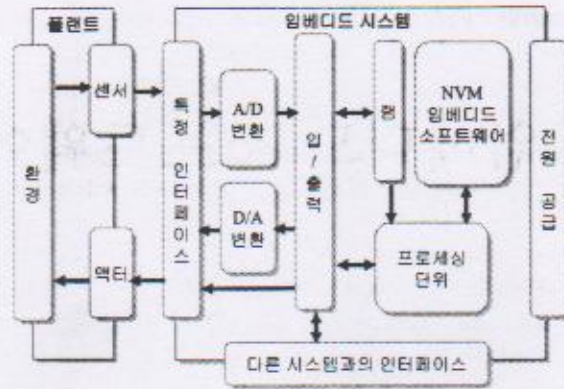
현재 많은 임베디드 소프트웨어들이 개발되고 있으며, 임베디드 소프트웨어에 있어 신뢰성은 매우 중요하다. 임베디드 시스템을 위한 소프트웨어는 데스크톱 또는 메인프레임을 위한 어플리케이션 소프트웨어보다 훨씬 더 다양한 문제를 다룰 수 있어야 하고, 외부 이벤트에 응답을 해야 한다. 또한, 임베디드 시스템은 사람의 도움 없이 갑작스러운 상황에 대처할 수 있어야 한다. 그리고 임베디드 시스템의 작업들은 제한 시간 내에 수행되어야 한다.[1] 이를 살펴보면 임베디드 소프트웨어는 높은 신뢰성을 갖고 있어야 한다는 것을 알 수 있다. 우리는 임베디드 소프트웨어에 대하여 간략히 설명하고 임베디드 소프트웨어의 신뢰성을 측정하기 위해 기존 소프트웨어 품질의 6항목 중 신뢰성에 초점을 맞추었으며 그 표준을 살펴본다. 그리고 최근 통합적인 품질 표준인 ISO/IEC 25000 에 대해 알아본다. 그리고 우리는 ISO/IEC 9126[2]을 기반으로 임베디드 소프트웨어의 품질 특성에 따른 요구 사항을 체계화하고 이에 따른 임베디드 소프트웨어 품질평가를 위해 매트릭스를 제안한다.

II. 관련연구

1. 임베디드 시스템

임베디드 시스템은 여러 범위의 시스템을 포함하는 일반적인 용어로 예를 들어 핸드폰, 철도 신호 시스템, 미사일 추적 시스템, 항공 운항 시스템 등이 있다. 그러나 모든 시스템은 몇몇 특정한 하드웨어를 제어함으로써 실 설계와 상호작용하는 공통된 특징을 가지고 있다. 다음 [그림 1]은 임베디드 시스템의 일반적인 컴포넌트 스키마를 보여준다. 이러한 스키마는 대부분의 임베디드 시스템에 적용 가능한 모델이다.

임베디드 시스템은 센서(sensor)를 통해 입력 신호를 주고받으며, 환경을 조작하는 액터(actor)에 출력 신호를 보낸다. 액터와 센서를 포함하는 임베디드 시스템을 플랜트라 칭한다.[3]



[그림 1] 임베디드 시스템의 일반적인 스키마

임베디드 소프트웨어는 임베디드 시스템을 제어하거나 혹은 입출력 값을 처리하는 소프트웨어로 일반적으로 ROM이라는 비 휘발성 메모리에 저장된다. 또한 소프트웨어는 플래시카드, 하드디스크 혹은 CD-ROM의 형태로 제공되거나 네트워크나 인공위성을 통해 다운로드 되기도 한다. 임베디드 소프트웨어는 특정 타겟 프로세서, 프로세싱 단위를 위해 컴파일되고, 작동하기 위해 일정량의 RAM을 요구한다. 프로세싱 단위는 전용의 입출력 레이어를 통해 모든 입력/출력 신호를 처리한다.[3]

