

## 웹 응용프로그램의 상호운용성 분석과 테스트 방법

김 익 수\* · 최 중 명\*\*

### *An Analysis and Testing Method for Interoperability of Web Applications*

Kim, Ik Su · Choi, Jong Myung

#### 〈Abstract〉

As the web technologies advance, most of applications have being developed as web applications. The web has advantages of high interoperability due to the web standards though the applications run on heterogenous platforms. However, as the systems are growing and have some interoperability problems, the government and the big companies require interoperability testing for new systems. In this paper, we introduce some interoperability issues for web applications and classify them into three groups according to 3-tier client/server architecture. We also propose interoperability testing process and tasks to be performed at each step in the process.

Key Words : Interoperability Testing, Web Application, Web Application Server

### I. 서론

지난 수십 년간 소프트웨어들 간의 호환성은 매우 큰 이슈였다. 호환성 때문에 플랫폼을 변경하지 못하거나, 다른 플랫폼에서 개발된 소프트웨어를 효과적으로 사용하지 못하는 것이 일반적이었다. 또한, 한 플랫폼에서 개발한 소프트웨어를 다른 플랫폼에 포팅하기 위해서 많은 노력과 비용을 들여야 하는 것이 당연하게 여겨졌다. 특히, 최근 들어서는 시스템이 매우 복잡해지고 분산 환경으로 발전하면서 개별적으로 개발된 시스템들을 통합하여 서비스를 제공하려는 추세이기 때문에 시스템 간의 상호운용성은 매우 중요하다[1]. 이에 정부에서도 한국정

보화진흥원을 중심으로 정부 시스템에 대한 상호운용성 확보를 위한 연구를 진행하고 있다[2].

시스템들 간의 상호운용성 문제는 최근 십여 년간 표준화 기술을 중심으로 하는 웹 응용프로그램과 웹 응용 프로그램 서버의 발전 덕분에 상당 부분 해소되었다. 즉, IBM의 웹스피어, 혹은 티맥스의 제우스를 미들웨어로 사용하는 시스템 상에서 구현된 대부분의 응용프로그램들은 일관된 결과를 기대할 수 있게 되었으며, 표준화를 기반으로 한 시스템 통합 환경을 통해 높은 상호운용성을 보장할 수 있게 되었다. 이와 같이 표준화 기술은 웹 응용프로그램 서버뿐만 아니라 운영체제 혹은 하드웨어에도 적용되어 이를 따르는 시스템 구현들은 높은 호환성을 보장할 수 있다.

웹과 인터넷으로 대변되는 표준화된 프로토콜의 발전으

\* 숭실대학교 컴퓨터학부 조교수

\*\* 목포대학교 컴퓨터공학과 조교수(교신저자)

로 웹 응용시스템 간의 호환성에는 상당한 발전이 있었지만, 아직 해결해야 할 문제들이 여전히 남아있다. 예를 들면, 웹 응용프로그램 개발에서 사용되는 프로그램 언어의 비표준 확장 기능과 플랫폼 간의 데이터 타입 문제 등은 상호운용성에 있어 커다란 문제점을 야기한다. 예를 들어, 비표준 확장 기능을 이용하여 구현된 사용자 인터페이스는 특정 브라우저에서 올바르게 동작하지 않을 수 있다. 또한, 플랫폼 간의 상이한 데이터 타입의 사용은 서버 간에 교환된 데이터가 올바르게 처리되지 않기 때문에 시스템 오동작 혹은 서비스 정지 등의 문제를 유발시킬 수 있다.

본 논문에서는 웹 응용프로그램의 상호운용성 문제를 해결하기 위해 시스템 구조를 기반으로 각 분야에서 고려해야 할 상호운용성 점검 항목을 소개하고, 상호운용성 향상을 위한 단계별 테스트 방법을 제시한다. 제안된 테스트 방법에 통과하는 웹 응용프로그램은 브라우저의 종류에 상관없이 사용자에게 일관된 서비스를 제공하며, 플랫폼이 서로 다른 웹 응용 시스템간의 상호운용을 가능하게 한다. 특히, 본 논문은 상호운용성 테스트 관련 업무를 효과적으로 수행할 수 있는 기반을 마련하며, 나아가 최근 이슈가 되고 있는 서비스 지향 아키텍처(SOA)에서의 상호운용성 테스트를 위한 연구에 활용될 것으로 기대된다.

