

# 웹 서비스 프로토콜 기술에 대한 비교 분석

최종석\*, 유승학, 김보승, 신용태

\*송실대학교 컴퓨터학과

{jschoi\*, shryu, bskim}@cherry.ssu.ac.kr, shin@ssu.ac.kr

## The Comparison analysis of web service protocol

Jong-Seok Choi\*, Seung-Hak Ryu, Bo-Seung Kim, Yong-Tae Shin

\*Dept of Computing, Soong-Sil University

### 요 약

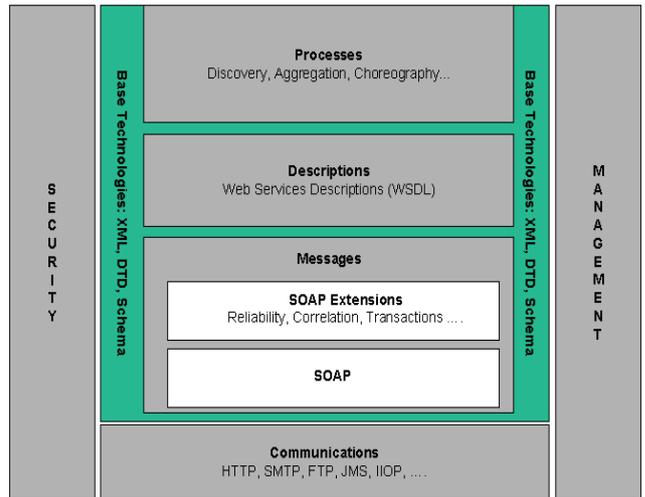
웹 서비스는 표준화된 XML 메시지를 통하여 네트워크에 연결된 기계간에 상호작용 위하여 고안된 분산 컴퓨팅을 지원하는 기술이다. 일반적으로 웹 서비스를 구성하는 프로토콜 중 SOAP은 현재 웹 서비스의 메세징 프레임워크를 정의하는 프로토콜의 표준으로 사용되고 있지만 REST의 경우 전송방식이 개발에 용이하고 사용이 편리하여 앞으로 구축이 활발해질 전망이다. 또한 SRW/U, OAI-PMH는 웹 기반 정보검색을 표준화된 방식으로 수행하도록 규정하는 웹 서비스 프로토콜이다. 본 논문에서는 웹 서비스의 다양한 프로토콜에 대한 기술을 소개하고 이에 대한 비교 분석을 통하여 장단점을 살펴본다.

### 1. 서론

웹 서비스(Web Service)는 ‘웹’과 ‘서비스’라는 두 단어가 결합해 생겨난 용어인 만큼 단순하게 해석하면 웹을 통해 서비스를 교환하는 것이다. 웹은 이미 널리 사용되고 있는 표준인 URL(Uniform Resource Locator), HTTP(Hypertext Transfer Protocol), XML(Extensible Markup Language) 등에 기반하고 있으며 표준 방식으로 분산되어 있는 정보자원들을 공유하고 호환시키는 인터넷의 응용 역할을 하고 있다. 서비스란 사용자에게 세부적인 사항은 감추고 추상적인 관점에서 제공되는 기능을 의미한다. 종합적으로 웹 서비스는 분산되어 있는 콘텐츠를 서비스 형태로 추상화시켜 표준 방식으로 연계하거나 공유하는 기술이다. 이에 따라 웹 서비스를 구성하는 프로토콜 중 SOAP(Simple Object Access Protocol)은 현재 웹 서비스의 메세징 프레임워크를 정의하는 프로토콜의 표준으로 사용되고 있지만 REST(REpresentational State Transfer)의 경우 전송방식이 개발에 용이하고 사용이 편리하여 앞으로 구축이 활발해질 전망이다. 또한 SRW/U(Search and Retrieve Web Service/URL), OAI-PMH(Open Archives Initiative Protocol for Metadata Harvesting)는 웹 기반 정보검색을 표준화된 방식으로 수행하도록 규정하는 웹 서비스 프로토콜이다. 이러한 프로토콜에 대해서 본 논문에서는 웹 서비스의 다양한 프로토콜에 대한 기술을 소개하고 이에 대한 비교 분석을 통하여 장단점을 살펴본다. 본문의 구성은 2장에서는 웹 서비스에 대한 개념 및 구조를 정의하고, 3장에서는 실제 사용되고 있는 웹 서비스의 프로토콜인 REST, SOAP, SRW/U, OAI-PMH에 대하여 비교 분석하고 4장에서 결론을 맺는다.

