

다중 Application을 위한 Web Services 상호 운용성에 관한 연구

유윤식* · 송종철* · 최일선* · 임산송* · 정희경*

*배재대학교 컴퓨터공학과

Study of Web Services Interoperability for Multiple Applications

Yoon-Sik Yoo* · Jong-Cheol Song* · Il-Sun Choi* · San-Song Lim* · Hoe-Kyoung Jung*

*Dept. of Computer Engineering Paichai University

E-mail : {ys5315, choiilsun}@hanmail.net, jcsong@iita.re.kr, ssim@jungbu.ac.kr, hkjung@mail.pcu.ac.kr

요 약

웹의 이용이 급속하게 증가함에 따라, 웹 기반의 어플리케이션 대 어플리케이션간의 상호작용을 체계적으로 지원하는 적절한 모형과 새로운 분산 플랫폼을 기존 환경들과 효율적으로 통합할 수 있는 방안에 대한 요구가 발생하였으며 이에 대한 해결책으로 웹 서비스가 등장하였다.

현재 많은 소프트웨어와 하드웨어 업체들이 웹 서비스를 시장에 도입하려 하는 적극적인 시도를 보이고 있으며 어플리케이션들을 다양한 웹 서비스 제공자들로부터의 컴포넌트를 조합하여 구축하려고 하고 있다. 하지만 웹 서비스를 완벽히 이행하기 위해서는 반드시 상호 운용되어야 하며 다른 업체들로부터의 서비스뿐만 아니라 플랫폼, 어플리케이션과 프로그래밍 언어에 종속되지 않도록 하는 일련의 표준화 작업이 필요하다. 이에 웹 서비스 상호 운용성 표준화 기구인 WS-I(Web Services Interoperability Organization)에서는 XML, UDDI, WSDL 그리고 SOAP을 기본으로 웹 서비스 상호 운용을 위한 표준인 기본 프로파일(Basic Profile 1.0)을 제정하였으며 실제 적용하기 위한 웹 서비스 사용자 시나리오의 프로파일을 개발하고 있다.

이에 본 논문에서는 이질적인 두 어플리케이션간의 웹 서비스 상호 운용의 적합성을 검증하기 위하여 도서 정보 검색 웹 서비스를 J2SE 플랫폼의 웹 서비스 클라이언트와 .NET 플랫폼의 웹 서비스 서버를 기반으로 각각 설계 및 구현하고 서로 서비스가 이루어지도록 하여 이질적인 두 플랫폼 사이에서 상호 운용이 가능하다는 것을 보이며 두 어플리케이션 간의 전송 서비스를 WS-I 기본 프로파일을 적용하여 테스트 및 검증하였다.

ABSTRACT

According as utilization for web increases rapidly, it is demanded that model about support interaction between web-based applications systematically and solutions can integrate new distributed platforms and existing environment effectively, accordingly, Web Services appeared by solution in reply.

These days, a lot of software and hardware companies try to adoption of Web Services to their market, attempt to construct their applications associating components from various Web Services providers. However, to execute Web Services completely, it must have interoperability and need the standardization work that avoid thing which is subject to platform, application as well as service and programming language from other companies. WS-I (Web Services Interoperability organization) have established Basic Profile 1.0 based on XML, UDDI, WSDL and SOAP for Web Services interoperability and developed Usage Scenario Profile to apply Web Services in practice.

In this paper, to verify suitability Web Services interoperability between heterogeneous two applications, have design and implements the Book Information Web Services that based on the Web Services Client of J2SE platform and the Web Services Server of .NET platform, so that analysis and verify the service by adaptation of WS-I Basic Profile.

키워드

웹 서비스, 어플리케이션, 상호 운용성

1. 서론

웹의 이용이 급속하게 증가함에 따라, 웹 기반의 어플리케이션들 사이의 상호운용성을 체계적으로 지원하는 적절한 모형과 새로운 분산 플랫폼을 기존 환경들과 효율적으로 통합할 수 있는 방안에 대한 요구가 발생하였으며 이에 대한 해결책으로 웹 서비스가 등장하였다.