본 논문의 구성은 다음과 같다. 2장에서는 상호운용성 테스트와 관련된 기존 연구에 관해 기술하며, 3장에서는 웹 응용 시스템 구조를 기반으로 상호운용성 유지를 위해 요구되는 사항들을 분석한다. 그리고 4장에서는 상호운용성 향상을 위한 테스트 방법을 제시하며, 마지막으로 5장에서는 결론을 맺는다.

## II. 관련연구

컴퓨팅 환경의 발전으로 대기업들은 이기종 시스템간의 상호운용성 지원을 위해 많은 노력을 기울이고 있다.

특히 MS는 네트워크, 데이터, 애플리케이션, 관리 등과 같이 다양한 영역에서의 상호운용성을 위한 제품 개발에 힘쓰고 있으며, 최근에는 이러한 상호운용성 지원 환경에서 가능한 클라우드 서비스 시장에 MS, 구글, IBM 등의 대기업들이 많은 자본과 기술을 투입하고 있다[3].

과거에는 상호운용성 테스트에 관련된 연구가 주로 통신프로토콜 분야에서 이루어졌는데[4], 이는 네트워크 상에서의 상호운용성이 시스템 간의 상호운용성 유지를 위한 기본이 되기 때문이다. 이외의 또 다른 분야로는 전자상거래를 위한 ebXML 분야에서 서로 다른 서비스들 간 상호운용성에 관련된 연구들이다[5]. 이와 관련하여 국내에서는 한국정보통신기술협회 주관으로 국내 표준화 관련 6개 기관 및 유관 기관과 공동으로 여러 분야에 걸친 상호운용성 테스트 행사를 실시하고 있다.

웹 서비스 간에 높은 상호운용성을 유지하기 위해서는 WS-I[6]에서 정의한 메시지 교환에 관한 프로토콜을 준수해야 한다. WS-I는 플랫폼, 운영체제, 프로그래밍 언어간에 웹 서비스 상호운용성을 증진하기 위한 오픈 산업 조직으로서, 상호운용할 수 있는 웹 서비스를 개발하고 그 결과가 WS-I가 제안하는 표준을 따르는지를 증명하려는 웹 서비스 개발자를 위한 Profile, Sample application, Testing tool을 제공한다.

- Profile : 최적의 상호운용성을 위해서 관련 웹 서비스 명세를 사용하도록 구현 가이드라인을 제공
- Sample Applications : WS-I Profile을 따르는 웹 서비스 응용프로그램 샘플
- Testing tool : 웹 서비스와 상호 교환되는 메시지들이 WS-I Profile을 따르는지를 검사하는 도구

하지만 WS-I에서 제공하는 Basic Profile은 상당한 내용을 포함하고 있으며, 웹 서비스 간의 상호운용성을 고려하면서 서비스를 개발하기란 여간 어려운 일이 아니다. 따라서 다양한 서비스를 웹 상에 배치할 때에는 서비스 간 상호운용성을 유지하기 위한 별도의 상호운용성

테스트가 요구되며, 이를 위한 다양한 상호운용성 테스트에 관한 연구가 진행되고 있다[7-8].

상호운용성 테스트를 위한 항목들의 집합을 상호운용성 테스트 수트라 하는데 이는 소프트웨어 개발 과정에서 생산되는 시스템 명세, 설계 명세, 구현 소스 코드로부터 도출될 수 있다. 하지만 개발업체들은 구현 소스 코드를 공개하려 하지 않기 때문에 상세한 화이트 박스(White Box) 테스트가 불가능하다. 따라서 상호운용성 테스트를 위해서는 프로그램의 명세를 자세히 기술할 필요가 있다. 대부분의 테스트 자동화에 관한 연구들은 시스템 명세로부터 테스트 데이터를 생성하는 방식이며, 이를 위해서는 정형적인 명세 기술이 필수적이다.

이와 같이 네트워크 프로토콜 개발, 웹 서비스 등과 같이 각 기술들 혹은 분야별로 상호운용성을 위한 기술적인 내용들을 다루는 연구 논문들은 있지만, 현재까지 웹 응용시스템을 위한 상호운용성에 관한 연구는 거의 없었다. 특히, 대부분의 기존 연구들은 웹 응용프로그램 간의 상호운용성 테스트와 관련된 상세한 작업 절차 및 방법에 관련된 연구가 거의 없었기 때문에 이에 관한 연구는 중요하고 가치있는 연구라고 판단된다.