2. 임베디드 소프트웨어의 특성

임베디드 소프트웨어는 일반적인 소프트웨어와 많은 가지 차이점을 갖고 있다. 차이점을 모두 나열하기는 것은 어렵지만 대표적인 특징들을 요약하면 다음과 같다.[3]

1) 개발 목적의 한정성

일반적으로 범용으로 개발된 프로세스와 메모리는 어떤 프로그램이 수행될 지 알 수 없다. 따라서 일반적인 프로그램이 빠르게 실행될 수 있는 구조를 갖게 된다. 일단 설계된 프로세스의 성능은 결국, 운영체제 및 컴파일러 등 많은 구성 소프트웨어에 의해 평가된다. 이는 먼저 하드웨어가 개발되고 그 구조에 적합한 소프트웨어를 설계하는 순차적인 방법을 말한다.

반면 특정 목적을 수행해야하는 임베디드 소프트웨어의 경우 각각의 목적에 맞게 수행하기 위해 최소의 하드웨어와 최소의 소프트웨어를 이용

하여 상황에 맞게 기능을 갖도록 특화시켜야 한다. 그에 따라 임베디드 시스템은 제한된 하드웨어에 맞게 개발된 소프트웨어에 따라 각각의 성능을 갖게 된다.

2) 실시간 처리

범용 시스템의 처리는 일반적으로 주어진 자원을 활용하여 최대한 빠르게 수행하는 것을 목적으로 한다. 이것을 수행 기한에 대한 제한이 없이 가장 효율적으로 수행되도록 설계되었음을 의미한다. 하지만 임베디드 시스템의 경우 처리시간에 제한이 있는 실시간 처리가 많다.

3) 내구성

임베디드 시스템은 고온이나 다습한 환경, 또는 충격이 가해지거나, 일부 기능에 이상이 있어도 기본 기능은 정상적으로 동작되도록 요구된다. 예를 들어, 미사일 제어 시스템 혹은 자동차의 ABS시스템의 경우 특정 기후나 환경에 의해 영향을 받아 오동작 될 경우 사람의 생명에 영향을 미칠 수 있다.

3. 기존 소프트웨어의 품질 평가

3.1 ISO/IEC 9126

소프트웨어 제품평가에 관한 국제표준인 ISO/IEC 9126은 [그림 2]과 같이 품질특성과 부특성의 체계를 가지고 있다. 품질특성은 기능성, 신뢰성, 사용성, 효율성, 유지보수성, 이식성의 6개의 특성으로 구성되어 있으며, 각 품질특성에 대해 대응하는 부특성들로 구성된다. 부특성들에는 매트릭스들이 대응하고 있으며 외부메트릭스(ISO/IEC 9126-2) 과 내부메트릭스(ISO/IEC 9126-3), 그리고 사용품질 매트릭스(ISO/IEC 9126-4)으로 나뉜다.

3.2 ISO/IEC 14598

품질 모델은 다음과 같이 정의되었다. 요구사항의 명세와 품질평가를 위한 기준을 제공하는 품질특성집합과 그들 상호간의 관계를 의미한다. 또한 품질모델은 품질요소를 적절하게 배치시킨

결과로 나타난다.

품질에 관한 정의를 살펴보면 외부품질 (External Quality), 내부품질 (Internal Quality), 사용중 품질 (Quality in Use)로 나뉘는데, 외부품질은 소프트웨어 제품이 규정된 조건에서 사용될 때 명시적 또는 암묵적 요구사항을 만족시키는 정도를 말하고, 내부품질은 소프트웨어 제품이 규정된 조건에서 사용될 때 명시적 또는 암묵적 요구사항을 만족시키기 위한 능력을 측정하는 속성들의 총체를 말한다. 사용 중 품질은 소프트웨어 제품이 특정 환경에서 사용될 때 사용자의 작업 효율성, 생산성, 만족도등 사용자의 요구를 충족시키는 정도를 말한다.

