# RESTful SOAP 기반 전표 데이터 처리 시스템 설계 및 구현

## Design and Implementation of a RESTful SOAP-based Slip Data Processing System

**황의철**
광주여자대학교 교육미디어학과

Eui-Chul Hwang(euhwang@mail.kwu.ac.kr)

**요약**

본 논문에서는 SOAP(Simple Object Access Protocol)기반 웹 서비스의 문제점인 확장성을 REST(REpresentational State Transfer)모델을 이용하여 해결한 RESTful SOAP 기반 웹 서비스 방식을 제안한다. 또한 웹 서비스 응용으로 전표 데이터 처리에 대한 웹 서비스를 SOAP 및 RESTful SOAP 방식으로 구현하여 서로의 장단점을 비교하고, 추후 웹 서비스 구축에 하나의 프로토타입으로 제시한다. REST는 분산 컴퓨팅 모델이며, 세계에서 가장 큰 분산 응용인 웹에서 사용하고 있는 웹 구조 스타일 모델이다. REST에서 리소스의 식별은 URI로, 상태는 상태가 표시된 문서(리소스)로써 HTTP를 통해 전달된다. 리소스의 내용은 XML로 기술하며, 리소스 탐색 및 참조에는 HTTP의 표준 메소드인 GET, PUT, POST, DELETE 등만을 이용하는 것으로 분산컴퓨팅을 모델링하고 있다. 본 논문의 결과는 전자 상거래 시스템 구축의 주요 플랫폼인 웹 서비스의 효율적인 개발에 기여할 것으로 전망된다.

■ 중심어 : | RESTful SOAP | 웹 서비스 | 웹 애플리케이션 | 전표 데이터처리 | 전자 상거래 |

**Abstract**

In this paper, we propose a design methodology of RESTful SOAP-based web services which try to solve extensibility problem of SOAP-based Web service by adopting REST concept into SOAP. And, we apply the proposed RESTful SOAP-based web services design methodology in constructing slip data processing web services, compare it with simple SOAP-based slip data web services, and try to propose a model for building Web Services in the future. REST is a model about architecture type of web distributed computing , the biggest distributed application in the world. In REST, identification of resources are made by URI, states(resources) are represented by document described by XML and transferred through HTTP. Our proposed RESTful SOAP-based web services are expected to contribute to constructing useful world wide web services which are essential in building E-Commerce society.

■ Keyword : | RESTful SOAP | Web Services | Web Application | Slip Data Processing | e-Commerce |

접수번호 : #041115-001
접수일자 : 2004년 11월 15일
심사완료일 : 2005년 01월 28일
교신저자 : 황의철, e-mail : euhwang@mail.kwu.ac.kr

## I. 서 론

웹 서비스는 웹에서 이미 널리 수용되어 사용되고 있는 표준인 URL, HTTP, XML 등에 기반하고 있으며, 대형 소프트웨어 개발 회사인 IBM, 마이크로소프트, SUN, HP 등이 적극적으로 지원하고 있기 때문에 현재 인터넷상에서 분산 응용 구축에 매우 유망한 웹 미들웨어로 각광을 받고 있다. 웹 서비스는 데이터 표현에 XML을 사용하고 있는데 XML은 데이터 구조를 스스로 기술할 수 있다[1]. 메시지에 데이터만 실어 전송하는 CORBA, DCOM, RMI 등의 경우에는 해당 데이터 타입을 이해하지 못하는 수신자는 수신된 데이터를 이해할 수 없어 상호 동작하기 어렵다. 그러나 웹 서비스의 메시지는 데이터뿐만 아니라, 데이터 표현을 포함하고 있기 때문에 XML 파싱이 가능한 어느 누구도 메시지 내용을 이해할 수 있으며 따라서 인터넷상의 여러 응용 사이의 데이터 교환에 매우 유리하다. 또한, 특정 포트가 규정되어 있지 않아 파이어 월 통과가 쉽지 않은 CORBA 객체 서비스에 비해, 웹 서비스는 보통 SOAP 메시지를 HTTP로 전송하는 데 HTTP는 보통 파이어 월을 통과하므로 서비스 사용이 용이하다. 따라서 현재 CORBA 등의 종래 미들웨어로 구축된 분산 응용이 웹 서비스로의 포팅 또는 통합이 적극적으로 추진되고 있는 추세이다[3]. 웹 서비스는 메시지 패케징 프로토콜로 SOAP을 사용하며, 현재 구축되고 있는 웹 서비스는 SOAP 기반으로 기존 분산 응용 프로그래밍 모델인 프로시저 형태를 지향하고 있다.