### 2. 웹서비스

웹 서비스는 표준화된 XML 메시지를 통하여 네트워크에 연결된 기계간에 상호작용 위하여 고안된 분산 컴퓨팅을 지원하는 기술이다. 기계가 처리 가능한 형태로 기술된 인터페이스(WSDL(Web Services Description Language))를 가지며, 다른 시스템들은 주로 HTTP 프로토콜 상에서 SOAP메시지를 사용하여 인터페이스에 명시된 방식으로 웹 서비스와 상호작용을 하게 된다[1]. 이러한 웹 서비스 아키텍처 스택의 구성은 (그림 1) 과 같다.



(그림 1) 웹서비스 아키텍처 스택

웹서비스 아키텍처 스택의 계층 구조는 아래와 같다.

- 커뮤니케이션 계층에서는 네트워크 프로토콜로써 HTTP, HTTPS, SMTP, FTP 등의 인터넷 프로토콜들을 사용할 수 있도록 정의되어 있다.
- 메시지계층의 SOAP은 XML을 사용한 메시지 프로토콜

이다. SOAP 메시지 콘텐츠를 묘사하기 위한 Envelope 과 인코딩 규칙, 원격 프로시저 호출(RPC)을 제공하기 위한 메커니즘으로 구성되어 있다.

- 기술계층은 웹 서비스를 호출하는 수단에 대한 상세 정보를 제공한다. WSDL이 기술 계층에서 서비스 인터페이스와 실행을 정의하는데 쓰이는 기본 표준이다.
- 프로세스계층은 개별적인 서비스의 상위에 존재하여, 서비스들을 함께 유통하고 활용하는 방법을 제공하는 층이다. 이 층에서는 서비스 유통에 필요한 UDDI(universal description, discovery, and integration)와 서비스들의 협업을 위한 표준, 비즈니스 정책이나 프로파일들이 포함된다.
- 보안계층은 웹 서비스의 안전한 서비스 유통을 위하여 필요한 인증, 암호화, 권한부여 등의 표준을 다룬다.
- 관리계층은 웹 서비스가 개발되고 실제 운영될 때, 그 안전성과 신뢰성을 확보하기 위해서는 웹 서비스에 대한 관리가 중요하다. 관리 계층에서는 이러한 웹 서비스의 관리에 대한 표준 안을 제공한다.