ISO/IEC 14598 에 따른 품질 평가 (Quality Evaluation)는 다음과 같이 3가지 측면에서 고려되는데, 내부 품질평가(Internal Quality Evaluation), 외부품질평가 (External Quality Evaluation), 사용중 품질 평가 (Quality in Use Evaluation)로 나뉜다[4].

3.3 IEEE 표준 982.2-1988



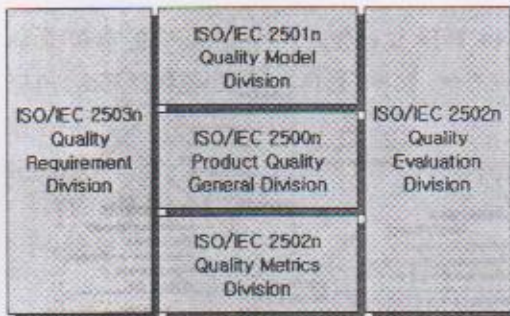
[그림 2] 신뢰성에 영향을 주는 품질 요소

[그림 2]에서와 같이 다른 라이프 사이클에 신뢰성의 관계를 나타내는 것은 단계적으로 체계를 따른다. 또한 IEEE Std 982.2-1988에서 주장한다. “소프트웨어 신뢰성 관리 프로그램은 사용자 품질 목표의 균형이 잡힌 집합의 설립을 요구합니다. 그리고 중간의 품질 목표의 식별은 그 사용자 품질 목표를 달성하는 것을 그만큼 도움 것이다.”[6]

3.4 SQaRE (Software product Quality Requirements and Evaluation)

기존의 존재하던 소프트웨어 제품 평가가 기준들의 일관성을 부여하고 실현가능하도록 하는 노력이 진행되어 진행하고 있는 소프트웨어 제품 평가에 대한 ISO/IEC 9126(소프트웨어 품질특성 및 매트릭스)시리즈 및 ISO/IEC 14598시리즈의 국제표준을 통합하여 ISO/IEC 25000시리즈로 재편하고 있다.

SQaRE는 소프트웨어 품질을 위한 품질 모델과 매트릭스를 정의하고 품질 모델과 매트릭스의 표준화, 소프트웨어 제품품질은 품질요구사항에 대응한 품질모델과 매트릭스를 기반으로 평가하게 된다. 그리고 소프트웨어 품질에 대한 기술지침 및 국제표준의 이해와 통합을 제공한다. 또한 품질모델과 매트릭스뿐만 아니라 품질요구사항과 평가에 대한 기술적 지침 및 국제표준을 포함하여 제정한다.



[그림 3] SQaRE Architecture

SQaRE는 [그림 3]과 같이 소프트웨어 제품 품질에 대한 기술적 지침 및 국제표준을 제안하는데 다음과 같은 4개의 품질영역과 1개의 포괄적 영역으로 구성된다.

- 품질요구사항(Quality Requirement Division)
- 품질평가(Quality Evaluation Division)
- 품질모델(Quality Model Division)
- 품질 매트릭스(Quality Metrics Division)
- 포괄적 영역(Product Quality General Division)

용어정의, 참조모델 및 지침, 요구사항과 권고사항, SQaRE 사용에 대한 각각의 지침 등을 포함하고 있다.

4. 기존 소프트웨어 품질측정과 매트릭스

품질평가에 있어 측정은 매우 중요하다. 모든 제품은 각 특성에 대한 측정이 이루어져야 기술적인 평가가 가능하기 때문이다. 또한 소프트웨어의 품질개선을 위해서는 모든 특성들이 측정되어야 한다. 그리고 측정에 사용되는 매트릭스는 객관적이고 경험적이며 재생산적이어야 한다.

소프트웨어 제품 매트릭스는 품질요구사항을 정의하고 품질 개선과 측정, 품질관리과 예측, 제품 인도와 승인의 결정을 그 목적으로 한다.

