

테스트 활동 그래프를 사용한 효율적인 웹 응용 분석 및 테스트 기법

김진수*, 김남중*, 심지은*, 채진석*

*인천대학교 컴퓨터공학과

E-mail: {devillove, water09z, simjiuen, jschae}@incheon.ac.kr

An Efficient Analysis and Testing Scheme for Web Applications Using Test Activity Graph

Jinsu Kim*, Namjoong Kim*, Jiuen Sim*, Jinseok Chae*

*Dept. of Computer Science and Engineering, University of Incheon

요 약

복잡하고 난해한 웹 응용을 이해하고 테스트하는 것은 사용자가 요구하는 품질을 만족시키기 위한 필수적인 과정이지만 실제로 이 과정을 제대로 수행하는 것은 쉬운 일이 아니다. 지금까지 웹 응용의 테스트를 위해 여러 가지 프레임워크가 제시되어 사용되고 있지만 이러한 프레임워크의 효율성은 어떤 접근법을 사용하느냐에 의존적인 경우가 많다.

본 논문에서는 웹 응용 상태를 표현하는 도메인 모델과 테스트 활동 그래프를 통해 객체의 활동과 의존성을 표현하여 웹 응용의 정확성을 검증하는 테스트 기법을 제안하고자 한다. 이 기법은 웹 응용을 사용자가 이해하기 쉬운 활동으로 모델링하여 테스트할 수 있도록 도와줄 수 있으므로 업무의 효율성과 편의성을 증대시킬 수 있을 것으로 기대된다.

1. 서론

웹 응용은 인터넷이 발달하고 대중화가 성공적으로 이루어짐에 따라서 많은 사용자와 밀접한 관계를 맺으며 인터넷 기반에서 웹 브라우저를 통해 동작하는 소프트웨어 프로그램이다[1]. 따라서 이러한 웹 기반 시스템은 사용자의 요구사항을 만족시키기 위해 잦은 변경이 불가피하다[2][3]. 이러한 잦은 변경에 따라 기능의 성공 여부를 판단할 수 있는 테스트도 자주 수행되지만 이를 위해 웹 응용을 분석하는 것과 테스트 코드를 만들고 유지하는 것은 쉽지 않다. 따라서 점점 복잡하고 난해한 웹 응용을 이해하기 쉽게 분석하고 테스트하여 품질을 보장하기 위해 이를 위한 접근법과 테스트 도구의 필요성 역시 증대되어지고 있다[4].

이를 위해 JUnit, NUnit, CxxTest, JsUnit 등의 프레임워크를 기반으로 테스트가 많이 사용되지만 테스트의 효율성은 프레임워크보다는 접근법에 더 의존적이다[5][6]. 본 논문에서는 웹 응용과 사용자간의 동작을 활동이라 정의하고 이 활동을 주체로 하여 웹 응용의 분석 및 테스트를 수행하는 기법을 제안한다. 이러한 활동 지향 접근법은 웹 응용의 이해를 돕고 테스트를 용이하게 하기 위해 요청/응답 기반의 HttpUnit을 사용하며 도메인 모델과 테스트 활동 그래프로 구성된다.

논문의 구성은 다음과 같다. 2장에서는 관련연구에 대해 기술하고, 3장에서는 본 논문에서 제안하는 활동-지향 접근법에 대하여 설명한다. 4장에서는 이 접근법을 기반으로

한 테스트의 구현 결과를 제시하며 5장에서 결론 및 향후 과제와 함께 끝맺도록 한다.

2. 관련 연구

웹 응용의 분석을 통해 이해를 돕고 테스트를 효율적이고 용이하게 하기위해 많은 연구가 진행 중이다.

먼저, Ricca, Tonella는 정적 웹 페이지 분석을 위해 UML을 이용하여 웹 응용 분석 모델과 관련 테스트 전략을 제시하였다[2]. 하지만 점점 웹 기술의 발전하면서 동적인 콘텐츠가 증가하게 되어 동적 웹 페이지 분석을 위한 연구가 더 필요하다.