### III. 웹 응용프로그램 상호운용성 분석

#### 3.1 웹 응용프로그램의 구성 및 특징

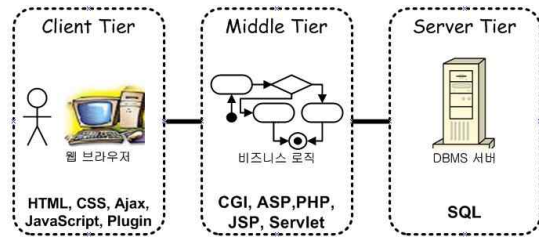
웹 응용프로그램(Web application)이란 인터넷과 같은 네트워크 환경에서 웹 브라우저를 통해서 접근할 수 있는 응용프로그램[9]을 의미한다. 상호운용성은 다양한 플랫폼으로 구성된 시스템(예를 들어, 클라이언트/서버 시스템, 분산 객체 시스템, 웹 시스템 등)에서 중요한 이슈로 등장한다[10]. 또한 시스템들이 점차 이기종 시스템들로 구성된 분산 환경에서 동작하는 형태로 발전하고 있으며, 반대로 각 시스템들이 제공하는 서비스를 결합해서 새로운 서비스로 활용하는 형태로 발전하기 때문에

상호운용성에 대한 중요성은 더욱 커지고 있다.

웹 응용프로그램은 다양한 플랫폼에서 개발 및 사용할 수 있으며, 많은 관련 기술들로 구성된다. 일반적으로 웹 응용프로그램은 <그림 1>과 같은 구조를 갖는다. 클라이언트 tier는 웹 브라우저에서 실행되는 내용들이며, 주로 HTML과 CSS를 통해서 사용자 인터페이스를 결정하며, 상호작용을 지원하기 위해서 Ajax, 자바스크립트, 플러그 인 등의 기술을 활용한다.

미들 tier는 실질적으로 비즈니스 로직을 구현하는 부분이다. 일반적으로 JSP 혹은 서블릿을 사용하는 경우에 미들 tier는 WAS라고 불리는 서버 응용프로그램에서 실행된다.

엔터프라이즈 tier(혹은 서버 tier)는 데이터와 관련된 부분으로서 DBMS 서버 혹은 레거시 시스템이 이에 해당한다. 엔터프라이즈 tier에는 엔터프라이즈 데이터를 저장 및 관리하는 DBMS 서버가 있거나 기존 엔터프라이즈 서비스를 관리하는 레거시 시스템이 존재한다.



<그림 1> 웹 응용프로그램 구조

<그림 1>에서 볼 수 있듯이 웹 응용프로그램이 3-tier 구조로 되어 있기 때문에 상호운용성 테스트도 각 단계별로 구분해서 진행되어야 한다.

#### 3.2 구성 요소별 상호운용성 분석

##### 3.2.1 웹 클라이언트 tier

웹 클라이언트 tier의 핵심은 웹 브라우저이며, 웹 브

라우저에서 활용되는 기술들로는 기능에 따라 <표 1>과 같이 분류할 수 있다. 웹 클라이언트 tier에서 상호운용성 측면의 문제점은 각 브라우저가 기본적인 표준 프로토콜 이외의 확장 기능들을 제공함으로써 발생한다. 이러한 결과로 일부 HTML 태그나 자바 스크립트는 특정 브라우저에서만 실행되기도 한다. 다음은 Ajax에서 자바 스크립트 표준을 따르지 않고 확장된 기능을 사용함으로써 발생하는 몇 가지 상호운용성 문제들이다.

1) XMLHTTP의 상이한 구현

Ajax에서 XMLHTTP는 HTTP 요청을 보내고 응답을 비동기로 처리할 수 있는 방법으로 현재 대부분의 브라우저(익스플로러, 파이어폭스, 오페라 등)에서 지원한다. 기본적으로 XMLHTTP는 현재 페이지가 놓은 호스트 안의 리소스에 대한 URL만 사용이 가능하다. 그런데 익스플로러에서는 브라우저의 보안 레벨을 낮춤으로써 다른 호스트로 요청을 보낼 수 있다. 하지만 크로스 브라우저 환경을 고려하면 XMLHTTP를 통해 다른 호스트로의 요청을 보내는 스크립트를 작성해서는 안 된다.