제품 매트릭스는 내부 매트릭스 (Internal Metrics), 외부 매트릭스 (External Metrics), 사용중 매트릭스 (Quality Metrics)로 나뉜다.

다음 [그림 4]는 측정 결과를 판정하기 위한 등급 설정을 보여준다. 측정된 결과 값이 품질이 초과되거나 목표범위 안에 들어오거나 향후 품질이 향상될 것이면 만족하고 측정된 값이 받아들여지지 않으면 불만족하는 두 개의 값으로 표현된다[5].



[그림 4] 측정결과를 판정하기 위한 등급설정

III. 임베디드 소프트웨어 신뢰성 측정 매트릭스 제안

<표 1>과 같이 IEEE 표준 982.2의 신뢰성 품질 요소를 각 품질 요소 단계에서 필요로 하는 항목들을 표로 만들어 보았다.

앞서 품질 측정 매트릭스에서는 품질 모델에서 품질특성 과 부특성을 통해 측정항목을 선정하였으나, 본 논문에서는 신뢰성에 초점을 두어 신뢰성에 맞는 단계를 사용하는 것을 제안한다.

왜냐하면 일반적인 소프트웨어와 달리 임베디드 소프트웨어 특성상 높은 신뢰성을 요구하고 소프트웨어 개발단계 또한 기존의 일반 소프트웨어 개발과는 다르기 때문이다.

<표 1> 평가항목에 따른 신뢰성 품질요소 평가 매트릭스

신뢰성 품질 요소	평가 항목	양적 / 질적
개념 (CONCEPT)	사용자의 요구/목적	
	문서화 표준	
요구사항 (REQUIREMENTS)	요구(Needs)의 고수	
	아키텍처	
	운영환경	
	완전성	
디자인 (DESIGN)	쉬운 사용	
	복합성	
	모듈화	
	인터페이스	
	확장성	
	타이밍, 분배	
구현 (IMPLEMENTATION)	완전성	
	복합성	
	인터페이스	
	개발 표준	
	완전성	
테스트 (TEST)	유지보수성	
	성능 범위	
	안정 범위	
	원문본 인터페이스	
설치 및 점검 (INSTALLATION & CHECKOUT)	상호 측정	
	실제 운용	
	구성 범위	
	인터페이스	
운영 및 유지보수 (OPERATION & MAINTENANCE)	변화의 용이성	
	리그리션 테스트 범위	
	배출과 사용의 용이성	
말수 (RETIREMENT)	양도성, 변형성, 통합성	
	명절 검사	

매트릭스를 살펴보면 다음과 같다. 개념, 요구 사항, 디자인, 구현, 테스트, 설치 및 점검, 운영 및 유지보수, 철수로 구분된다. 기존 품질 측정과는 달리 품질 측정 매트릭스에 임베디드 소프트웨어 개발 단계를 적용하였다.

소프트웨어 개발 단계중 요구사항 단계에서 품질에 관련된 요소를 고려하고 이를 기반으로 제안된 품질 측정 매트릭스에서는 비기능적 요소가 측정 가능하다.

IV. 결론

현재 소프트웨어의 품질의 인종은 국제적으로 많이 확산되고 있으며 그 중요성 또한 많이 인식되고 있다. 그리고 임베디드 시스템을 개발하는데 있어 소프트웨어의 비중이 커지고 있다. 일부 시스템의 경우 구성의 절반 이상이 임베디드 시

스템을 제어하는 소프트웨어로 되어 있다. 그에 따라 임베디드 소프트웨어 중요성도 커지게 되었다. 본 논문에서는 임베디드 소프트웨어의 신뢰성 측정을 위한 매트릭스를 제안하였다. 또한 임베디드 소프트웨어 신뢰성을 측정하는데 체계적으로 수행할 수 있을 것으로 예상된다.

