

## 활동-지향 접근법을 사용한 웹 응용 테스트 기법의 설계

김진수<sup>o</sup> 김남중 김재환 심지은 채진석

인천대학교 컴퓨터공학과

{devillove<sup>o</sup>, water09z, jhkim, simjiuen, jschae}@incheon.ac.kr

### A Design of Web Application Test Method

### Using Activity-Oriented Approach

Jinsu Kim<sup>o</sup>, Namjoong Kim, Jaehwan Kim, Jiuen Sim, Jinseok Chae

Dept. of Computer Science and Engineering, University of Incheon

#### 요 약

웹의 발전과 함께 증가해온 웹 응용은 점점 복잡하고 난해하여 이를 쉽게 이해하기 위한 방법이 필요하다. 또한 양적 성장과 함께 사용자의 질적 만족도를 충족시키기 위한 요구도 증가하고 있다. 따라서 웹 응용에 대한 이해를 돕는 모델과 웹 응용 테스트는 점차 중요하게 되어가고 있지만 웹 응용을 모델링하는 작업과 테스트 코드를 생성하고 유지하는데 필요한 비용과 노력이 만만치 않아 이를 줄이기 위한 노력이 절실하다.

본 논문에서는 웹 응용의 상태를 표현하기 위한 도메인 모델과 웹 응용의 정확성을 검증하는 테스트 활동 그래프를 이용한 활동-지향 접근법으로 웹 응용의 이해를 돕도록 하며 이를 기반으로 하는 웹 응용 테스트 방법을 제안한다. 이것은 웹 응용을 테스트하기 위한 소스 코드를 생성, 저장, 호출하여 사용함으로써 개발자 및 테스터에게 업무의 효율성과 편의성을 제공할 것이라 기대되어지며 웹 응용의 품질을 향상시키는 데 기여할 것으로 예상된다.

#### 1. 서 론

네트래프트는 2006년 11월 1일 전 세계 웹사이트 도메인의 수가 1억을 돌파했다고 전했다. 1995년 전 세계에 1만8000여개에 불과했던 웹 사이트가 2004년도에 5천만 사이트가 개설되어 운영되었고 그 후로 30여 개월이 지난 2006년 11월에 1억 사이트로 폭발적인 발전을 계속해가고 있다[1]. 그림 1에서 볼 수 있듯이 인터넷을 기반으로 웹이 급속도로 발전하고 대중화되면서 웹 응용의 역할 및 중요성은 점점 증대되어지고 있으며, 양적 성장뿐만 아니라 질적인 성장도 함께 이루어지고 있다.

사용자의 요구사항을 만족시키고 웹 응용의 품질을 보증하기 위해서 테스트를 수행하는 것은 불가피하다. 하지만 웹 응용은 점점 복잡해질 뿐만 아니라 이해하기 어렵고, 웹 페이지가 정적인지 동적인지에 따라서 그 테스트 방법이나 절차가 달라진다. 이러한 이유로 웹 응용을 통합 테스트하는 것 또한 어렵다. 뿐만 아니라 개발자나 테스터의 입장에서 웹 응용에 대한 테스트 코드의 생성과 유지는 매우 성가신 일이며, 이를 위한 단순하고 반복적인 작업은 업무의 효율성과 동기를 저하시키는 요인이 된다.

따라서 웹 응용을 보다 쉽고 편리하게 테스트할 수 있도록 하는 방법을 연구하고 개발하는 것은 매우 의미있

는 일이며, 이를 자동화하는 것은 개발자 및 테스터에게도 편의성과 효율성을 제공해준다.

본 논문에서는 활동-지향 접근법(activity-oriented approach)을 기반으로 웹 응용을 모델링하여 쉽게 이해할 수 있도록 도움을 주며 이를 기반으로 테스트 코드를 생성하여 보다 쉽게 테스트할 수 있는 방법을 제안한다.