그런데, 프로시저 지향형인 SOAP 기반 웹 서비스는 확장성에서 여러 문제점을 가지고 있으며, 현재 웹 구조 모델에도 잘 부합되지 않는 측면이 많다. SOAP 기반 웹 서비스에서, 서버의 서비스는 보통 객체의 메서드 형태로 구현되고 있으며 이러한 메서드들의 모임이 하나의 웹 서비스로 제공되고 있는 데, 웹 서비스 종단점에 대해서만 하나의 URI(Uniform Resource Identifier)가 할당된다. 따라서 SOAP의 종단점과 매핑되는 응용의 내부 리소스는 직접 어드레싱이 불가능하고, 이들 리소스 조작을 위해 응용 각자의 독자적인 메서드들을 제공한다. 그러나 이러한 메서드들은 HTTP의 CRUD 같은 메서드(PUT, GET, POST, DELETE 등)들처럼 표준적이 아니기 때문에 클라이언트는 반드시 이러한 메서드들을 이해하고 있어야 한다. 추후 기능 확장 등을 위해서 메서드들 내용이 변경되어야 하는 경우, 클라이언트는 이 변경된 메서드를 이용하여야 하기 때문에 기존 클라이언트 응용 프로그램의 변경이 불가피하고, 이러한 점에서 확장성이 부족하다. 따라서 SOAP 클라이언트는 웹 서비스의 인터페이스를 알아야 하며, 변경될 가능성이 있으므로 반드시 현재의 인터페이스를 알아야 한다. 물론 응용마다 제공되는 이러한 메서드 인터페이스에 대해서는 WSDL(Web Services Description Language)에 기술되며, 이는 개별적으로 또는 UDDI (Universal Description, Discovery, and Integration)를 통해 인터페이스에 대한 정보 획득이 가능하다[3]. 그러나 UDDI는 확장성이 부족하다. 이러한 점은 프로시저형 미들웨어인 CORBA 및 DCOM 등이 광범위하게 전개되지 못한 이유와도 같다.

SOAP은 현재 웹 서비스의 메시지 프로토콜로 확정되어 있으며, SOAP 지원이 산업계에서 적극적으로 이루어지고 있으므로 SOAP 기반 웹 서비스 구축이 활발히 이루어 질 전망이다.

본 논문의 구성은 다음과 같다. I 장 서론에서는 웹 서비스의 기본정의와 웹 미들웨어, 프로시저 지향형인 SOAP, 메소드 인터페이스인 WSDL 및 UDDI에 대한 설명과 SOAP의 확장성 부족에 대한 문제 제기 부분이다.

II장에서는 REST의 확장성, 웹의 REST형식 및 REST 기반 웹 서비스의 설계에 대하여 설명한다. III장에서는 전표데이터 처리를 위한 웹 서비스의 구조 및 시스템에 대하여, IV장에서는 REST 기반 웹 애플리케이션의 구현과 SOAP 및 REST 기반 웹 애플리케이션을 비교 분석하였다. V장은 결론 및 향후 연구방향을 제시하였다.

## II. REST 기반 웹 서비스

### 1. REST의 확장성

REST(REpresentational State Transfer)는 분산

컴퓨팅 플랫폼 모델이며, 세계에서 가장 큰 분산 응용인 Web에서 사용하고 있는 웹 구조 스타일 모델이다. 현재의 웹의 기본 요소는 URI, HTTP, XML(HTML) 이며, REST는 이러한 인터넷 표준만을 이용한다. REST에서 리소스의 식별은 URI로, 상태는 이를 표현된 문서(리소스)로써 HTTP를 통해 전달된다. 리소스의 내용은 XML로 기술하며, 리소스 탐색 및 참조에는 HTTP의 표준 메서드인 GET, PUT, POST, DELETE 등을 이용하는 것으로 분산 컴퓨팅을 모델링하고 있다[4].