또 Lee와 Offutt은 웹 기반 소프트웨어 시스템 컴포넌트사이의 데이터 상호작용의 신뢰성을 검증하는데 초점을 맞추어, 변경 분석 폼을 사용하여 XML 기반의 웹 응용 테스트 케이스를 생성하는 방법을 연구하였다[7].

Elbaum, Karre, Rothermel은 웹 응용의 테스트 케이스를 생성하기 위해 HTML 폼으로부터 얻고 이름-값의 쌍을 포함하는 “사용자 세션 데이터”를 사용하는 방법을 제안하였다[8].

Nilawar는 UML을 기반으로 웹 응용을 설명하고 Use-case 기반 테스트와 Web compounds 단위 테스트로 구성된 “combo testing”을 사용하여 웹 응용을 테스트하는 접근법을 제시하고 있다[1].

Kung은 BDI 모델과 UML을 기반으로 하는 에이전트 기반 프레임워크를 발표하는 한편[9], 그래프로 웹 사이트

를 설명하기 위한 설계방안인 그래프 기반 테스트 방법을 연구하였다[12][13].

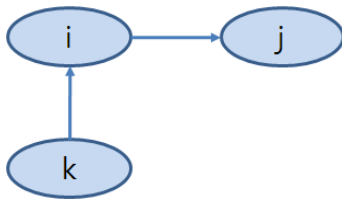
3. 활동-지향 접근법

활동은 웹 응용과 사용자간의 동작을 의미하는 것으로 독립적인 것과 의존적인 것으로 구분한다. 여기서 독립적인 활동이라는 것은 시스템의 처음 상태를 의미하여 다른 상태에 의존하지 않는 활동을 말하는 반면 의존적인 활동은 실행을 위해서 시스템이 어떠한 상태(이를 test fixture 라고 한다)를 이루고 있어야만 하는 활동을 말한다.

이러한 활동을 기반으로 하는 접근법을 우리는 활동-지향 접근법이라고 하며, 이 접근법은 웹 응용을 테스트하기 위해서 다음의 4단계 과정을 따르게 된다.

첫 번째 단계는 활동의 식별단계이다. 테스트는 시스템의 모든 상황에 대해서 수행할 수 없기 때문에 결함이 없다는 것을 보증할 수 있을만한 활동을 결정해야 한다. 이렇게 결정된 활동을 test fixture가 필요한지 아닌지로 구분하게되면 활동은 독립적인 것과 비독립적인 것으로 구분된다.

두 번째 단계에서는 활동과 의존성을 나타내기 위해 업무 흐름 그래프의 화살표를 반대로 한 테스트 활동 그래프를 만든다. 그림 1에서 볼 수 있듯이 그래프에서 i가 j에 의존하면 화살표는 활동 i에서 j로 향하고, k가 i에 의존하면 화살표는 활동 k로부터 i로 향하게 된다.



(그림 1) 테스트 활동 그래프

테스트 활동 그래프는 중복을 제거함으로써 간략화되고 이로 인해 더 효율적으로 된다. 또 이 그래프에서 활동은 대상 시스템의 기능을 테스트하게 되므로 테스트 케이스를 나타낸다.

세 번째 단계는 활동 지향 테스트 모델을 개발하는 과정이다. 이 테스트 모델은 두 번째 단계에서 나온 테스트 활동 그래프를 기반으로 만들어지게 되는데 크게 DomainObject와 Activity 두 개의 클래스로 만들어진다.

DomainObject 클래스는 웹 응용에서 사용자가 이해할 수 있는 객체들을 표현하는 서브 클래스를 정의하는데 사용되며, 이 클래스는 도메인 객체들을 식별하기 위한 id 속성과 임의의 값을 생성하는 createRandom 메서드를 가지고 있다. Activity 클래스는 도메인 객체를 사용하는 활동을 나타내는 서브 클래스를 정의하는데 사용된다. 이것의 각 서브 클래스는 활동을 수행하는 추상 메서드인 execute를 구현해야 한다.

```

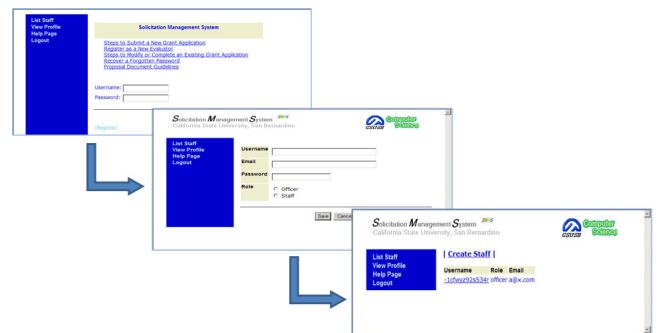
test()
{
    if status != UNTESTED
        return
    for each dependent activity a
        a.test()
    if a.status != PASSED
        status := UNREACHABLE
    if status = UNREACHABLE
        return
    if execute() = TRUE
        status := PASSED
    else
        status := FAILED
}
    
```

