

웹 응용 신뢰성 확보를 위한 UML 순차도 기반의 시험사례 추출 및 시험

UML Sequence Diagram Based Test Case Extraction and Testing for Ensuring Reliability of Web Applications

정기원(Kiwon Chong)*, 조용선(Yongsun Cho)**

초 록

시간적 금전적 압박으로 최근의 웹 응용에서는 체계적인 시험작업이 종종 간과되곤 한다. 게다가 효과적인 웹 응용 시험 방법이 많지 않아 웹 응용에 기존의 시험 방법들을 적용하면서 개발자들은 많은 어려움을 겪고 있다. 빠르고 효과적인 시험을 위하여 웹 페이지간의 메시지를 모델링한 UML 순차도(Sequence Diagram)로부터 웹 응용을 위한 시험사례를 추출하는 방법을 제안한다. 웹 응용을 위한 시험사례들은 UML의 순차도에서 자기 호출 메시지(Self-Call Messages)를 포함한 메시지들로부터 추출된다. 시험사례는 시험할 대상인 메시지와 스크립트 함수, 서버 페이지와 추가적인 시험값들로 구성된다. 또한, 웹 응용 시험을 지원하기 위한 간단한 도구를 제시한다. 이 시험 도구는 웹 응용을 시험하기 위한 URL을 생성하고 이를 실행하여 그 결과를 보인다. 생성된 URL은 시험할 서버 페이지의 주소와 추가적인 시험값들로 구성된다. 이 도구는 마이크로소프트 비주얼 베이직으로 작성하였다. 회원 관리 기능을 수행하는 웹 응용 개발 과정에 제안하는 시험사례 추출방법과 시험 지원 도구를 적용하였으며 그 유용성을 확인하였다.

ABSTRACT

The systematic testing is frequently regretted in recent web applications because of time and cost pressure. Moreover developers have difficulties with applying the traditional testing techniques to testing web application. The approach of creating test cases for a web application from a sequence model is proposed for the rapid and efficient testing. Test cases for web application are extracted from call messages (including self-call messages) of UML (Unified Modeling Language) sequence diagram. A test case consists of messages, script functions, or server pages and additional values. Moreover a simple testing tool for web application is proposed. A URL for testing web application is created and executed by this testing tool. The URL consists of server page address and additional values. This test tool is made using Microsoft Visual Basic. The efficiency of proposed method and tool has been shown using a practical case study which reflects the development project of the web application for supporting member management

키워드 : 웹 응용 테스트, 시험 사례, 시험 도구

Web Application Testing, Test Case, Test Tool, UML

본 연구는 숭실대학교 교내연구비 지원으로 이루어졌음.

* 숭실대학교 정보과학대학 컴퓨터학부

** 숭실대학교 대학원 컴퓨터학과

1. 서 론

인터넷 환경이 확산되면서 현재의 업무들은 웹에서 많은 부분이 이루어지고 있다. 초보자들도 손쉽게 사용할 수 있는 웹이 보급되면서 많은 기업 및 조직들은 단순히 보여주는 웹이 아니라 사용자와 의사소통을 할 수 있는 웹을 개발하고 있으며, 자사의 광고, 물품 판매, 고객 지원 등의 목적을 위해 다양한 분야에서 지속적으로 웹 응용 개발이 수행되고 있다[1]. 또한, 웹을 통한 비즈니스와 서비스가 다양해지고 상호 경쟁하게 되면서 최근의 웹 응용 개발은 정확하면서도 빠른 개발과 이를 통한 시장의 선점을 요구하고 있다[2].

이러한 환경에서 웹 응용의 정확성은 더욱 강조되고 있다. 특히 인터넷에서의 전자상거래를 통하여 수익을 얻는 기업의 경우, 웹 응용이 정확하지 않게 작동하였거나 잘 못된 동작으로 웹 응용을 통한 서비스가 중단되었을 경우 기업의 수익에 큰 손실을 가져올 수 있을 뿐만 아니라 고객들에게 나쁜 이미지를 심어주게 되어 장기적으로도 기업에 큰 손실을 가져올 수 있다. 이러한 상황을 미리 방지하기 위해서는 웹 응용의 정확성에 대한 시험이 필수적이다.

현재에도 웹 응용 시스템을 시험하기 위한 방안들이 연구되고 있으나, 인터넷 환경이 아니었던 전통적인 시스템에 대한 시험방법에 비해서는 아직 많은 연구가 이루어지지 않았다. 초창기에 제시된 웹 응용을 위한 시험 방법들은 대부분 웹 응용 시스템을 구성하는 html, jsp, asp 파일들의 문법 및 구조를 검사하는 방법이나, 이 페이지들이 연결하는 링크

들이 정확히 존재하는 지를 시험하는 것이 대부분이다. 최근에는 웹 응용의 기능을 시험하기 위한 연구도 활발히 진행되고 있다. 그러나 현재로는 단순히 단일 웹 페이지의 기능을 분석하거나, 웹 페이지간의 호출관계를 분석하는 것이 대부분이다. 그러나, 웹 응용의 정확한 시험을 위해서는 이 두 가지는 물론 웹 페이지들의 협력에 통하여 수행하는 기능을 시험하기 위한 클러스터 시험도 필요하다. 국내에서 개발한 한국전자통신연구원(ETRI)의 마르미 시리즈[3]등과 같은 많은 범용 방법론의 프로세스에서는 단위 시험, 통합 시험, 시스템 시험 등을 모두 차례로 시험하기를 명시하고 있다. 또한, 고객에게 인도하는 경우에는 사용자 승인 시험, 혹은 인도 시험 등을 추가적으로 수행하여야 한다.

이에 본 논문에서는 UML[4] 모델의 순차도(sequence diagram)를 기반으로 하여 시험 사례를 추출하는 방안을 제시한다. 웹 페이지간의 메시지의 순서를 표현하는 UML 모델의 순차도에서 각각의 메시지들을 시험사례로 추출한다. 자기자신에게 발생하는 메시지(self-call message)는 웹 페이지 내에 존재하는 스크립트 함수를 호출하는 경우이거나, 변수 값을 달리하여 서버 페이지를 재수행하는 경우이다. 이러한 메시지들은 단일 웹 페이지 시험을 위한 시험사례로 추출한다. 웹 페이지간에 전달되는 메시지는 상호 웹 페이지 시험을 위한 시험사례로 추출한다. 시스템의 사용자인 액터(actor)가 시스템으로 보낸 메시지와 그 응답으로 시스템으로부터 받은 메시지는 통합 웹 페이지 시험을 위한 시험사례로 추출한다. 이러한 방법으로 순차도의 메시지

들을 기반으로 연관된 함수, 서버페이지, 상호 웹 페이지, 통합 웹 페이지 시험을 수행한다.

