임베디드 소프트웨어 평가 모델 구축 방안

최현미[0], 성아영[0], 조나경[0], 최병주[0], 반효경[0], 김재웅[1]
[0]이화여자대학교, [1]한국정보통신기술협회
{hmchoi , aysung, nkcho, bjchoi, bahn}@ewha.ac.kr, jwkim@tta.or.kr

The Evaluation Model for Embedded Software

Hyunmi Choi[0], Ayoung Sung[0], Nakyoung Cho[0], Byoungju Choi[0], Hyokyung Bahn[0], Jaewoong Kim[1]
[0]Ewha Womans Univ., [1]Telecommunications Technology Association

요 약

임베디드 소프트웨어는 탑재할 대상에 맞게 구성하고 맞춤화(customize)하는 작업이 필요하므로 그 종류가 다양하고 시스템을 구성하는 컴포넌트들이 밀접하게 연관되어 있기 때문에 이를 평가하기 위해서는 일반 소프트웨어와 다른 관점이 필요하다. 본 논문에서는 임베디드 소프트웨어의 구성을 분석하였고, 이를 기반으로 임베디드 소프트웨어의 특징을 반영하여 품질 평가와 기능평가를 포함한 체계적인 임베디드 소프트웨어 평가 모델의 구축 방안에 대해 제시하였다.

1. 서론

최근 임베디드 소프트웨어 산업은 기술 개발 및 시장 경쟁에 의해 빠르게 변화하고 있다. 마이크로프로세서의 가격 저하, 소형화 및 고성능화가 진행되고, 제품 경쟁력의 핵심이 하드웨어 생산에서 소프트웨어 최적화 기술로 이동함에 따라 상품의 가치가 소프트웨어에 의해 좌우되고 있으며, 홈 네트워크 및 유비쿼터스 산업과 같은 컴퓨팅 패러다임의 등장으로 임베디드 소프트웨어의 응용 분야는 더욱 넓고 다양해졌다.

임베디드 소프트웨어에 대한 관심 및 비중이 커짐에 따라 임베디드 소프트웨어의 평가 또한 중요한 이슈로 부각되고 있다. 하지만, 임베디드 소프트웨어의 다양성과 확고한 표준의 부재로 임베디드 소프트웨어의 개발에 비해 임베디드 소프트웨어에 대한 평가는 그 진척이 미흡한 상황이다.

임베디드 소프트웨어의 경우, 탑재할 대상(target)에 맞게 구성되어야 하므로 그 종류가 매우 다양하며, 임베디드 운영체제, 임베디드 응용 프로그램과 같은 소프트웨어 컴포넌트 간 또는 소프트웨어 컴포넌트와 하드웨어 컴포넌트 간의 긴밀한 상호작용이 존재하기 때문에 일반 소프트웨어 평가와는 다른 관점을 필요로 한다. 그러므로 본 논문에서는 이러한 특성을 고려하여 임베디드 소프트웨어를 체계적으로 평가할 수 있는 평가 모델 구축 방안에 대해 제시하고자 한다.

본 논문의 구성은 다음과 같다. 2장에서는 평가 모델의 대상이 되는 임베디드 소프트웨어에 대해 기술하고, 3장에서는 제안하는 임베디드 소프트웨어 평가 모델의 전체 구조에 대해 기술하며, 4장에서는 결론 및 향후 과제에 대해 기술한다.

2. 임베디드 소프트웨어

초기의 임베디드 소프트웨어는 군사 및 산업용 기기를 제어하는 간단한 펌웨어 형태였으나, 임베디드 시스템의 응용이 디지털 가전 기기, 음향·영상 기기, 모바일 기기, 센서 기기 등 여러 분야로 다양화되고 그 기능에 대한 요구사항도 복잡해짐에 따라, 이를 지원하기 위한 임베디드 소프트웨어의 종류 및 기능이 다양해지고 복잡해졌다. 본 장에서는 이런 다양한 임베디드 소프트웨어를 일관된 관점으로 다루기 위해, 평가 모델의 대상이 되는 임베디드 소프트웨어를 체계화 하여 다시 정의한다.

2.1 임베디드 소프트웨어의 구성 요소

임베디드 소프트웨어는 다음과 같은 세 종류의 소프트웨어 요소(component)로 이루어져 있다.