REST는 분산 응용을 '상태 엔진'으로 본다. 클라이언트의 응용에 대한 요청은 응용의 현재 상태를 변화시키고, 이 변화된 상태는 클라이언트 요청에 대한 결과 정보를 포함하고 있다. 이 상태는 representation을 갖는데 보통 한개 또는 여러 개의 문서로 표현되며(XML 문서), 이 문서는 하이퍼텍스트로 다른 상태와의 연결고리(링크)를 갖는다. 즉, 응용의 상태들은 응용의 현재 정보를 나타내며 하이퍼텍스트 문서들로 표현된다. 이 하이퍼텍스트 문서들은 리소스로써 유일무이한 URI를 갖는다. 링크는 새로운 상태로의 이동을 지시한다. 다음 상태로의 전환은 다음 상태 정보를 표현하는 리소스로의 이동을 가르키는 링크를 선택함으로써 이루어진다. 또한, REST에서 리소스의 전달은 그 표현인 문서 형태로 HTTP를 통해 이루어진다. 응용의 클라이언트는 응용에 서비스를 요청하고, 이 서비스의 요청 결과는 응용의 요청 결과 상태를 표현하는 문서로써 클라이언트에 전달된다. REST에서 리소스에 대한 오퍼레이션은 보편적이며 보다 단순한 GET(문서 가져오기), PUT(문서 쓰기), POST(문서에 추가 정보 전달하기), DELETE(문서 삭제), HEAD(문서의 헤더만을 가져오기) 등의 메서드만을 지원하며, 이러한 메서드는 HTTP 에서 지원한다[9].

이상의 논의에서, REST 모델의 분산 컴퓨팅 응용은 확장성에 매우 유리함을 알 수 있다. 웹의 REST 형식은 표 1을 사용한다. 현재의 웹의 기본 요소는 URI, HTTP, XML(HTML) 이다. 이들 3가지 요소는 모두 확장성을 갖는다. XML은 자신의 문서 구조를 스스로 표현 할 수 있는 능력을 가지고 있어 그 확장성은 잘 알려져 있다.

**표 1. 웹의 REST 형식**

| | |
|---|---|
| Network Protocol | HTTP |
| Address Scheme | URIs-URLs and URNs |
| Resource Representations | XML, HTML, GIF etc. |
| Resource types | MIME Types-ext/xml, text/html, image/gif |
| Hyperlinks | XML's xlink attributes, HTML(a) tags |
| Resource search<br>Resource reference | Get, Put, Post, Delete |

## 2. SOAP 기반 웹 서비스의 한계

SOAP과 WSDL은 모든 객체를 관리하는 무거운 한 개의 종단점만을 참조하는 하나의 URI 만을 기술한다. 따라서, SOAP의 종단점과 매핑되는 응용의 내부 리소스는 직접 어드레싱이 불가능하고, 이들 리소스 조작을 위해, 응용에 따른 독자적인 메서드들을 제공한다. 그러나 이러한 메서드들은 HTTP의 CRUD 같은 메서드(PUT, GET, POST, DELETE 등)들처럼 표준적이 아니기 때문에 클라이언트는 반드시 이러한 메서드들을 이해하고 있어야 한다. 그런데, 추후 기능 확장 등을 위해서 메서드들 내용이 변경되어야 하는 경우, 클라이언트는 이 변경된 메서드를 이용하여야 하기 때문에 기존 클라이언트 응용 프로그램의 변경이 불가피하고, 이러한 점에서 확장성이 부족하다. 따라서 SOAP 클라이언트는 웹 서비스의 인터페이스를 알아야 하며, 변경될 가능성이 있으므로 반드시 현재의 인터페이스를 알아야 한다. 물론 응용마다 제공되는 이러한 메서드 인터페이스에 대해서는 WSDL에 기술되며, 이는 개별적으로 또는 UDDI를 통해 인터페이스에 대한 정보 획득이 가능하다. 그러나 UDDI는 확장성이 부족하다. 이러한 점은 CORBA 및 DCOM 등이 광범위하게 사용되지 못한 이유와도 같다.

## 3. REST 기반 웹 서비스의 설계[6]

제 2세대 REST 웹 서비스는 HTML보다 오히려

XML을 사용한다. 웹 서버로 공급되는 디렉토리의 목록 자원에 추가나 삭제할 수 있다. REST 웹 서비스의 본질적인 목표는 웹 애플리케이션을 형성하는 것, 기존의 웹 클라이언트, 중개자, 서버에서 동작하는 것 및 별개의 요청을 별개의 URI를 캐싱하기 위해 사용한다는 것이다.