### 3. 웹서비스 프로토콜 분석

#### 3.1 REST

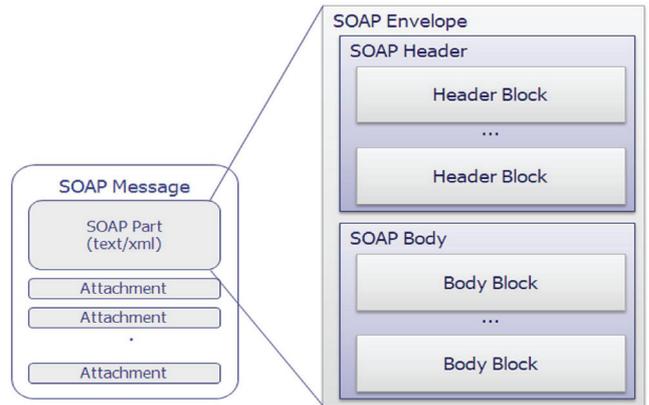
REST(REpresentational State Trasfer)는 대규모 네트워크 시스템을 위한 아키텍처 원리의 모음으로써 2000년 Roy Fielding에 의해서 최초로 제안되었다. 이 아키텍처는 웹상의 모든 자원들을 URL로 표현하며 이러한 URL 표현 방식을 통해 웹 어플리케이션을 구동하고 그 결과를 전달 받아 처리하는 것에 중점을 두고 있다.[2] 이는 도메인 지향 데이터를 HTTP 위에서 SOAP이나 쿠키를 통한 세션 트래킹 같은 부가적인 전송 레이어 없이, 전송하기 위한 아주 간단한 인터페이스를 말한다. REST는 웹의 기본 요소인 URI, HTTP, XML(HTML)을 사용하여 리소스 식별은 URI로, 상태는 이를 표현된 리소스으로써 HTTP를 통하여 전달된다. 리소스의 내용은 XML로 기술되며, 리소스 탐색 및 참조에는 HTTP의 표준 메서드인 GET, PUT, POST, DELETE와 대응한 URL 표현 방식으로 요구 자원을 요청한다. 이러한 REST의 웹에서의 표현 형식은 (표 1)과 같다.

|                          |   |
|--------------------------|---|
| Network Protocol         | HTTP                                    |
| Address Scheme           | URIs-URLs and URNs                      |
| Resource Representations | XML, HTML, GIF etc.                     |
| Resource type            | MIME Types-ext/xml,text/html, image/gif |
| Hyperlinks               | XML's xlink attributes, HTML<a> tags    |
| Resource search          | GET, PUT, POST,                         |
| Resource reference       | DELETE                                  |

<표 1> REST의 웹에서의 표현 형식

#### 3.2 SOAP

SOAP(Simple Object Access Protocol)은 인터넷 상에서 XML 데이터를 교환하는데 사용되는 메세징 프레임워크를 정의하는데 사용된다. 메세징 프레임워크는 단순하고 개발하기 쉬우면서도 OS나 프로그래밍 언어와 같은 플랫폼에 독립적이어야 하며 분산 컴퓨팅 환경에 적합해야만 하는데, SOAP은 이와 같은 메세징 프레임워크의 조건을 충족시킴과 동시에 최소한의 가장 기본적인 전송에 사용될 수 있게 만들어진 프로토콜이다[3].



(그림 2) SOAP의 메시지 구조

SOAP의 메시지 구조는 (그림 2)와 같이 주로 HTTP POST 메소드를 통하여 XML 문서를 전달하기 위해서 메시지의 시작과 끝을 정의하는 봉투(Envelope), 메시지의 모든 조건적 속성들을 포함하는 헤더(header), 전송될 메시지를 포함한 모든 XML 데이터를 포함하는 바디(body) 등 세 개의 주요 부분으로 구성되어 있다. SOAP 메시지는 이메일과 같이 MIME 타입을 사용하여 하나 이상의 첨부파일을 보낼 수 있다. SOAP의 기본적인 기능은 다음과 같다.

- SOAP Envelop(필수) : SOAP에서 사용하는 요소를 위한 네임스페이스 선언과 메시지에 쓰이는 데이터를 위한 인코딩 스타일 속성을 정의한다. SOAP 1.1과 1.2 메시지는 서로 다른 네임스페이스를 사용한다.
- SOAP Header(옵션) : 메시지 내용에 대한 추가적인 기능(발신자 인증, 트랜잭션 관리 등)을 기술한다.
- SOAP Body(필수) : RPC(Remote Procedure Call) 호출 내용을 XML로 인코딩하거나 메시지 안에 오류 혹은 상태정보(fault)를 전달하는데 사용한다.