2) XML DOM과 HTML DOM의 비표준 확장 기능

XML DOM과 HTML DOM은 W3C의 표준이지만 브라우저 개발 업체가 제공하는 확장 기능에 의해 호환성의 문제가 발생한다. 예를 들어 익스플로러에서 지원하는 document.all과 innerText 자바스크립트 문법은 W3C DOM 표준에 포함되어 있지 않지만 자주 사용되는 기능들이다. 하지만 모든 브라우저에서 올바르게 동작하는 스크립트를 작성하기 위해서는 비표준 확장 기능을 사용하지는 안 된다.

이처럼 표준을 따르지 않고, 브라우저 고유의 기능을 사용하는 경우에 한번 개발된 웹 응용프로그램의 상호운용성은 떨어진다.

<표 1>은 웹 클라이언트 tier 기술과 상호운용성 이슈를 나타낸다. 이와 같이 클라이언트 측면에서 웹 응용프로그램의 상호운용성 문제를 해결하기 위해서는 브라

<표 1> 웹 클라이언트 tier 기술 및 상호운용성 이슈

분류	기술	이슈
플랫폼	웹브라우저	브라우저간의 호환성
프리젠테이션	HTML, CSS, 플래시, 플러그인	표준화, 브라우저 고유의 기능
제어 및 인터랙션	JavaScript, Ajax, 플래시, 플러그인	표준화, 브라우저 고유의 기능

우저와 상관없이 동작하는 표준화된 태그 및 스크립트를 사용해야 한다.

3.2.2 웹 미들 tier

웹 미들 tier는 비즈니스 로직을 수행하는 부분으로서 다양한 기술들이 사용된다. 대표적으로 많이 사용되는 것으로는 자바 기반의 JSP와 서블릿이 있으며, 그 이외에 ASP, PHP, CGI 등의 기술들이 사용된다. <표 2>는 웹 미들 tier에서 사용되는 기술들과 상호운용성 이슈이다.

<표 2> 웹 미들 tier 기술 및 상호운용성 이슈

분류	기술	이슈
플랫폼	J2EE, Microsoft. NET	각 응용프로그램 서버간의 호환성
언어	JSP, PHP, ASP, C, Perl	표준화, 레거시 시스템 통합

썬 마이크로시스템사의 J2EE는 컴포넌트 기반의 멀티 tier 엔터프라이즈 응용프로그램 개발을 위한 표준을 정의하고 있으며, 가용성, 보안성, 신뢰성, 확장성 구현을 위한 API 들을 제공하고 있다. 특히, J2EE 명세는 WS-I Basic Profile의 지원을 통해 웹 서비스 상호운용성을 보증한다.

J2EE는 윈도우, 유닉스, 리눅스 등에서 동작하며, 썬 마이크로시스템사는 벤더 간의 일관된 구현을 보증하기 위한 호환성 테스트 수트를 제공한다. 이러한 J2EE 명세를 구현하고 있는 웹 응용 서버에는 웹스피어, 제우스, 웹 로직 등이 있다.

웹 미들 티어 간에 발생하는 상호운용성 문제는 다음과 같이 요약된다.

1) 기본 데이터 타입의 매핑 문제

여러 개발 언어에서 동시에 지원되는 데이터 타입이라 할지라도 다른 플랫폼 상에서 올바르게 인식되지 않을 수 있다.

2) 전용 데이터 타입 문제

특정 개발 언어에만 지원되는 데이터 타입을 사용할 경우 상호운용성에 문제가 발생한다. 예를 들면, 엘리먼트의 집합을 표현할 때 사용되는 벡터 데이터 타입이 모든 언어에서 지원되는 것은 아니다.

3.2.3 엔터프라이즈 티어

엔터프라이즈 티어는 백엔드의 레거시 시스템이 존재하거나 비즈니스 데이터를 위한 DBMS 시스템이 존재한다. DBMS 구성은 단일 시스템도 있지만, 시스템이 점차 복잡해지면서 분산된 형태의 DBMS에서 데이터를 저장 및 관리하는 역할을 수행할 수 있다.

