

임베디드 소프트웨어 테스트 도구에 관한 연구

장선재*, 김지영*, 정란**, 김행곤*

*대구가톨릭대학교 컴퓨터정보통신공학과

**삼척대학교 컴퓨터공학과

e-mail : *{jsjsid, kimjy, hangkon}@cu.ac.kr,

**jungran@samchock.ac.kr

A Study on Embedded Software Testing Tool

Seon Jae Jang*, Ji-Young Kim*, Ran Jung**, Haeng Kon Kim*

*Dept of Computer Engineering, Catholic University of Daegu

**Computer Engineering, Samcheok National University

요 약

기존의 임베디드 소프트웨어 테스트는 하드웨어 지원을 필요로 하거나, 단순한 인터페이스로 제공되는데 반해, 최근의 임베디드 시스템은 복잡하고 높은 수준의 기능을 요구함으로써 기존의 테스트를 그대로 적용하는 것은 매우 비효율적이다. 그러므로, 상세한 지식 없이도 사용 가능하고, 자동적으로 주어진 단계의 시뮬레이션을 수행하고 유효성을 확인하며, 분석의 원활한 수행을 위한 자동화 도구가 필요하게 되었다. 본 논문에서는 기존 기술과의 차이점을 제시하고, 임베디드 테스트 작업이 용이한 임베디드 소프트웨어 자동 테스트 기술을 개발하는 것을 목표로 한다. 이를 위해 임베디드 소프트웨어 테스트 지원 도구의 요구사항을 분석하고, 테스트 도구 ESTE(Embedded Software Testing Environments)의 전체 시스템 구조를 제시하고자 한다.

1. 서론

최근 마이크로프로세서의 저가화, 소형화, 고성능화 및 소프트웨어의 최적화에 따라 모든 정보 산업 기기에 IT기술이 내장되는 “Embedded Everywhere” 시대가 도래하고 있다. 이러한 시대에서 소프트웨어의 비중은 높아지고 있으며, 경제적 잠재성과 파급 효과가 매우 큰 기술인 임베디드 소프트웨어 활용 영역은 향후 Post-PC 시대에서 주요산업으로 대두되고 있다. 이에 따라, 사용자의 요구가 다양하고 변경이 빈번한 특성을 적시에 반영하는 새로운 방식의 소프트웨어 개발 기술 필요하며, 임베디드 소프트웨어와 같은 국가 주력 산업에서 국제 경쟁력 확보를 위한 품질 및 생산성 향상을 위한 소프트웨어 개발 기술과의 접목이 필수적이다.

임베디드 시스템은 사용자의 외부 입력에 대해 어떤 응답과 처리를 하도록 설계된다. 이것은 시스

템 내에 논리적으로 정의되었던 기능들이 순서에 따라 정확하게 수행되어야 하며, 실시간을 중요시 여기는 임베디드 시스템인 경우 기능의 수행이 정확한 시간 내에 이루어져야 한다는 조건을 만족시켜야 한다. 따라서 임베디드 시스템은 일반적인 응용 소프트웨어와는 다른 방식으로 개발되며, 또한 하드웨어에 대한 지식이 요구되기 때문에 신뢰성 있는 소프트웨어의 개발이 매우 어렵다. 이에 따라 시스템의 특성에 따른 요구가 반영된 기능들을 효과적으로 테스트하는 환경 및 도구의 개발이 절실히 필요하다. 본 논문에서는 임베디드 소프트웨어 테스트 도구의 요구사항을 분석하고, 플랫폼-독립적인 임베디드 소프트웨어 테스트 엔진 및 타겟 에이전트를 제공하는 임베디드 소프트웨어 테스트 도구 ESTE에 관해 연구한다. 먼저 관련연구로서 국내외 임베디드 시스템 기술의 개발현황과 기존 S/W 테스트 기술을 살펴보고, 다음으로 임베디드 소프트웨어 테스트 도구와 결론 및 향후연구를 제시한다.