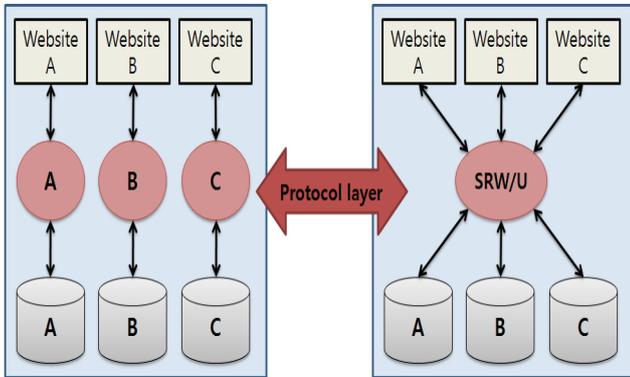
SOAP은 표준규격 자체로 보면 분산 환경에 있는 피어(Peer)들 사이에 정형화된 정보와 구조를 주고 받을 수 있도록 경량화된 프로토콜을 정의한 것이지만 실제 사용에서는 이러한 범위를 뛰어넘는 적용 가능성을 보여준다. SOAP은 LAN(local area network)에 있는 컴퓨터들 사이의 의사소통 수단으로도 사용할 수 있으며 프로그래밍 모델을 정의하지 않고 있으며 API(application programming interface)와 같은 구현 의존적인 부분에 대한 의미론을 따로 정해두지 않았기 때문에 확장 가능성이 더 크다. SOAP을 지원하는 웹 사이트와 웹 어플리케이션은 웹 서

비스를 제공할 수 있는 기본 자격을 갖추게 되고, 추가적인 프로그래밍을 하지 않고도 다른 컴퓨터의 서비스들과 연계가 가능하다.

3.3 SRW/U

SRW/U 프로토콜은 SRW(Search/Retrieve Web Service)와 SRU(Search/Retrieve via URL)로 나눌 수 있으며 정보검색 서비스에 대한 표준 규격(Information Retrieval : ANSI/NISO Z39.50, ISO 23950)인 Z39.50 프로토콜을 웹 환경에 맞게 재구성한 것이다. SRW/U 프로토콜은 Z39.50 프로토콜과 같이 서로 다른 데이터베이스, 검색항목, 데이터의 형식 등을 사용하는 이기종간의 정보 검색을 위한 표준화된 통신규칙을 규정하고 있다[4].

SRW/U는 클라이언트가 작동하는 특정 컴퓨터에서 서버 역할을 하는 다른 컴퓨터에 탐색 명령어를 보내고 그 결과를 제공받는 분산형 클라이언트/서버 환경의 웹 기반 정보검색을 표준화된 방식으로 수행하도록 규정하는 것이다.



(그림 3) 개별 검색 프로토콜과 SRW/U 프로토콜 적용 비교

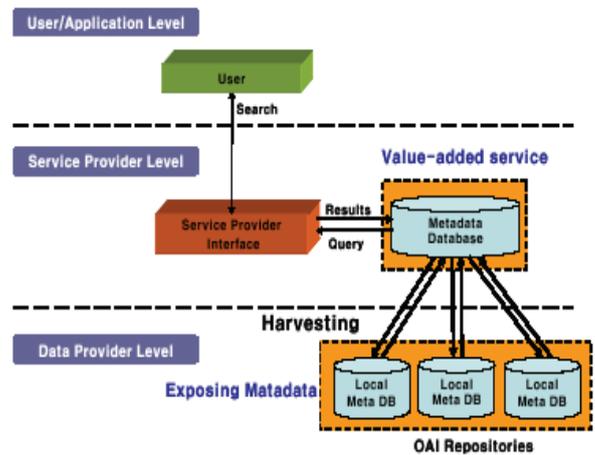
위의(그림 3)으로 비교해 보았을 때 SRW/U 프로토콜의 기능을 보여주는 것으로서 개별 검색 프로토콜은 다른 데이터베이스 검색을 위해서는 별도의 프로토콜을 다시 구현하여야 한다. SRW/U 프로토콜의 각 웹 페이지는 모든 데이터베이스에 접근 할 수 있는데, 이는 공통된 검색 프로토콜을 사용하였기 때문이다. SRW와 SRU는 모두 데이터베이스를 탐색하고 그 결과를 보내기 위한 웹 서비스 기반 프로토콜이다. 두 가지 프로토콜에 사용되는 요청(request)과 응답(response) 파라미터들은 유사하나, 클라이언트와 서버 응용 프로그램간 교환되는 질의와 결과 형식에서 차이가 있다. SRW는 SOAP 방식의 웹 서비스 프로토콜로서, 클라이언트가 서버에 요청 시와 서버에서 클라이언트로 결과를 보낼 때 모두 SOAP 형식으로 처리한다. SRU는 REST 방식의 웹 서비스 프로토콜로서 클라이언트가 서버에 요청시에는 URL 형식으로 파라미터가 인코딩되어 HTTP GET 요청 방식으로 보내지고 서버가 클라이언트로 결과를 보낼 때에는 SOAP의 envelope 부분을 제외한 XML 형식만을 전달한다.

