

XML 테스트 스크립트를 이용한 내장형 시스템 소프트웨어 테스팅 도구

곽동규^o 조용운 유재우

송실대학교 일반대학원 컴퓨터학과

{coolman^o,yycho}@ss.ssu.ac.kr, shkim@dmclab.co.kr, cwyo@comp.ssu.ac.kr

Embedded System Software Testing Tool Using XML Test Script

Dong-Gyu Kwak^o Yong-Yoon Cho Sang-Heon Kim Chea-Woo Yoo

Dept of Computing, Soongsil University

요 약

내장형 시스템의 요구사항이 복잡해짐에 따라 신뢰성이 높은 소프트웨어 생산이 어려워지고 있다. 본 논문은 신뢰성 높은 내장형 시스템의 소프트웨어를 생산을 위해 교차 컴파일 환경에서 사용 가능한 소프트웨어 테스트 도구들을 제안한다. 일반적으로 테스트 도구는 독자적인 테스트 스크립트를 사용한다. 그러므로 테스트를 하고자하는 개발자는 테스트 도구에서 사용하는 테스트 스크립트를 학습하여야 한다. 즉, 개발자가 기존의 테스트 도구를 사용하기 위해서는 새로운 스크립트 언어를 학습해야 하는 부담을 가지고 있다. 본 논문에서 제안하는 시스템은 이러한 단점을 극복하기 위해서 개발자에게 친숙한 XML을 이용하여 테스트 스크립트를 설계한다. XML은 마크 업 언어의 표준으로 다양한 응용을 가지고 있고 다른 형태의 포맷으로 쉽게 변환 가능하다는 장점을 가지고 있다. 또한, GUI 기반의 테스트 스크립트 생성기를 제공하여 개발자에게 직관적인 테스트 스크립트 작성을 할 수 있도록 제안한다. 그리고 기존의 테스트 스크립트와 달리 테스트 스크립트 언어 레벨에서의 테스트 분기를 제공하고 있어 테스트 결과에 따른 다양한 테스트를 실시할 수 있다. 본 테스트 도구는 개발자에게 테스트 드라이버 작성을 위한 노력을 줄여 더욱 질 좋은 프로그램을 생성하는데 기여할 것으로 기대된다.

1. 서 론

현재 내장형 시스템을 채용한 정보 기기들에 대한 사용자의 요구사항이 복잡해지고, 보다 다양한 주변장치들과 연결되어야 하는 필요에 따라 내장형 소프트웨어 또한 더욱 복잡해져서 신뢰성 높은 소프트웨어 생산이 어려워지고 있다. 신뢰성 높은 소프트웨어란 요구사항에 합당한 동작을 하는 소프트웨어를 의미한다. 소프트웨어는 요구사항에 따라 다수의 모듈로 이루어진다. 그러므로 개발 단계에서 각 모듈의 단위 테스트를 통해 전체 소프트웨어의 신뢰성을 증가시킬 수 있다. 각 모듈의 단위 테스트를 위해서는 요구사항을 분석하여 모듈의 입력 데이터와 출력 데이터인 테스트 데이터를 작성하고 작성된 테스트 데이터를 바탕으로 모듈 테스트를 위한 프로그램 즉, 테스트 드라이버를 생성해야 한다. 본 논문은 모듈 테스트를 위한 테스트 드라이버를 자동으로 작성할 수 있는 테스트 도구를 제안한다. 일반적인 테스트 도구 [1][2]는 모듈의 요구사항에 합당한 입력과 출력을 테스트 스크립트로 작성하여 테스트 드라이버를 생성한다. 특히 내장형 시스템에서는 생성된 테스트 드라이버를 교차 개발 환경에서 컴파일하여 타겟 시스템에서 실행시킨다. 기존의 방법은 테스트 스크립트 작성을 위해 스크립트 언어를 학습해야 사용할 수 있고 테스트 결과에 따른 분기를 제공하고 있지 않아 다양한 테스트를 실시하기 어렵다.