현대 웹 구조는 구성요소 상호작용의 확장성, 인터페이스의 보편성, 구성요소와 중간의 구성요소의 독립적인 전개를 상호작용 호출 시간을 줄이고, 안전을 실시하고, 레거시 시스템을 캡슐에 넣기 위해 강조한다[5].

## III. 전표 데이터처리 웹 서비스

### 1. 전표 데이터처리 웹 서비스 구조

웹 서비스는 애플리케이션 개발의 마지막 단계이며, 어떠한 플랫폼에서도 개발자가 작업을 할 수 있게 해준다. 웹 서비스가 인터넷 또는 인트라넷을 통해 RPC (Remote Procedure Call ; 원격 프로시저 콜)를 만들어 준다.

전표 데이터 처리를 위하여 사용자는 웹 서비스를 어디 에서나 접근할 수 있는 컴포넌트로 생각할 수 있으며, 웹 서비스는 잘 정의된 메서드들을 노출하는 인터페이스로 표준 인터넷 프로토콜을 사용하여 메서드를 호출하고 XML 형식으로 매개변수를 넘기며, XML 형식으로 응답을 받도록 한다.

그림 1은 전표 입력 사용자 웹 인터페이스를 통하여 입력된 전표 데이터가 전표 데이터베이스 서버(DBMS; SQL server, MS-Access)에 저장된다. 웹 서비스는 회계 담당자의 애플리케이션이 자유롭게 중앙에서 관리되는 비즈니스 객체들의 메서드를 실행할 수 있도록 하기 위하여 HTTP를 통해서 회계담당자 애플리케이션을 조작하는 비즈니스 로직을 만드는데 SOAP과 HTTP를 사용한다. 클라이언트 Request는 인터넷을 통해서 SOAP을 사용하여 메시지를 주고받는다. 웹 서비스는 HTTP를 통해서 SOAP 메시지로 클라이언트 애플리케이션에 응답을 보낸다.

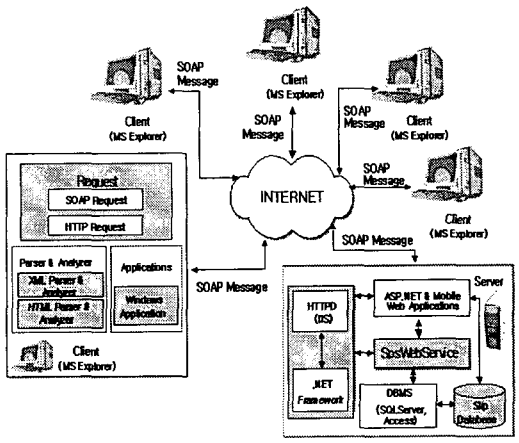


그림 1. 전표 데이터 처리의 웹 서비스 구조도

### 2. XML 기반 전표 데이터 처리 시스템

XML 기반 전표 데이터 처리 시스템의 구현은 MS Windows 2000 운영체제 하에서 웹 서버 IIS5.0을 기반으로 하고, DBMS로는 MS SQL 2000, 프로그래밍 환경으로는 .NET Framework을 기반으로 한 ASP.NET[11], C#를 이용하였다. MS-Explorer는 HTTP Request, XML Parser & Analyzer,SQL XML Query로 구성되는"http://hec.kwu.ac.kr/cs/websps/input_slip_aspx"의 실행을 요청하면 Web 서버의 Daemon, ASP.NET의 .dll 파일에 의해 전표 처리 시스템에의 전표입력(input_slip_aspx)이 될 수 있도록 MS-Explorer에 전표 입력화면이 뜬다. 웹 서버에서 실행 결과는 XML Parser & Analyze 후에 브라우저에 표현된다. 클라이언트에서 MS-SQL 또는 MS-ACCESS의 데이터베이스로 질의하면 질의 해석을 하여 질의 생성기에서 생성된 질의 결과를 MS-Explorer에 XML형식으로 표현된다. 전표 처리 시스템에서는 XML builder, Visual Studio .net 등으로 .XSD(XML Schema)나 .XSLT(출력양식)로 변환이 가능하다.

전표 데이터 처리를 구현할 시스템의 전체 구조로서, 다음 세 부분으로 나누어 볼 수 있다.

