스츠럿츠2 프레임워크를 이용한 웹 어플리케이션의 테스트 자동화 방법

이천희*, 박석천**, 김종현***
*가천대학교 모바일소프트웨어학과
**가천대학교 컴퓨터공학과(교신저자)
***위세아이텍 대표이사
e-mail : yplch203@gc.gachon.ac.kr

Automatic Test Method of Web Applications by Struts2 Frameworks

Chun-Hee Lee*, Suck-Chun Pack**, Sun-Young Kim*
*Dept. of Mobile Software, Ga-Chon University
**Dept. of Computer Engineering, Ga-Chon University(Corresponding Auths)
***Representative President, WiseItech Co., Ltd.

요 약

최근 많은 웹 어플리케이션은 오픈소스 프레임워크를 이용하여 개발되면서 개발환경과 개발방법이 표준화 되어 개발 생산성이 향상되었다. 그러나 여전히 운영 및 유지보수 측면에서의 비용은 줄어들고 있지 않다. 하지만 테스트를 자동화 하면 시스템을 수정할 때 좀더 안전하게 수정하고 또한 수정하는 부분이 시스템 전체에 미치는 영향을 최소화 할 수 있다. 이에 본 논문에서는 오픈소스 프레임워크인 Struts2를 이용하여 웹 어플리케이션의 테스트 자동화 방법을 제안한다.

I. 서론

최근 많은 웹 어플리케이션은 오픈소스 기반의 웹어플리케이션 프레임 워크를 이용하여 개발되고 있다. 많이 알려진 오픈소스 프레임워크로는 스프링 프레임워크와 스트럿츠 프레임워크가 있다.

스프링 프레임 워크는 엔터프라이즈급의 어플리케이션에서 필요로하는 기능을 제공하는 프레임워크이다.

스트럿츠 프레임워크는 자바 기반으로한 웹언어인 Jsp만을 위한 프레임워크로서 MVC기반으로 모델과 컨트롤러, 뷰를 레이어로 구성하고 있다. 또한 각 부분의 연결을 느슨하게 하여 쉽고 빠르고 유지보수성을 쉽게 하도록 도와주는 대표적인 자바 웹 어플리케이션의 프레임워크다.

그러나 이와같이 프레임워크기반으로 개발된 어플리케이션이라 해도 개발단계부터 단위테스트기법을 적용하여 개발하지 않으면 여전히 운영유지보수에 많은 어려움이 따른다.

이를위해 본 논문에서는 스트럿츠 프레임워크로 개발된 웹 어플리케이션에 테스트를 자동으로 할 수 있는 방법을 제안하고자 한다.

II. 관련연구

1. 스트럿츠2 프레임워크

스트럿츠 프레임워크는 웹 어플리케이션을 쉽게 개발하고 쉽게 운영 유지 보수할 수 있도록 개발된 자바 프레임워크이다.

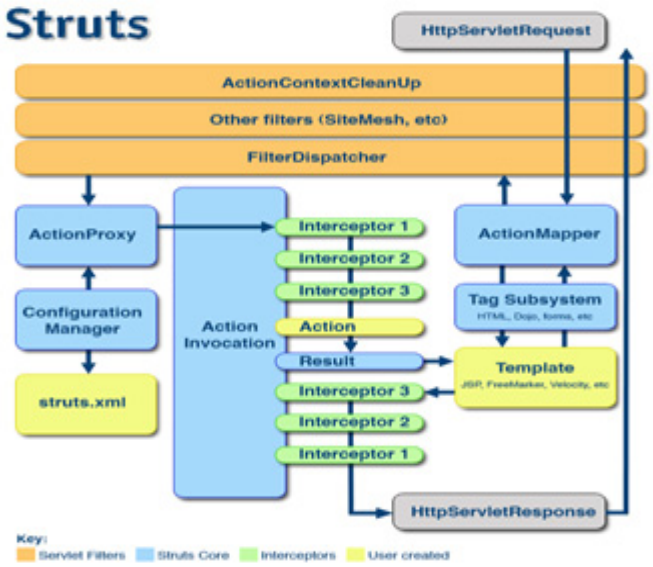
Java EE 웹 어플리케이션을 개발하기 위한 오픈 소스 프레임워크인 스트럿츠는 MVC 아키텍처를 적용하였고 개발자를 지원하기 위하여 자바 서블릿 API를 사용하여 확장하였다.

하지만 초창기 스트럿츠 프레임워크를 사용하기 위해서는 기존에 개발된 Bean기반의 자바 웹 어플리케이션을 재사용할 수 없었다.

또한 프레임워크를 사용하기 위해서 강제하는 사항들과 프레임워크 설정에 관련된 설정파일인 Config.xml을 작성하는 어려움 때문에 많은 개발자들로부터 외면을 받아왔다. 이와같은 사항들을 개선하여 나온 것이 스트럿츠2프레임워크다. 스트럿츠2 프레임워크는 기존 스트럿츠 프레임워크와 완전히 다른 구조를 가지고있다. 기존 스트럿츠 프레임워크가 서블릿 API기반의 컨트롤러를 사용한 반면 스트럿츠2프레임워크는 필터API기반의 컨트롤러를 사용한다.