또한, Microsoft Visual Basic을 이용하여 작성한 웹 테스트 지원 도구를 제시한다. 이 도구는 시험할 서버 페이지의 주소와 추가적인 변수 값들로 구성된 시험사례로 받아 시험사례를 직접 실행하고 그 결과를 보여주는 도구이다. 이 도구를 통하여 손쉽게 개발한 웹 응용을 시험할 수 있으며, 특히 웹 응용의 수정 후 수행하는 회귀시험(regression test)을 간편하게 실행할 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 관련연구로 기존에 제시된 웹 기반 시스템을 위한 시험사례 추출방법들에 대하여 알아본다. 3장에서는 적용사례로 개발하였던 설계모델과 개발시스템을 예로 들며 시험사례 추출방법에 대하여 알아본다. 4장에서는 웹 응용 시험을 지원하기 위하여 개발한 테스트 지원 도구인 OnlineTestWeb에 대하여 알아본다. 5장에서는 기존의 웹 응용 시험방법과 본 논문에서 제시하는 방법을 비교 평가하고, 마지막으로 6장에서는 제시한 방법에 대한 결론과 향후 연구 방향을 기술한다.

2. 관련 연구

본 절에서는 기존에 연구되었던 웹 응용의 시험사례 추출방법에 대하여 알아본다. 본 논문에서는 다른 연구에서 많이 참고되고 있는 RUP(Rational Unified Process)[5]의 시험사례 추출방법과, Ye Wu와 Jeff Offutt가 제안한 웹 기반 응용의 모델링 및 시험방법[6]과

Filippo Ricca와 Paolo Tonella가 제안한 웹 응용 분석 및 시험방법[7]에 대하여 알아본다.

2.1. RUP의 시험사례 추출방법

RUP의 경우에는 기능적 시험을 위하여 사용사례(use case)로부터 시험사례를 추출하는 기법을 지침(guidelines)으로 제공하고 있다. 각 사용사례 내부의 이벤트 흐름(flow of event)에는 기본 흐름(basic flow)이외에도 여러 가지 대체 흐름(alternate flow)들이 있을 수 있는데, 이들을 조합하여 여러 가지 시나리오(scenario)들을 생성하게 된다. 이러한 시나리오들에 관련된 변수들을 위한 시험 데이터를 추가하고, 필요한 경우 조건을 추가하여 시험사례들을 추출한다. 그러나, RUP에서는 단위 시험, 통합 시험, 시스템 시험 등의 시험 수준에 대한 언급은 있으나 이러한 시험 수준별로 시험사례를 추출하는 방법은 제시하지 않고 있다.

다음의 <표 1>은 RUP 지침에 따라 사용사례로부터 시험사례들을 추출한 예이다.

기능적 시험을 위하여 사용사례를 이용하는 반면 비 기능적 시험을 위해서는 그 밖의 지원 명세(Supplementary specification)들을 기반으로 추출한다. 그러나 비 기능적 시험의 경우 RUP에서는 명시적인 기법을 제공하지 않고 있으며, 휴리스틱(heuristic)한 방법으로 시험사례를 추출하게 된다. 단지 시험사례 추출에서 참고할 점검사항들만을 제시한다.

〈표 1〉 RUP기법에 따른 시용사례로부터의 시험사례 추출 예

시험 사례 식별자	시나리오 /조건	시험 데이터					예상결과
		비밀 번호	계좌번호	요청금액	계좌잔액	ATM잔액	
CW1	시나리오1-성공적인 인출	4682	461-123456-12-001	50,000	500,000	2,000,000	성공적으로 인출 수행, 계좌잔액을 450으로 변경
CW2	시나리오2-현금없음	4682	461-123456-12-001	100,000	500,000	0	예금인출 기능을 사용불능 상태로
CW3	시나리오3-현금부족	4682	461-123456-12-001	100,000	500,000	70,000	잔액부족을 알림. 인출금입력 오류
...

2.2. Ye Wu와 Jeff Offutt의 웹 응용 시험 기법

Ye Wu와 Jeff Offutt는 servlet을 section으로 나누어 분석하고 이러한 section들간의 흐름을 시험사례로 추출하여 시험하는 방법을 제시하고 있다. Section은 servlet 코드의 일부를 개별적으로 식별하는 단위로 최소 단위인 atomic section과 이들의 집합으로 구성된 composite section 등이 있다.

Atomic section은 servlet코드에서 항상 같이 실행되거나, 항상 같이 실행되지 않는 부분을 각각 분리한 단위이다.

위의 〈표 2〉에서 p1~p4는 atomic section이다. 각 atomic section은 항상 같이 실행되거나, 조건에 따라 항상 같이 실행되지 않을 수 있다. 예제로 제시한 Servlet을 P라고 할 때, 정의한 atomic section을 기반 composition rule에 따라 다음과 같이 표현할 수 있다.

〈표 2〉 Servlet의 분리

p1 =	<pre> PrintWriter out = response.getWriter(); out.println("<HTML>"); out.println("<HEAD><TITLE>" + title + "</TITLE></HEAD>"); out.println("<BODY>"); </pre>
p2 =	<pre> for (int I=0; I<myVector.size(); I++) if (myVector.elementAt(i).size > 10) out.println("<P>" + myVector.elementAt(i) + "</P>"); else </pre>
p3 =	<pre> out.println("<P>" + myVector.elementAt(i) + "</P>"); </pre>
p4 =	<pre> out.println("</BODY></HTML>"); out.close(); </pre>

$$P \rightarrow p_1 \cdot (p_2 \mid p_3)^* \cdot p_4$$

atomic section간의 작은 점은 순차적인 수행을 의미하며, (|)은 괄호안에 “|” 으로 구분된 section들 중 하나가 선택적으로 수행됨을 의미하며 위 첨자로 표시된 * 는 반복을 의미한다. 이러한 방법으로 예제로 제시한 servlet의 내부 수행을 식으로 표현하였으며, 이는 p_1 이 수행되고, p_2 혹은 p_3 중 하나가 반복적으로 수행된 후 p_4 가 수행되고 마침을 의미한다.

알파벳 대문자로 표현된 서버 프로그램 사이의 관계를 식으로 표현하는 방법도 제공하고 있다.

$$S \Rightarrow \text{GradeServlet} \rightarrow p_1 \cdot p_3 \cdot p_4 \Rightarrow S$$

위의 식은 초기 웹 페이지인 S에서 GradeServlet로 링크 혹은 submit을 통하여 연결되었으며, GradeServlet이 수행된 후에는 p_1 , p_3 , p_4 인 atomic section들이 차례대로 수행되고, 마지막으로 다시 초기 페이지인 S로 연결됨을 의미한다. 위의 식에서 두줄 화살표는 HTML의 <A> 태그나 <FORM> 태그의 submit 등을 통한 사용자의 요청에 의해 다른 페이지로 연결됨을 의미하며, 한줄 화살표는 좌항이 실행됨으로써 우항의 section들이나 페이지가 프로그램에 의해 실행됨을 의미한다.