본 연구는 2005년도 ITR&D 연구센터 지원사업에 의해 연구됨

2. 관련연구

이 장에서는 현재 개발되고 있는 국내외의 임베디드 소프트웨어 기술들의 개발 현황과 기존의 소프트웨어 테스트 기술들을 살펴본다.

2.1 국내·외 임베디드 소프트웨어 기술 개발 현황

이 절에서는 미국과 유럽, 우리나라의 기술개발 현황을 살펴보고, 기존의 소프트웨어 테스트 기술을 살펴본다. 국내외 임베디드 소프트웨어 기술 개발 현황은 <표 1>과 같다.

현재, 국내의 임베디드 소프트웨어 관련 기술은 미흡한 편이며, 양성되는 고급 전문 인력의 양적 규모는 상당히 미흡한 편이다. 또한 모든 분야를 위한 기술들을 개발하는 것은 불가능하므로, 몇 분야를 선택하여 그에 대한 집중적 연구와 지원을 하는 것이 바람직하다. 따라서, 이들 분야의 개발과 지원은 장기적으로 이루어져야 하며, 이에 따라 고급 전문 인력의 양성이 이루어져야 한다.[1]

개발 되는 임베디드 시스템의 품질은 소프트웨어의 품질로 정의되며, 개발 단계의 많은 시간이 소프트웨어 테스트에 투자되고 있다. 이는 임베디드 소프트웨어가 하드웨어와 동시에 개발되어 제품에 영향을 미치기 때문이다.

2.2 기존 임베디드 테스트 도구

이러한 임베디드 소프트웨어 테스트 도구들 중 CodeScroll사의 Embedded Tester와 hitex사의 Tessy를

<표 1>국내·외 임베디드 소프트웨어 기술 경향과 개발 현황

경향	대표적인 기술
<ul style="list-style-type: none"> ● 개발적으로 활발히 연구가 이루어지고 있음 ● 임베디드 소프트웨어 분야의 기초 핵심 기술들에 대한 연구 개발 지원은 여러 특성들이 복합적으로 필요한 경제 시스템 차원에서 연구로 확장됨 ● 다양한 응용들의 실행을 지원하기 위한 시스템 기술과 그러한 응용들의 개발을 촉진시킬 수 있는 소프트웨어 공학적 설계 기술에 초점을 맞추어 다양한 형태로 진행되고 있음 	<ul style="list-style-type: none"> ● Yamacraw : 현대의 전진 SW 공학 기술들을 임베디드 SW 개발의 영역으로 도입하기 위한 포괄적인 통합 환경 구축, 기존 자산들의 재사용 접근과 컴포넌트 기반 개발 방법 적용 ● McBES : 응용 도메인마다 가지는 특정 요구에 따라 시스템 컴포넌트들을 조합되거나 맞춤 하는 차세대 시스템 SW co-design 기술 개발이 목적, 여러 연구 기관에서 목적 달성을 위한 세부 기술들을 분리하여 수행 ● VEST : 컴포넌트 기반의 실시간, 임베디드 시스템 구축을 위한 통합 환경 개발 프로젝트, 컴포넌트 기반 real-time 시스템 개발과 구현, 평가의 실질적 향상 도모
<ul style="list-style-type: none"> ● 발달된 자동차 및 항공 산업과 연계되어 임베디드 소프트웨어 기술에 대한 연구가 진행되어 있음 ● 대량 생산을 필요로 하는 Consumer Electronics 분야에 대한 임베디드 소프트웨어 기술 개발에도 연구 지원을 강화하고 있음 ● 최근 e-Europe이란 슬로건을 걸고 달성을 위한 임베디드 소프트웨어 및 응용 기술의 연구 개발 지원 강화 추세 ● 기초 핵심 기술 개발에 역점을 두고 있음 	<ul style="list-style-type: none"> ● MOOSE : Software Engineering Methodologies for Embedded Systems의 개발을 목표로, Generic embedded systems development framework를 만들고 다양한 embedded software projects에 적용해 검증하는 것이 최종 목표 ● PECOS : 임베디드 시스템의 Component-based software development를 가능하게 하기 위한 소프트웨어 컴포넌트기반의 임베디드 시스템의 명세화, 조립, 항상 점검 및 매치를 지원하는 환경을 제공 ● KOALA : 소비 전자 제품을 위한 임베디드 SW 개발을 위한 컴포넌트 모델로서, 컴포넌트 저장소로부터 제품에 대한 큰 다양성 구축 지원할 수 있는 아키텍처 정의 언어이며, Microsoft의 COM을 참조모델로 하고 있음
<ul style="list-style-type: none"> ● 임베디드 소프트웨어 관련 기술수준은 하드웨어 기술 수준에 비해 매우 낮은 편임 ● 운영체제 및 응용 미들웨어 계층의 독자적인 기술이 거의 없음 ● 대부분을 해외 제품들을 사용하고 있다. ● 현재 임베디드 SW 기술은 차세대 IT 산업을 이끌어 나갈 주요 핵심 분야로서 인식됨 ● 정부 및 국제 연구 기관을 중심으로 연구 자원의 투자가 크게 증가하고 있음 	<ul style="list-style-type: none"> ● 공개 소스 소프트웨어 특히 Embedded Linux를 활용한 독자 기술 개발과 고급 인력양성에 대한 정부의 지원이 이루어짐 ● 국가가 추구하고 있는 신성장 동력 산업 발전이 원활히 이루어 지게하기 위한 연구 개발이 이루어지고 있음