본 논문에서는 소프트웨어의 품질 평가 기술과 그에 따른 표준들에 대해 알아보았으며, 임베디드 소프트웨어의 품질 측정방안을 제안하였다. 임베디드 소프트웨어 신뢰성 측정을 통해 신뢰성의 수준화를 알아보고자 한다. 하지만 현재의 신뢰성 평가 모델만으로는 임베디드 소프트웨어의 품질 평가가 어렵다. 기존 소프트웨어와는 달리 하드웨어와의 연관성을 생각해야 한다. 또한 임베디드 시스템이 특정 목적으로 만들어질 경우 일반적인 품질 평가 방법으로 품질을 측정하기가 어렵다. 좀 더 다양한 사례연구를 통해 일반적인 임베디드 소프트웨어의 품질요소를 추출하고 이를 바탕으로 임베디드 소프트웨어 품질평가가 이루어져야 할 것이다. 현재 임베디드 소프트웨어의 일반적인 품질 요소를 추출하여 정형화하는 연구가 진행 중에 있으며 이를 기반으로 측정된 데이터를 통해 임베디드 소프트웨어의 신뢰성을 향상시키는 임베디드 소프트웨어의 신뢰성 평가 모델의 개선이 진행 중에 있다. 그리고 임베디드 소프트웨어 개발 프로세스에 품질측정 매트릭스를 적용하여 임베디드 소프트웨어의 품질을 개발 전 단계에 거쳐 측정하고자 하는 연구가 진행 중이다.

V. 참고문헌

- [1] David E. Simon, "An Embedded Software Primer", Addison Wesley 1999
- [2] ISO/IEC 9126 "Information Technology -Software Quality Characteristics and Metrics - Part 1,2,3
- [2] 한국정보통신기술협회, "Embedded S/W의 품질평가 모델 개발연구", 2002.
- [4] ISO/IEC 14598 "Software Engineering Evaluation", 2001
- [5] ISO/IEC 25000 SQuaRE(Software product Quality Requirements and Evaluation).
- [6] 박인수 "국제 표준의 소프트웨어 품질평가 기술과 임베디드 소프트웨어" 임베디드 소프트웨어 테스트 세미나. 2005

김기두



e-mail : kdkim@tta.or.kr

2003년 홍익대학교 컴퓨터정보통신 (학사)
2005년 홍익대학교 일반대학원 소프트웨어공학전공(석사)
2005년 ~ 현재 홍익대학교 일반대학원 박사과정
2005 ~ 현재 한국정보통신기술협회 SW시험인증연구소 연구원

관심분야 : 테스트 성숙도 모델, 테스트 프로세스, 임베디드 소프트웨어 테스트, 소프트웨어 신뢰성 테스트

김장한



E-mail: jkim1@hongik.ac.kr

현재 홍익대학교 컴퓨터정보통신 정교수

관심분야 : 영상신호처리

김동호



e-mail : ray@selab.hongik.ac.kr

2003년 홍익대학교 컴퓨터정보통신 (학사)
2005년 홍익대학교 일반대학원 소프트웨어공학전공(석사)
2006년 ~ 2007년 한국학술진흥재단 문제해결형 인력양성 연구원
2007년 ~ 현재 홍익대학교 일반대학원 박사과정

관심분야 : 사용자 행위 분석 방법론, 임베디드 소프트웨어 개발 방법론, 모델 기반 아키텍처(MDA) 임베디드 소프트웨어 신뢰성 평가

김영철



e-mail : bob@hongik.ac.kr

2000년 : Illinois Institute of Technology(공학박사)
2000년 ~ 2001 : LG 산전 중앙연구소 Embedded system 부장
2001년 ~ 현재 : 홍익대학교 컴퓨터정보통신 조교수

관심분야 : 소프트웨어 성숙도 모델, Use Case 방법론 및 도구 개발, CBD, BPM, 사용자 행위 분석 방법론