

임베디드 소프트웨어 테스트 자동화 도구 성능향상을 위한 테스트슈트 제너레이터의 설계

박제원*, 박진호, 이남용
송실대학교 컴퓨터학과

jwpark@ssu.ac.kr, jhpark@selab.ssu.ac.kr
Nylee@Computing.ssu.ac.kr

A Design of Test Suite generator for Improving the Embedded Software testing Automation tool

Jae-won Park, Jin-ho Park, Nam-yong Lee
Department of Computing Graduate School, Soong-Sil University

요 약

임베디드 소프트웨어는 다른 일반 시스템과는 여러 가지 다른 특성을 가지고 있다. 우선 프로세서 자원과 메모리 자원을 적게 사용해야 하며, 소프트웨어의 오류에 대해 테스트 탭 또는 기타 서버환경에 비해 확실한 소프트웨어의 검증을 요구한다. 이러한 소프트웨어의 검증은 임베디드 소프트웨어의 오류허용이 다른 환경에 비해서 매우 다르기 때문에, 고난도의 임베디드 소프트웨어의 응용을 빠르고 안정되게 개발하기 위해서는 사용자가 쉽게 사용할 수 있는 기술이 절실히 필요한 실정이다. 본 논문에서는 기존의 테스트 자동화도구의 핵심 이슈인 테스트 데이터 제너레이터와 테스트 스크립트의 기능을 포함한 모듈인 테스트슈트 제너레이터의 성능향상 위한 방법을 제시한다. 제시된 테스트슈트 제너레이터 모듈은 임베디드 소프트웨어 테스트 자동화도구의 개발 시 활용하여 개발한다면 임베디드 소프트웨어를 보다 효과적으로 테스트 할 수 있을 것으로 예상된다.

1. 서론

좋은 품질 및 신뢰성을 절대적으로 기대하고 있다. 이러한 임베디드 소프트웨어에서 좋은 품질 및 신뢰성을 기대하기 위해서 필요한 것이 테스트 자동화도구이며, 현재 Rational TestRealtime등과 같은 대표적인 임베디드 테스트도구 등이 범용적으로 사용되고 있다. 하지만, 이러한 범용적 테스트 도구에는 특정 임베디드 개발 환경에 적용하거나 개발 조직이 직접 테스트 시스템을 커스터마이징 해야 하기 때문에, 고도의 전문 인력 및 별도의 컨설팅이 필요하다는 단점과, 또한 임베디드 소프트웨어를 테스트 할 때 필요한 요구사항을 모두 만족하지 못하고 있다.

1.1 연구의 배경

최근 임베디드 소프트웨어 산업은 기술 개발과 시장 경쟁에 의해 빠르게 변화하고 있으며, 마이크로프로세서의 가격이 낮아지고 소형화 및 고성능화가 진행함에 따라 제품 경쟁력의 핵심인 하드웨어 생산에서 소프트웨어 최적화 기술로 이동함에 따라 상품의 가치가 소프트웨어에 의해 좌우되는 기술집약적 고부가 가치 산업으로 발전하고 있다.

본 논문에서는 임베디드 소프트웨어 자동화도구에서 핵심모듈인 테스트슈트 제너레이터의 성능 개선의 필요성과 성능개선 방법을 알아보고 임베디드 소프트웨어 테스트 자동화도구의 전반적인 성능 개선을 위한 방법을 제안한다.

초기의 임베디드 소프트웨어는 산업 기기를 제어하는 간단한 제어 프로그램이 대부분이었으나 최근 복잡한 기능 및 네트워크 기능을 제공하는 임베디드 운영체제가 등장하고, 다양한 플랫폼이 출현함에 따라 실시간성, 신뢰성, 하드웨어 최적화와 같은 임베디드 소프트웨어에 대한 비 기능적 요구사항이 증가하고 있다. 이런 결과로, 사용자들은 성능 좋은 임베디드 소프트웨어 개발요구와