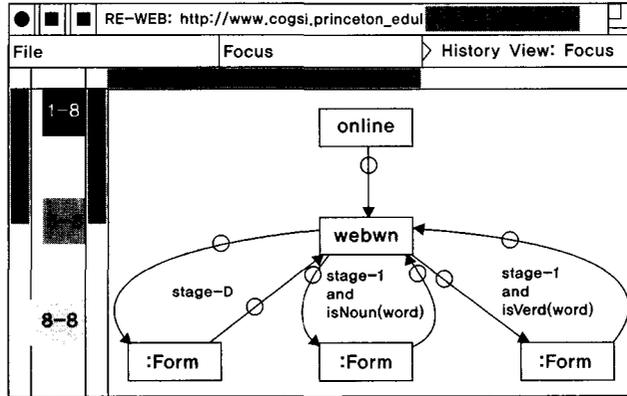
이 방법에서는 서버 프로그램의 코드를 분석하여 가능한 실행흐름들을 기호를 이용한 식을 통하여 정형적으로 표현하였으며, 이러한 식 각각이 시험사례로 사용될 수 있다. 그러나 이러한 식만으로는 시험시의 조건들이나 시험 데이터 값을 표현하기에는 어려움이

있다. 또한, white box testing을 기반으로 가능한 모든 수행흐름을 고려하고 있어 서버 프로그램이 조금만 복잡해져도 무수히 많은 시험사례들이 도출될 수 있으며, 여러 페이지들의 통합시험이나 시스템시험에 적용할 경우 매우 복잡한 수많은 식들이 도출된다. 특히, 이 연구에서는 이러한 식들을 사용하여 시험에 활용하는 방안들은 향후 연구로 남겨두고 있어 실제로 적용하기에는 많은 어려움이 남아있다.

2.3. Filippo Ricca와 Paolo Tonella의 웹 응용 시험 기법

Filippo Ricca와 Paolo Tonella는 웹 응용 시험을 위하여 크게 static verification과 dynamic verification 등 두 가지 방법을 설명하고 있다. Static verification은 일반적으로 html파일을 대상으로 수행하는 검사로 html문서의 문법상의 오류나 다른 파일이나 리소스의 링크가 정상적인지, 해당 링크와 연결된 파일이나 리소스가 실제 존재하는 지를 검사하는 것이다. 이러한 검사는 현재 출시되어있는 많은 도구들을 통해서도 손쉽게 확인할 수 있다.

Dynamic verification에서는 웹 응용을 구성하는 웹 페이지들 사이의 관계를 분석하여 시험사례를 생성한다. 우선 ReWeb 이라는 도구의 지원을 통하여 개발된 웹 사이트를 분석하여 웹 페이지간의 연관관계를 표현하는 그래프를 작성한다. 이후 TestWeb 이라는 도구의 지원을 통하여 그래프 이론을 바탕으로 웹 페이지간의 모든 경로를 시험 사례로 추출하여 시험을 수행한다. 다음의 <그림 1>[7]은 이 기법을 기반으로 웹 페이지 사이의 관계를 그



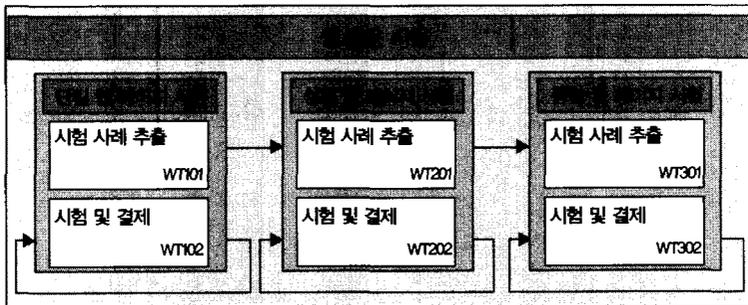
〈그림 1〉 ReWeb을 이용한 웹 사이트 분석모델

래프로 표현한 예이다.

그러나, 이 기법에서는 변수값이 조금씩 바뀌면서 웹 페이지들이 서로 지속적으로 연결되는 경우에는 많은 시험사례가 도출될 수 있다. 또한, 웹 페이지 내부에 존재하는 스크립트 함수를 개별적으로 시험하는 데에는 어려움이 있다. 가장 작은 단위인 웹 페이지 내부의 스크립트 함수와 웹 페이지 내부 논리의 정확성이 시험된 이후에 웹 페이지간의 시험을 수행하는 것이 합당하다.

3. 웹 응용을 위한 시험사례 추출방법

웹 응용을 시험하기 위해서는 가장 작은 단위로 부터 시험을 수행하여 시스템 전체로의 시험으로 확장해나간다. 하위 모듈이 정확하게 동작하는지 신뢰할 수 없는 상태에서 이를 포함한 상위 모듈을 시험한 결과는 신뢰할 수 없기 때문이다. 이러한 단계적인 시험절차는 반드시 필요하다. 웹 응용을 위한 시험은 단일 웹 페이지 시험, 상호 웹 페이지 시험, 통합 웹 페이지 시험 등 크게 3가지로 구분된다.



〈그림 2〉 웹 응용 시험 단계

정확한 시험을 위해서는 다음의 <그림 2>와 같이 3가지 시험이 차례로 수행되어야 한다.

이 때, 수준이 다른 각 시험들은 반복적으로 수행하게 된다. 시험을 통하여 오류를 발견하고 수정한 경우 다시 동일한 환경에서 회귀시험을 수행하여 정확히 정제되었는지 판단하여야 하기 때문이다. 또한, 시험 후 정제는 웹 응용의 다른 부분에 영향을 줄 수도 있으므로, 관련된 부분들은 시험을 수행하여야 한다. 시험을 언제까지 수행하여야 하는가의 물음에는 Logarithmic Poisson Execution-time Models[8]을 참조하여 발견한 누적 오류의 수와 예상되는 누적 오류의 수가 미리 정의한 일정 수준에 이르렀을 때 시험을 종료하고 다음 단계로 이동한다. Logarithmic Poisson Execution-time Models의 표현형식은 다음과 같다.

$$f(t) = (1/p)\ln[10 pt+1]$$

$f(t)$ = 실행 시간 t 동안 시험하였을 때 발생할 것으로 예상되는 누적 오류 수

p = 발견한 오류의 수정을 통한 오류 정도의 지수적 감소 비율

10 = 시험 시작시 초기의 시간 단위당 고장 정도

본 장에서는 <그림 2>에서 제시한 웹 응용 시험 단계를 바탕으로 웹 페이지간의 메시지 전달을 UML의 순차도로 표현한 다이어그램에서 웹 응용을 위한 단일 웹 페이지 시험, 상호 웹 페이지 시험, 통합 웹 페이지 시험을 위한

시험사례를 추출하는 방안을 차례로 제시한다.

3.1 단일 웹 페이지 시험사례 추출

웹 응용의 시험을 위하여 가장 먼저 해야 할 시험은 단일 웹 페이지 시험이다. 단일 웹 페이지 시험에서는 개별 웹 페이지에 대한 시험을 시작한다. Html과 같은 언어로 만들어진 정적인 페이지는 구문검사와 포함하는 리소스 검사를 수행한다. 구문검사는 html파일이 문법에 맞게 작성되었는지 확인하는 것이며, 리소스 검사는 html파일이 불러오거나 연결하는 url의 파일들이 실제로 존재하는 지를 확인하는 작업이다. 이러한 작업들은 기존에 나와있는 html 작성도구나 html 파일 검사 도구들을 통해서도 손쉽게 시험할 수 있다.