논문의 구성은 다음과 같다. 2장에서는 테스트에 대한 관련 연구에 대해 기술하고, 3장에서는 제안하고자 하는 활동-지향 접근법과 테스트 코드를 생성하기 위한 아키텍처를 설명한다. 4장에서는 이에 따른 기대효과를 기술하고, 5장에서는 결론 및 향후과제를 제시한다.

Total Sites Across All Domains August 1995 - August 2007

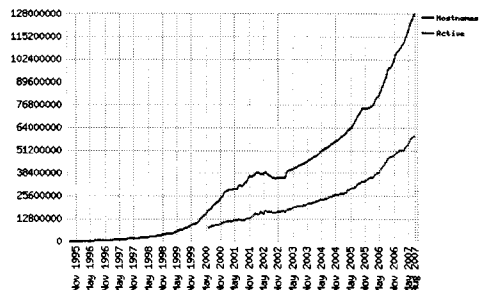


그림 1. 1995. 8 ~ 2007. 8까지 도메인 누적통계

## 2. 배경 연구

### 2.1 블랙박스 테스트

블랙박스 테스트는 요구사항에서 정의한 기능을 만족시키는지 확인하는 테스트로 기능 테스트라고 할 수 있다. 즉, 주어진 입력에 대한 결과를 검사하는 방법으로 모듈의 내부에 대해서는 시험하지 않으며 웹 응용의 기능 시험에 주로 활용이 되는 테스트 기법으로 웹 응용안의 논리적인 흐름을 관찰함으로써 구조적인 테스트를 하는 화이트박스 테스트와는 반대라 할 수 있다.

화이트 박스 테스트는 프로그램 안에서 확정되어진 실행 흐름을 이용하여 수행되어지므로 보다 확실하게 자동화할 수 있으며 모니터링도 쉽게 가능하지만 기능의 잘못된 구현이나 실행되지 않는 코드(dead code)를 발견하기 어렵고 인터페이스나 다중 프로세스, 스레드, 인터럽트 등의 동적인 부분에 대한 시험이 어렵고 원시코드의 규모가 커지며 실행 경로가 무수히 많은 경우가 많다는 점이 한계이다. 따라서 단위 테스트의 규모를 벗어난 시스템 수준의 테스트에서는 블랙박스 테스트 기법을 주로 사용한다[2].

블랙박스 테스트는 테스트케이스를 작성하여 수행하는데, IEEE에서는 이 테스트케이스를 "특정 프로그램의 경로나 소프트웨어의 요구사항을 확인하는 등의 일정한 목적을 위하여 개발된 테스트 입력, 실행 조건, 예상 결과의 집합체"라고 정의하고 있다[3]. 따라서 블랙박스 테스트를 위해서 테스트의 입력사항과 실행 조건을 정하고 이에 따른 예상 결과를 정의하고 수행된 테스트 결과와 비교하여 프로그램의 요구사항의 만족 정도와 품질 등을 평가하게 된다.

### 2.2 SMS (Solicitation Management System)

미국의 샌 버나디노 대학에서 개발한 SMS는 입찰제안서 관리 및 승인할 수 있도록 한 웹 기반 시스템이다.

이 시스템은 크게 5개의 다른 사용자 역할을 가지고 있다: admin, officer, staff, applicant, evaluator. admin의 역할은 officer와 staff 계정의 생성과 관리이며 officer의 역할은 evaluator가 제안서를 결정하기 위해 입찰제안서를 생성하고 수정하는 권한 등을 가지고 있다. staff은 시스템 데이터에 읽기 전용으로 접근할 수 있으며, applicant는 하나이상의 입찰 제안에 제안서를 제출하는 사용자이다. evaluator는 제안서를 읽고 평가결과를 전송하는 사용자이다.