(1) 사용자가 사이트를 사용하여 등록과 로그인 인증과 같은 일을 할 때 WebSps 웹 애플리케이션은

.NET 컴포넌트 WebSpsLib의 메서드를 실행하며, 등록의 세부사항은 WebSps 데이터베이스에 저장된다.

(2) WebSps는 다음의 작업을 위하여 인터넷을 통해 웹 서비스의 메서드를 실행한다.
  ① SpsWebService ; 전표 데이터의 내용 처리 및 그 결과값을 XML로 응답 해준다.
  ② SlipXmlService ; 전표자료의 XML 출력 웹 서비스

(3) 윈도우 서비스로 활용되는 윈도우즈 애플리케이션인 waSlipXml의 SlipXmlForm를 통하여 전표의 분개현황을 XML 파일로 기록하고 그 URL 및 XML 파일을 볼 수 있도록 구현한다.

(4) 윈도우 서비스로 활용되는 웹 애플리케이션인 매입매출 정보 검색 웹 서비스(SpsWebService의 SelectPurSalesSlip)를 활용하여 매입매출현황을 검색하는 웹 애플리케이션을 구현한다.

## Ⅳ. REST 기반 웹 애플리케이션 구현 및 분석

웹 서비스는 상이한 플랫폼 상에서 많은 다양한 방법으로 구현될 수 있으며[8], 본 논문에서는 웹 애플리케이션 웹 서비스의 활용과 윈도우즈 애플리케이션의 웹 서비스 활용에 대하여 구현된 경우를 설명한다.

그림 2는 전표 데이터 처리 응용 프로그램 구조로서, 본 논문에서 구현된 웹 서비스 및 웹 애플리케이션에 대한 다이어그램이다. 전표 데이터는 ①의 전표입력 사용자 웹 인터페이스 창에서 입력한다. 이 창에서는 입력된 내용의 목록, 세금계산서 및 거래내역서 등의 증빙자료 이미지를 보면서 확인이 가능하여 입력 시 발생되는 오류를 최소화할 수 있다. 입력된 전표자료는 ②의 MS Access 및 SQL Server의 데이터베이스에 저장되어 전표자료의 공유, 재사용 가능, 회계전문가로 하여금 거래내용의 투명성 여부를 확인할 수 있다. 공통모듈(*Common Module*)은 ③,④의 웹 서비스로서 ③의 웹 서비스는 ②에 저장되어 있는 데이터들의 내용 중 거래회사의 대변, 차변, 매입매출별 세금들의 합 등의 가공처리된 내용을 호출해주는 메서드들이다. ④의 웹 서비스는 전표 데이터의 company Code, SendDate, sendNo를 입력하면, DBMS에 저장된 세부정보를 XML 웹 서비스의 URL로 받는다. 또 ②의 데이터베이스에 XML 파일로도 저장된다.