시험을 위한 주요 대상이 되는 것은 논리를 포함하고 있는 동적인 파일들로 servlet이나 jsp, asp같은 server page와 server page에 연결된 bean 클래스 등 client 측에 넘겨줄 html 파일을 생성하는 파일들이다. 이 단계에서는 동적인 페이지들 내에 존재하는 독립적인 실행 모듈들이나 웹 페이지의 재실행을 시험한다.

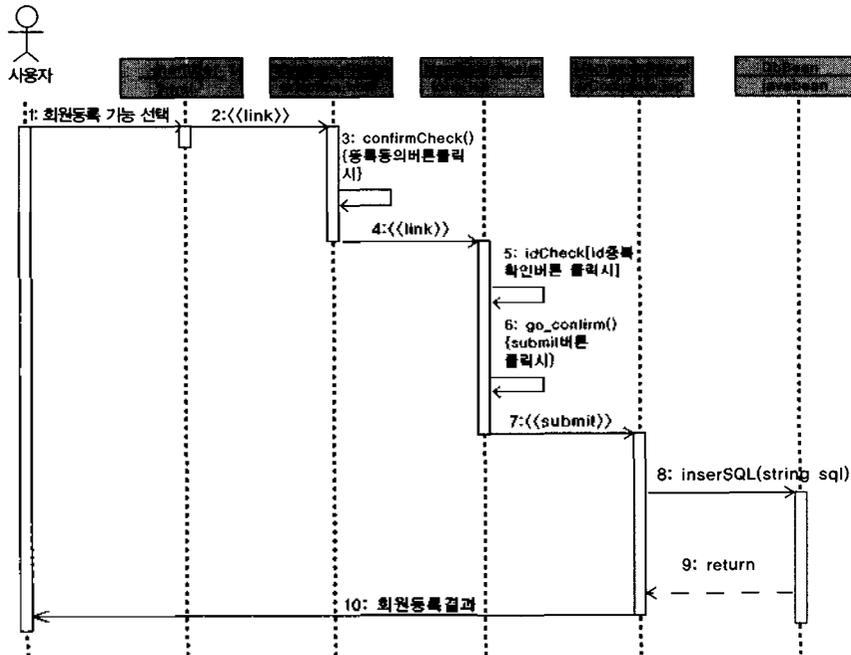
단일 웹 페이지 시험을 위해서는 각 웹 페이지의 자기호출 메시지(self-call message)를 시험사례로 추출한다. 이 경우는 서버 페이지가 서버 페이지 내에 존재하는 스크립트 함수를 호출하는 경우이거나, 웹 페이지가 스스로를 재실행하는 경우이다. 웹 페이지가 자신을 재실행하는 경우는 대부분 여러 기능이 중복되어 있는 경우이다. 예를 들어, 하나의 서버 페이지 내에 정보를 등록하는 기능과 등록결과를 보여주는 기능이 함께 있으며 작업의 종

류를 지정하는 변수의 값에 따라 두 가지 기능 중 하나를 수행하는 경우에, 정보를 등록하는 작업을 마친 이후에 등록결과를 보여주는 기능을 수행하기 위해 작업 지정 값을 달리하여 스스로를 재실행하게 된다.

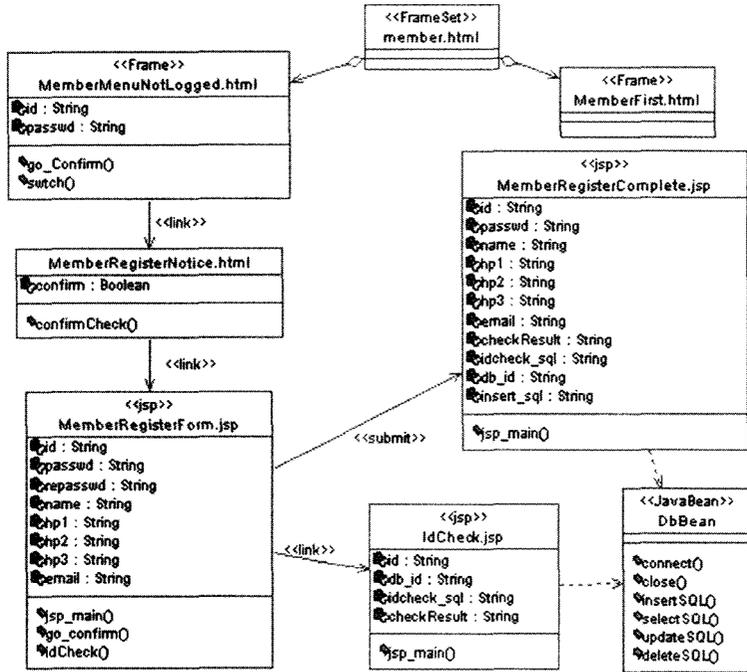
스크립트나 웹 페이지가 변수값을 입력 받는 경우에는 시험사례에 시험테이터들을 추가한다. 또한, 스크립트 함수가 직접 매개변수로 테이터를 전달 받지 않더라도, 내부코드에서 자바스크립트의 document 객체[9]를 이용하여 서버 페이지에 Form에 존재하는 데이터 값을 참조하여 수행하는 경우에는 이러한 데이터 들도 시험 테이터로 고려한다. <그림 3>에서는 3, 5, 6번 메시지가 이에 해당한다.

메시지를 통하여 시험항목을 식별하는데, 시험항목과 관련한 웹 페이지, 스크립트 함수, 관련변수는 페이지 다이어그램[10]에 명세한 내용을 참조한다. 페이지 다이어그램은 UML 클래스도 표기법을 사용하여 웹 페이지와 이들 사이의 관계를 표현한 다이어그램이다. 페이지 다이어그램을 작성하는 방법은 다음과 같다.

- 추출한 웹 페이지들을 클래스로 나타낸다. 페이지 작성에 사용할 언어에 따라 <<HTML>>, <<JSP>>, <<ASP>>, <<PHP>> 등의 스테레오타입으로 웹 페이지의 종류를 구분한다. 여러 페이지간의 기능의 재사용이나 유지보수의 편의성 등을



<그림 3> 순차도 - 회원등록



〈그림 4〉 페이지 다이어그램 - 회원등록

위하여 기능의 일부를 bean형태로 추출한 경우에는 <<JavaBean>> 등의 스테레오타입을 갖는 클래스로 표현한다.

- 페이지간의 관계는 클래스간의 관계로 표현한다. 페이지 사이의 관계의 종류에 따라 <<build>>, <<link>>, <<redirect>>, <<submit>> 등의 스테레오타입을 사용한다. 이때, FrameSet과 Frame의 관계는 aggregation으로 표현한다.
- 웹 페이지에서 사용하는 데이터 필드들과 웹 페이지 내부의 논리에서 사용된 변수들을 클래스의 속성으로 표현한다.
- Server page의 논리와 Script 함수들은 클래스의 함수로 나타낸다. JSP, ASP 등

의 서버 페이지를 구성하는 기본 논리는 jsp_main(), asp_main() 등의 미리 정의한 이름으로 표시하며, script 함수들은 각각의 함수 이름을 사용한다.