이미 존재하는 분산 컴퓨팅 기술 즉, 컴포넌트 기반 기술인 DCOM, CORBA, JAVA RMI들을 상호 운용성 측면에서 살펴보면 지극히 연결 지향적이고 표준화되지 않은 통신 포트를 사용하며, 보안 메커니즘 또는 방화벽으로 인하여 접근이 불가능해 질 수도 있다. 또한 JAVA RMI와 DCOM은 플랫폼에 독립적이지 못하여 서로 통신하기에는 많은 어려움이 따른다.

이에 대한 해결책으로 웹 서비스가 등장하였으며 상호 운용성 촉진을 위해 IBM과 마이크로소프트를 주축으로 전 세계의 46개 업체가 모여 만든 웹 서비스 상호 운용성 표준화기구인 WS-I(Web Services Interoperability Organization)가 설립되었고 현재 140여 개의 IT 업체들이 참여하고 있다[1].

WS-I에서는 2003년 3월 WS-Basic Profile 1.0을 발표하였고 이 문서에는 기본적으로 웹 서비스를 서로간에 통신하는 데 있어 기본이 되는 요소들을 각 웹 서비스 명세를 적용하여 설명하고 있다[2].

따라서, 이질적인 두 어플리케이션간의 웹 서비스 상호 운용성의 적합성을 검증하기 위하여 WS-I 기본 프로파일을 분석하고 이에 대한 테스트베드로서 J2SE 플랫폼의 웹 서비스 클라이언트와 .NET 플랫폼의 웹 서비스 서버를 기반으로 각각 설계 및 구현하고 서로 서비스가 이루어지도록 하였고 제공하기 위한 서비스로는 쉬운 사용자의 쉬운 이해를 위해 일상생활에서 접할 수 있는 도서 정보 검색 시스템을 구현하여 이질적인 두 플랫폼 사이에서 상호 운용성이 가능하다는 것을 보였으며 두 어플리케이션 간의 전송 서비스를 WS-I에서 제공하는 테스트 도구를 이용하여 WS-I 기본 프로파일에서 권고하는 웹 서비스 상호 운용을 위해 지켜야 할 규칙에 대한 적합성 여부를 검증하였다.

II. 관련 연구

2.1 웹 서비스 아키텍처

웹 서비스는 SOAP, UDDI, WSDL 등과 같은 개방형 표준 기술을 기반으로 서비스 제공자(Service Provider)와 서비스 저장소(Service Registry), 서비스 요청자(Requester) 등이 유기적으로 결합하여 구현된다[3].

그림 1은 웹 서비스가 작동하는 원리를 개념적으로 나타내고 있다. 먼저 서비스 제공자는 자신이 제공하는 서비스에 대한 목록 정보를 서비스 레지

스트리 (Service Registry)에 등록(publish)을 한다. 다음으로 서비스를 요청하는 프로그램에서는 해당 서비스의 정보를 수집하기 위해 인터넷 디렉토리 시스템을 사용한다. 보통 URL을 통해 웹 자원을 검색하듯 이러한 디렉토리 시스템은 인터넷 곳곳에 산재하여 있으며 이를 서비스 레지스트리라고 부른다. 여기에서 "검색"이란 서비스 요청자가 원하는 서비스의 내용을 탐색하고 필요한 정보를 획득하는 동작을 말한다. 서비스 레지스트리를 통하여 원하는 서비스의 존재 여부와 위치, 그리고 호출에 필요한 각종 정보를 얻은 서비스 요청자는 서비스 제공자와 직접적인 통신을 시도한다. 따라서 서비스 제공자가 서버를 다른 기계로 옮기든 서비스 로직이 바뀌든 간에 서비스 요청자 입장에서는 서비스의 내용만 이해하고 있다면 별다른 수정 없이 프로그램 스스로 상호 통신이 가능하게 된다.