1.2 관련연구

1.2.1 테스트 자동화 도구

테스트 자동화란 더 많은 코드를 보다 짧은 시간에 테스트 하고자 수동식의 테스트 프로세스를 자동화한 것으로 수동으로 테스트하기 어렵거나 불가능한 경우에 바람직한 방법이다. 이러한 테스트 자동화를 위해서는 형식을 갖춘 테스트 프로세스와 기대되는 결과를 예측할 수 있는 세분화되고 반복적인 테스트 케이스 생성이 요구된다. 테스트 자동화를 이용할 경우 사람이 직접 테스트 케이스를 만들고 실행시킬 때보다 훨씬 많은 수의 테스트 케이스를 가지고 테스트를 수행하므로 테스트 커버리지를 높여 신뢰성을 향상시킬 수 있으며 테스트 케이스를 생성하고 실행하기 위해 소요되는 많은 시간을 단축시킬 수 있다. 현재 임베디드 시스템을 테스트 할 수 있는 대표적인 테스트 자동화 도구로는 국외로는 Rational TestRealTime라는 제품과, Telelogic Tau TTCN Suite 등이 있다. 전자의 제품은 자동화된 SCI기법을 적용하여 Test Harness를 생성하고 타겟기반의 테스트를 수행하는 시스템이며 후자는 통신장비업체에서 사용되며 일반적인 테스트 환경을 제공하는 시스템이다. 다음 그림은 국내의 테스트 자동화 도구의 제품을 나타내고 있다.

제품명	내용
Astonliunx의 CodeMaker	윈도우 환경에서 리눅스 기반 타겟 시스템을 개발하는 통합 개발환경
MacroImpact의 codeScroll	일반 패키지S/W를 대상으로 컴포넌트 테스트를 지원하는 자동테스트 도구
Rational TestRealtime	테스트하니스를 생성하고 타겟기반 테스트를 수행하는 시스템
TestQuest Pro	기존 핸드폰 단말기의 Manual Testing Procedure를 자동화 하는 테스트 시스템

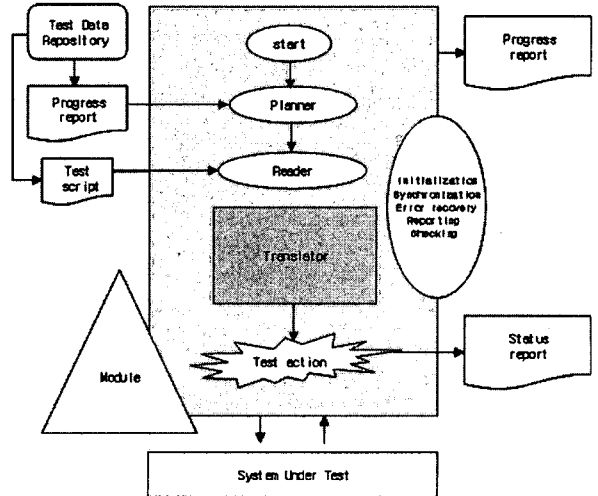
[표 1] 국내 외 테스트 자동화 도구 현황

1.2.2 테스트슈트 제너레이터

테스트슈트 제너레이터란 일련의 함수 호출 시퀀스에 의해 수행될 개별 테스트들의 집합, 또는 패키지들을 생성하며 테스트스크립트를 생성하는 모듈을 말한다. 다음 그림은 테스트슈트 제너레이터의 일반적인 형태를 설명하고 있다.

테스트슈트 제너레이터는 테스트 자동화 도구의 핵심 모듈로, 일련의 함수 호출 시퀀스에 의해 수행될 개별 테스트의 집합, 또는 패키지들을 생성하며, 테스트 스크립트를 생성하는 일을 하며, 그 구성모듈로는 테스트 데이터 제너레이터, 테스트스텝 제너레이터, 테스트 드라이버 제너레이터로 구성되어있다. 테스트슈트 제너레이터의 테스트스텝과, 테스트 드라이버는 테스트 스크립트에 대한 템플릿을 자동으로 생성할 때, 소스 코드를 분석하는 기능을 한다. 분석 결과는 소스 모듈 내 Identifier를 추출하여 테스트 데이터로 선정하게 되고, 그중 의존적인 모듈을 테스트 스텝으로 처리해 이 스크립트에서 필요한 모듈에 대한 인터페이스의 구체적인 값을 지정할

수 있게 한다. 핵심 모듈중 하나인 테스트 드라이버는 소스코드 분석과 동시에 테스트 하고자 하는 모듈을 구동시키고자 하는 테스트 드라이버 모듈을 생성하는 기능을 한다.