다음의 <표 3>은 단일 웹 페이지 시험을 위하여 <그림 3>의 순차도에서 시험사례의 대상으로 선정된 3, 5, 6번 메시지를 기반으로 시험사례들을 추출한 예이다. 첫 번째 시험사례인 UT01은 MemberRegisterNotice.html 파일에서 사용자가 회원등록에 동의함을 확인하고 회원등록을 요청하였을 때의 반응을 시험하는 것이고, 두 번째 시험사례인 UT02는 MemberRegisterForm.jsp 파일에서 입력한 ID

3.2 상호 웹 페이지 시험사례 추출

단일 웹 페이지에서는 javascript 함수와 같이 각 웹 페이지 내에 존재하는 독립적인 실행 모듈들과 웹 페이지의 재실행을 시험하였다. 단일 웹 페이지 시험이 끝난 이후에는 웹 페이지들 간의 상호관계에서 웹 페이지들이 정상적으로 수행되는지를 시험하는 상호 웹 페이지 시험을 수행한다. 상호 웹 페이지 시

험의 목적은 웹 페이지가 다른 페이지로 설계한대로 정확히 연결되는지, 또한 연결되면서 관련정보들이 정확히 전달되며 훼손으로 인한 오류가 발생하지 않는지를 시험하는 것이다.

상호 웹 페이지 시험을 위해서는 순차도 상에서 각 웹 페이지가 액터 혹은 다른 웹 페이지로부터 수신하는 메시지를 시험사례로 추출한다. <그림 3>에서는 1, 2, 4, 7, 8번 메시지가 이 경우에 해당한다. 웹 페이지가 변수값

<표 4> 상호 웹 페이지의 시험사례 추출 예

단일 웹 페이지 시험 : 시험대상 - 회원등록 순차도							
메시지 식별자	시험사례 식별자	시험 항목	입 력				통과 여부
			(형식)	(이름)	(값)	(형식)	
1	IT01	Member.html	N/A	N/A	N/A	브라우저에 페이지 로드	
2	IT02	MemberRegisterNotice.html	N/A	N/A	N/A	브라우저에 페이지 로드	
4	IT03	MemberRegisterForm.jsp	N/A	N/A	N/A	브라우저에 페이지 로드	
7	IT04	MemberRegisterComplete.jsp	String	id	"yongsun"	회원 등록에 실패하였습니다. 모든 필드가 제대로 입력되었는지 확인한 후 다시 등록하십시오. 라는 메시지를 보이고 이전 화면으로 이동하는 버튼을 보임	
			String	passwd	"123456"		
			String	repasswd	"123456"		
			String	name	"조용선"		
			String	hp1	"011"		
			String	hp2	"9876"		
			String	hp3	" "		
			String	e-mail	"yongsuns@shinbiro.com"		
8	IT05		String	document.MemberInfo.email.value	(표 하단에 별도 표시)		
...	

IT05 시험사례의 sql = "insert into member(ID, PASSWD, NAME, HP1, HP2, HP3, EMAIL) values('yongsun', '123456', '조용선', '011', '9876', '5432', 'yongsuns@shinbiro.com')"

들을 입력 받는 경우에는 시험사례에 시험데이터들을 추가한다.

시험 사례에는 시험에 필요한 추가적인 입력값들이 필요하다. 시험 목적에 따라 시험 데이터들은 달라지며, 시험 데이터에 따라 웹 페이지의 실행에서 다른 분기가 발생하기도 한다. 시험 데이터를 작성하는 대표적인 방법에는 균등 분할(equivalence partitioning)과 경계값 분석(boundary value analysis)[11, 12] 등이 있다. 균등 분할은 프로그램의 입력 영역을 시험 사례가 유도될 수 있는 자료 형태의 클래스로 분류하는 것으로 일반적으로 유효한 입력 영역과 유효하지 않은 입력 영역을 모두 시험하도록 시험 데이터를 준비한다. 경계값 분석은 많은 수의 오류들이 입력 영역의 중앙보다는 경계에서 발생한다는 사실을 기반으로 입력 영역의 경계값 주변의 값들을 시험 데이터로 사용하는 방법이다. 이러한 기본적인 규칙들을 바탕으로 시험할 대상과 목적에 적합한 시험 데이터를 시험사례에 포함시켜야 한다.

다음의 표 4는 상호 웹 페이지 시험을 위하여 <그림 3>의 순차도에서 시험사례의 대상으로 선정한 1, 2, 4, 7, 8번 메시지를 기반으로 시험사례들을 추출한 예이다. IT01, IT02, IT03은 각각 해당 페이지가 링크를 통하여 정상적으로 브라우저에 로드 되는지를 시험하기 위한 것이다. IT04는 회원정보를 입력 받은 후 등록하는 페이지에서 전화번호의 일부가 생략되었을 때, 이를 정확히 판단하여 사용자에게 알리는지 시험하기 위한 것이며, IT05는 입력 받은 sql문을 이용하여 데이터베이스에 정보를 입력하는 javabean 클래스인 DbBean의 insertSQL 함수가 정상적으로 작동하는지 시험하기 위한 시험사례이다.

3.3 통합 웹 페이지 시험사례 추출

단일 웹 페이지 시험과 상호 웹 페이지 시험을 마친 이후에는 통합 웹 페이지 시험을 수행한다. 통합 웹 페이지 시험에서는 액터의

<표 5> 통합 웹 페이지의 시험사례 추출 예

단일 웹 페이지 시험 : 시험대상 - 회원등록 순차도						
메시지 식별자	시험사례 식별자	시험 상황	입력			
			(형식)	(이름)	(값)	(형식)
1. 10	CT01	회원등록 기능이 정상적으로 작동하는가.	정상적인 회원등록 정보		"회원 등록이 성공적으로 수행되었습니다." 메시지 표시	
1. 10	CT03	이미 존재하는 ID로 등록요청을 하였을 때, 이 오류를 정확히 처리하는가.	이미 존재하는 ID를 포함한 회원 등록 정보		"이미 존재하는 아이디입니다. 다른 아이디를 입력하십시오." 라는 메시지를 보이고 이전 화면으로 이동하는 버튼을 보임	
...	

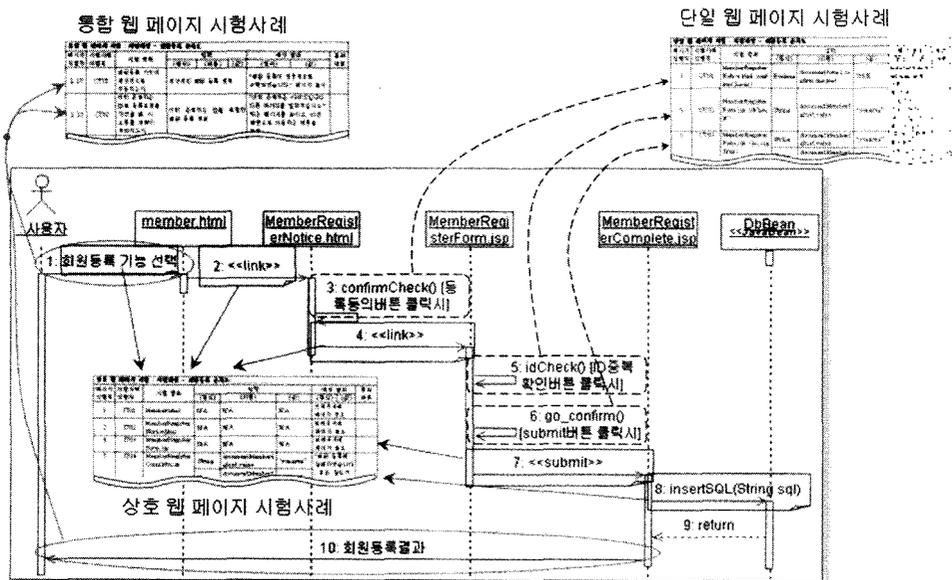
요청이 여러 웹 페이지들을 거치면서 수행된 후 예상된 결과가 도출되는 지를 시험하는 것이다.

