XML 테스트 스크립트 기반 단위 테스팅 도구의 설계 및 구현

김재현° 조용윤 유재우
숭실대학교 컴퓨터학과
{jaehyun°, yycho}@ss.ssu.ac.kr, cwyoo@computing.ssu.ac.kr

A Design and Implementation of the Unit Testing Tool based on a XML Test Script

Jaehyun Kim° Yongyoon Cho Chaewoo Yoo
Dept. of Computing, Soongsil University

요 약

소프트웨어 개발 생산성을 높이고 신뢰성 있는 프로그램을 개발하기 위하여 수많은 테스트 기법 및 도구들이 연구되고 있다. 본 논문은 효율적인 소프트웨어의 단위 테스팅을 위해 XML 기반의 테스트 스크립트 언어를 설계하고 테스팅 도구를 제안한다.

제안하는 테스팅 도구는 테스트 대상 소스를 기반으로 테스트 스크립트를 생성해주는 테스트 스크립트 생성기와 테스트 스크립트를 대상 언어로 작성된 테스트 드라이버로 변환해 주는 테스트 드라이버 변환기를 제공함으로서 보다 간편한 테스트 환경을 제공한다. 테스트 스크립트를 XML 형태로 기술함으로서 개발자들은 새로운 스크립트 언어의 학습이 불필요하며 대상 언어에 독립적인 테스트 스크립트를 작성 할 수 있다. 또한 테스트 실행 후 테스트 평가 결과를 XML로 제공함으로서 다양한 형태의 리포트 뷰(View)를 가능하게 한다. 본 XML 기반의 테스트 스크립트 언어와 테스팅 도구는 프로그래밍 언어에 독립적인 부분과 종속적인 부분을 분리하여 여러 가지 프로그래밍 언어의 단위 테스트 환경을 하나로 통합 할 수 있게 해주어 관련 소프트웨어 테스팅 분야의 발전에 크게 기여할 것으로 기대된다.

1. 서 론

소프트웨어 개발의 생산성을 높이는데 있어서 테스팅의 중요성이 강조되고 있다. 최근 각광을 받고 있는 개발 방법론중의 하나인 XP(eXtreme Programming)방법론에서는 무엇보다 테스팅의 중요성이 강조 되고 있으며[1] 개발해야 할 소프트웨어의 테스트 케이스를 먼저 작성하고 프로그램을 작성하기를 권고한다.[1][2] 소프트웨어의 개발 단계에서 작은 부분의 단위 테스팅은 프로젝트의 실패의 위험성을 감소시킨다. 소프트웨어 개발자 혹은 개발 관리자들은 소프트웨어를 개발 하는 과정에서 쉽고, 효율적인 단위 테스팅을 할 수 있기를 요구한다.

본 논문은 소프트웨어 단위 테스팅을 위한 XML형태의 테스트 스크립트 언어를 설계하고 본 테스트 스크립트 언어를 이용한 단위 테스팅 도구를 제안한다. 테스트 스크립트를 XML로 기술함으로서 기존의 단위 테스트 도구와 비교하여 가독성이 높고 익히기 용이하다는 장점이 있다. 테스트 대상 소스로부터 테스트 스크립트의 기본 골격을 생성하는 테스트 스크립트 생성기와 테스트 스크립트를 대상 언어로 작성된 테스트 드라이버로 변환해 주는 변환기를 제공함으로서 테스트 작업 용구(Test harness)를 쉽게 생성할 수 있도록 한다. 개발자는 테스트 스크립트 생성기에 의해 자동으로 생성된 테스트 스크립트에 테스트 값을 입력하고, 테스트 값이 입력된 테스트 스크립트를 테스트 드라이버 변환기를 이용하여 대상 언어의 테스트 드라이버로 변환한다. 이렇게 생성된 테스트 드라이버는 테스트 평가 결과를 추출하는 모듈을 포함하고 있다. 테스트를 수행한 후 생성되는 테스트 평가 결과는 데이터의 표현, 변환이 용이한 XML형태로 생성되어 다양한 형태의 보고서 뷰를 가능하게 한다.