(그림 2) 활동 테스트 알고리즘

마지막 단계로 활동 테스트 알고리즘 코드를 작성하는 단계이다. 그림 2에서 활동 테스트 알고리즘을 보이고 있는데, 이 테스트 함수는 모든 의존적인 활동에 대해 재귀적으로 실행되기 때문에 다른 활동을 의존하지 않은 활동에서 test 메서드를 실행하는 것만으로 충분하다. 그리하면 test 기능은 시스템의 모든 활동에서 적어도 한번은 실행될 것이다.

4. 활동 지향 접근법 기반의 테스트 구현

구현은 본 논문에서 제안한 도메인 모델과 테스트 활동 그래프를 사용하는 활동-지향 접근법을 기반으로 웹 응용을 분석하고 이를 이용하여 웹 응용을 테스트하도록 한다. 이 테스트를 위해서 선택한 웹 기반 응용 시스템은 제안서 관리 시스템이다. 이 제안서 관리 시스템의 간단한 운용 방식은 Admin이 Officer 계정을 만들면 Officer가 제안서를 모집하고 Applicant가 제안서를 제출한다. 제출된 제안서는 Evaluator에 의해서 평가되고 Staff이 평가 자료를 기반으로 보고서를 작성하도록 하는 것이다.



(그림 3) Officer 생성 및 정보 수정 테스트 흐름

그림 3과 같은 웹 응용의 흐름에서 그림 4의 구현 코드는 Admin을 test fixture로 생성하고 이 Admin이 Officer 계정을 생성한 후, 생성된 Officer의 정보 중 email 정보를 변경함으로써 수정이 제대로 이루어지는지 확인하는 소스 코드이다.

```

// Use domain model to create test fixture.
Admin admin = new Admin();
Officer officer = Officer.createRandom();
admin.login();
admin.createOfficer(officer);

// Use low-level HttpUnit library to test the activity
// (admin modifies officer email).
String url = resourcePrefix + adminEditStaffResource + "?id=" + officer.getId();
WebRequest request = new GetMethodWebRequest(url);
WebConversation wc = admin.getWebConversation();
WebResponse response = wc.getResponse(request);
WebForm form = response.getForms()[0];
SubmitButton submitButton = form.getSubmitButton("submit-button");
assertNotNull(submitButton);
String newEmail = "a@x.com";
officer.setEmail(newEmail);
form.setParameter("email", newEmail);
response = form.submit(submitButton);

// Use the domain model to verify results.
admin.viewOfficer(officer);

```

(그림 4) 정보 변경 기능을 테스트하는 코드

또 그림 5에서는 일반적인 웹 페이지에서 사용자의 로그인 기능을 수행하는 기능을 테스트하는 코드이다. 이는 생성된 웹 페이지 정보를 기반으로 폼 정보를 채우는 방식으로 쉽게 웹 응용을 테스트 할 수 있다.

```

User tester = new User();
WebConversation wc = tester.getWebConversation();
WebResponse response = wc.getResponse(userLogin);
ASSERT(response.getResponseCode() == 200); // 200은 ok sign
WebForm form = response.getForms()[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.getURL().toString();
ASSERT(currentUrl.equals(userLoginOk));

```

(그림 5) 로그인 기능을 테스트하는 코드

5. 결론 및 향후 연구 과제

오픈 소스 기반의 HttpUnit을 이용한 요청/응답 기반의 테스트 프레임워크는 웹 응용을 빠르게 테스트하지만 테스트의 효율성과 테스트 코드의 유지는 프레임워크보다는 접근법에 더 의존적이다. 따라서 본 논문에서는 테스트 접근법중의 하나로 사용자와 웹 응용과의 상호작용을 기반으로 블랙박스 테스트를 수행하는 활동-지향 접근법을 제안하였다.