<표 3> 엔터프라이즈 티어 기술 및 상호운용성 이슈

분류	기술	이슈
플랫폼	DBMS, 레거시 시스템	DBMS간 호환성
언어	SQL, 프로그래밍 언어	DBMS에 따른 SQL 표준화

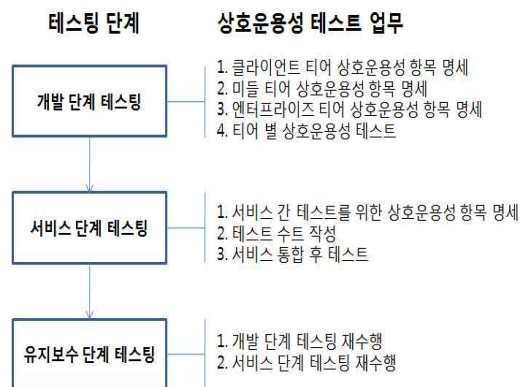
<표 3>은 엔터프라이즈 티어에서의 기술과 상호운용성 이슈를 나타낸 것이다. J2EE 플랫폼은 여러 기술과 API를 제공하는데 특히 Java Database Connectivity (JDBC)를 통해 이기종의 다양한 데이터베이스로의 접근과 트랜잭션을 지원한다. 따라서 J2EE 명세를 준수하는 웹 응용 서버는 다양한 DBMS로부터 데이터를 다룰 수 있다. 반면, 그 외의 레거시 시스템과의 연동을 위해서는 다양한 래퍼(Wrapper) 프로그램의 구현이 요구된다. 래

퍼는 웹 서비스를 실행시키기 위한 런타임 환경과 SOAP-RPC 요청에 대한 기존 레거시 시스템상의 함수 매핑을 담당하여 상호운용성을 지원한다.

IV. 웹 응용프로그램 상호운용성 테스트 방법

4.1 상호운용성 테스트 단계

대규모의 공공기관 및 대기업 등에서는 통합 서버를 통한 서비스를 제공할 때 반드시 상호운용성이 보장되어야 한다. 하지만 표준화 없이 구현된 통합된 서버들의 상호운용성 수준은 매우 낮기 때문에 요구된 서비스 결과를 보장하기가 매우 어렵다. 따라서 상호운용성 테스트는 시스템을 구축한 후에 기가하는 단기간의 과정이라기보다는 <그림 2>에서 보는 바와 같이 시스템 개발, 서비스, 유지보수의 전 부분에 포함되는 과정으로 진행되어야 한다. 각 단계 별 테스트 업무의 특징을 살펴보면, 개발 단계에서는 티어별로 상호운용성을 만족하는지 테스트하며, 서비스 단계에서는 서비스가 통합된 후 전체 테스트가 이루어지며, 마지막으로 유지보수 단계에서는 수정된 서비스에 대한 개발 단계 및 서비스 단계 테스트가 이루어진다.



<그림 2> 상호운용성 테스트 단계 및 단계별 업무

## 4.2 체크리스트와 단계별 테스트 업무

<표 4>는 개발 단계, 서비스 단계, 그리고 유지보수 단계에서 상호운용성 유지를 위해 요구되는 체크리스트 예제 항목들이다. 체크리스트의 테스트 결과는 도메인의 상호운용성 중요도 수준에 따라 Pass/Fail 혹은 상/중/하로 구분될 수 있다. 예를 들어 국방, 금융, 행정 분야와 같이 도메인이 광범위하며, 상호운용성이 중요한 분야에서는 Pass/Fail 등급만을 부여하도록 한다. 반대로 상호운용성 도메인 범위가 협소하거나, 상호운용성 확보에 소요되는 비용 및 시간이 부족한 상황, 기 개발된 시스템 및 응용프로그램의 수정이 매우 어려운 상황에서는 상/중/하 등급을 부여할 수 있다.

### 4.2.1 개발 단계 상호운용성 테스트 업무

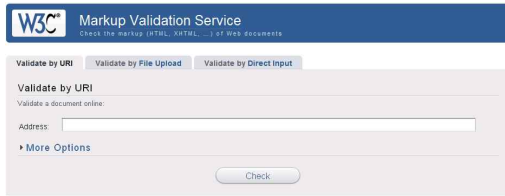
개발 단계 테스트에서는 상호운용성에 대한 요구 항목을 파악할 필요가 있다. 웹 응용프로그램의 장점은 PC에 응용소프트웨어를 설치하지 않고도 웹 응용프로그램을 사용할 수 있으며, 웹 응용프로그램의 유지 관리가 단지 웹 응용프로그램 서버에서만 이루어진다는 것이다. 아울러 여러 웹 응용프로그램 서버에서 실행되는 웹 응용프로그램 간의 상호운용을 통해 더욱 효과적인 서비스를 제공할 수 있다는 장점이 있다. 이러한 장점을 이용하기 위해서는 웹 응용프로그램이 다양한 브라우저에서 올바르게 동작해야 하기 때문에 이를 위한 상호운용성 요소들을 점검해야 한다.