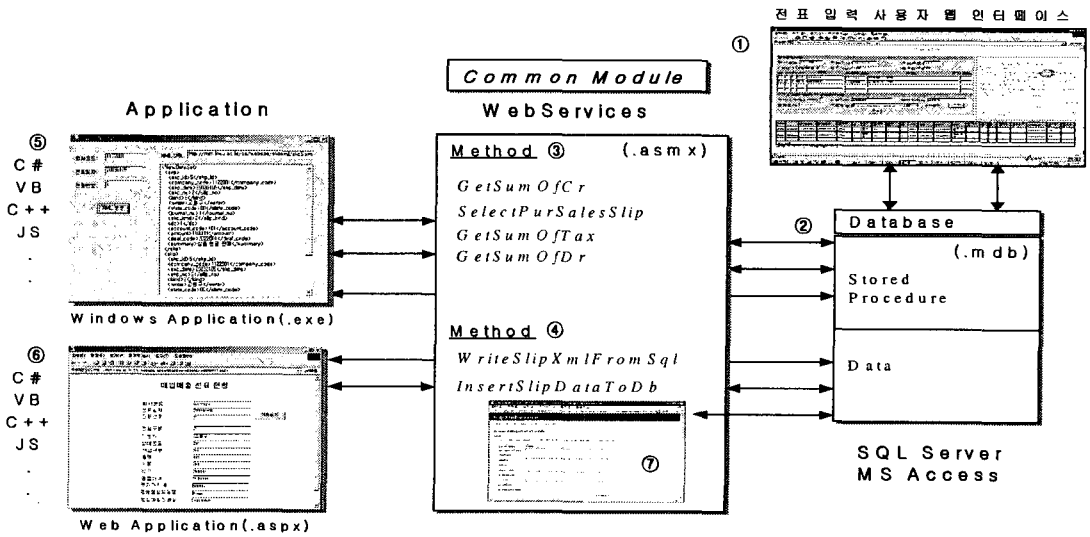


그림 2. 전표 데이터 처리 응용프로그램 구조의 다이어그램

InsertSlipDataToDb 메서드를 실행하면 ⑦의 폼이 생성되어 companyCode, slipDate, slipNo, writer, stateCode, drAccount, amount, dealCode, summary Code 값을 입력하여 ②의 데이터베이스에 추가할 수 있는 메서드이다. ⑤,⑥은 애플리케이션으로 ⑤는 윈도우즈 애플리케이션으로 VB, C++, C#, JScript 등의 언어로 구현할 수 있으며, waSlipXml.exe가 실행되어 ⑤의 화면에 회사코드, 전표일자, 전표번호를 입력하면 해당 URL의 XML 파일을 호출하여 보여준다. ⑥의 웹 애플리케이션에 회사코드, 전표일자, 전표번호를 입력하면 폼의 매입매출 전표에 해당 데이터의 정보가 나타난다.

## 1. REST 기반 웹 애플리케이션의 구현

최근의 REST 웹 서비스는 HTML 보다는 XML을 사용한다. 따라서 이러한 서비스는 사용자에게 다음과 같은 오퍼레이션을 제공할 수 있도록 고려한다.

- 웹 서버에 의해 제공되는 디렉토리에 있는 자원의 목록화 및 디렉토리에 있는 각 자원을 가져옴.(GET)
- 새로운 자원이나 서브디렉토리를 디렉토리에 추가(PUT) 및 디렉토리 및 서브디렉토리 삭제(DELETE)

이러한 REST 서비스는 doGet(), doPut(), doDelete()와 같은 .NET의 웹 서비스의 3개의 메서드로 구현된다.

그림 3은 REST 기반의 웹 애플리케이션 구현으로, company 에서 검색하려고 하는 회사코드(1122001)를 클릭하면 매입매출 전표현황으로 연결되고, 거래발생일 및 거래가 빈번하게 발생되므로 확인하고자 하는 전표일자의 전표식별번호(5)를 클릭하면 매입매출 전표내역을 상세히 확인할 수 있다. 전표 데이터 처리를 위해 다음과 같은 웹 서비스를 배포하여 일반 사용자가 이용할 수 있도록 하고자 한다.

(1) 전표 데이터 처리 회사의 목록을 표시한다.(①)
(2) 특정회사의 매입매출 전표목록을 표시한다.(②)
(3) 매입매출 전표 상세 내역을 표시한다.(③)