이러한 활동-지향 접근법은 DomainObject와 Activity 클래스로 구성되어 웹 응용의 상태를 표현하기 위한 도메인 모델과 활동 및 의존성을 나타내어 웹 응용의 정확성을 검증하는 테스트 활동 그래프로 이루어진다. 이렇게 사용자가 이해할 수 있는 활동을 기초로 하는 접근법이

때문에 복잡하고 난해한 웹 응용 시스템을 보다 쉽게 이해하고 테스트할 수 있도록 도움을 줄 것으로 기대된다.

향후 연구 과제로는 본 논문에서 제시된 활동-지향 접근법을 기반으로 모델링하는 부분과 테스트 코드를 생성하는 부분에 있어서 수동적이고 반복적이므로 기존의 접근법이나 테스트 방법과 마찬가지로 불편하고 효율성이 떨어진다. 따라서 이러한 문제를 해결하기 위해서 웹 응용을 분석하여 활동-지향 접근법을 기반으로 모델링하고 테스트 코드를 자동으로 생성하여 편의성과 효율성을 더욱 증대시켜야 하겠다.

참고문헌

- [1] Manish Nilawar, "A UML-Based Approach for Testing Web Applications," Master's Thesis, University of Nevada, 2003.
- [2] Filippo Ricca, Paolo Tonella, "Analysis and Testing of Web Applications," 23rd International Conference on Software Engineering (ICSE'01), pp. 25-34, 2001.
- [3] Giuseppe Antonio Di Lucca et al., "Testing Web Applications," Proc. ICSM, pp. 310-319, 2002.
- [4] 김현수, 최은만, "다양한 웹 기반 소프트웨어의 테스트를 위한 효율적인 테스트 케이스의 생성," 한국정보처리학회논문지D, 12D, pp. 569-582, 2005.
- [5] David Thomas, Andrew Hunt, *Pragmatic Unit Testing in Java with JUnit*, O'REILLY, 2004.
- [6] 서광익, 최은만, "AJAX를 위한 자바스크립트 시험 도구 제안," 한국정보과학회 가을 학술발표논문집, 제33권, 제2호(C), pp. 458-461, 2006.
- [7] S. C. Lee, Jeff Offutt, *Generating test cases for XML-based Web Component interactions using mutation analysis*, Proc. of the 12th International Symposium on software Reliability Engineering, pp 200-209, IEEE Computer Society Press, 2001.
- [8] S. Elbaum, S. Karre, G. Rothermel, *Improving Web application testing with user session data*, Proc. of the 25th International Symposium on software Reliability Engineering, pp. 49-59, IEEE Computer Society Press, 2003.
- [9] David Kung, "An Agent-Based Framework for Testing Web Applications," Proc. COMPSAC, pp. 174-177, 2004.
- [10] IEEE, *SWEBOK(Software Engineering Body of Knowledge) 2004 Version*, IEEE Computer Society Press, 2004.
- [11] Ye Wu, Jeff Offutt, Xiaochen Du, "Modeling and Testing of Dynamic Aspects of Web Applications," Submitted for publication, 2004. Technical Report ISE-TR-04-01, 2004.
- [12] D. Kung, Chien-Hung Liu, Pei Hsia, "An

object-oriented web test model for testing Web applications," Proc. of IEEE 24th annual International Computer Software and Applications Conference COMPSAC2000), pp. 537-542, 2000.

- [13] C. H. Liu, D. Kung, P. Hsia, C. T. Hsu, "Structural testing of Web applications," Proc. of the 11th International Symposium on Software Reliability Engineering, pp. 84-96, IEEE Computer Society Press, 2000.
- [14] Cem Kaner, James Bach, Bret Pettichord, *Lessons Learned in Software Testing : a Context-Driven Approach*, WILEY, 2002.