본 논문은 2장에서 관련연구를 소개하고 3장에서 XML 기반의 테스트 스크립트 언어를 제안하고 4장에서 테스트 도구를 설계하고 5장에서 C언어로 작성된 프로그램을 대상으로 하는 테스트 도구를 구현한다. 그리고 6장에서 결론 및 향후 연구 과제를 기술한다.

2. 관련연구

2.1 단위 테스팅(Unit Testing)

Unit testing에서 "units"는 서브프로그램 혹은 서브프로그램의 집합을 의미한다.[3] 이 레벨은 테스트하기에 적합하며, 이보다 더 큰 시스템 레벨의 테스팅은 의미를 분석하는데 너무 많은 코드가 관여하여 테스트 레벨로서 부적합 하다.[3] 우리가 일반적으로 많이 사용하는 언어에서는 함수가 하나의 단위(unit)로서 분류 될 수 있다.

2.2 XML을 이용한 테스트 스크립트

미군(U.S. Army)의 자동 테스트 시스템 팀(ATST - Automated Test Systems Team's)이 주도한 XML기반의

테스트 스크립트 언어를 이용한 테스팅 방법은 XML의 장점을 살려 구현하기 쉽고, 유연하며, 다루기 쉽고, 이식성이 뛰어난 테스트 스크립트 언어를 보여주고 있다.[4] 그러나, 이 테스트 스크립트 언어에는 UI, 테스트 진행 절차 등의 광범위한 내용을 테스트 스크립트에 기술함으로서 테스트 스크립트가 복잡하다. 본 논문에서는 단위 테스팅에 필요한 요소만을 테스트 스크립트에 기술함으로서 보다 쉽게 익힐 수 있는 테스트 스크립트 언어를 제안한다.

2.3 Rational Test RealTime - C Unit Testing[5]

IBM의 Rational Test RealTime은 소프트웨어 개발 중에 필요한 성능측정을 위한 실시간 프로파일링, 테스팅 기능을 제공하는 툴이다. 이 툴의 단위 테스팅 부분에서는 자체적인 테스트 스크립트 언어를 제공 하는데, 개발자는 테스트 스크립트를 새롭게 익혀야 한다는 부담이 있다. 테스트 스크립트의 전역에 특수문자를 사용한 라인 주석 형태의 표현을 이용하여 대상 언어 코드를 포함시킬 수 있다. 이 특수문자를 이용한 대상 언어 코드의 삽입은 유연하고 광범위한 테스팅을 가능하게 하지만, 스크립트 언어의 어디든지 포함 될 수 있기 때문에 테스트 스크립트가 불필요 하게 복잡해지고, 테스트 드라이버가 잘못 작성될 위험이 있다.

2.4 JUnit[6]

Java언어로 작성된 프로그램의 단위 테스팅 도구로서 JUnit이 널리 쓰이고 있다. JUnit은 전체적인 프레임워크가 자바언어로 구현되어 있으며, 개발자는 패키지에서 제공되는 "assertEquals(x, y)" 등의 "assertXXX"형태의 코드를 테스트 드라이버의 함수에 추가해서 테스트 드라이버를 작성하면 된다. JUnit은 객체지향언어의 상속을 이용하여 손쉽게 간단한 구조의 테스트 드라이버를 생성하고, 실행할 수 있는 환경을 제공해 주지만 Java언어로 작성된 프로그램의 테스트에만 사용 될 수 있다.

3. XML 기반의 테스트 스크립트 언어 설계

테스트 스크립트 언어는 학습과 사용이 용이하고, 대상언어에 대해 독립적으로 기술할 수 있어야 한다. 대상언어에 따라 테스트 스크립트가 다르면 개발자는 언어마다 새로운 스크립트 언어를 익혀야 하는 부담이 발생한다. 현재 XML은 W3C표준으로 데이터를 저장하고 기술하는 용도로서 많은 응용 분야에서 사용되고 있다.

```
<!ELEMENT testDriver
        ( globalDeclaration?, environment*, test*, run? ) >
<!ELEMENT test
        ( declaration?, element+ ) >
<!ATTLIST test
        id      CDATA   #REQUIRED >
<!ELEMENT element
        ( (var | array | structure)*, execution ) >
<!ELEMENT var EMPTY>
<!ATTLIST var
        name    CDATA   #REQUIRED
        index   CDATA   #IMPLIED
        init    CDATA   #IMPLIED
        ev      CDATA   #IMPLIED
>
```