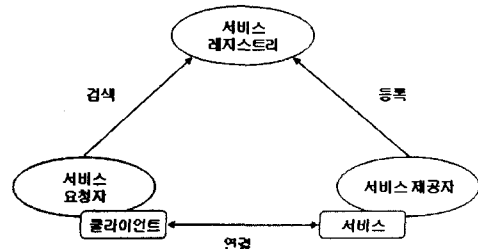


그림 1. 웹 서비스 아키텍처

2.2 WS-I 테스트 도구

WS-I는 기본 프로파일의 요소를 상호 운용성을 테스트하기 위한 도구로서 Monitor와 Analyzer를 제공한다[4].

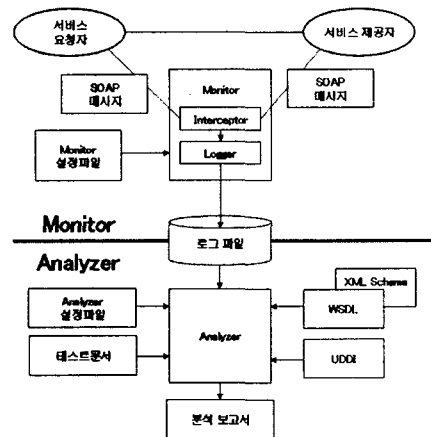


그림 2. WS-I 테스트 도구 구조도

그림 2와 같이 Monitor는 웹 서비스의 서버와 클라이언트의 중간에 위치하여 Monitor 내부의

Interceptor를 사용하여 교환되는 메시지를 모니터링하고 메시지 명세의 합리성을 평가하며, Logger를 이용하여 메시지 로그 파일이 생성된다. Analyzer는 생성된 로그 파일을 이용하여 웹 서비스의 WSDL과 UDDI를 기준으로 웹 서비스가 상호 운용되는지를 분석하며, 분석 보고서를 WS-I 기본 프로파일 1.0을 기준으로 작성하고 오류를 상세하게 기록한다.

III. 상호 운용성을 위한 웹 서비스 설계

웹 서비스 상호 운용성 테스트를 위하여 웹 서비스 제공자와 클라이언트를 각각 .NET과 J2SE 기반으로 설계하였고 이로써 상호 이질적인 어플리케이션간에 웹 서비스가 연동하도록 하였다. 테스트를 위한 서비스로는 사용자의 쉬운 이해를 위해 일상생활에서 접할 수 있는 도서관에서 도서 정보를 검색하여 그 결과를 얻는데 이용될 수 있는 도서 검색 웹 서비스를 설계 하였다.

본 시스템은 J2SE 기반의 도서 정보 검색 클라이언트를 구축하고 이를 이용하여 검색할 도서의 정보를 입력하여 서버의 .NET 기반의 도서 정보 검색 웹 서비스를 호출하며 이 웹 서비스는 서버의 데이터베이스로부터 도서 정보를 검색하여 웹 서비스 클라이언트로 다시 데이터를 전송하도록 한다.

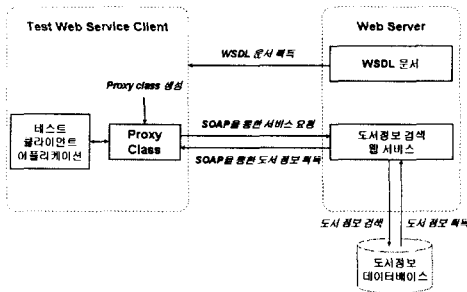


그림 3. 전체 시스템 구성도

그림 3은 전체 시스템 구성도 이다. 웹 서비스 서버와 클라이언트간의 모든 데이터 전송 즉, 클라이언트에서의 웹 서비스 호출과 웹 서비스에서 클라이언트로의 데이터 전송은 HTTP 상의 SOAP으로 이루어지며 웹 서비스 클라이언트는 서버의 웹 서비스에 의해 생성된 WSDL을 기준으로 구축하여 .NET 웹 서비스와 J2SE 기반의 웹 서비스 클라이언트간의 상호 운용을 보이도록 설계하였다.

IV. 구현 및 검증

4.1 .NET 웹 서비스 구현

웹 서비스의 구현 환경은 IBM-PC 호환 컴퓨터 (Pentium IV-2.4G)에서 개발하였으며, Windows XP Professional 운영체제에서 Microsoft Visual C# .NET을 사용하여 구현하였고, 개발 도구는 Microsoft Visual Studio .NET 2003을 사용하였다.