예로 들 수 있다.

2.2.1 Embedded Tester

CodeScroll사의 Embedded Tester는 화이트 박스 테스트 제품으로, Host와 Target 탑재 소프트웨어 부분으로 구성된다. 비정상 상황에 대한 대처를 시험하기 위한 테스트 케이스를 추가로 제공하여, 테스트 인력의 효율적 활용과 테스트 성숙도의 향상 기회를 제공한다.

2.2.2 Tessy

hitex사의 Tessy는 임베디드 소프트웨어의 자동화된 유닛 테스트를 지원한다. 이 유닛 테스트에 의해 오류를 빨리 발견하고 쉽게 격리할 수 있으며, 테스트 케이스 설명의 복잡성을 줄일 수 있다. Tessy는 명령어를 배울 필요 없이 쉽게 사용하고, 재사용을 위한 시험 정보를 저장할 수 있는 것을 장점으로 한다.

2.3 White / Black box 테스트 기술

테스팅 과정에서 가장 중요한 요소 중 하나는 효과적인 테스트 케이스를 설계하는 것이다. 따라서 최적 최소의 테스트 케이스 설계를 위한 방법은 크게 프로그램의 외부 명세에 근거한 방법(블랙 박스 테스트)과 프로그램 내부의 논리적 명세에 근거한 방법(화이트 박스 테스트)로 나눌 수 있다.

2.3.1 화이트 박스 테스트

알고리즘을 기본으로 테스트하는 방법. 구조 테스트라고도 한다. 궁극적인 목표는 프로그램 내부 구조에 존재하는 모든 경로의 실행이나, 루프를 포함한 프로그램의 경우 경로의 수개 매우 크기 때문에 불가능하다. 따라서 합리적인 방법으로 테스트 케이스를 선정해야한다.

2.3.2 블랙 박스 테스트

모듈의 알고리즘을 블랙 박스화하여 모듈의 기능 사항이나 인터페이스 관련 사항을 기본으로 테스트하는 방법이며, 기능 테스트라고도 한다.