[그림 1] 테스트슈트 제너레이터

2. 본론

본론에서는 현재 사용되고 있는 테스트 자동화 도구에서 구현된 테스트슈트 제너레이터의 문제점을 분석하여, 이를 해결할 수 있는 방법을 제시하고, 제시된 방법을 통해 임베디드 시스템에 적합한 테스트슈트 제너레이터를 제안하였다.

2.1 테스트슈트의 성능개선을 위한 핵심 요구사항

기존의 테스트슈트 제너레이터를 임베디드 시스템에 적합한 모듈로 만들기 위한 요구사항은 크게 세 가지가 있다. 이 요구사항은 범용적으로 사용하던 테스트 자동화 도구에서 지원하지 않거나 혹은 임베디드 소프트웨어를 테스트 할 때 필요한 사항이다. 요구사항은 첫째, 테스트 테스트가 스크립트 언어의 용이한 사용으로 테스트 할 수 있어야 한다. 둘째, 타겟에서 사용하는 임베디드 소프트웨어에 적합한 테스트 데이터 제너레이터 생성기가 필요하다. 셋째, 테스트 수행정보에 대해 보다 상세 모니터링을 할 수 있는 테스트 수행엔진이 있어야 하며, 마지막으로 적절한 테스트케이스 생성이 이루어져야 한다.

2.2 제안하는 테스트슈트 제너레이터의 개선사항

임베디드 소프트웨어에서 일반적인 테스트 수행을 위해 필요한 요소는 테스트 케이스생성과 테스트 드라이버, 그리고 테스트 스텝이 있다. 여기에서 가장 핵심이 되는 부분은 테스트 케이스생성에 관한 부분인데 그 이유는 다른 모듈인 드라이버와 스텝은 범용적 테스트 자동화 도구와 생성방법 크게 다르지 않기 때문이다. 따라서 테스트 케이스 생성에 관련된 부분 중 핵심인 테스트

스크립트를 임베디드 소프트웨어에서 필요한 요구사항에 맞게 개선한다면, 보다 향상된 기능을 제공할 수 있을 것으로 보인다. 본 논문에서는 이러한 점을 테스트슈트 제너레이터의 성능개선을 통해 해결할 수 있도록 세 가지 사항을 제안하였다.

첫째, 테스트 케이스를 생성하기 위한 언어로 만든 테스트 스크립트 언어를 xml 형태로 만들어, 테스트 수행 시 스크립트 작성 및 데이터 재사용의 편의성을 제공하며, 스크립트의 내부적인 기능 중 입력과 예상결과의 설정과, 테스트 경로설정을 통해 임베디드 소프트웨어 테스트의 기능을 확장할 수 있는 기능을 제안한다. 스크립트 언어의 작성 시 보통 일반 프로그래밍 언어를 이용해서 작성하거나 개별 프로그램으로 작성되는 것이 보통이다. 개별 프로그램으로 작성하면 테스트 케이스 하나하나가 독립적이라는 면에서 장점이 있으나, 테스트 스크립트 작성에 시간이 많이 걸리는 단점이 있다. 반면에 회사별 독자언어는 테스트를 위해서 준비된 라이브러리를 충분히 사용할 수 있다는 장점이 있으나, 호환성이 좋지 않다. 이러한 점을 보완하기 위해 본 논문에서는 xml로 테스트스크립트의 작성을 통해, 기존 회사에서 사용한 호환성 문제의 해결과, 사용자가 편의성을 제공하여, 사용자의 용이한 스크립트를 사용으로, 양질의 테스트케이스 생성의 바탕이 된다. 또한 입력 값 과 예상 결과 설정 기능을 확대함으로써, 테스트 수행시 결과 값의 예상을 통한 정확한 테스트가 가능해진다. 그리고 테스트 경로설정의 기능은 사용자가 불필요한 경로는 삭제하고 필요한 부분만을 선택하여 테스트 할 수 있기 때문에 보다 시간의 단축과, 효율적인 테스트를 가능케 한다.

