

---

# REST 웹 서비스를 위한 WADL2Stub 시스템의 구현

이화민\*

## An Implementation of WADL2Stub System for REST Web Services

Hwa-Min Lee\*

### 요 약

최근 Web 2.0의 등장과 함께 SOAP 기반의 웹 서비스에 비해 가볍고 구현이 쉬운 REST 웹 서비스가 많은 주목을 받고 있다. REST 웹 서비스의 API를 정의하고 기술하기 위한 다양한 방법들이 제안되었고, WSDL도 2.0 버전에서 SOAP 뿐만 아니라 REST도 기술할 수 있게 확장되었지만 너무 복잡하다는 문제점이 있었다. 이에 썬 마이크로시스템은 WSDL에 비해 간략하면서도 범용성이 뛰어난 WADL을 발표하게 된다. WADL의 목적은 Web 2.0 어플리케이션을 보다 쉽게 생성하고 동적으로 관리할 수 있도록 인터넷 상의 웹 서비스들을 기계가 처리할 수 있는 방법으로 기술하는 것이다. 그러나 WADL의 경우 자동으로 클라이언트 코드를 생성해주는 툴이 없어 개발자들이 클라이언트 코드를 직접 생성해야 하는 문제점이 있다. 이에 본 논문에서는 개발자들이 WADL 기반의 REST 웹 서비스를 쉽고 편리하게 이용할 수 있도록 클라이언트 코드를 생성하는 WADL2Stub 시스템을 설계하고 구현하였다.

### ABSTRACT

Recently, RESTful web service is getting attention with the emergence of Web 2.0. The main advantages of REST web services are lightweight and easy implementation compared to SOAP based web services. In order to define and describe RESTful APIs, various methods have proposed. WSDL 2.0 can describe both SOAP-based and RESTful services and but it is too complicated. WADL is championed by Sun Microsystems and it is lightweight, easier to understand and easier to write than WSDL. The purpose of WADL is to allow web services to be described in a machine readable way, to make it easy to create Web 2.0 applications and create a dynamic way of creating and configuring services. However, WADL does not provide utility tools to generate client code automatically and the developers describe client code manually. In this paper, we design and implement a WADL2Stub system to generate client code from WADL files automatically so that the developers use REST web services to easily and conveniently.

### 키워드

WADL, REST Web Service, Client Code, Utility Tools

## I. 서 론

웹 서비스는 서비스 지향적인 분산 컴퓨팅 기술의 하나로, 네트워크 상에서 서로 상이한 컴퓨터들 간에 상호작용을 하기 위한 소프트웨어 시스템이다[1]. 웹 서비스 프로토콜 스택은 XML-RPC (Extensible Markup Language Remote Procedure Call)[2], SOAP(Simple Object Access Protocol)[3], UDDI(Universal Description, Discovery and Integration)[4], WSDL(Web Services Description Language)[5] 등으로 구성된다. WWW은 사람과 컴퓨터간의 상호작용을 위한 시스템인데 반해, 웹 서비스는 컴퓨터와 컴퓨터 간의 상호작용을 위한 시스템이다. XML-RPC의 확장 또는 발전형이라고도 할 수 있는 SOAP 을 통한 웹 서비스는 XML 메시지 기반의 프로토콜이었기 때문에 플랫폼 독립적이고 기술 습득에 용이하다는 장점 때문에 많은 인기를 끌었다. SOAP의 등장 이후 이를 이용하여 공개된 웹 서비스가 등장하였으나, 표준화된 기술 방법이 존재하지 않았기에 이를 사용하기 위해서는 일일이 각 서비스의 설명을 참고해야 했다. 따라서 인터페이스를 기술하기 위한 표준이 필요하게 되었고, 이에 따라 XML 기반의 WSDL이 등장하였다. 이후 WSDL을 분석하여 Stub/Skeleton code를 자동으로 생성해주는 여러 툴들이 등장하였고 이는 웹 서비스를 더욱 사용하기 쉽게 해주었다. SOAP를 이용한 웹 서비스는 기존의 분산 객체 기술인 CORBA(Common Object Request Broker Architecture), DCOM(Distributed Component Object Model), Java RMI(Remote Method Invocation) 등의 대안으로, 트랜잭션, 보안, 정책(policy), 신뢰성 있는 메시징 등의 다양한 요구사항을 지속적으로 추가하여 특히 엔터프라이즈 환경에서 각광받았다.