제안하는 시스템은 스크립트 언어로 XML을 사용하여

스크립트 언어에 대한 학습이 쉬운 장점을 가지고 있다. XML은 마크 업 언어의 표준으로써 다양한 분야에서 데이터를 표현하는 방법으로 많이 사용되고 있다. 그리고 XSLT [3]와 같은 다양한 응용을 가지고 있어 다른 형태의 포맷으로도 변환하기 쉽다. 또한, GUI 기반의 스크립트 작성 도구를 제공하고 있어 초보자도 편리하게 스크립트를 작성할 수 있는 장점을 가지고 있다. 또한, 테스트 스크립트에서 테스트 결과에 따른 분기를 제공하고 있어 테스트 결과에 따라 다양한 테스트를 실시할 수 있는 장점을 가지고 있다. 그림 1은 내장형 시스템에서의 소프트웨어 테스트 도구의 개념도이다.

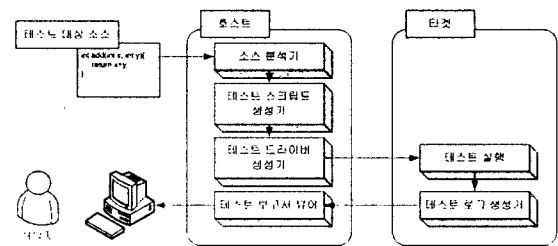


그림 1 내장형 시스템에서의 소프트웨어 테스트 도구 개념도

내장형 시스템에서 소프트웨어를 개발 환경은 교차 개발 환경을 사용한다. 제안하는 소프트웨어 테스트 도구는 그림 1과 같이 호스트에는 소스 분석기와 테스트 소

크립트 생성기, 테스트 드라이버 생성기, 테스트 보고서 뷰어를 가지고 있고 타겟은 테스트 실행과 테스트 로그 생성기를 가진다.

본 논문은 2장에서 관련 연구를 소개하고 3장에서 제안하는 시스템 구조를 보인 후 4장에서 테스트 스크립트 생성기에 대해 설명한다. 그리고 5장에서 결론 및 향후 연구 과제에 대해 기술한다.

2. 관련 연구

2.1 Rational Test RealTime[1]

RTRT(Rational Test RealTime)는 소프트웨어의 성능 측정과 테스트를 위한 프로파일 및 테스트를 제공하는 툴이다. RTRT는 트레이스와 메모리, 퍼포먼스, 커버리지 프로파일을 제공하고 테스트 스크립트를 통해 테스트 드라이버를 생성하고 결과를 제공한다. RTRT는 테스트 스크립트의 전역에 특수 문자를 사용하여 테스트 대상 언어와 함께 입력 값과 출력 값을 설정할 수 있도록 제공하고 있다. 하지만 이 방법은 개발자가 새로운 테스트 스크립트 언어를 학습해야 하는 부담을 가지고 있고, 스크립트 언어가 어디든지 포함할 수 있어 스크립트가 불필요하게 복잡해지고 테스트 드라이버가 잘못 작성될 위험이 높다.

2.2 JUnit[2]

JUnit은 Java로 작성된 프로그램을 테스트하기 위한 툴이다. JUnit은 자바로 구현되어 있으며 개발자는 패키지에서 제공되는 코드를 테스트 드라이버의 함수에 추가해서 테스트 드라이버를 작성한다. JUnit은 객체지향언어의 상속을 이용하여 손쉽게 테스트 드라이버를 생성하는 장점을 가지고 있다. 하지만 Java로 작성된 프로그램의 테스트를 지원한다.

3. 시스템 구조

제안하는 시스템은 소스 분석기와 테스트 스크립트 생성기, 테스트 드라이버 생성기, 테스트 실행기, 테스트 결과 보고서 뷰어로 나누어져 있다. 그림 2는 전체 시스템에서 각 모듈의 데이터 관계를 보여준다.

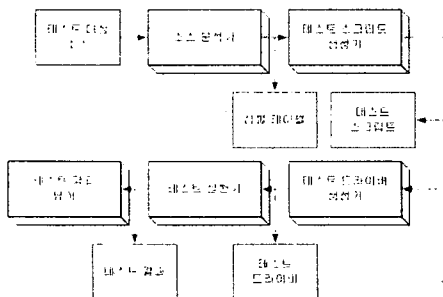


그림 2 각 모듈의 데이터 관계도

테스트 대상이 되는 소스는 분석기를 통해 대상 프로그램의 함수와 변수를 분석한다. 소스 분석기는 테스트 대상 소스를 입력으로 하여 함수와 함수의 인자, 전역 변수를 심볼 테이블로 작성한다. 작성된 심볼 테이블은 테스트 스크립트 생성기의 입력이고 사용자에게 GUI를 통해 제공된다. 개발자는 테스트 스크립트 생성기를 이용하여 직관적으로 테스트 스크립트를 작성하고 테스트 스크립트는 테스트 드라이버 생성기를 통해 실행 가능한 테스트 드라이버를 생성한다. 테스트 드라이버는 컴파일하여 실행되고 테스트 결과는 테스트 뷰어를 통해 사용자에게 직관적으로 제공된다.