클라이언트 tier에서는 사용자에게 일관된 뷰를 제공하기 위해 HTML, CSS, Javascript의 표준 및 확장 기능 목록들을 파악해야 한다. 그리고 기 작성된 문서들이 표준을 준수하고 있는지의 여부를 판단해야 하며, 응용프로그램 확장 기능을 지원하기 위한 다양한 브라우저들의

<표 4> 단계별 상호운용성 테스트 체크리스트

테스팅 단계	체크리스트	비고	테스트 결과
개발 단계	CSS 표준을 따르는가?	클라이언트 tier 상호운용성 요구사항	Fail/Pass
	HTML 표준을 따르는가?		Fail/Pass
	자바스크립트 표준을 따르는가?		Fail/Pass
	다양한 브라우저에 맞는 플러그 인을 제공하는가?		Fail/Pass
	다양한 브라우저상에서 일관된 결과를 생성하는가?		Fail/Pass
	이기종 플랫폼에 독립적인 표준 데이터 타입을 사용하는가?	미들 tier 상호운용성 요구사항	Fail/Pass
	통합하려는 레저시 시스템에 대한 래퍼가 존재하는가?	엔터프라이즈 tier 상호운용성 요구사항	Fail/Pass
이기종의 다양한 데이터베이스로의 접근과 트랜잭션을 지원하는가?	Fail/Pass		
서비스 단계	시스템 간 상호운용성 테스트를 위한 테스트 베드가 구축되었는가?	상호운용성 테스트를 위한 요구사항	Fail/Pass
	시스템 간 인코딩 방식이 동일한가?		Fail/Pass
	시스템 간 데이터 교환을 위해 XML을 사용하는가?		Fail/Pass
	데이터 교환 정의가 DTD나 스키마로 명세 되었는가?		Fail/Pass
	상호 간의 원활한 데이터 처리를 위한 공통의 파일 포맷을 지원하는가?		Fail/Pass
	여러 기관에서 제공하는 테스트 수트를 이용하여 상호운용성 테스트를 하였는가?		Fail/Pass
유지보수 단계	업그레이드 컴포넌트가 개발 단계 및 서비스 단계의 체크리스트를 통과하였는가?	상호운용성 테스트를 위한 요구사항	Fail/Pass
	업그레이드 컴포넌트의 단위별 상호운용성 테스트가 수행되었는가?		Fail/Pass
	업그레이드 시스템의 전체 상호운용성 테스트가 수행되었는가?		Fail/Pass

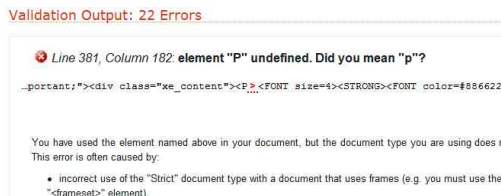
플러그인 제공 여부를 파악해야 한다. 이후 실제로 다양한 브라우저상에서 일관된 결과를 생성하는지에 대한 테



<그림 3> W3C에서 제공하는 웹 기반 유효성 검사기

스팅이 이루어져야 한다.

<그림 3>은 웹 표준 기구 W3C에서 제공하는 유효성 검사기이며[11], 이를 이용하면 기 작성된 문서가 HTML, XHTML 등의 표준을 따르고 있는지의 여부를 알 수 있어 상호운용성 테스트에 유용하게 사용할 수 있다.



<그림 4> 특정 페이지에서 발생한 유효성 검사 결과

<그림 4>는 유효성 검사기를 이용하여 기 작성된 문서의 표준 준수 여부를 검사한 결과로써 현재 페이지에 존재하는 에러 수와 해당 오류에 대한 상세한 설명을 보여주고 있다. 클라이언트 tier 측의 일관성을 보장하기 위해서 테스트 과정에서는 검사기에 명시된 에러 목록들을 참조하여 표준을 준수하는 문서로 수정한다.