(그림1)은 스트럿츠2 프레임워크의 구성도다. 스트럿츠2 프레임워크의 가장 두드러진 특징은 Pojo기반의 액션을 사용하는데 이는 기존에 JSP MVC2기반의 bean 객체를 추가 상속없이 바로 사용할 수 있다는 말이다. 이는 곧 기존의 래거시 시스템을 스트럿츠2 프레임워크로 적용하기 용이하다는 이야기다. 또한 Pojo기반의 액션, 폼을 사용함으로 해서 프레임워크에 종속되지 않고, Mock객체에 의존하지 않고 테스트가 손쉽게 가능하다. Zero Configration을 지향하는데 이는 기본값을 지원함으로 해서 많은 설정

을 생략할 수 있다. 이는 기존 스트럿츠 프레임워크에서 config.xml파일을 작성하는 어려움을 어느정도 해소하는 결과를 가져왔다.



(그림 1) 스트럿츠2 프레임워크 구조

2. Junit 프레임워크

Junit 프레임워크는 독립된 단위 테스트를 할 수 있도록 도와주는 자바언어로 된 프레임워크다. 기존에는 main() 함수에서 테스트대상이 되는 함수를 이용한 테스트용 코드를 무작위로 나열하여 사용하였다. 이 방식은 코드를 지저분하게 만들뿐만 아니라 유지보수비용을 증가시키는 결과를 가져왔다. 하지만 Junit 프레임워크를 이용하여 각각의 테스트목적에 따라 함수를 만들어 각각의 테스트를 분리하여 관리가 가능해졌고 향후 모듈의 수정이나 추가개발시에 목적모듈의 명세서 역할도 할 수 있게되었다.

Junit 3.x 이하 버전에서는 테스트를 위해 TestCase를 상속받는 클래스를 생성하고 또한 단위테스트를 수행할 함수명을 test로 시작해야하는 등의 제약사항들이 있었지만 4.x 버전에 오면서 어노테이션 기능을 이용하여 Pojo 기반의 클래스를 이용하도록 변경되었다.

또한 단위테스트를 수행하는 함수에는 간단하게 @Test 어노테이션만 추가하면 단위테스트를 수행할 수 있도록 변경되었다.

III. 테스트 자동화 설계

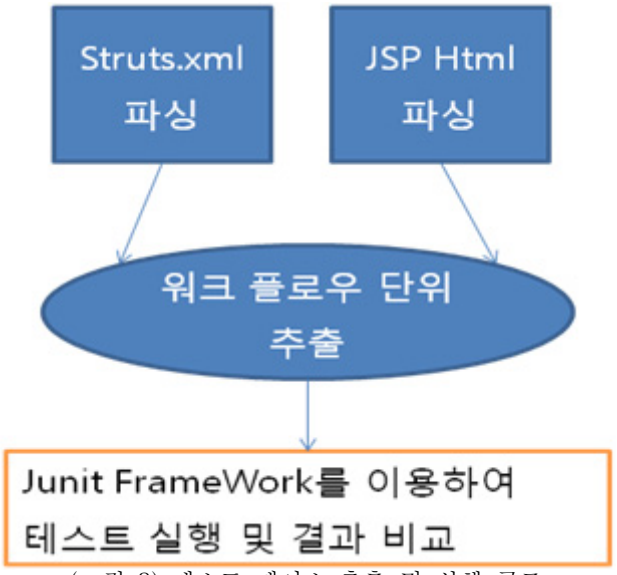
스트럿츠2 프레임워크는 struts.xml이라는 설정파일을 갖는다. (그림 2)는 스트럿츠2 에서 예제로 제공하는 메일리더 어플리케이션의 설정 파일이다.

설정파일에는 웹 어플리케이션의 워크 플로우 정보가 담겨있는데 액션(action) 테그에 해당 처리 모듈의 정보가 있고 해당모듈의 처리결과 값에 따라 행해질 결과 정보가 리절트(result) 테그에 담겨져 있다. 이 설정파일을 파싱하여 기본적인 업무 워크 플로우 단위를 추출하고 추출된 워크 플로우 단위로 Junit기반의 테스트 케이스를 생성한다.

```
<struts>
    <package name="mailreader-support" namespace="/" extends="mailreader-default">

        <action name="Tour">
            <result>/tour.html</result>
            <interceptor-ref name="guest"/>
        </action>

        <action name="Welcome" class="mailreader2.Welcome">
            <result>/Welcome.jsp</result>
            <interceptor-ref name="guest"/>
        </action>

        <action name="Logout" class="mailreader2.Logout">
            <result type="redirectAction">Welcome</result>
        </action>

        <action name="Login_*" method="{1}" class="mailreader2.Login">
            <result name="input">/Login.jsp</result>
            <result name="cancel" type="redirectAction">Welcome</result>
            <result type="redirectAction">MainMenu</result>
            <result name="expired" type="chain">ChangePassword</result>
            <exception-mapping
                    exception="org.apache.struts.apps.mailreader.dao.ExpiredPasswordException"
                    result="expired"/>
            <interceptor-ref name="guest"/>
        </action>

        <action name="Registration_*" method="{1}" class="mailreader2.Registration">
            <result name="input">/Registration.jsp</result>
            <result type="redirectAction">MainMenu</result>
            <interceptor-ref name="guest"/>
        </action>
    </package>

</struts>
```