4. 테스트 스크립트 생성기

본 논문은 테스트 스크립트를 개발자에게 친숙한 XML로 작성하고 이 XML을 XTS(XML Test Script)라 한다. 그림 3은 간단한 예제 XTS의 일부로 XTS의 구조를 보인다.

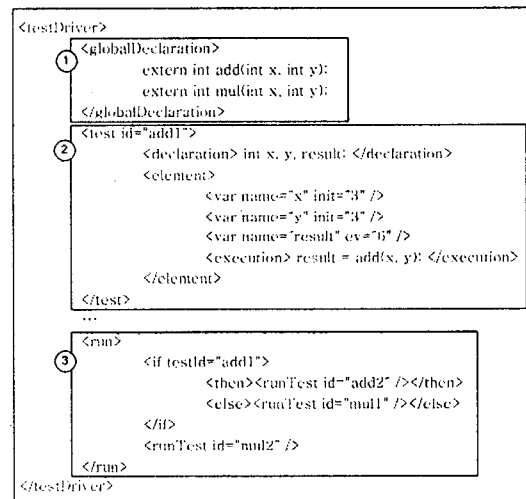


그림 3 XTS의 구조

XTS는 그림 2에서와 같이 ① 전역 변수 영역과 ② 단위 테스트 영역, ③ 테스트 실행 영역으로 나누어져 있다. ① 전역 변수 영역은 테스트 스크립트에서 사용할 전역 변수를 선언하고 테스트 대상이 되는 함수를 테스트 스크립트에서 사용할 수 있도록 포함시킨다. ② 단위 테스트 영역은 테스트를 실시하는 단위로서 테스트 데이터에 해당하는 모듈의 입력과 출력을 기술한다. ③ 테스트 실행 영역은 단위 테스트의 실행 순서를 기술하고 “<if>” 태그를 이용하여 단위 테스트의 결과에 따른 분기를 설정할 수 있다. 테스트 스크립트 언어 레벨에서의 분기는 복잡한 모듈간의 관계를 테스트 스크립트에 적용할 수 있는 장점을 가지고 있다. XTS는 사용자에게 친숙한 XML로 설계되어 있으나 초급 사용자에게는 더욱 직관적인 인터페이스가 필요하다. 그림 4는 XTS를 작성하

기위한 XTS 생성 마법사를 보여준다.

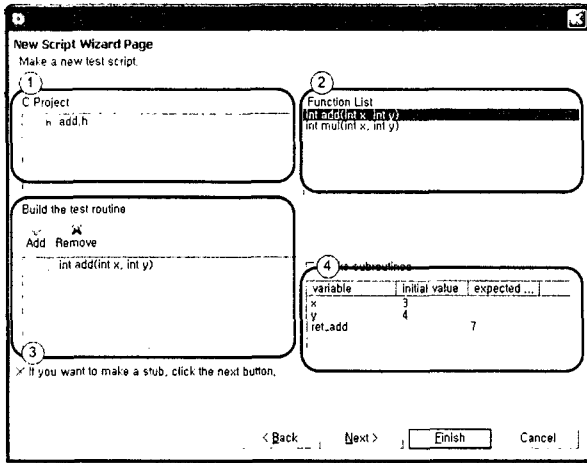


그림 4 XTS 생성 마법사

그림 4와 같이 XTS 생성 마법사는 네 가지로 구성되어 있다. XTS 생성 마법사를 통한 XTS 생성 과정은 다음과 같다.

- XTS 생성 과정

- ① 테스트 대상 프로젝트에서 소스 파일을 선택한다.
- ② 소스 파일에서 테스트 대상이 되는 함수를 선택한다.
- ③ 테스트의 분기를 설정한다.
- ④ 테스트의 입력과 출력을 셋팅한다.