- 하드웨어에 의존적인 부분 (hardware dependent part, HP)
- 운영체제에 의존적인 부분 (OS dependent part, OP)
- 어플리케이션에 의존적인 부분 (application dependent part, AP)

2.1.1 HP

HP는 임베디드 소프트웨어에서 입출력 장치 및 주변 장치들을 제어할 수 있는 코드, 탑재되는 마이크로 프로세서에 대한 제어 코드 등과 같이 대상 하드웨어를 직접적으로 제어할 수 있는 코드를 의미한다.

2.1.2 OP

운영체제가 탑재된 임베디드 시스템에서 존재하는 부분으로써, 임베디드 시스템에 탑재하는 운영체제 자체보다는 개발자가 AP를 개발하기 위해 운영체제의 일부를 취하여 맞춤한(customized) 모듈과, AP를 개발할 때 운영체제에서 제공하는 기능을 이용할 수 있도록 하는 API 호출에 초점을 둔다.

2.1.3 AP

AP는 도메인의 특징과 사용 목적에 따라 특정 기능을 구현하는 부분으로 경우에 따라서는 서로 다른 임베디드 소프트웨어 사이에서 조정 및 중개 역할을 하는 미들웨어를 포함하기도 한다.

2.2 임베디드 소프트웨어의 구성

HP, OP, AP 각각이 임베디드 소프트웨어의 구성 요소로서의 개별적인 위상을 가지지만, 많은 경우 임베디드 소프트웨어는 HP, OP, AP 각 요소가 긴밀하게 연결되어 있다. 대부분의 임베디드 소프트웨어는 그림 1-1에서 보는 바와 같이 HP, OP, AP가 모두 긴밀하게 연결되어 구성되는 경우와 그림 1-2에서 보는 바와 같이 HP와 AP만 구성된 형태로 구분할 수 있다.

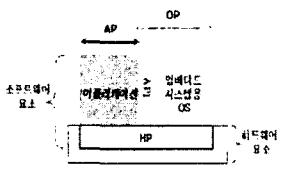

그림 1-1. HP + OP + AP

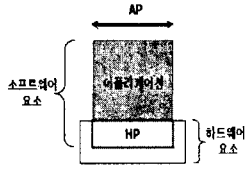

그림 1-2. HP + AP

일반적으로 아주 간단한 제어 기능을 수행하기 위해 8비트 MCU(*micro controller unit*)와 같은 프로세서로 개발되는 임베디드 시스템에서는 운영체제를 사용하지 않는다. 그러나 점차 모든 기기들의 기능이 복잡해지고 하드웨어 기술이 발전함에 따라 임베디드 운영체제를 사용하는 그림 1-1의 형태를 가지는 추세이다.

3. 임베디드 소프트웨어 평가 모델

3.1 평가 모델의 구조

평가 모델은 그림 2와 같이 평가 레벨, 평가 부분, 평가 항목으로 이루어져 있으며, 각 용어의 정의 및 모델에서의 의미는 다음과 같다.

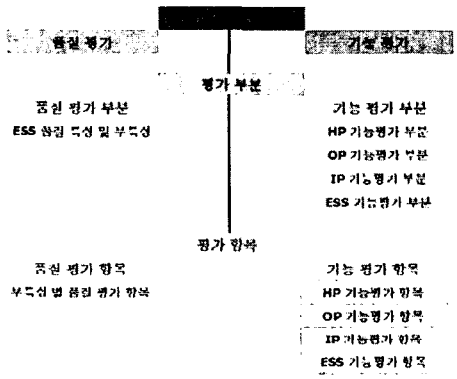

그림 2. 평가 모델의 구성

3.1.1 평가 레벨

임베디드 소프트웨어 평가를 위한 평가 레벨은 그 목적에 따라 품질 평가 레벨과 기능 평가 레벨로 나누어진다.

▪ 품질 평가 레벨

탑재되는 임베디드 소프트웨어 전체 관점에서 품질 특성을 평가하기 위함이며 이를 ESS(embedded software system) 품질 평가라고 정의한다.

▪ 기능 평가 레벨

임베디드 소프트웨어의 기능적인 측면을 평가하기 위함이며, 여기서 "기능적인 측면"이란 "임베디드 소프트웨어에 주어진 특정 기능을 올바르게 발현시킬 수 있도록 지원하는 측면"의 의미를 가진다. 기능 평가를 위해서는 HP 기능 평가, OP 기능 평가, IP 기능 평가, ESS 기능 평가가 있다. 임베디드 소프트웨어의 구성요소인 AP의 경우, 그것 자체가 제품에 대한 기능을 구현한 부분으로써 일반적인 소프트웨어의 기능적인 측면 평가와 같은 맥락을 가지기 때문에 본 모델에서는 고려하지 않는다.