그러나 2003년 Web 2.0의 바람과 함께 사용자들은 Web을 통해 상호작용할 수 있는 보다 간단한 기술을 원하기 시작했다. 웹 2.0은 단순한 웹 사이트 집합체에 불과했던 웹에서 벗어나 웹 어플리케이션을 제공하는 하나의 완전한 플랫폼으로 동적인 웹을 의미한다. SOAP은 XML을 사용하여 상호운용성은 높지만 초기의 간단하고 사용하기 쉬운 장점은 복잡한 확장 표준에 의해 그 의미가 희석되어갔다. 사람들은 다시 간단한 것을 원했고, Web 2.0의 등장과 함께 엔터프라이즈 환경에서의 요구사항들은 충족시키지 못하지만 HTTP의 기본 기능인

GET, POST 등을 적극 활용하고 XML 형식의 메시지 전송도 지원하는 REST(Representational State Transfer)[6] 웹 서비스가 대두되었다[7, 8, 9]. 웹 2.0과 함께 SOAP를 이용하지 않는 웹 서비스들을 REST 스타일 웹 서비스라고 하며, 이러한 REST 스타일 웹 서비스들은 현재 많은 주목을 받고 있다. 현재 Yahoo, Flickr, del.icio.us의 모든 웹 서비스들은 REST를 사용하고 있으며, eBay와 Amazon은 SOAP 기반의 웹 서비스와 함께 REST 웹 서비스를 제공하고 있다. 특히 아마존의 경우 OPEN API의 85%를 REST로 지원하고 있다.

REST 웹 서비스가 개발자들 사이에서 각광을 받으면서 REST 웹 서비스의 인터페이스를 기술하기 위한 다양한 방법들이 제안되고 있다. REST API를 기술하기 위한 언어들도 여러 가지 있지만, 아직까지 주류가 형성되지 않은 상황이다. 기존의 WSDL도 2.0 버전에서 SOAP은 물론 REST도 기술할 수 있게 확장 되었지만 복잡성 때문에 많이 쓰이지는 못하고 있다. 이에 썬 마이크로시스템은 간략하면서도 범용성이 뛰어난 WADL(Web Application Description Language)[10]를 발표하였다. WADL의 장점으로 인해 개발자들 사이에서 WADL의 사용은 점차 증가하고 있으며, Google의 경우 REST Describe & Compile[11]이라는 이름으로 WADL를 이용하는 프로젝트를 진행 중이다. 그러나 WADL의 경우 WSDL처럼 WADL 문서를 분석하여 자동으로 클라이언트 코드를 생성해주는 툴이 없어 개발자들이 직접 클라이언트 라이브러리를 만들어야 하는 문제점이 있었다. 이에 본 논문에서는 WADL 문서를 분석하여 해당 문서가 기술하고 있는 웹 서비스를 이용할 수 있도록 자동으로 다양한 프로그래밍 언어로 클라이언트 코드를 생성하는 WADL2Stub 시스템을 설계하고 구현하였다.

본 논문은 모두 5장으로 구성되며, 각 장의 내용은 다음과 같다. 1 장에서는 본 논문의 연구 배경 및 목적에 대해서 기술한다. 2 장에서는 본 논문과 관련된 REST 웹 서비스와 WADL에 대해 설명한다. 3 장과 4 장에서는 WADL2Stub 시스템의 설계 및 구현에 대해서 기술한다. 마지막으로 5 장에서는 결론 및 향후 연구과제에 대해 설명한다.

## II. 관련연구

본 장에서는 REST Web Services와 이를 기술하는 WADL에 관한 관련 기술을 설명한다.