3. 임베디드 소프트웨어 테스트 도구 개발

이전의 소프트웨어 테스트 작업은 기존의 하드웨어 지원을 필요로 하거나 단순한 사용자 인터페이스를 통해 소프트웨어 특성에 따른 간단한 테스트 기



(그림 2) 임베디드 소프트웨어의 평가 사항

능들 제공하는 하드웨어 의존적인 시스템의 소프트웨어들을 대상으로 삼았다. 이때의 임베디드 시스템 응용분야는 기존의 산업이나 군사분야에서의 제어 및 관계 장치였으나 최근에는 정보가전, 지능형 건물 등에 사용되는 각종 다양한 장치로 빠르게 범위가 넓혀지고 있으며, 따라서 다양한 기능의 처리가 필요한 응용 소프트웨어의 개발 요구가 빠르게 증가되고 있다. 이에 따라, 규모가 큰 테스트 탑 수준의 복잡한 기능을 가지는 다양한 고급언어에 의해 개발된 임베디드 응용 소프트웨어에 대한 수요가 증가하고 있다. 이러한 복잡한 기능을 가지는 고급언어로 개발된 임베디드 응용 소프트웨어들을 하드웨어 지원이나 단순한 사용자 인터페이스에 의한 간단한 테스트를 제공하는 기존의 테스트 도구들에 그대로 적용하는 것은 매우 비효율적이다.

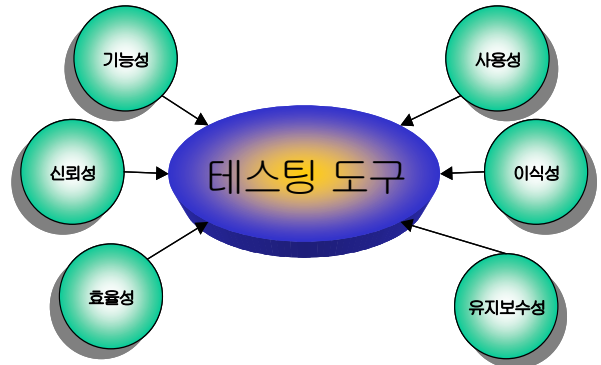
이장에서는 임베디드 소프트웨어 테스트를 지원하는 환경과 도구에 관한 요구사항과 연구과제로서 수행할 ESTE(Embedded Software Testing Environments)의 구조를 소개한다.

3.1 요구사항 분석

본 연구에서 제시하는 임베디드 소프트웨어 테스트 도구의 요구사항을 분석하기 전에 임베디드 소프트웨어에 요구되는 사항은 (그림 2)와 같다.

<표 2> 임베디드 소프트웨어의 품질 평가 요구사항

특성	내용
신뢰성	하드웨어와 연동하여 동작하기 때문에 오류발생 시 적용 환경에 따라 큰 문제점을 발생시키며 수정하기 어려움
이식성	하드웨어 플랫폼의 쉬운 변경에 의한 다양한 플랫폼 지원과 소프트웨어 변경이 쉽고 간단하게 이루어져야 함
재사용성	신제품 개발 주기가 빨라지는 문제점 보완 대책
실시간성	내부 또는 외부에서 발생하는 이벤트에 대해 실시간으로 반응, 동작을 수행해야 함
호환성/확장성	오랜 기간 사용됨으로 기존 임베디드 소프트웨어에 대한 호환과 기능확장이 용이해야 함
하드웨어 최적화	소프트웨어와 하드웨어가 동시에 개발되므로 서로간에 최적화가 이루어져야 함
강한 내구성	외부 요소에 의해 일부 기능에 이상이 생겨도 기본적인 중요 기능은 동작할 수 있어야 함
초소형/초경량	임베디드 시스템은 주로 System-on-chip 기술이 적용되고 있으며, 소프트웨어는 이에 최적화되어 개발되어야 함
사용성	임베디드 시스템은 주로 일반 사용자들을 대상으로 하는 경우가 대부분으로 전문 지식 없이도 사용할 수 있는 편리성이 요구됨
상호운용성	하나의 임베디드 시스템이 다른 임베디드 시스템과 연결되어 작동할 때 이상 없이 작동하여야 함