미들 tier에서는 플랫폼 간에 발생하는 데이터 타입의 비호환성 문제를 해결하기 위해 각 플랫폼에서 지원 가능한 데이터 타입과 비지원 데이터 타입 목록들을 파악한 후, 실제 구현 코드에서 이기종 플랫폼에 독립적인 표준 데이터 타입을 사용하는지 확인하여 올바른 데이터가 생성되도록 한다.

마지막으로 엔터프라이즈 tier에서는 통합하려는 레

저서 시스템들의 레퍼 목록들과 이기종의 데이터베이스 접근과 트랜잭션을 보장하기 위한 드라이버 목록들을 확인한다. 특히, 레저서 시스템에 대한 레퍼의 구현은 상당한 시간과 비용이 요구되기 때문에 공개된 소프트웨어나 COTS 제품을 도입하여 상호운용성 테스트를 수행하는 것이 좋다[12].

이와 같이 개발 단계에서는 각 tier별로 상호운용성에 요구되는 목록들을 도출하고 이를 준수하는 프로그램을 작성해야 하며, 사용자에게 일관된 뷰와 정확한 결과를 보증하기 위한 단위별 테스트가 이루어져야 한다. 이러한 과정은 요구되는 상호운용성 조건을 만족할 때까지 반복적으로 수행된다.

#### 4.2.2 서비스 단계 상호운용성 테스트 업무

서비스 단계에서의 상호운용성 테스트를 위해서는 테스트 환경에서 요구되는 목록들을 확인한다. 기본적으로는 서비스 시스템 간의 원활한 데이터 교환을 위해 인코딩 방식과 XML, DTD, 스키마, 표준 파일 포맷 사용 여부를 확인해야 한다.

이후 서비스 단계 테스트에서는 개발 단계에서 도출된 미들 tier 상호운용성 항목 명세에 따라 테스트 수트를 작성한다. 테스트 수트 작성은 명세서를 기반으로 도출할 수 있지만 상세한 수준까지는 기대할 수 없다. 다시 말해 상세한 수준의 테스트 수트는 소스코드를 기반으로 생성할 수 있지만 소스코드의 확보는 개발자 측의 동의에 달려있기 때문에 블랙박스(Black Box) 테스트 수준으로 제한된다. 따라서 효과적인 웹 응용 시스템 통합을 위해서는 개발자의 책임이 매우 중요하다 할 수 있다.

통합 서비스를 제공하는 공공기관이나 대기업에서는 개발자에게 미들 tier간의 상호운용성을 위해 플랫폼에 독립적인 구현 및 매크로들을 요구한다. 아울러 앞서 소개했듯이 썬 마이크로시스템사는 상호운용성 테스트를 위한 테스트 수트를 제공하며, WS-I에서는 상호운용성 테스트 도구들을 제공하기 때문에 서비스 단계 테스트에

서는 이를 이용하여 상호운용성을 검증할 수 있다.

#### 4.2.3 유지보수 단계 상호운용성 테스트 업무

유지보수 단계 테스트에서는 각 시스템을 유지 보수 함으로써 훼손될 수 있는 상호운용성을 테스트해야 한다. 유지보수 단계에서는 새로운 서비스가 추가되거나 기존의 서비스가 수정될 수 있기 때문에 상호운용성 테스트를 통해 검증된 시스템 간에서도 문제가 발생할 수 있다.

만일 시스템의 업그레이드가 요구된다면, 안정적인 서비스 제공을 위해 업그레이드 컴포넌트에 대한 개발 단계 테스트가 선행되어야 한다. 이 때 업그레이드된 컴포넌트 전체에 대한 테스트를 수행할 경우, 상호운용성에 문제가 되는 컴포넌트 식별이 매우 어려워지기 때문에 단위별 테스트 후 통합 방법을 선택하는 것이 효과적이다. 또한, 상호운용성 문제가 발생할 것을 대비하여 기존에 구현된 서비스를 지속적으로 제공하면서 별도의 테스트 베드 상에서 유지보수 테스트가 이루어져야 한다.

## V. 결론

시스템이 복잡해지면서 상호운용성 테스트의 중요성은 더욱 커지고 있다. 이에 따라 공공기관과 대기업을 중심으로 상호운용성을 위한 테스트 항목을 별도로 도입하고 있으며 그 관심은 매우 높아지고 있다.