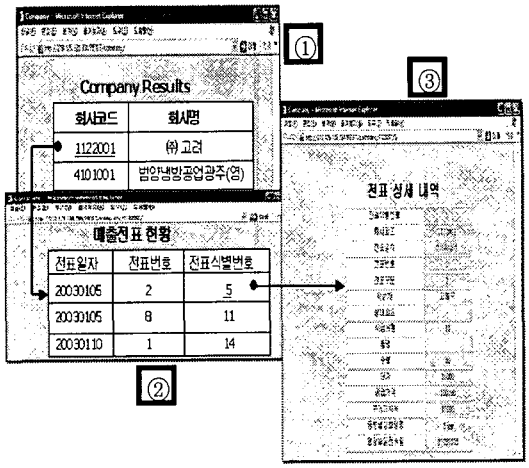


그림 3. REST 기반의 웹 애플리케이션 구현

## 2. SOAP & REST 기반 웹 애플리케이션 비교분석

REST는 분산 하이퍼미디어 시스템을 위한 구조이므로, 전표 데이터베이스내의 정보를 서로 연관된 정보에 대해 효과적이면서 비순차적인 검색이 가능한 하이퍼텍스트의 장점과 다양한 매체를 통해서 효율적인 정보를 제공할 수 있는 멀티미디어의 장점을 모두 갖춘 새로운 매체환경으로 정의할 수 있다. 또한 멀티미디어, 하이퍼텍스트, 하이퍼미디어 등은 컴퓨터를 통하여 제어되는 상호작용적인 정보환경을 제공할 수 있다. REST 기반의 전표 데이터 웹 애플리케이션은 수많은 회사의 거래내역을 신속하고 정확하게 확인하거나, 거래 내역의 투명성 차원에서 검색을 용이하도록 응용하는데 매우 효율적일 것이다. 또한 멀티미디어 응용의 발전에 따라 분산 환경에서의 접근과 배포(distribution)등의 멀티미디어 활용 및 응용간의 정보교환을 효과적으로 수행하기 위하여 멀티미디어 정보의 공통된 교환 형식이 필요하게 되었다. 표준 인터넷 프로토콜인 HTTP, 모든 웹 서비스를 통해 데이터와 메시지를 표현하는 표준인 XML, 어떤 웹 서비스에도 접속이 가능하게 해주는 SOAP, SOAP에 덧붙여서 웹 서비스를 실용적인 솔루션으로 만드는 표준인 WSDL, DISCO, UDDI 등의 표준이 서버와 클라이언트 사이에서 정보교환이 가능하도록 해준다. 웹 서비스는 차후 전자상거래 애플리케이션에 적용이 되어 요구되는 기능을 수행하게 될 것이다.

점점 웹 서비스가 비즈니스를 위해서 사용되면 될수록 애플리케이션의 개발은 기존의 솔루션을 최대한 재사용하여 특정 비즈니스 문제를 해결하고 비즈니스측면이 아닌 특별한 문제와 모든 하부 구조를 위한 웹 서비스에 접근하는 형태를 띠게 될 것이다.

본 논문을 통하여 전표 데이터, XML, SOAP, WSDL, UDDI, REST 등의 웹 서비스 프로토콜, 전표 데이터베이스 테이블 설계 및 XML 스키마 설계, 전표 데이터 처리를 위한 웹 서비스 설계 및 구현을 위한 관련 지식과 실제 구현한 내용을 바탕으로 SOAP 기반의 웹 서비스와 REST 기반의 웹 서비스에 대하여 비교하였다.

**표 2. SOAP & REST 기반 웹 애플리케이션 비교**

| 웹 애플리케이션 | SOAP 기반 웹 애플리케이션 | REST 기반 웹 애플리케이션 |
|---|---|---|
| Addressing | 표준화된 방법이 없으며, SOAP application은 자기 자신의 주소 지정 스키마를 정의 | 표준화된 방법을 사용하며, 기존 웹 주소 지정 모델을 모두 활용 |
| SOAP Interface | 각 애플리케이션이 자신의 오퍼레이션 집합을 정의 | SQL, File system, Registry 사용. HTTP 통한 PUT, GET, POST, DELETE 서비스 제공 |
| | 일반적인 오퍼레이션 제공하지 않음 | 표준화되고 일반화된 오퍼레이션 제공 |
| 상태관리 | Message들이 단지 data만 포함하고 있으며 SOAP application이 상태정보를 가지지 못함 | 자원이 data와 유효한 상태 전이에 대한 link 포함하고 있고, REST application이 상태정보를 가지고 있음 |

SOAP 기반의 웹 서비스는 SOAP의 종단점과 매핑되는 응용의 내부 리소스는 직접 어드레싱이 불가능하고, 이들 리소스 조작을 위한 독자적인 메서드의 사용으로 추후 기능 확장을 위해서는 프로그램을 변경해야 하므로 확장성이 부족하다.

REST는 분산 컴퓨팅 플랫폼 모델이며, Web에서 사용하고 있는 웹 구조 스타일 모델이다. 현재의 웹의 기본 요소는 URI, HTTP, XML(HTML) 이며, REST는 이러한 인터넷 표준만을 사용한다. REST 에서 리소스의 식별은 URI로, 상태는 이를 표현된 문서(리소스)로써 HTTP를 통해 전달된다. 리소스의 내용은 XML로 기술하며, 리소스 탐색 및 참조에는 HTTP의 표준 메서드인 GET, PUT, POST, DELETE 등만을 이용하는 것으로 분산 컴퓨팅을 모델링하고 있다.