통합 웹 페이지 시험을 위해서는 액터가 시스템에 전달한 메시지와 그 응답으로 시스템으로부터 받은 메시지를 시험사례와 시험사례의 결과로 추출한다. <그림 3>의 순차도에서는 1, 10번 메시지가 시험사례가 되며, 그 응답인 10번 메시지가 시험사례의 결과가 된다.

다음의 <표 5>는 통합 웹 페이지 시험을 위하여 순차도에서 시험사례의 대상으로 선정한 1, 10번 메시지를 기반으로 시험사례들을 추출한 예이다. 첫 번째 시험사례인 CT01은 정상적으로 수행되는 회원등록 기능 전체를 시험하기 위한 것이며, 두 번째 시험사례의 CT02는 이미 등록된 ID로 등록요청을 하였을 때 여러 웹 페이지들을 통해 수행되는 회

원등록 기능이 예상 결과와 같이 등록된 ID임을 알리고 이전 페이지로 돌아갈 수 있게 하는지 확인하기 위한 시험사례이다.

이와 같이 단일 웹 페이지 시험, 상호 웹 페이지 시험, 통합 웹 페이지 시험을 차례대로 수행하였다. 각각의 시험을 위하여 순차도의 메시지들로부터 시험사례를 추출하였으며, 시험사례와 관련된 웹 페이지, 함수, 변수에 대한 정보는 <그림 4>와 같은 페이지 다이어그램을 참조하였다. <그림 5>는 지금까지의 시험사례 추출방법들을 하나의 그림으로 표현한 것이다. 순차도의 자기호출 메시지로부터 단일 웹 페이지의 시험사례들을, 페이지 사이의 메시지로부터 상호 웹 페이지 시험사례들을, 액터와의 메시지에서 통합 웹 페이지 시험사례들을 추출하였다.



<그림 5> 순차도에서의 시험사례 추출

4. OnlineTestWeb을 이용한 적용

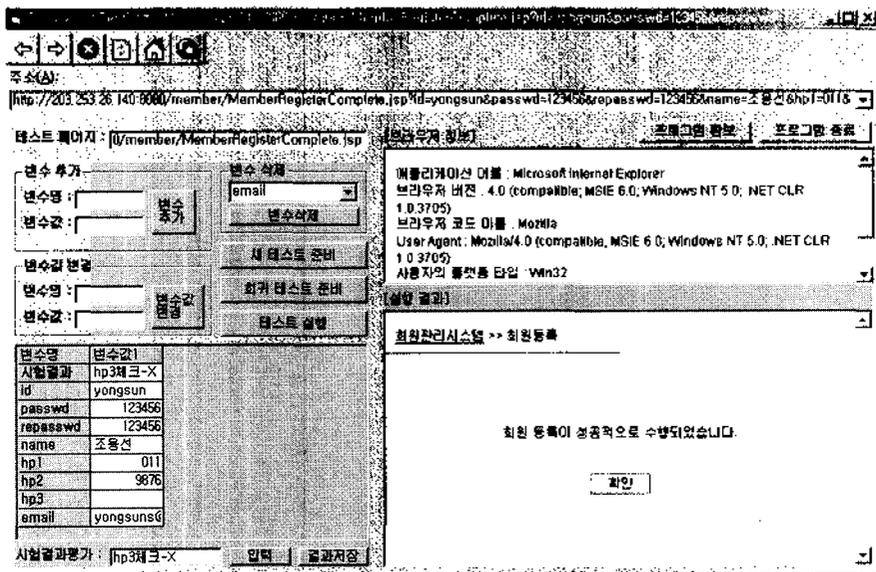
본 장에서는 웹 응용 시험을 지원하기 위한 도구로 개발한 OnlineTestWeb에 대하여 설명한다. OnlineTestWeb은 본 논문에서 제시한 기법을 바탕으로 웹 응용을 위해 추출한 시험 사례들의 시험작업을 지원하는 도구이다. 이 도구는 사용자가 입력한 시험사례를 바탕으로 웹 서버 및 어플리케이션 서버에 있는 웹 응용을 온라인상에서 실행하여, 그 결과를 사용자에게 보여주며, 이전 시험 데이터와 결과들이 도구에 누적되면서 여러 차례의 시험결과들을 비교하여 개발한 웹 응용의 기능에 대한 분석을 더욱 효과적으로 수행할 수 있다.

이 도구는 Microsoft Visual Basic 6를 이용하여 작성하였다. <그림 6, 7>은 OnlineTestWeb

의 모습을 보이고 있다. OnlineTestWeb 화면의 좌측은 테스트 시험사례를 입력하는 부분이다. 우선 "테스트 페이지" 항목에 시험의 대상이 되는 서버 페이지의 주소를 입력하고 시험을 위해 필요한 변수들의 이름과 값을 차례대로 입력한다. 이 때 입력된 "변수명"과 "변수값"들은 좌측하단에 스프레드시트 형태로 한눈에 볼 수 있도록 표시하여 준다.

테스트 실행 버튼을 눌러 시험을 시작하면 우측의 2개의 미니 브라우저로 시험결과를 보여준다. 우측 상단의 미니 브라우저는 시험에 사용하고 있는 브라우저의 정보를 보여주며, 우측 하단의 미니 브라우저는 시험사례를 웹 브라우저에서 실행한 결과를 보여준다. 시험결과를 확인한 후, 좌측 하단의 "시험결과 평가" 항목에 간단히 시험결과를 입력한다.

새로운 시험을 시작하기 위해서는 "새 테



<그림 6> OnlineTestWeb - <표 4>의 IT04시험 후 오류 확인

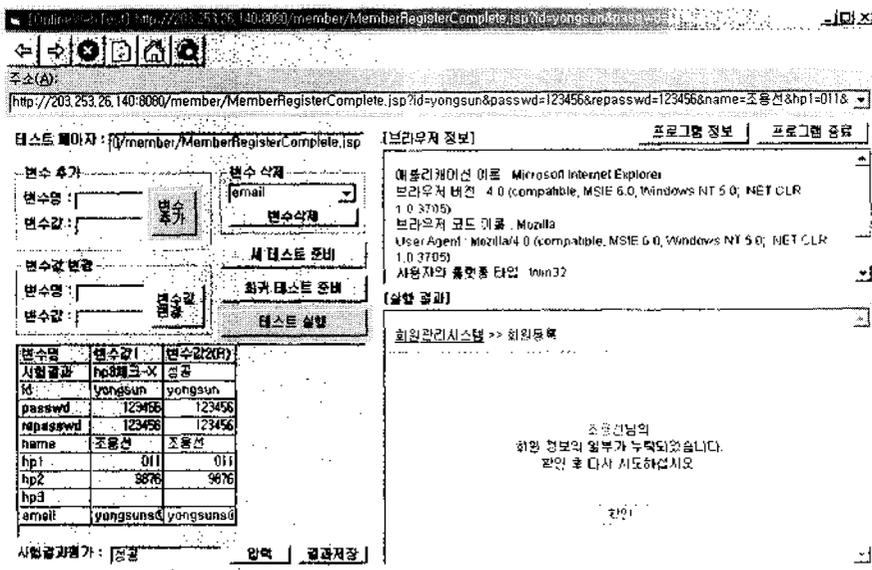
스트 준비” 버튼을 누른다. 이때는 이미 실행한 시험사례의 변수값들이 회색으로 바뀌고, 새로운 변수값을 받을 수 있도록 스프레드시트 우측에 새로운 열이 추가된다.