본 논문에서는 웹 응용시스템 환경에서 상호운용성을 위해서 점검해야 할 항목들과 상호운용성 점검을 위한 절차 및 방법에 대해 소개하였다. 그리고 웹 응용시스템은 3-tier 형태의 클라이언트/서버 구조를 갖기 때문에 각 부분에 대해서 기술적인 항목에 대해서 고려해야 할 상호운용성 항목들을 소개하였다. 또한 테스트를 개발, 서비스, 유지보수 단계로 구분하였으며, 각 단계에서 진행해야 할 상호운용성을 위한 작업 내용들을 제시하였다.

본 논문은 향후 상호운용성을 점검해야 하는 테스트 작업 및 테스트 업무를 효과적이고 체계적으로 수행하는데 큰 도움을 줄 수 있으며, 최근 이슈가 되고 있는 서비스 지향 아키텍처에서의 상호운용성 테스트를 위한 연구에 활용될 것으로 기대된다.

## 참고문헌

- [1] <http://www.microsoft.com/korea/technet/interop/ndam.msp>
- [2] 이상학 외, *상호운용성 기반 확보를 위한 정보시스템 구조 표준화 방안 연구*, 한국정보사회진흥원, 2009, 1.
- [3] 안원호, *클라우드 서비스 활성화를 위한 정책방향*, TTA Journal, 2009.
- [4] Soonuk Seol, Myungchul Kim, Sungwon Kang, and Jiwon Ryu, "Fully automated interoperability test suite derivation for communication protocols," *Computer Networks*, Vol. 43, No. 6, Dec., 2003, pp. 735-759.
- [5] Peng Liang, Keqing He, Bing Li, and Jin Liu, "Interoperability Test of ebXML e-Business Solutions," *Proceedings of the International Conference of Computer Information Technology (CIT2004)*, 2004, pp. 1004-1007.
- [6] WS-I Deliverables, available at <http://www.ws-i.org/deliverables/matrix.aspx>
- [7] A. Bertolino and A. Polini, "The audition framework for testing web services interoperability," In *Proceedings of the 31st EUROMICRO International Conference on Software Engineering and Advanced Applications*, 2005, pp. 134-142.
- [8] G. Canfora and M. D. Penta, "Service-Oriented



Architectures Testing: A Survey”, LNCS 5413, 2009, pp. 78-105.

- [9] [http://en.wikipedia.org/wiki/Web\\_application](http://en.wikipedia.org/wiki/Web_application)
- [10] James D. Kindrick, John A. Sauter, and Robert S. Matthews, “Improving Conformance and Interoperability Testing,” StandardView, Vol. 4. No. 1, Mar, 1996, pp. 61-68.
- [11] Markup Validation Service, available at <http://validator.w3.org/>
- [12] 박병호, 강윤희, “웹서비스 기반 국방 정보체계 구성”, 한국정보기술학회, 정보기술학회 논문지 제3권, 제 6호, 2005, pp. 1-8.

논문접수일 : 2010년 1월 18일
수정일 : 2010년 2월 15일
게재확정일 : 2010년 2월 25일

■ 저자소개 ■



김 익 수  
Kim, Ik Su

2009년 9월~현재  
 송실대학교 컴퓨터학부 조교수  
 2008년 2월 송실대학교 컴퓨터학과 (공학박사)  
 2002년 2월 송실대학교 컴퓨터학과 (공학석사)  
 2000년 2월 송실대학교 컴퓨터학과 (공학사)  
 관심분야 : 시스템 보안, 네트워크 보안, 모바일  
 보안, 시스템 소프트웨어  
 E-mail : skycolor@ss.ssu.ac.kr



최 종 명  
Choi, Jong Myung

2004년 3월~현재  
 국립목포대학교 컴퓨터공학과 교수  
 2003년 8월 송실대학교 컴퓨터학과 (공학박사)  
 1996년 8월 송실대학교 전자계산학과  
 (공학석사)  
 1992년 2월 송실대학교 전자계산학과 (공학사)  
 관심분야 : 프로그래밍 언어, 유비쿼터스  
 컴퓨팅, 컨택스트-인지 시스템  
 E-mail : jmchoi@mokpo.ac.kr