SRW/U는 3가지 기능인 Explain, Scan, SearchRetrieve을 가지며 이를 통한 주요한 특징으로는 모듈성, 확장성, 신뢰성 등이 뛰어난 무상태 연결(stateless connection)과

사람이 이해하기 쉬운 질의어로 CQL(Common Query Language)을 사용하고 있으며 레코드 구문은 XML만을 사용하고 있다.

3.4 OAI-PMH

OAI-PMH(Open Archives Initiative Protocol for Metadata Harvesting)은 REST 기반의 웹서비스의 프로토콜이다. OAI-PMH은 개방 아카이브(Open Archive)의 메타 데이터를 수집하기 위한 프로토콜이며, 웹 기반 자원들에 대한 상호운용성을 고려하는 것이다. OAI-PMH는 표준 기술인 XML, HTTP, Dublin Core에 기반을 두고 있다[5].



(그림 4)OAI-PMH 프로토콜 구조

OAI-PMH는 (그림 4)와 같이 이용자가 정보를 검색하고 활용하는 영역인 이용자(User/Application) 계층, 실제 콘텐츠에 대해 수집과 보유하고 있는 서비스 제공자의 요청에 대응하는 적합한 메타데이터를 제공하는 데이터 제공자(DP;Data Provider)계층, DP로부터 수확한 메타데이터를 기반으로 부가가치 서비스를 제공하는 서비스 제공자(SP;Service Provider)계층으로 이루어져있다. OAI-PMH는 SP의 요청과 DP의 응답에 의해 이용자가 원하는 자료를 검색하고 획득할 수 있으며, SP의 요청은 HTTP 표준(GET,POST방식)을 사용하고, DP의 응답은 XML로 인코딩되어 SP에 전달되는 방식이다.

4. 결론

본 논문에서는 효율적인 표준 기술로 평가를 받는 웹서비스의 프로토콜인 REST, SOAP, SRW/U, OAI-PMH 프로토콜을 비교하고 <표 2>에서 웹서비스 프로토콜의 장단점을 분석하였다. 분산 컴퓨팅 환경에서 정보를 교환하기 위한 XML 기반 웹 서비스 프로토콜인 SOAP은 복잡한 트랜잭션 처리의 적합성을 가지고 표준화를 통한 확장성이 뛰어나고 REST는 SOAP에 비해 단순하고 메세징이 가벼운 장점을 통하여 OpenAPI활용으로 매쉬업 서비스를 창출할 수 있다. OAI-PMH는 디지털 콘텐츠의 개방형 지식 저장소 구축 시 사용하여 메타데이터를 수확하고, 정보검색을

위한 SRW/U를 적용함으로써 시스템의 상호 운용성, 정보공유 및 검색의 효율성을 크게 높일 수 있게 될 것이다. 이러한 웹 서비스의 프로토콜 분석의 근간을 활용하여 좀 더 활용적이고 효율성 있는 프로토콜로 발전 할 것으로 기대된다.