시험을 수행한 후 오류를 발견하면, 이 오류를 수정한 후 다시 같은 시험을 반복하는 회귀시험을 수행하여야 한다. 이때에는 “회귀 테스트 준비” 버튼을 눌러, 이전의 시험에 사용된 변수값들을 회색으로 만들고 동일한 변수값들을 갖는 새로운 시험을 위한 열이 스프레드시트 우측에 생성된다. 이때 열의 제목의 마지막에는 “(R)” 이라는 표시를 삽입하여 회귀시험 (regression test)임을 알린다.

〈그림 6〉은 〈표 4〉에서 제시하였던 상호 웹 페이지 시험을 위한 시험사례 IT04를 실행한 예이다. 〈표 4〉에서 예상결과로 기술한 것처럼 전화번호 마지막 4자리를 지칭하는 “hp3”

의 값이 누락되어 있으므로 회원등록이 이루어지지 않고 오류메시지를 내보여야 하였으나, 시험결과 회원등록이 정상적으로 이루어졌다. 이는 시험대상인 MemberRegisterComplete.jsp 파일에서는 등록정보의 완전성을 확인하는 코드가 누락되어 있었기 때문이었다. MemberRegisterComplete.jsp 파일에 등록정보를 전달하는 MemberRegisterForm.jsp 파일에서 미리 등록정보의 완전성을 확인하였기 때문에 생략하였던 것이다. 그러나, 이러한 경우에는 이전 페이지나 페이지간의 연결상에서 코드의 오류로 등록정보가 누락되거나, 사용자가 임의로 URL을 직접 조작하여 잘못된 등록정보로 회원 등록이 가능하게 된다. 시험을 통하여 이러한 오류를 식별하고 웹 응용을 수정하였다.

〈그림 7〉은 IT04시험을 통해서 발견한 회



〈그림 7〉 OnlineTestWeb - 〈표 4〉의 IT04시험의 오류 확인 및 수정 후 회귀시험

원등록상의 등록정보확인 작업의 오류를 수정하고 이전의 시험을 통하여 데이터베이스에 입력되었던 회원정보를 삭제한 후, 다시 회귀시험을 한 결과를 보여준다. "회귀 테스트 준비" 버튼을 눌러 이전의 시험 데이터들을 똑같이 사용하여 시험을 하였다. 이번에는 오류를 수정한 이후로 회원 정보의 일부가 누락되었음을 알리고 이전 페이지로 돌아갈 수 있도록 하였으며, 비정상적인 회원 등록이 이루어지지 않았다.

제시한 도구를 통하여 웹 응용을 손쉽게 시험해볼 수 있으며, 누적된 시험결과들을 비교하여 웹 응용의 분석을 더욱 효과적으로 수행할 수 있다. 또한, 오류를 확인하고 수정한 이후 회귀시험의 수행도 간편하게 행할 수 있다.

5. 시험사례 추출기법 비교 평가

본 장에서는 제시한 시험사례 추출기법이 기존의 연구와 비교하여 어떤 장점이 있는지 알아본다. 2장의 관련 연구에서는 다른 연구에서 많이 참고되고 있는 RUP(Rational Unified Process)[5]의 시험 사례 추출방법과, Ye Wu와 Jeff Offutt가 제안한 웹 기반 응용의 모델링 및 시험방법[6], Filippo Ricca와 Paolo Tonella가 제안한 웹 응용 분석 및 시험방법[7]에 대하여 알아보았다.

본 논문에서 제시하는 방법은 단일, 상호 통합 웹 페이지 시험의 세가지 수준으로 분류하여 각각 시험사례 추출기법을 제시하였으나, 관련연구의 다른 방법들에서는 모두 시험

수준을 분류하지 않고, 임의의 수준의 시험을 위한 시험사례 추출기법을 제시하였으며, 웹 응용에서 논리를 수행하는 최소 단위가 되는 스크립트 함수에 대한 시험방안도 부재하였다. RUP에서는 사용사례의 이벤트 흐름들의 조합으로 시험사례를 추출하여 통합 웹 페이지 시험 수준의 시험사례를 추출하였고, Wu, Offutt의 방법에서는 소스 코드 내의 흐름과 웹 페이지간의 흐름들의 조합을 시험사례로 추출하여 모든 수준을 같이 표현하여 복잡한 결과를 낳았다. Ricca, Tonella의 방법에서도 시험수준을 분류하지 않았으며, 웹 페이지간의 연관관계 모델을 기반으로 통합 웹 페이지 시험 수준의 시험사례 추출기법만을 제시하였다.

시험데이터에 대해서는 본 논문의 방법과 RUP에서는 시험사례에 정상적 실행 및 오류를 발생시키는 시험데이터를 추가하여 사용할 수 있었으나, Wu, Offutt의 방법에서는 논리간, 페이지간의 흐름만을 표현하여 시험데이터를 표현하기가 곤란하였다. Ricca, Tonella 방법에서는 시험데이터의 수에 따라서 시험사례가 크게 늘어날 우려가 존재하였다.

시험데이터의 상이함을 고려하지 않고 파악한 기본 시험사례의 개수를 추정해보면 본 논문의 방법에서는 순차도 메시지들의 개수가 기본 시험사례의 개수가 된다. 그러나 다른 방법들은 흐름들의 조합을 사용하므로 그 수가 크게 늘어날 우려가 있다. RUP에서는 각 사용사례의 이벤트 흐름들의 조합을, Wu, Offutt의 방법은 각 소스 코드 내부 흐름들과 페이지간 흐름들의 조합을, Ricca, Tonella의 방법은 페이지간 흐름들의 조합의 수를 사용한다.