### 2.1 REST

웹 서비스의 새로운 패러다임으로 등장한 REST는 WWW과 같은 분산 하이퍼미디어 시스템을 위한 소프트웨어 아키텍처의 한 형식이다. REST는 2000년에 HTTP의 주요 저자들 중 한 명인 Roy Fielding의 박사 학위 논문[6]에서 소개되었다. 엄격한 의미에서 REST는 자원을 정의하고 자원에 대한 주소를 지정하는 방법에 대한 네트워크 구조 원리들의 모음이다[7]. 즉, 도메인 지향 데이터를 HTTP에서 SOAP이나 쿠키를 통한 세션 트래킹 같은 부가적인 전송 레이어 없이 전송하기 위한 아주 간단한 인터페이스를 말한다. REST 원리에 따르는 시스템은 RESTful이란 용어로 지칭되기도 한다.

REST Web Services는 현재 웹 서비스 스택인 WSDL의 스펙이 복잡하여 개발자들이 사용이 쉽지 않다는 점에서 등장하게 된다. REST 웹 서비스는 복잡한 SOAP 메시지를 호출하지 않고 HTTP의 기본 기능(GET, POST, DELETE)을 최대한 활용하면서 XML로 데이터를 주고 받는 장점을 갖기에 현재 Ajax 등의 기술과 함께 많이 사용되고 있다. REST의 등장에 따라 기존 엔터프라이즈 환경에서는 SOAP이 그 외의 환경에서는 REST가 각자의 입지를 굳히면서 발전하고 있다.

### 2.2 WADL

WADL은 HTTP 기반의 웹 어플리케이션들에 대한 기계 관독이 가능한 기술(description)을 제공하는 XML 기반의 파일 형식이다[8, 10]. WADL의 목적은 Web 2.0 어플리케이션을 보다 쉽게 생성하고 동적으로 관리할 수 있도록 인터넷 상의 서비스들을 기계가 처리할 수 있는 방법으로 기술하는 것이다. WADL은 WSDL과 같이 플랫폼과 언어에 독립적이며, 어플리케이션의 재사용성 증대를 목적으로 한다. WADL은 서비스에 의해 제공되는 자원들과 자원들 사이의 관계를 모델링할 수 있으며, 자동으로 Stub/Skeleton 코드를 생성함으로써 REST 웹 서비스를 배포하고 이용하는 개발자들에게 유용하다.

WADL 문서의 구조는 <그림 1>과 같다. WADL 문서

에서 application 요소(element)는 최상위 요소로, 하위에 doc, grammars, resources, resource\_type, method, representation, fault 등의 요소를 포함할 수 있다[10].

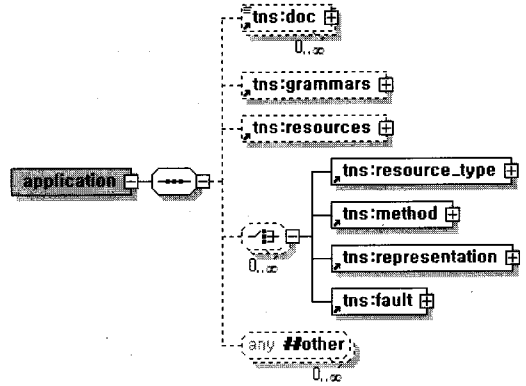


그림 1. WADL 문서 구조  
Fig. 1 WADL document structure

```

1 <?xml version="1.0"?>
2 <application xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3   xsi:schemaLocation="http://research.sun.com/wadl/2006/10 wadl.xsd"
4   xmlns:tns="urn:yahoo:yn"
5   xmlns:xsd="http://www.w3.org/2001/XMLSchema"
6   xmlns:yn="urn:yahoo:yn"
7   xmlns:ya="urn:yahoo:api"
8   xmlns="http://research.sun.com/wadl/2006/10">
9   <grammars>
10    <include href="NewsSearchResponse.xsd"/>
11    <include href="Error.xsd"/>
12  </grammars>
13  <resources base="http://api.search.yahoo.com/NewsSearchService/V1.1">
14    <resource path="newsSearch">
15      <method name="GET" id="search">
16        <request>
17          <param name="appid" type="xsd:string"
18            style="query" required="true"/>
19          <param name="query" type="xsd:string"
20            style="query" required="true"/>
21          <param name="type" style="query" default="all">
22            <option value="all"/>
23            <option value="any"/>
24            <option value="phrase"/>
25          </param>
26          <param name="results" style="query" type="xsd:int" default="10"/>
27          <param name="start" style="query" type="xsd:int" default="1"/>
28          <param name="sort" style="query" default="rank">
29            <option value="rank"/>
30            <option value="date"/>
31          </param>
32          <param name="language" style="query" type="xsd:string"/>
33        </request>
34        <response>
35          <representation mediaType="application/xml" element="yn:ResultSet">
36            <fault status="409" mediaType="application/xml" element="ya:Error"/>
37        </response>
38      </method>
39    </resource>
40  </resources>
41 </application>