1) HP 기능 평가
하드웨어와의 제어 측면에서 임베디드 소프트웨어의 요소 단위인 HP가 고려해야 할 기능을 포함하였는지를 평가한다.

2) OP 기능 평가
운영체제의 서비스 이용이라는 측면에서 임베디드 소프트웨어의 요소 단위인 OP에서 필요한 기능을 포함하였는지를 평가한다.

3) IP(Interaction part) 기능 평가
임베디드 소프트웨어를 구성하는 각 요소들을 통합하는 관점 및 임베디드 소프트웨어와 대상 하드웨어를 통합하는 관점에서 각 대상간의 상호작용에 대한 고려가 있었는지를 평가한다.

4) ESS(Embedded software system) 기능 평가
임베디드 소프트웨어 제품 전체 레벨인 ESS에서 고려해야 할 기능적 요구사항이 포함되었는지를 평가한다.

3.1.2 평가 부분

평가 부분은 평가 항목보다 한 단계 상위의 카테고리를 제공함으로써, 평가 항목 추출의 토대를 제공한다.
각 평가에 대한 평가 부분 및 평가 부분 추출의 근거는 표 1과 같다.

표 1 각 평가의 평가 부분 및 근거

평 가 (기호)		평가 부분	평가 부분의 근거
품질 평가 (E1)		임베디드 소프트웨어 품질 특성 및 부특성	ISO/IEC 9126[1]를 임베디드 소프트웨어의 특성을 고려하여 재구성
기능 평가	HP 기능 평가 (E2)	메모리 제어, 입출력 디바이스 제어, 타이머 제어	RTOS 문서[2], 임베디드 리눅스[3], 임베디드 시스템 문서[4,5,6]
	OP 기능 평가 (E3)	태스크 관리, 태스크간 통신 관리, 시간 관리, 인터럽트·시그널·예외처리, 메모리 관리, 입출력 관리, 네트워킹, 파일시스템	POSIX[7], KELPS[8], RTOS문서, 임베디드 리눅스, 일반 운영체제[9], 임베디드 시스템 문서
	IP 기능 평가 (E4)	하드웨어와 HP, OP와 AP	RTOS문서, 임베디드 시스템, 펌웨어 문서[10]
	ESS 기능 평가 (E5)	임베디드 소프트웨어 전체	RTOS문서, 임베디드 시스템 문서

3.1.3 평가 항목

평가 항목은 각 평가 부분에 대하여 평가하고자 하는 구체적인 항목을 의미하며, 표 2 에서는 이러한 평가 항목의 일부분을 제시한다.

표 2. 평가 항목 (일부분)

평가	평가 부분	평가 항목
E1	적합성	▪기능 구현의 구조가 올바른가? ▪예외 처리에 대한 메커니즘을 구현하였는가? ....
	정확성	▪빈번도를 포함하여 수행에 적용될 조건이 열거된 문서가 존재하며 이에 따라 수용 가능한 결과가 나오는가? ....
	....	....
E2	메모리 제어	▪메모리 맵에 의거하여 유효한 번지 값을 할당해야 한다 ▪메모리에 쓰거나 지우기를 할 때 해당 섹터를 보호모드로 설정하는 작업을 먼저 해야 한다. ....
	입출력 디바이스 제어	▪ 능동형 I/O 디바이스 사용시, 인터럽트 발생 처리 모듈을 포함해야 한다 ▪ 수동형 I/O 디바이스 사용시, 폴링 요청을 받아들일 수 있는 모듈을 포함해야 한다. ....
	타이머 제어	▪단발성 및 주기성 사용 여부에 따라 올바른 타이머 컨트롤 레지스터 값이 설정되어야 한다. ...
E3	태스크 관리	▪응용 프로그램 태스크가 시스템 태스크의 우선순위를 변경해서는 안 된다 ▪우선순위 역전을 피하기 위해 우선순위 계승, 상한 우선순위, 우선순위 상한 프로토콜 등의 자원 접근 제어 프로토콜을 사용해야 한다. ...
	태스크간 통신 관리	▪한 태스크가 크리티컬 섹션 수행 시 상호배제 알고리즘이 구현되어야 한다 ▪데드락이 발생하지 않도록 데드락 탐지, 복구, 회피, 방지 알고리즘을 구현해야 한다 ...
	시간 관리	▪타이머 인터럽트 발생을 감지하는 인터럽트 처리 루틴이 있어야 한다. ...
	인터럽트·시그널·예외처리	▪인터럽트 금지를 사용한 태스크는 대기 상태가 되지 않도록 해야 한다 ...
	메모리 관리	▪메모리 접근 예외를 발생시키는 메모리 정렬 문제에 대한 처리를 수행해야 한다. ...
	입출력 관리	▪여러 태스크가 하나의 I/O 디바이스 접근을 요청하는 경우, 데이터의 무결성을 유지하기 위해 자원 관리 모듈이 구현되어야 한다. ...
	네트워킹	▪호스트 사이의 다양한 에러와 상태 메시지를 운반하는 메커니즘이 구현되어 있어야 한다. ...
	파일시스템	▪전원 차단으로 인한 데이터 손실을 막기 위해 트랜잭션 베이스로 작업을 수행해야 한다. ...
E4	하드웨어와 HP	▪메모리 및 레지스터 값 세팅 시, 하드웨어 시그널로의 전환이 올바로 이루어져야 한다 ...
	OP와 AP	▪응용 프로그램 태스크가 시스템 태스크로써 의미를 가지게 하는 메커니즘이 존재해야 한다. ...
E5	임베디드 소프트웨어 전체	▪같은 일을 하는 두 개의 함수를 사용해선 안 된다. ...