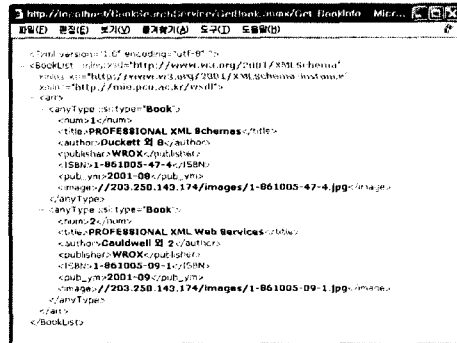


그림 4. 파싱된 XML 결과문서

그림 4는 실제로 웹 서비스를 호출하여 응답 SOAP 메시지를 구문 분석하고 도서 정보 목록 부분만을 추출하여 얻어낸 XML 문서이다. 가장 상위 엘리먼트는 도서 정보 목록을 가지고 있는 BookList이며, 그 하위로 반환 값이 배열형식이란 의미를 가진 arr 엘리먼트, 그리고 그 하위로는 배열로써 반환된 도서목록이 2개의 값을 가지며 이 목록 안에는 도서 정보인 num, title, author, publisher, ISBN, pub_ym, image 엘리먼트들을 각각 가지고 있다.

4.2 J2SE 웹 서비스 클라이언트 구현

J2SE 웹 서비스 클라이언트의 구현 환경은 IBM-PC 호환 컴퓨터 (Pentium IV-2.4G)에서 개발하였으며, Windows XP Professional 운영체제에서 개발 언어는 JDK(Java Development Kit) 1.4.1을 사용하여 구현하였고, 그리고 WSDL로부터 프록시 및 스킴레톤을 구축을 위하여 Apache 그룹의 Axis 1.0을 사용하였다[5, 6]. 그림 5는 구현된 클라이언트 화면이다.

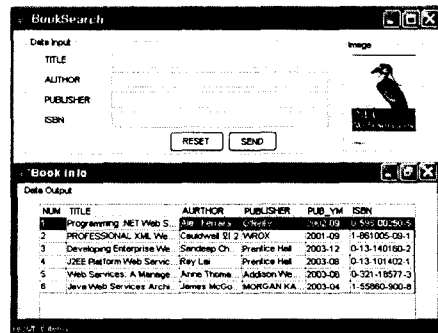


그림 5. J2SE 클라이언트

클라이언트의 좌측 상단에는 도서 검색 키워드 입력을 위한 창을 두어 제목, 저자, 출판사, ISBN으로 검색을 할 수 있도록 하였고 검색한 결과는 하단에 내부 프레임 형식으로 출력되도록 구성하여 여러 검색 결과를 비교 할 수 있도록 하였다. 또한 우측 상단에 이미지 보기 창을 두어 검색된 도서의 이미지를 볼 수 있도록 구성하였다.

4.3 웹 서비스 상호 운용성 테스트

구현한 .NET 웹 서비스 서버와 J2SE 웹 서비스 클라이언트 간의 상호 운용성의 테스트는 WS-I Testing Tool 1.0 Java 버전을 사용하였다[7]. 테스트 도구는 Monitor와 Analyzer로 구성되며 이는 그림 6과 같이 웹 서비스 서버와 클라이언트 사이에 위치하게 된다.

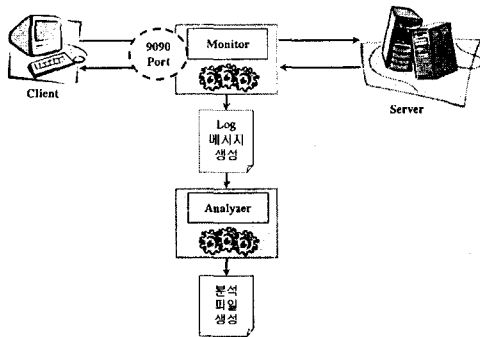


그림 6. 상호 운용성 테스트 구조도

WS-I Analyzer는 기본 파일 적용 규칙 별로 적용성의 단계 별로 Passed, Not Applicable, Warning, Failed, Missing Input으로 나뉜다.