SMS는 officer가 입찰제안서를 생성하면 applicant가 회원 가입 후 제안서를 제출한다. 그러는 동안 officer가 제안서를 evaluator에게 할당하며 evaluator는 할당받은 제안서를 읽고 평가서를 제출한다. staff은 선택된 제안

서를 사용할 수 있도록 보고서를 작성하는 순서로 SMS는 동작한다.

## 3. 제안 방법

### 3.1 활동-지향 접근법

웹 응용을 테스트하기 위해서 본 논문에서 사용하는 HttpUnit은 오픈 소스로 전형적으로 JUnit 테스트 프레임워크와 함께 사용되는 request/response 기반의 테스트 프레임워크이다. HttpUnit을 사용하는 웹 사이트의 기능을 테스트하고 검증하는 것은 매우 빠르고 쉽게 수행되지만 테스트 코드를 유지하는 것은 테스트 프레임워크보다 테스트 접근방법이 더 중요하다.

본 논문에서 제안하는 활동-지향 접근법은 사용자가 보는 웹 응용의 객체이며 웹 응용의 상태를 표현하기 위한 도메인 모델을 정의하고, 웹 응용의 정확성을 검증하는데 테스트 활동 그래프(test activity graph)를 사용한다. 활동-지향 접근법에서 말하는 활동은 웹 응용과 사용자간의 동작을 정의한다. 이러한 활동의 예로는 위의 SMS에서 admin이 로그인하는 과정을 생각할 때, admin의 홈 페이지가 반환되는지 로그인하고 검증하는 admin 사용자를 들 수 있다. 활동은 독립적인 활동과 의존적인 활동으로 구분할 수 있다. 독립적인 활동은 그것을 수행하는데 웹 응용의 상태에 의존하지 않는 활동을 말하며, 의존적인 활동은 주어진 활동을 실행하기 위해서 요구되는 웹 응용의 상태 즉, test fixture가 필요한 것을 의미한다.

활동-지향 접근법은 웹 응용을 테스트하기 위해서 다음의 4단계 과정을 따른다.

첫 번째 단계는 활동을 인식하는 하는 단계이다. 시스템과의 가능한 모든 상황에 대한 동작을 테스트하는 것은 불가능하기 때문에 개발자는 결함이 없다는 것을 보증하기 위해 어떤 활동이 시스템 기능과 품질을 보여줄 수 있는지 분석하고 결정해야 한다.

두 번째 단계는 테스트 활동 그래프를 개발하는 단계이다. 개발자는 활동들과 그들의 의존성을 나타내기 위해서 테스트 활동 그래프를 만든다. 테스트 활동 그래프는 원래 순차적인 제약보다는 의존성을 보이기 위해 화살표를 반대로한 작업 흐름 그래프이다. 그림 2는 앞서 설명한 SMS를 테스트 활동 그래프로 표현한 것이다.

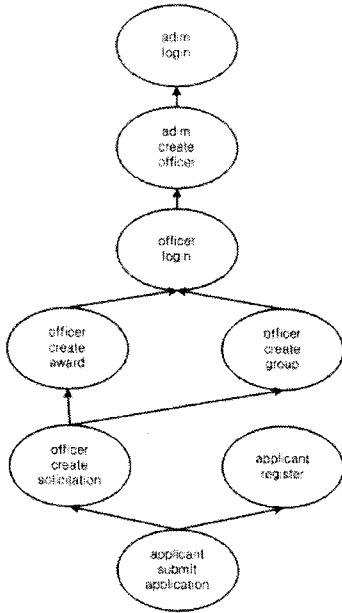


그림 2. 테스트 활동 그래프

세 번째 단계는 활동-지향 테스트 모델을 개발하는 것이다. 테스트 모델은 테스트 활동 그래프로부터 생성되어진다. 이 모델은 DomainObject 클래스와 Activity 클래스를 사용하여 만들어진다.

DomainObject 클래스는 officer, applicant 등의 사용자가 이해하는 웹 응용안의 객체들을 표현하는 서브 클래스들을 정의하기위해 사용된다.