(그림 3) 임베디드 테스트 도구의 품질 평가 요구사항

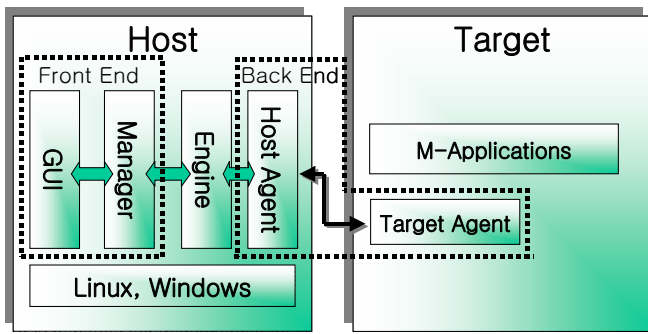
요구사항들은 임베디드 소프트웨어 전체 품질을 평가하는 근거로 사용되는 ISO/IEC 9126을 사용하였다. 이 외에도 기능적인 부분을 평가하는 요소들이 있는데 크게 하드웨어와의 제어, 임베디드 소프트웨어의 필요기능 포함여부, 소프트웨어 구성 요소들의 통합 관점과 임베디드 소프트웨어와 대상 하드웨어간의 통합관점에서의 상호작용 고려 여부, 마지막으로 임베디드 소프트웨어 전체에서 고려해야 하는 기능적 요구사항을 평가하는 4가지 요소로 나누어지는 2가지 구조로 되어 있다.[2-3]

<표 2>는 이러한 임베디드 소프트웨어 품질평가 요소들 중 품질 평가 요소들을 나타낸 것이다. 표의 요구사항들에 의해 나타난 임베디드 소프트웨어 테스트 도구의 품질 평가 요구사항은 (그림 3)과 같이 표시되며, 그 세부내용은 <표 3>과 같다.[5-6]

이외에도 임베디드 소프트웨어의 응용 분야가 더욱 다양해지고 수요가 증가하여 개발이 증가할수록

<표 3> 임베디드 소프트웨어 테스트 도구의 품질 평가 요구사항

주특성	부특성	적용
기능성	적합성	다양한 타겟 보드에 적용 가능 여부
	정확성	테스트 결과의 정확성
	상호운용성	타겟보드와의 상호운용성
신뢰성	보안성	테스팅 결과의 보호
	성숙성	자체 진단의 여부
	오류허용성	타겟 보드의 오동작, 사용미숙 등
효율성	복구성	오류 발생시 작업 복구 또는 결과 보호
	시간반응성	실시간 정보 처리
	자원효율성	타겟 보드의 CPU, 메모리, 전력 사용정도
사용성	이해성	테스팅 도구의 이해 여부
	습득성	테스팅 도구 사용지식 획득의 용이성
	운용성	도구의 운용이 쉬운가?
이식성	친밀성	사용자가 도구 사용 방식에 잘 적응하는가?
	적용성	다양한 기기에서 사용 가능한가?
	설치성	설치가 용이한가?
유지보수성	공존성	타겟 내의 어플리케이션과의 충돌은 없는가?
	대체성	기존의 제품과의 차이점은 무엇인가?
	해석성	도구 자체의 오류 진단이 가능 유무
	변경성	도구의 변경이 용이한가?
	안정성	변경 후의 도구에 발생하는 문제 최소화



(그림 4) ESTE 시스템의 구조

그에 따른 요구사항들이 나타날 것이며, 이러한 요구들에 의해 테스트의 중요성은 더욱 커질 것이다.[4]

3.2 ESTE시스템 구조

(그림 4)는 임베디드 소프트웨어 테스트 도구인 ESTE (Embedded Software Testing Environments) 시스템의 구조이다. ESTE는 임베디드 시스템의 특징적인 요소들을 만족시킬 수 있는 실시간 임베디드 소프트웨어 테스트 지원을 위한 환경도구이다.

개발할 시스템의 목표는 모바일 기반-응용 시스템의 효과적인 개발을 지원하고 양호한 품질의 제품을 생산하고 새로운 호스트나 타겟에 호환성이 있고 사용자가 시스템에 관한 상세한 지식을 가지고 있지 않더라도 사용할 수 있는 환경을 구축하는 것이다. 시스템의 구성 요소는 다음과 같다.