그림1. XML 테스트 스크립트 언어의 DTD 일부

그림1과 같이 테스트 기술부에는 test엘리먼트가 반복적으로 올 수 있으며, 하위 자식 노드로 테스트를 통과하기 위한 요소인 element엘리먼트가 반복적으로 올 수 있다. element엘리먼트는 실제 테스트 대상의 호출 등 대상언어로 기술하는 execution엘리먼트를 가지고 있으며, 테스트 성공, 실패의 조건이 되는 테스트 변수의 초기화 및 예상 값을 기술하는 var엘리먼트를 포함한다.

```
<testDriver>
        <test id="add">
                <declaration>
                        int x, y, rval_add;
                </declaration>
                <element>
                        <var name="x" init="3">
                        <var name="y" init="4">
                        <var name="rval_add" ev="7">
                        <execution>
                                rval_add = add(x, y);
                        </execution>
                </element>
        </test>
        <run>
                <runTest id="add">
        </run>
</testDriver>
```

그림2. add(x, y)함수의 테스트 스크립트 예제

테스트 스크립트의 구조는 그림2에서 보는 바와 같이 크게 테스트 기술부와 테스트 실행부로 나누어져 있다. 테스트 기술부에서는 test엘리먼트를 하나의 테스트 단위로 테스트를 기술하며, 테스트 실행부에서는 test엘리먼트의 성공, 실패의 여부에 따라 분기하는 테스트 플로우를 설정 할 수 있다.

4. 테스트 도구 설계

본 논문에서 제안하는 테스트 도구의 구성은 그림3과 같이 테스트 스크립트 생성기와 테스트 드라이버 변환기로 구성되며, 생성된 테스트 드라이버는 테스트 대상 코드와 함께 실행되어 테스트 결과를 보고서를 출력한다.

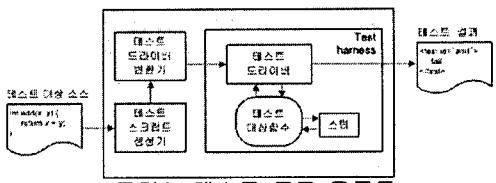

그림3. 테스트 도구 구조도

4.1 테스트 스크립트 생성기 설계

테스트 스크립트는 개발자가 직접 작성해도 되지만, 사용자의 편의를 위하여 기본 골격은 자동으로 생성 될 수 있도록 테스트 스크립트 생성기를 제공한다.

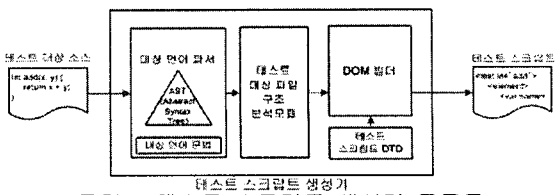

그림4. 테스트 스크립트 생성기 구조도

테스트 대상 파일을 대상 언어의 파서로 파싱하여 심볼 테이블 정보 및 함수 원형 정보를 추출한다. 테스트의 대상이 되는 함수의 개수, 매개변수의 개수, 타입 정보를 알 수 있으며, 이 정보를 바탕으로 DOM빌더의 API를 이용하여 테스트 스크립트의 XML DOM(Document Object Model) 트리를 생성한다.

4.2 테스트 드라이버 생성기 설계

테스트 대상이 C언어로 작성된 프로그램의 경우 결과적으로 테스트를 수행하는 테스트 드라이버는 C언어로 작성되어야 한다. 그러므로 XML형태의 테스트 스크립트는 대상언어로 작성된 테스트 드라이버로 변환해야 한다.