REST는 중개자에 의해 상호작용, 구성요소의 동적인 대용 가능성과 행동 처리의 캐싱과 재사용을 가능하게 하며, 인터넷의 확산으로 하이퍼미디어 시스템의 필요성을 갖는다. 현대의 웹은 REST 스타일 구조의 한 인스턴스이다. 비록 웹에 기반을 둔 적용이 다른 스타일의 상호작용으로의 접근을 포함할 수 있지만, 그 프로토콜과 실행 중심의 초점은 분배되었던 하이퍼미디어이다

따라서 서비스마다 다양한 메서드를 기억하여야 하는 SOAP 기반 웹 서비스에 비해 REST 모델의 분산 컴퓨팅 응용은 확장성에 매우 유리함을 알 수 있었다.

## V. 결 론

REST는 리소스의 식별은 URI 로, 상태는 이를 표현된 문서(리소스)로써 HTTP를 통해 전달된다. 리소스의 내용은 XML로 기술하며, 리소스 탐색 및 참조에는 HTTP의 표준 메서드인 GET, PUT, POST, DELETE 등을 이용하는 것으로 분산 컴퓨팅을 모델링하고 있다.

서비스마다 다양한 메서드를 기억하여야 하는 SOAP 기반 웹 서비스에 비해 REST 모델의 분산 컴퓨팅 응용은 확장성에 매우 유리함을 알 수 있었다. SOAP 기반 웹 서비스는 현재, 웹 서비스의 대세로 굳어져 가고 있고, SOAP 지원이 산업계에서 적극적으로 이루어지고 있으므로, SOAP 기반 웹 서비스 구축이 활발히 이루어 질 전망이다.

본 논문에서는 SOAP 기반 웹 서비스의 문제인 확장성을 REST 모델을 이용하여 해결한 RESTful SOAP 기반 웹 서비스 방식을 제안하였다. 그리고 이러한 방식의 응용으로 전표 데이터 처리에 대한 웹 서비스를 SOAP 방식과 RESTful SOAP 방식으로 구현하고, 장단점을 비교하여 추후 웹 서비스 구축에 하나의 사례로 제시하였다.

본 논문의 결과는 전자 상거래 시스템 구축의 주요 플

랫폼인 웹 서비스의 효율적인 개발에 기여할 것이며, 향후 ebXML차세대 전자상거래의 웹 서비스 구축에 대한 연구로 이어져야 할 것이다.

## 참 고 문 헌


[1] JAVA & XML, Sun Newletter May 2002, pp. 34-35.

[2] S. vinoski, "CORBA: Integrating Diverse Applications Within Distributed Heterogeneous Environments," IEEE Comm., Vol.35, No.2, pp. 46-55.

[3] "Web Service Description and Discovery Using UDDI, Part I," http://msdn.microsoft.com/library/enus/nservices/html/

[4] Roy Fielding, "Architectural Styles and the Design of Network-based Software Architectures," http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm, 2000.

[5] Roy T. Fielding and Richard N. Taylor, "Principled Design of the Modern Web Architecture," http://www. ics.uci.edu/~fielding/pubs/webarch_icse2000.pdf

[6] Hamish Taylor, "Network Applications Lecture 19," http://www.cee.hw.ac.uk/courses/5nn2/19/index.htm, 2003.

[7] Scott Short, "Bulding XML web services for the microsoft.NET platform," Microsoft Press, pp. 137-149, pp. 267-268, 2002.

[8] Lonnie Wall, Andrew Lader, "Building Web Services and .NET Applications," McGrawHill, pp. 258-265, 2002.

[9] Roger L. Costello, "Bulding Web Services the REST Way," http://www.xfront.com/REST-Web-Services.html

[10] Peter Drayton, "REST & SOAP," developmentor, http://www.clrinternals.com/slides/RESTfulSOAP.pdf

[11] Richard Anderson, "PROFESSIONAL ASP. NET," pp. 1220-1221, 2002.


## 저 자 소 개


**황 의 철(Eui-Chul Hwang)** 정회원

- 1986년 : 서울산업대학교 전자공학과(공학사)
- 1988년 : 연세대학교 전자공학과(공학석사)
- 2004년 : 숭실대학교 전자공학과(공학박사)
- 1992년~현재 : 광주여자대학교 교육미디어학과 교수

<관심분야> : 멀티미디어, 디지털제어회로, ebXML 컴퓨터네트워크,