```

그림 2. Yahoo News Search를 위한 WADL 문서  
Fig. 2 WADL document for Yahoo News Search

<그림 2>는 Yahoo News Search[12]를 위한 WADL 문서의 예시이다.

### III. WADL2Stub 설계

본 논문에서 제안하는 WADL2Stub은 WADL 문서를 분석하여 해당 WADL 문서가 기술하는 REST 웹 서비스를 사용할 수 있는 Stub(Client Code)을 생성하는 시스템이다. WADL2Stub 시스템의 구조는 <그림 3>과 같다.

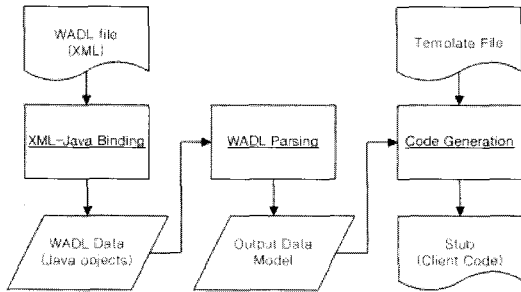


그림 3. WADL2Stub 시스템의 구조  
Fig. 3 Structure of WADL2Stub system

#### 1) XML-Java Binding

XML-Java Binding Library는 XML 문서를 자바에서 사용하기 위해 XML Schema에서 정의한 형식과 비슷한 자바 객체로 XML 문서를 변환한다. 이를 위한 XML-Java Binding Library는 여러 가지가 있지만 이 시스템에서는 그 성능과 범용성이 이미 검증된 XMLBeans [13]를 사용한다. XMLBeans를 이용하기 위해서는 먼저 XML Schema를 XMLBeans에서 제공하는 스키마 컴파일러를 통해 XMLBeans 클래스와 메타데이터로 변환해야 한다. 변환 후 생성된 XMLBeans 클래스들은 해당 XML Schema에서 정의한 데이터 형식과 유사한 형태의 자바 클래스로 정의된다. 이를 이용하여 XML 문서와 자바 객체간의 변환이 간단하게 이루어진다.

#### 2) WADL Parsing

WADL2Stub은 XML로 작성된 WADL 문서를 자바 객체로 바인딩 한 뒤 그 구조를 분석하게 된다. 이를 위해 본 논문에서는 WADL2Code의 wadl.parse 패키지를

구현하였다. wadl.parse 패키지에는 WADL 문서 구조의 주요 요소별로 각각의 Parser 클래스를 정의하였다.

#### 3) Code Generation

본 논문에서 제안하는 WADL2Stub 시스템은 사전에 작성되어 있는 템플릿 파일과 WADL 문서를 분석하여 얻은 데이터를 입력받아 클라이언트 소스 코드를 생성하게 된다. 이를 위해 본 논문에서는 템플릿 엔진을 사용하여 미리 정의된 템플릿 파일의 형식에 맞게 소스 코드를 생성한다. 소스 코드의 생성에 템플릿을 이용하고 있으므로, 미리 정의된 템플릿 파일 이외에도 지정된 형식에 맞도록 템플릿을 작성하면 사용자가 원하는 형식과 프로그래밍 언어로도 소스 코드를 생성할 수 있다. 본 논문에서는 템플릿 엔진으로 Free Marker[14]를 사용한다. 본 논문에서는 템플릿 엔진을 보다 편리하게 사용하기 위해 Template Helper Class와 ClientCodeManager Class를 설계하였다. 템플릿과 관련된 작업은 모두 Template Helper Class를 통해 이루어지고, 템플릿 파일의 경로 설정 및 템플릿 파일 로딩, 그리고 템플릿을 이용하여 실제 출력의 과정을 수행한다. ClientCodeManager Class는 앞서 WADL 문서를 분석하여 얻은 데이터 구조를 가지고 있다.