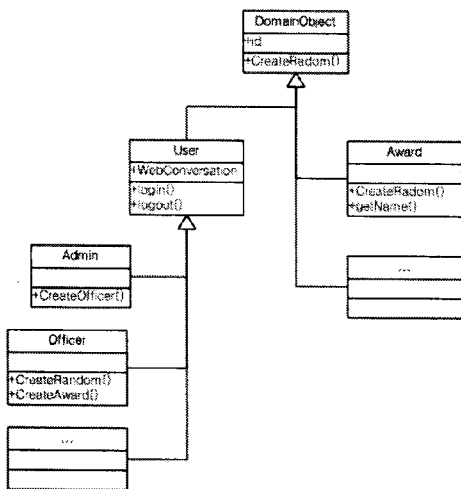


그림 3. Domain Object 클래스

Activity 클래스는 도메인 객체에서 사용되는 활동들을 나타내는 서브 클래스들을 정의하는데 사용된다.

Activity 클래스는 테스트 활동 그래프에서 나타나는 의존성을 포함하고 있으며 테스트 활동 그래프에서 보이는 각 활동에서 Activity 클래스의 서브클래스를 정의한다.

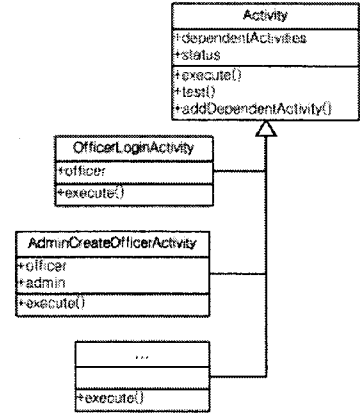


그림 4. Activity 클래스

위의 그림 3과 그림 4는 SMS를 DomainObject 클래스와 Activity 클래스로 각각 모델링한 것이다.

테스트 모델을 설계한 후 마지막 네 번째 단계로 도메인 객체와 활동들의 모든 필요한 인스턴스들을 나타내면서 테스트 활동 그래프에 포함된 의존성을 나타내는 코드를 작성하는 것이다.

### 3.2 활동-지향 접근법기반의 자동화된 테스트

위의 네 번째 단계에서 테스트 코드를 작성할 때 테스트하고자 하는 것에 따라서 일정한 패턴을 가지고 있다. 예를 들면, 웹 페이지에서 사용자가 입력 폼을 채우고 페이지를 전송하게 되는 경우의 테스트를 위해서는 입력 페이지에서 'input' 태그를 찾아서 해당 태그의 이름에 임의의 값을 생성하여 입력하는 방법으로 페이지를 채운 후 전송하여 그 결과로 성공여부를 결정한다. 따라서 이러한 패턴을 자동으로 생성하여 개발자나 테스터 등의 사용자에게 제공해야 할 것이다.

이러한 코드를 사용자에게 제공하기 위해서 페이지의 입력 폼을 채워서 페이지를 전송하는 것과 같은 대표적인 기능의 코드는 사용자에게 제공하고 또 사용자가 원하는 패턴의 코드를 저장해두었다가 사용자의 호출에 의해 제공될 수 있도록 한다.

아래의 그림 5에서 보이는 것과 같이 사용자에게 테스트 코드를 생성하여 제공하기 위한 모듈 CodeGen 클래스는 test fixture를 생성하기 위한 FixtureCodeGen 클래스와 HttpUnit 라이브러리를 사용하여 활동을 테스트하기 위한 코드를 생성하는 TestCodeGen 클래스로 일반화되어진다.