(그림 2) 스트럿츠2 프레임워크 설정파일 struts.xml

테스트를 위한 입력값에 대한 정보는 [1]에서 제시한 방법을 이용하여 각 jsp파일의 html테그를 파싱하여 <form>테그와 <input>테그정보, 그리고 <a>테그의 href 속성값을 가지고 입력값 정보를 추출한다. 이렇게 Jsp 파일에서 추출된 정보와 Struts.xml 설정파일에서 추출된 정보를 조합하여 각 액션별 입력값을 알아내고 해당액션의 처리 결과를 비교한다. 본 논문에서 제안하는 테스트케이스 추출 및 실행구조는 (그림 3)과 같다.



(그림 3) 테스트 케이스 추출 및 실행 구조

1. Struts.xml 파싱

액션(action)테그의 정보를 사용하여 각 단위 모듈의 클레스 명과 호출함수명을 추출한다. 이 정보는 추후 Junit 프레임워크를 이용하여 단위테스트를 실행할 때 이용된다.

리절트(result)테그의 정보를 사용하여 해당 액션의 결과에 따라 호출될 페이지정보를 추출한다. 이 정보는 추후 Junit프레임워크에서 해당 모듈을 호출한 후 결과리턴 정보의 url정보와 비교하여 테스트의 정합성을 체크하는데 사용된다.

2. Jsp Html 파싱

Html 폼(form)테그의 인풋(input)테그의 값(value)정보가 요청바디에 입력되면 서버모듈은 이 요청바디의 값을 파싱하여 처리한다. 즉 폼테그의 인풋테그의 정보가 각 모듈의 입력값 정보가 된다.

이 입력값 정보를 가지고 해당 모듈의 입력파라메터 정보를 추출하여 테스트 데이터 생성시에 사용한다.

3. 워크플로우 단위 추출 및 테스트 데이터 생성

struts.xml 파일에서 추출한 각 단위 모듈의 정보와 Jsp html 테그정보를 파싱하여 추출한 각 모듈의 입력값 정보를 통해서 Junit 프레임워크의 단위테스트용 함수를 만든다. 입력값은 추출된 변수명에 직접 입력한다.

이와같이 생성된 Junit Test Case는 배치프로그램으로 등록하여 주기적으로 실행되도록 하여 운영시스템에 프로그램을 적용하기 전에 테스트프로그램에 먼저 적용하여 모든 테스트가 이상없이 실행됨을 확인한 후에 운영에 적용한다.

IV. 결론 및 향후 과제

웹 어플리케이션을 운영할때 가장 어려운 부분은 테스트다. 현제의 웹 어플리케이션은 다양한 컴포넌트들을 연결하여 거대한 시스템으로 구성하는데 한 컴포넌트를 수정할 때 다른 컴포넌트들과의 연관관계를 알고있지 못하면 운영중에 큰 문제를 일으킬 수 있다. 따라서 테스트는 시스템 운영에 있어서 가장 중요한 요소라고 할 수 있다. 이와같이 테스트를 자동화하는 시스템을 이용하면 좀더 쉽게 운영 및 수정작업을 테스트할 수 있다.

본 논문에서는 스트럿츠2 프레임워크를 이용한 웹어플리케이션의 테스트 자동화방법에 대해 연구하였다. 설정파일인 struts.xml파일과 Jsp파일을 이용하여 각 모듈정보와 입력값을 추출하여 Junit을 이용한 자동화 테스트 방법을 제안하였다.

그 결과 시스템의 테스트를 원활하게 할수있었고, 자동으로 테스트가 실행됨으로서 개발자의 실수로 테스트를 하지않는 문제등을 해결할수 있었다.

하지만 여전히 데이터베이스를 이용하는 DAO단의 테스트에는 취약한 부분이 있다. 이를 위해 향후 데이터베이스 테스트 프레임워크인 DBUnit도 함께 이용하는 방법을 연구하여 제시하고자 한다.

V. 사서의 글

본 연구는 2013년도 지식경제부의 SW전문인력양성사업의 재원으로 정보통신산업진흥원의 고용계약형 SW석사과정지원사업(HB301-13-1003)으로부터 지원받아 수행되었습니다.

참고문헌

[1] 임정희, 이시현, 장진아, 최병주, 황상철 “사용자 화면 중심의 블랙 박스 테스트와 웹인터페이스 테스트 커버리지를 통한 웹 어플리케이션 테스트 방법” 정보과학회논문지: 소프트웨어 및 응용 제 36권 제 9호, (2009.09)

[2] Struts, Struts2 http://struts.apache.org

[3] Junit http://www.junit.org/