**참고문헌**

[1] David Booth, Hugo Haas. et al., “Web Services Architecture”, W3C Standard, <http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/>, February 2004.  
 [2] Lori MacVittie , “REST(Representation State Transfer): Tasking a REST From SOAP”, <http://www.networkcomputing.com/other/rest-representational-state-transfer.php/>, 2006.  
 [3] “SOAP Tutorial”, W3C, [http://www.w3schools.com/soap/soap\\_intro.asp](http://www.w3schools.com/soap/soap_intro.asp)  
 [4] Library of Congress, “SRU/SRW(Search and Retrieve URL/Web Service)”, <http://www.loc.gov/standards/sru/>, August 2009.  
 [5] “The Open Archives Initiative Protocol for Metadata Harvesting”, Open Archives Initiative <http://www.openarchives.org/AI/openarchivesprotocol.html/>, 2004.

<표 2> 웹서비스 프로토콜의 장단점

| 구분      | 장점   | 단점  |
|---------|--|---|
| REST    | <ul style="list-style-type: none"> <li>· 언어, 플랫폼 중립적</li> <li>· SOAP보다 개발이 용이함</li> <li>· 매쉬업을 통한 새로운 웹서비스 개발이 용이함</li> <li>· 분산 컴퓨팅 응용으로</li> <li>· 단순한 데이터 포맷</li> <li>· 가벼운 프로그래밍 모델</li> </ul> | <ul style="list-style-type: none"> <li>· 분산 컴퓨팅 환경에서 메시지가 중간 경유지를 여러 번 통과하는 경우 사용하기 어려움</li> <li>· 보안, 정책, 안정적인 메시지 전달을 지원하기 위한 표준이 부족함</li> <li>· 복잡한 서비스 개발에 부적합</li> <li>· 보안이 상대적으로 취약</li> </ul> |
| SOAP    | <ul style="list-style-type: none"> <li>· 언어, 플랫폼, 전송 중립적</li> <li>· 분산 컴퓨팅 환경에서 사용하기 위한 디자인</li> <li>· 표준화 되어있으며 확장성이 뛰어나며 다른 Vendor를 통한 지원이 이루어짐</li> </ul>                                       | <ul style="list-style-type: none"> <li>· XML태그 형태로 인하여 COBRA같은 미들웨어 기술과 비교하여 상대적으로 느림</li> <li>· REST보다 어렵고 무거움</li> <li>· 개발 난이도가 높고 Tool이 필요한 경우가 많음</li> <li>· 지원 환경 구축에 비용소요</li> </ul>           |
| SRW/U   | <ul style="list-style-type: none"> <li>· 분산된 데이터의 검색이 강력하고 상호운용성이 뛰어나</li> <li>· 지속적인 결과물 셋을 통합하여 클라이언트가 원하는 레코드를 정확하게 검색</li> <li>· 규모가 크고 주제가 광범위한 데이터 셋에 적합하다.</li> </ul>                       | <ul style="list-style-type: none"> <li>· 요청과 응답 시 모두 버전을 명시하고 확인하여야함</li> <li>· 검색을 위한 질의어 작성에 많은 노력 필요</li> </ul>  |
| OAI-PMH | <ul style="list-style-type: none"> <li>· 분산된 데이터의 검색이 강력하고 상호운용성이 뛰어나</li> <li>· 데이터의 집적과 분배가 용이함</li> <li>· 데이터 중복을 검색하여 메타데이터 수집가능</li> </ul>  | <ul style="list-style-type: none"> <li>· 수집된 레코드에 대해서 예전 데이터로 돌아갈 수 없다.</li> <li>· 데이터수집에서 불필요한 정보까지도 수집됨</li> <li>· 수집된 자료를 대상으로 이용자가 실제 필요한 정보 검색 불가능</li> </ul>                                    |