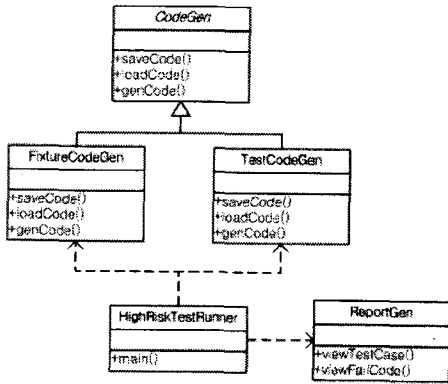


그림 5. 코드 생성을 위한 클래스 구조

웹 응용을 테스트는 다음의 그림 6과 같이 이루어진다. 간단하게 만들어진 테스트 페이지에 값을 입력하고 페이지를 전송하여 이상없이 페이지가 전송된다면 테스트는 성공한다. testLogin 메서드 부분을 보면 알 수 있듯이 페이지를 테스트하기 위한 코드는 일정한 패턴으로 이루어질 것이라는 것을 예측할 수 있다. 따라서 그림에는 나오지 않았지만 그림 5의 구조를 기반으로 테스트 코드를 자동으로 생성하도록 하여 개발자나 테스터에게 제공하도록 한다.

```

* 프로젝트명 : ProtoTest2
package prototest;

import static prototest.user.User.ASSERT;
import prototest.user.User;

import com.netware.httpunit.SubmitButton;
import com.netware.httpunit.WebConversation;
import com.netware.httpunit.WebForm;
import com.netware.httpunit.WebResponse;

public class HighRiskTestRunner implements ApplicationOris {

    public static void main(String[] args) throws Exception {
        System.out.println("Main Start");
        testLogin();
        System.out.println("Main End");
    }

    private static void testLogin() throws Exception {
        System.out.println("Login Test Start");
        User user = new User();
        WebConversation wc = context.getWebConversation();
        WebResponse response = wc.getResponse(ajaxLogin);
        ASSERT(response.getResponseCode() == 200); // 200 OK sign.
        WebForm form = response.getForm(0);
        ASSERT(form != null);
        ASSERT(form.hasParameterNamed("txtId"));
        ASSERT(form.hasParameterNamed("txtPwd"));
        form.setParameter("txtId", "test");
        form.setParameter("txtPwd", "test");

        SubmitButton submitButton = form.getSubmitButton("", "submit");
        ASSERT(submitButton != null);
        response = form.submit(submitButton);
        ASSERT(response.getResponseCode() == 200);
        String currentUrl = response.getCurrentUrl().toString();
        ASSERT(currentUrl.equals("http://localhost:8080/"));

        System.out.println("Login Test Done.");
    }
}

```

그림 6. 대상 웹 페이지를 테스트하기 위한 코드

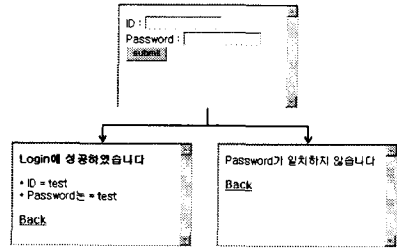


그림 7. 테스트 대상 웹 페이지

#### 4. 기대 효과

웹 응용을 위한 테스트 기법을 연구하고 이러한 기법을 기반으로 일련의 테스트 과정을 자동화하는 시스템을 설계 및 구현하는 것은 업무의 효율성을 증대시키는 것에서 웹 응용의 신뢰성을 증가시키는 부분까지 현재의 단순하고 반복적이며 비효율적인 업무를 제거하여 웹 응용의 발전에 기여하는 역할을 할 수 있을 것이다.

또한 본 논문에서 제안하는 활동-지향 접근법을 통해서 웹 응용을 모델화하여 개발자 및 테스터에게 개발하고자 하는 웹 응용을 이해하고 테스트 케이스를 개발하는데 도움을 준다. 또한 이러한 접근방법을 기반으로 테스트 코드의 생성을 자동화하여 테스트 과정에서의 단순하고 반복적인 과정을 줄여 효율성을 증대하고 편의성을 제공할 것으로 기대된다.