표 1 테스트 결과 XML

```
<testResult><test name="add_1" result="fail" >
<element runTime="0" result="fail" >
<var name="x" init="3" initResult="3" evResult="" result="success" />
<var name="y" init="3" initResult="3" evResult="" result="success" />
<var name="ret_add" ev="9" initResult="" evResult="6" result="fail" />
</element></test>
<test name="add_2" result="success" >
<element runTime="0" result="success" >
<var name="x" init="1" initResult="1" evResult="" result="success" />
<var name="y" init="1" initResult="1" evResult="" result="success" />
<var name="ret_add" ev="2" initResult="" evResult="2" result="success" />
</element>
</test>
<runPath><if testId="add_1" result="fail" >
<then><runTest id="mul_1" /></then>
<else><runTest id="add_2" result="success" /></else>
</if></runPath>
</testResult>
```

그림 4와 같은 직관적인 XTS 생성 마법사는 테스트 스크립트를 잘 모르는 사용자도 쉽게 스크립트를 작성할 수 있도록 도와준다. XTS 생성 마법사를 통해 작성된 스크립트는 테스트 드라이버 생성기를 통해 테스트 대상과

동일한 언어로 번역되어 컴파일한다. 컴파일된 소스는 타겟에서 실행하여 테스트 결과를 로그로 생성하고 로그를 분석하여 다양한 형태로 변환 가능하고 응용이 풍부한 XML의 형태로 출력된다. 표 1은 테스트 결과를 보여준다.

테스팅 결과 화면은 사용자가 설정한 테스트 경로에 따라 테스트의 성공/실패를 XML 엘리먼트로 표현하여 사용자가 요구하는 형태로 변환 가능하다.

5. 결론 및 향후 연구과제

내장형 시스템의 소프트웨어의 요구사항은 내장형 시스템이 가지는 기능이 증가함에 따라 날로 복잡해지고 있다. 이에 따라 신뢰성 높은 소프트웨어를 생산하기 위해 소프트웨어 테스트의 중요성이 부각되고 있다. 기존의 테스트 툴은 사용자가 제공되는 라이브러리를 이용하여 직접 테스트 드라이버를 작성하거나 독자적인 테스트 스크립트를 작성하여 테스트를 실시한다. 이러한 방법은 사용자가 테스트를 위해 제공되는 라이브러리를 테스트 스크립트를 학습해야하는 부담을 가지고 있다. 또한, 테스트 스크립트 언어 레벨에서의 분기를 제공하고 있지 않아 다양한 테스트를 실시하기 어려운 단점을 가지고 있다.

본 논문에서 제안하는 시스템은 이러한 단점을 극복하기 위해 테스트 스크립트 언어로 개발자에게 친숙한 XML을 사용하고 초보 개발자를 위해 GUI 기반의 스크립트 생성기를 제안한다. 제안하는 시스템은 내장형 시스템 개발 환경인 교차 개발 환경에 적합하도록 테스트 결과를 통신을 통해 호스트에 전달하도록 설계한다. 사용자가 테스트 스크립트 생성기를 통해 작성한 테스트 스크립트는 자동으로 테스트 드라이버로 변환되고 컴파일하여 타겟에서 실행된다. 또한, 테스트 스크립트 언어 레벨에서 테스트 결과에 따른 분기를 제공하고 있어 보다 다양한 테스트를 실시할 수 있는 장점을 가지고 있다. 하지만 본 논문은 테스트를 실시하기 위한 테스트 데이터 생성 방법에 대해서는 고려하지 않았다. 테스트 데이터 생성은 사용자가 지정한 경로에 합당한 제약식의 연산을 통해 생성해야 하는데 이 단계가 가장 노력이 많이 들고 자동화하기 어렵다.[4] 현재 다양한 방향으로 진행되고 있는 테스트 데이터 생성의 자동화에 대한 연구와 함께 본 논문에서 제안한 시스템은 신뢰성 높은 소프트웨어 생산에 기여할 것으로 기대한다.

참고 문헌

- [1] Rational Test RealTime, <http://www-306.ibm.com/software/awdtools/test/realtime>.
- [2] JUnit, <http://junit.org>.
- [3] XSLT, <http://www.w3.org/TR/xslt>.
- [4] 정인상, "소프트웨어 테스트를 위한 테스트 데이터의 자동 생성", *한국정보과학회 논문지, 제9권 11호*, pp.10-18, 2001.11.