### IV. WADL2Stub 구현

본 논문에서 제안하는 WADL2Stub 시스템은 Microsoft Windows XP Professional, The Java SE Development Kit 6 Update 2, NetBeans IDE 5.5.1 환경에서 구현하였다.

WADL2Stub 시스템에서는 기본 템플릿 파일로 자바 스크립트용을 제공하며, 자바 스크립트 이외의 언어로 클라이언트 코드를 생성하려면, 해당 언어에 맞는 템플릿 파일을 정의하여 tp1 디렉터리에 저장하고 실행시 옵션을 통해 해당 템플릿 파일을 지정하면 된다. WADL2Code 시스템을 이용하기 위해서 지정하는 옵션들이 존재하며, 이는 다음과 같다.

- wadl: WADL 문서의 경로를 파일명까지 지정
- out: 생성된 클라이언트 코드 파일의 경로를 파일명까지 지정

- **template**: 별도의 템플릿 파일을 이용하여 클라이언트 코드를 생성할 때 템플릿 파일의 경로를 파일명까지 지정

본 논문에서 구현한 WADL2Stub 시스템의 실행 화면은 <그림 4>와 같다.

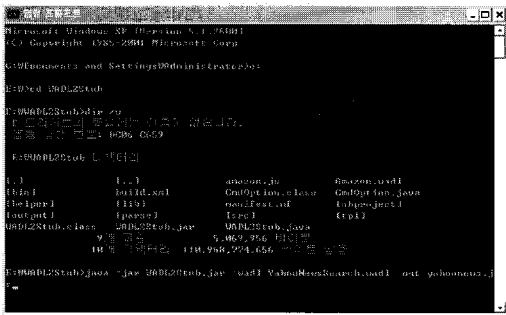


그림 4. WADL2Stub 시스템의 실행 화면  
Fig. 4 Execution of WADL2Stub system

<그림 5>은 WADL2Stub 시스템이 <그림 2>의 Yahoo News Search를 위한 WADL 문서를 분석하여 yahoonew.js 자바 스크립트 파일로 클라이언트 코드를 생성한 결과를 보여준다. 이를 통해 사용자는 생성된 js 파일을 참조하여 정의된 search 함수를 호출하여 Yahoo News Search 서비스를 이용할 수 있게 된다.

### V. 결 론

본 논문에서는 REST Web Services를 이용할 수 있도록 WADL 문서를 분석하여 자동으로 클라이언트 코드를 생성하는 WADL2Stub 시스템을 구현하였다.

본 논문에서 구현한 WADL2Stub 시스템을 이용할 경우 얻을 수 있는 기대효과는 다음과 같다.

첫째, 기존에 개발자가 직접 만들어야 했던 클라이언트 라이브러리를 WADL2Code를 이용하여 편리하게 생성하여 웹 서비스를 이용할 수 있다.

```

1  function createXMLHttpRequest()
2  {
3    var xmlhttp = null;
4    if (window.ActiveXObject) {
5      xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
6    }
7    else if (window.XMLHttpRequest) {
8      xmlhttp = new XMLHttpRequest();
9    }
10   return xmlhttp;
11 }
12 function search(appid, query, type, results, start, sort, language)
13 {
14   var xmlhttp = createXMLHttpRequest();
15   if ( xmlhttp ) {
16     var baseUrl =
17 "http://api.search.yahoo.com/NewsSearchService/V1/newsSearch?";
18     var param = "&appid=" + encodeURIComponent(appid) + "&query=" +
19 encodeURIComponent(query) + "&stype=" + encodeURIComponent(type) + "&results="
20 + encodeURIComponent(results) + "&start=" + encodeURIComponent(start) +
21 "&sort=" + encodeURIComponent(sort) + "&language=" +
22 encodeURIComponent(language);
23     xmlhttp.open("GET", baseUrl + param, true);
24     xmlhttp.onreadystatechange = function() {
25       if (xmlhttp.readyState == 4) {
26         // TODO: parse xmlhttp.responseText
27         document.write("<xmp>");
28         document.write(xmlhttp.responseText);
29         document.write("</xmp>");
30       }
31     }
32     xmlhttp.send(null);
33   }
34 }