#### 5. 결론 및 향후과제

웹 브라우저의 성공적인 보급은 웹 응용의 대중화를 극적으로 증가시켰으며 이와 같은 웹 응용의 양적 성장과 함께 질적 성장의 중요성이 필요하게 되었다. 따라서 웹 응용의 품질과 신뢰성을 보증하기 위한 테스트 방법론과 도구에 대한 요구가 높아지고 있다.

하지만 웹 응용에 대한 테스트는 그 중요도에 비해 연구가 부족한 실정이다. 특히 웹 응용이 점점 복잡해지면서 이를 테스트하는 방법도 어려워지고 있는 실정이다. 또한 웹 응용을 테스트하는 방법은 대체로 반복적이며 비효율적인 작업이 빈번하여 개발자 및 테스터에게는 매우 성가시면서도 비효율적이다. 따라서 이러한 웹 응용에 대한 자동화된 테스트 방법론과 도구를 연구할 필요가 있다.

본 논문에서는 웹 응용의 기능적 신뢰성과 품질을 보증하기 위한 방법으로 블랙박스 테스트를 기반으로 한 활동-지향 접근법을 사용하여 웹 응용을 모델링하여 테스트를 자동화할 수 있는 방법을 제안하였다.

활동-지향 접근법은 활동을 기반으로 하는 자동화된 테스트 방법으로 테스트 활동 그래프와 도메인 모델을

통해 모델을 구축한다. 이러한 모델을 기반으로 테스트를 위한 코드 패턴을 분석 및 저장하여 사용자에게 테스트를 위한 코드를 제공하도록 한다. 이로 인하여 웹 응용을 보다 쉽게 이해하고 테스트를 수행하는데 있어 보다 효율적이면서 신뢰성 있는 테스트가 이루어질 것으로 기대한다.

향후 연구 및 보완과제로는 활동-지향 접근법을 기반으로 한 자동화된 웹 응용 테스트 시스템을 구축하여 테스트를 수행하여 테스트 비용, 성능, 품질을 확인하여 연구의 신뢰성을 높이는 것이 필요하겠다. 뿐만 아니라 활동-지향 접근법을 위해서는 개발자나 테스터 등의 사용자가 웹 응용을 분석하는 것이 불가피한데 이러한 부분을 어느 정도 자동화시켜 제공할 필요가 있겠으며 보다 다양한 테스트 환경에서 테스트 코드를 생성하기 위한 패턴을 수집할 필요도 있겠다. 또한 현재 많은 연구와 발전이 이루어지고 있는 시맨틱 웹 혹은 웹 2.0에 대한 테스트를 수행할 수 있도록 연구가 필요하다.

## 6. 참고문헌

- [1] <http://www.netcraft.com/>.
- [2] 서광익, 최은만, “블랙박스 테스트 케이스의 리엔지니어링”, 정보처리학회 논문지, 제13-D권, 제4호 pp. 573-582, 2006. 8.
- [3] IEEE Standards, “IEEE Standard Glossary of Software Engineering Terminology,” IEEE Std 61.12-1990, 1990.
- [4] Joseph Schmuller, *Teach Yourself UML in 24 Hours*, 3rd Edition, SAMS Publishing, 2001.
- [5] Manish Nilawar, “A UML-Based Approach for Testing Web Applications,” Master’s Thesis, University of Nevada, 2003.
- [6] 서광익, 최은만, “AJAX를 위한 자바스크립트 시험 도구 제안,” 한국정보과학회 가을 학술발표논문집, 제33권, 제2호(C), pp. 458-461, 2006.
- [7] David Kung, “An Agent-Based Framework for Testing Web Applications,” Proc. COMPSAC, pp. 174-177, 2004.
- [8] Giuseppe Antonio Di Lucca et al., “Testing Web Applications,” Proc. ICSM, pp. 310-319, 2002.
- [9] Luciano Baresi et al., “Towards Model-Driven Testing of a Web Application Generator,” Proc. ICWE, pp. 75-86, 2005.