표 2 에서 볼 수 있듯이 품질 평가 항목은 대상 임베디드 소프트웨어의 품질 부특성에 따라 구조화 되며, 그 부특성을 만족시키기 위해 어떤 항목들을 체크해야 하는가에 대해 제시한다. 기능 평가 항목은 각 평가 분류에 따라 제시된 평가 부분을 초점으로 할 때 어떤 조건들이 만족되어야 하는가에 대해 제시한다. 이러한 평가 항목들은 상호 관계를 고려한 전체적인 결과 분석 방법을 통하여 임베디드 소프트웨어 전체에 대한 평가 결과를 제시할 수 있게 된다.

4. 결론 및 향후 과제

최근 임베디드 소프트웨어 산업 기술은 기술 개발과 시장 경쟁에 의해 빠르게 변화하고 있으며, 하드웨어의 소형화 및 고성능화가 진행됨에 따라 제품의 경쟁력이 하드웨어에서 소프트웨어 최적화 기술로 이동하고 있다.

임베디드 소프트웨어는 탑재할 대상에 맞게 구성되어야 하므로 그 종류가 다양하고 여러 컴포넌트들이 밀접하게 연관되어 있기 때문에 이를 평가하는 것은 쉽지 않다. 그러므로 이러한 임베디드 소프트웨어의 특징을 반영한 임베디드 소프트웨어 평가 모델을 개발하는 것은 매우 중요하다. 본 논문에서는 이러한 특성을 고려하여 임베디드 소프트웨어를 체계적으로 평가 할 수 있는 평가 모델 구축 방안에 대해 제시하였다.

본 논문에서는 임베디드 소프트웨어의 구성 요소 및 특성을 파악하고, 이를 기반으로 임베디드 소프트웨어 평가 모델의 핵심 구성 요소인 품질 평가 및 기능 평가와 같은 평가 레벨을 정의하고 각 평가 레벨별로 구체적인 평가 부분 및 평가 항목을 추출하였다.

향후에는 각 항목에 대한 구체적인 메트릭을 개발하고, 본 논문에서 제안한 모델의 타당성을 검증하기 위하여 사례 연구를 수행할 예정이다.

5. 참고문헌

[1] ISO/IEC 9126-1.2 Information Technology - Software Product Quality, 1998

[2] Jean J. Labrosse, "MicroC/OS-II, The Real-Time Kernel", CMP Books, 1999

[3] Robert Love, "Linux Kernel Development", Developer's Library, 2003

[4] Qung Li and Caroline Yao, "Real-Time Concepts for Embedded Systems", CMP Books, 2003

[5] David E. Simon, "An Embedded Software Primer", Addison Wesley, 2003

[6] Bart Broekman and Edwin Notenboom, "Testing Embedded Software", Addison-Wesley, 2003

[7] IEEE Std 1003.1-2001, IEEE Standard for Information technology - Portable Operating System Interface(POSIX), 2001

[8] KESIC Embedded Linux Platform Specification(KELPS), 2004

[9] Avi Silberschatz, Peter Baer Galvin and Greg Gagne, "Applied Operating System Concepts", WILEY, 2001

[10] Ed Sutter, "Embedded Systems Firmware Demystified", CMP Books, 2002