WS-I 기본 프로파일 1.0에서는 메시지에 관련된 규칙을 총 40개의 항목으로 권고하고 있으며 본 논문에서 구현한 도서 정보 검색 웹 서비스 서버와 클라이언트 간에 전송된 메시지를 분석한 결과는 Passed 40 항목으로 요구되는 항목을 모두 만족한 결과를 보였다.

V. 결 론

현재 많은 소프트웨어와 하드웨어 업체들의 XML, SOAP, WSDL과 UDDI와 같은 일반 표준의 포괄적인 도입이 나타내는 바와 같이 웹 서비스를 시장에 도입하려 하는 적극적인 시도를 보이고 있다. 하지만 웹 서비스를 완벽히 이행하기 위해서는 반드시 상호 운용성이 존재하여야 하며 다른 업체들로부터의 웹 서비스뿐만 아니라 플랫폼, 어플리케이션과 프로그래밍 언어에 종속되지 않도록 하는 일련의 표준화 작업이 필요하다. 이에 웹 서비

스 상호 운용성 표준화 기구인 WS-I에서는 상호 운용을 위한 표준을 제정하고 있으며 실제 적용하기 위한 웹 서비스 사용자 시나리오의 프로파일을 개발하고 현재 이들 어플리케이션 프로파일을 개발하기 위한 프로세스를 규정 중에 있다.

이에 본 논문은 이질적인 두 어플리케이션 기반의 웹 서비스와 웹 서비스 클라이언트를 구축하여 서비스를 전송하고 메시지를 분석하는 웹 서비스 상호 운용성의 적합성 검증에 관한 것으로 .NET 기반의 웹 서비스와 J2SE 기반의 웹 서비스 클라이언트를 설계 및 구현하고 두 어플리케이션 간의 전송 서비스를 WS-I 기본 프로파일을 적용하여 테스트 및 검증하였다.

WS-I의 메시지 전송에 관한 규칙 사항 총 40항목을 만족하였으며 이로써 .NET과 J2SE를 기반으로 하는 웹 서비스 전송간에는 상호 운용이 가능함을 보였다.

본 연구 결과는 웹 서비스를 구현 및 소비하는데 있어 상호 이질적인 어플리케이션간 인터페이스의 기술에 표준화된 기술을 적용함으로써 효율적인 상호 운용성을 확보하는 테스트 베드가 될 것이며 다양한 기술들이 적용된 웹 서비스 시스템들 간에 효율적으로 정보 및 서비스를 연계하고 활용하기 위한 중요한 기반 기술이 될 것으로 사료된다.

향후 연구되어야 할 과제로서, 최근 제기되고 있는 웹 서비스통신 프로토콜인 SOAP 메시지 포맷 불일치 문제와 Apache SOAP, SOAP Lite, MS SOAP Toolkit과 같은 각 업체별로 제공하는 SOAP 메시지 저작도구에 의해 생성된 메시지의 버전에 대한 불일치로 인하여 일어날 수 있는 웹 서비스 상호 운용성 문제를 해결하여 향후 웹 서비스가 본격화 되어 이용되는 시점에서 상호 운용성의 문제를 극복하기 위한 기반 기술로서의 연구가 진행 되어야 할 것이다.

참고문헌

- [1] Web Services Interoperability Organization, <http://www.ws-i.org>
- [2] WS-I, Basic Profile Version 1.0a, <http://www.ws-i.org/Profiles/Basic/2003-08/BasicProfile-1.0a.html>
- [3] W3C, Web Services Architecture, <http://www.w3.org/TR/ws-arch>
- [4] WS-I, Implementation Tools, <http://www.ws-i.org/implementation.aspx>
- [5] Apache XML Project, Xerces 2Java Parser, <http://xml.apache.org/xerces2-j>
- [6] Apache Web Services Project, Web Services-Axis, <http://ws.apache.org/axis>
- [7] WS-I, Testing Tool TAD, <http://www.ws-i.org/Testing/Tools/2004/01/BasicProfileTestAssertions.xml>