둘째, 동적 테스트데이터 제너레이터의 생성기의 제공을 통해 각 스크립트에서 필요한 데이터를 동적으로 생성기 기능을 제공하는 것이다. 현재 사용되고 있는 범용적 자동화 도구에는 테스트데이터 제너레이터가 사용되지 않고 있다. 하지만, 임베디드 소프트웨어를 보다 신속하고 정확하게 테스트하기 위해서는 동적으로 테스트데이터를 생성하는 테스트데이터 제너레이터가 꼭 필요하다. 동적 테스트데이터 제너레이터가 제공하는 기능은 주어진 테스트 스크립트에 대해서 초기 테스트 데이터를 생성해서 테스트를 수행하고, 그 결과를 바탕으로 추가 데이터를 생성해서 테스트 수행을 계속할지 혹은 해당 테스트 스크립트에 대한 테스트 수행을 종료할지를 결정하는 기능을 하며, 테스트 수행에 필요한 데이터를 생성하여 테스트 수행시간의 감소와 테스트 결과의 빠른 모

니터링을 가능하게 해준다.

셋째, 테스트 케이스를 수행하고 수행결과를 모니터링 하는 테스트 수행엔진을 제공해야 한다. 임베디드 소프트웨어는 다른 소프트웨어와는 다른 오류와 메모리 사용 및 누수에 민감하기 때문에 기능 측면으로 테스트 수행 결과가 예상결과가 동일한지를 비교하여 정확한 예상결과를 제공과, 성능측면에서는 프로그램의 응답시간, 처리율 등 성능관련 요소를 측정할 수 있는 모니터링도구가 있어야 한다. 또한 메모리 관련해서는 메모리 누출 등 메모리 사용오류 등을 세밀하게 나타낼 수 있어야 한다.

3. 결론 및 향후 연구

임베디드 소프트웨어를 테스트하기 위해서는 기존의 테스트 도구보다는 보다 섬세하고, 시스템에 맞는 자동화 도구가 필요하다. 본 논문에서는 기존 테스트 자동화 도구에서 제공하지 못했던 기능적인 부분 중 테스트슈트 제너레이터에 필요한 세 가지 기능적인 부분의 추가를 통해 임베디드 소프트웨어에 필요한 테스트를 할 수 있도록 하였다. 이러한 기능적인 부분을 테스트슈트 제너레이터가 제공할 수 있다면 보다 효율적이고 애러 없는 소프트웨어 테스트가 진행될 수 있을 것이다.

향후 연구로는 XML을 사용한 테스트 스크립트를 설계와 테스트 데이터 제너레이터의 구현을 통해 임베디드 시스템에 적합한 테스트 자동화도구의 구현을 연구를 진행할 것이다.

참고문헌

- [1] E.ALEE, What's Ahead for Embedded soft-ware IEEE Computer, pp18-26 September,2000
- [2] Bart Brokman, Testing Embedded Software, Addison-wesley, Dec, 2002
- [3] L.Hatton, Embedded software testing, Software Testing Congress, 2000
- [4] Richard A. DeMillo, Richard J. Lipton, and Fre-drick G. Sayward, Hints on test data selection: Help for the practicing programmer. IEEE Computer, April 1978
- [5] Jean Alert 외, Validation Based Development of dependable systems, IEEE micro, Jul, 1999
- [6] J.R. Horgan and S. London, "ATAC" : A data flow coverage testing tool for C." in Proceedings of Symposium on Assessment of Quality Software Development Tools, New Orleans, LA, MAY, 1992

[표 2] 테스트 자동화도구의 개선사항

현재 사용되는 테스트슈트 제너레이터	테스트슈트 제너레이터의 개선사항	테스트슈트 제너레이터의 개선 내용
스크립트 언어를 자체적으로 만들거나 perl 등과 같은 범용적 언어로 작성	스크립트의 언어를 XML형태로 변경	기존의 범용스크립트 언어 및 독자적 스크립트 언어를 xml 형태로 개선하여, 사용의 편리성 제공
같은 기능을 제공하지 않음	동적 테스트데이터 제너레이터 생성기	데이터 생성기를 통한 테스트 기능 확대
비슷한 기능을 제공하지만 임베디드 소프트웨어에 특화된 기능 제공하지 못함	임베디드 시스템에 특화된 부분을 상세하게 모니터링 할 수 있음	임베디드 시스템에 특화된 메모리 관련 기능의 상세한 모니터링기능 제공