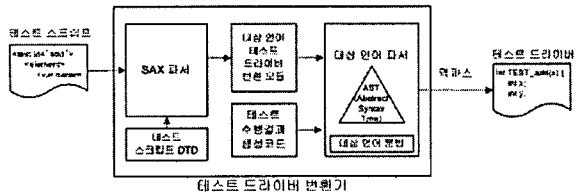

그림5. 테스트 드라이버 변환기 구조도

테스트 스크립트는 XML로 작성되었기 때문에 SAX파서를 이용하여 손쉽게 파싱 할 수 있다. 이 파싱된 엘리먼트 정보를 가지고 대상언어 테스트 드라이버 생성 모듈에서 변환 규칙에 따라 대상언어로 변환하여 대상언어 파서의 입력 스트림으로 입력한다. 또한, XML형태의 테스트 결과를 출력해 주는 모듈을 삽입한다. 대상 언어로 작성된 AST(Abstract Syntax Tree)는 역파스 루틴을 통하여 대상언어의 테스트 드라이버로 출력된다.

5. 구현 및 실험

C언어로 작성된 프로그램을 테스트하기 위한 테스팅 도구를 자바로 구현한다. 테스트 스크립트 생성기와 테스트 스크립트 변환기에서 사용될 C파서는 자바 언어 기반의 파서 제너레이터인 SableCC[7]에 C문법을 입력하여 C언어 파서를 생성한다. 생성된 C파서를 이용하여 C언어로 작성된 테스트 대상 프로그램의 변수 및 함수의 원형 정보를 분석하고, JAXP[8]의 DOM빌더를 이용하여 테스트 스크립트를 생성한다. 생성된 테스트 스크립트에 테스트 값을 입력 한 후, 테스트 드라이버 변환기를 이용하여 C언어의 테스트 드라이버로 변환한다.

그림6은 실험 예제로서 변수 두 개의 합을 구하는 C 프로그램의 테스트 드라이버를 생성한 결과이다.

```
extern int add ( int x , int y ) ;

int TEST_add ( ) {
        int TEST_RESULT;
        int x , y , rval_add ;
        x = 3;
        y = 4;
        CHECK_START_TIME();
        rval_add = add ( x , y ) ;
        CHECK_END_TIME();
        TEST_RESULT = CHECK_TEST_VAL (7, rval_add ) ;
        return TEST_RESULT;
}
int main ( int argc , char * * argv ) {
        OPEN_RESULT_FILE ( "REPORT_calc.xml" ) ;
        TEST_add ( ) ;
        CLOSE_RESULT_FILE ( "REPORT_calc.xml" ) ;
        return 0;
}
```

그림6. add(x, y)함수의 C언어 테스트 드라이버의 일부

6. 결론 및 향후 연구과제

본 논문에서는 소프트웨어 단위 테스팅을 위한 XML기반의 테스트 스크립트 언어를 설계하고 테스트 도구를 구현하였다. 테스트 스크립트 언어는 테스트 대상 언어에 독립적으로 설계하였으며, 테스트 대상 언어에 종속적인 부분은 대상언어를 스크립트의 PCDATA영역에 기술함으로서 테스트 스크립트의 언어 독립성을 유지했다. 그리고 테스트 스크립트는 대상 언어에 종속적인 변환 모듈에 의해서 대상언어의 테스트 드라이버로 변환 될 수 있도록 설계하였다.

향후 본 논문에서 제안한 테스트 스크립트 언어 및 테스트 도구는 보다 다양한 언어 및 환경에서 단위 테스팅을 수행 할 수 있는 소프트웨어 테스트 통합 환경으로 구축되어야 할 것이다.

참고문헌

[1] Lisa Crispin, Tip House, Testing Extreme Programming, Addison-Wesley, October. 2002

[2] Marc J. Balcer, William M. Hasling, Thomas J. Ostrand, Automatic Generation of Test Scripts from Formal Test Specifications, ACM, 1989

[3] Hamlet, R., Unit testing for software assurance, Computer Assureance COMPASS '89, Jun 1989

[4] Johnson, D.J. Roselli, P., Using XML as a flexible, portable test script language, AUTOTESTCON 2003. IEEE Systems Readiness Technology Conference. Proceedings, 187-192, Sept. 2003

[5] Rational Test RealTime, http://www-306.ibm.com/software/awdtools/test/realtime

[6] JUnit, http://www.junit.org

[7] SableCC, http://sablecc.org

[8] JAXP, http://java.sun.com/xml/jaxp