```

그림 5. WADL2Stub 시스템에 의해 생성된 클라이언트 코드 파일  
Fig. 5 Client code file generated by WADL2Stub System

둘째, WADL2Stub 시스템에서는 템플릿을 기반으로 하여 클라이언트 코드를 생성하므로 템플릿 파일만 작성해주면 다양한 프로그래밍 언어로 클라이언트 코드를 생성하여 이용할 수 있다.

향후 연구 과제로는 응답 메시지 자동 분석 기능 구현과 개발 환경과의 통합하는 것이다. 현재 WADL2Stub 시스템은 WADL 문서를 분석하여 REST Web Services를 호출하는 과정을 자동으로 생성해주지만, 이에 대한 응답 메시지를 자동으로 분석해서 필요로 하는 반환 값을 취하지는 못하기에 응답 메시지 자동 분석 기능을 추가하고자 한다. 또한 본 시스템을 Eclipse나 NetBeans와 같은 IDE의 플러그인으로 제작하여 개발 환경과 통합하고자 한다.

### 참고문헌

- [1] Ethan Cerami, Web Services Essentials, O'REILLY, 2002.
- [2] Dave Winer, XML-RPC Specification, 1999. <http://www.xmlrpc.org/spec>.
- [3] Simple Object Access Protocol(SOAP) V 1.2, W3C Recommendation, 2007. <http://www.w3.org/TR/soap/>.
- [4] T. Bellwood, L. Clement, D. Ehnebuske, A. Hatley, M. Hondo, Y. Leng Husband, K. Januszewski, S. Lee, B. McKee, J. Munter, and C. Riegen. UDDI version 3.0. Technical report, 2002.
- [5] Web Service Description Language (WSDL) Version 2.0, W3C Recommendation, 2007. <http://www.w3.org/TR/wsdl20/>.
- [6] Roy Thomas Fielding, Representational State Transfer (REST), Ph.D. Thesis, University of California, Irvine, 2000.
- [7] Leonard Richardson and Sam Ruby, RESTful Web Services, O'REILLY, 2007.
- [8] T. Takase, S. Makino, S. Kawanaka, K. Ueno, C. Ferris, and A. Ryman, Definition Languages for RESTful Web Services: WADL vs. WSDL 2.0, IBM White Paper, 2008.
- [9] M. Muenhlen, J. Nickerson, and K. Swenson, Developing Web Services Choreography Standards- the case of REST vs, SOAP, Decision Support Systems, 2005.
- [10] M. J. Hadley, Web Application Description Language(WADL), Sun microsystems Inc., 2006.
- [11] T. Steiner, REST Describe & Compile, <http://code.google.com/p/rest-api-code/>.
- [12] Yahoo Web APIs. Technical report, Yahoo, 2005. <http://developer.yahoo.net/>.
- [13] XMLBeans, <http://xmlbeans.apache.org>.
- [14] FreeMarker: Java Template Engine Library. <http://freemarker.org/>.

### 저자소개



이화민(HwaMin Lee)

2000년 고려대학교 컴퓨터교육과  
학사

2002년 고려대학교 대학원  
컴퓨터교육학과 석사

2006년 고려대학교 대학원 컴퓨터교육학과 박사

2006~2007년 특허청 통신사무관

2007년~현재 순천향대학교 컴퓨터학부 전임강사

※ 관심분야: 그리드&클라우드 컴퓨팅, 웹 서비스