〈표 6〉 시험사례 추출기법 비교 평가

비교 항목	본 논문의 시험사례 추출기법	RUP의 시험사례 추출기법	Wu, Offutt의 시험 사례 추출기법	Ricca, Tonella의 시험사례 추출기법
시험사례 출처	순차도 (Sequence Diagram)	사용사례 모델 (Use Case Model)	소스 코드	웹 페이지간 연관 분석 모델
시험수준별 시험사례 추출기법 제시	단일, 상호, 통합 웹 페이지 시험의 세가지 수준으로 분류하여 각각 시험사례 추출기법을 제시함.	시험수준을 분류하지 않았으며, 통합 웹 페이지 시험 수준의 시험사례 추출기법만을 제시함.	시험수준을 분류하지 않았으며, 각 시험수준들을 모두 하나의 시험사례로 표현하여 복잡해짐.	시험수준을 분류하지 않았으며, 통합 웹 페이지 시험 수준의 시험사례 추출기법만을 제시함.
시험 데이터 활용	추출한 시험사례에 적절한 시험 데이터 추가.	추출한 시험사례에 적절한 시험 데이터 추가.	시험 데이터 활용에 대한 명시적 방법 부재.	많은 수의 시험 데이터를 사용하기에는 곤란함.
스크립트 함수 시험	단일 웹 페이지 시험에서 수행.	명시적인 방법 없음	명시적인 방법 없음	명시적인 방법 없음
기본 시험사례 개수 예상치	순차도 메시지 개수.	각 사용사례의 이벤트 흐름들의 조합의 수.	각 소스 코드 내부 흐름들과 페이지간 흐름들의 조합의 수.	페이지간 흐름들의 조합의 수.

다음의 〈표 6〉은 제안한 시험사례 추출기법들을 다른 시험사례 추출기법들과 비교 평가한 내용을 정리한 것이다.

6. 결론 및 향후 연구방향

지금까지 UML의 순차도를 기반으로 웹 응용을 위한 시험사례 추출방법을 제시하였다. 웹 응용을 위한 시험을 단일 웹 페이지 시험, 상호 웹 페이지 시험, 통합 웹 페이지 시험의 세가지 수준으로 분류하였으며, 각각의 시험을 위한 시험사례 추출방법을 제시하였다. 개별 웹 페이지의 논리에 따른 재실행과 웹 페이지 내부의 스크립트 함수를 시험하는 단일 웹 페이지 시험의 시험사례를 위해서는 순차도의 자기호출 메시지를 활용하였다. 웹 페이지간의 연관관계에서 웹 페이지가 정상적으로

로 작동하는 지를 확인하는 상호 웹 페이지 시험의 시험사례를 위해서는 다른 웹 페이지로부터 전달 받는 메시지를 활용하였다. 또한, 여러 웹 페이지를 거치면서 사용자의 요청을 정확히 이행하는지를 시험하는 통합 웹 페이지 시험에서는 액터와 주고 받는 메시지를 활용하였다.

순차도는 클래스들 간의 메시지 흐름을 시간순으로 모형화한 것으로, 클래스를 웹 페이지로 대치하여 웹 페이지간의 연결을 표현하는 경우, 웹 페이지간의 논리의 흐름을 쉽게 파악할 수 있으며 이를 기반으로 시험 사례를 손쉽게 추출할 수 있다. 현재 웹 응용을 위한 시험방안들의 연구가 많이 부족한 상태이므로 유용하게 사용될 것이다. 특히 최근에는 웹 응용의 분석 및 설계에서도 UML의 표기법을 많이 사용하고 있으므로, 제시한 시험사례 추출기법의 더욱 유용할 것이다.

또한, Microsoft Visual Basic을 사용하여 웹 응용을 시험을 지원하는 도구인 OnlineTestWeb을 제시하였다. 이 도구는 시험 대상인 웹 페이지의 주소와 시험 데이터로 구성된 시험사례를 입력하면 이를 실행하고 결과를 보여주는 간단한 웹 응용 시험 지원 도구이다. 도구의 활용을 통하여 시험 데이터를 바꾸어 가면서 빠르게 시험을 행할 수 있으며, 오류의 수정 후 같은 시험을 다시 실행하는 회귀시험도 손쉽게 수행할 수 있다. 또한, 이전 시험 데이터와 결과들이 도구에 누적되면서 여러 차례의 시험결과들을 비교하여 개발한 웹 응용의 기능에 대한 분석을 더욱 효과적으로 수행할 수 있다.

본 논문에서 제시한 웹 응용을 위한 시험기법을 더욱 발전시키기 위하여, 웹 응용 분석 및 설계 작업과 시험작업을 연결하여 더욱 효율적인 시험이 되도록 할 것이다. 또한, 시험 결과들을 누적하여 비교하면서 더 많은 정보를 획득할 수 있도록 제시한 시험 지원도구를 개선해 나갈 것이다.

참 고 문 헌

- [1] Abhijit Chaudhury, et al., "Web channels in E-Commerce", Comm. ACM, Vol. 44, No. 1, pp. 99-104, 2001. 1.
- [2] Richard Baskerville, et al., "How Internet Software companies Negotiate Quality", IEEE Computer, Vol. 34, No. 5, pp. 51-57, 2001. 5.
- [3] 마르미(MaRMI) I / II / III / III v2.0x, 한국전자통신연구원(ETRI), 1996 ~ 2003
- [4] Unified Modeling Language Specification Version 1.4, OMG (Object Management Group), 2001. 9.
- [5] The Rational Unified Process, Rational Software Corporation (a wholly owned subsidiary of IBM), 2003.
- [6] Ye Wu, Jeff Offutt, "Modeling and Testing Web-based Applications", GMU ISE Technical ISE-TR-02-08, Nov. 2002.
- [7] Filippo Ricca, Paolo Tonella, "Analysis and testing of Web applications", Proceedings of the 23rd international conference on Software engineering, 2001.
- [8] Musa, J.D. and Ackerman, A.F., "Quantifying Software Validation: When to Stop Testing?", IEEE software, pp. 19-27, 1989. 5.
- [9] Mashito Hamba, Ryuichi Okakura, 자세히 설명한 HTML&Java Script 사전, 영진.com, 2000.
- [10] 조용선, 정기원, "고속 웹 개발을 위한 업무모델 중심의 접근 방법", e-Korea를 위한 전자거래 종합학술대회, pp.590-601, 한국전자거래(CALS/EC)학회 2002. 9.
- [11] Roger S. Pressman, Software Engineering : A Practitioner's Approach (5th Edition), McGraw-Hill, 2000
- [12] 왕창중, 소프트웨어 공학:개정판, 정의사, 2000.

저 자 소 개



정기원 (E-mail : chong@computing.ssu.ac.kr)
1967. 서울대 전기공학과(공학사)
1981. 미국 알라바미주립대 전산학과(석사)
1983. 미국 텍사스주립대 전산학과(박사)
1996~1968. 미8군 (IBM 기계정비 담당)
1971~1975. 한국과학기술연구소(책임연구원)
1975~1990. 국방과학연구소(책임연구소)
1990~현재 송실대학교 컴퓨터과학과 교수
관심 분야 소프트웨어 개발 프로세스, 방법론, 모델링, 실시간 응용, 전자거래, 정보시스템 개발 및 평가



조용선 (E-mail : youngsuns@hanafos.com)
1997. 2 송실대학교 소프트웨어공학과(공학사)
1992. 2 송실대학교 대학원 컴퓨터학과(석사)
2004. 2~현재 송실대학교 대학원 컴퓨터학과(박사)
관심 분야 소프트웨어 개발 프로세스, 방법론, 모델링, 웹 응용 분석 및 개발, 정보시스템 개발 및 평가