- Front-end : GUI, Manager
 - ◆ GUI : 사용자로부터 직접 입력을 받아 테스트 엔진으로 전달하고 처리결과를 받아 분석, 재가공하여 사용자에게 전달한다.
 - ◆ Manager : 테스트 엔진과 직접 연동하여 테스트 엔진으로부터 명령전달과 처리 결과 분석을 담당하며 명령어 해석기능을 가지게 한다.
- Engine : Front-end로부터 전달받은 명령을 해석하고 실행하며 호스트에서 처리 가능한 경우 직접 호스트에서 처리하고 타겟에서 처리가 필요한 경우 Back-end로 전달하여 처리하게 한다.
- Back-end : Agents
 - ◆ 타겟 에이전트는 호스트에서 전달된 디버깅 명령들을 타겟에서 실행하는 테스트에서 실제 적용한다.
 - ◆ 타겟 수준의 디버깅 정보를 파악하여 호스트로 전달하는 소프트웨어 테스트 모니터 역할도 담당한다.

4. 결론 및 향후연구

본 논문에서는 현재의 임베디드 시스템 기술 개발 현황과 이에 따른 테스트 도구들의 문제점과 해결책을 기술하였다. 결국 더 나은 임베디드 소프트웨어를 효과적으로 개발하는 데는 그에 맞는 테스트 도구가 필수적이며, 그러한 기술을 획득하는 것은 매우 중요한 것이다.

테스팅 도구의 사용에 의한 기여와 효과를 나타내면 학문적, 기술적, 경제적으로 나타낼 수 있다. 학문적인 관점에서는 프로덕트 라인 접근을 통한 임베디드 소프트웨어 개발과 전개 방법의 제시와 품질 향상, 개발비용 및 유지보수 비용이 감소가 있다.

기술적인 관점에서는, 테스트 도구 개발 분야에 최신 소프트웨어 개발 기술인 CBD기술 적용을 통한 독자적 원천 기술의 확보가 가능하다. 또한 효율성, 품질성 확보가 이루어지고 미래 유비쿼터스 컴퓨팅 모델의 기반 확립이 가능해 진다.

경제적 효과를 살펴보면, 다양한 사용자 요구에 신속히 대응하고 기술적 변화를 민첩히 수용하는 프로덕트 라인의 개발 방법 및 도구를 통한 새로운 임베디드 소프트웨어 시장 창출에 기여하며, 기술 생산국으로의 전환으로 글로벌 IT 리더십 확보가 가능해 진다.

향후, 제시된 개념과 구조를 바탕으로 소프트웨어를 구체적으로 설계하고, 이를 토대로 구현해 나갈 것이다.

참고문헌

- [1] 김문희, 김두현, 김정국, "임베디드 시스템과 소프트웨어공학", 정보과학회지 제22권 제6호, pp5-13, 2004.
- [2] 최현미, 성아영, 조나경, 최병주, 반효경, 김재용, "임베디드 소프트웨어 평가 모델 구축 방안", 한국정보과학회 가을 학술발표논문집 Vol.31, No2, pp349-351, 2004.
- [3] 이용호, 정창신, 신석규, "임베디드 소프트웨어 시험 사례", 정보과학회지 제22권 제6호, pp68-76, 2004
- [4] 성아영, 최병주, 최진영, 이나영, 이장수, "소프트웨어 관점에서 본 내장형 시스템의 테스트 프로세스", 한국정보과학회 추계학술대회 Vol. 30, pp271-273, 2003
- [5] Edward L. Jones, Christy L. Chatmon, "A PERSPECTIVE ON TEACHING SOFTWARE TESTING", JCSC(Journal of Computing Sciences in Colleges) Vol. 16 Issue 3, 2001
- [6] 배현섭, "텔레매틱스 분야의 임베디드 소프트웨어 테스트" 월간 임베디드월드, 슈어소프트테크, 2005
http://www.suresofttech.com