

# SOA 기반 웹서비스의 Library 2.0 적용방안에 관한 연구

## A Study on Applying Library 2.0 Using Service-Oriented Architecture and Web Services

구 중 역(Jung-Eok Gu)\*

이 응 봉(Eung-Bong Lee)\*\*

### < 목 차 >

- |                   |  |
|-------------------|--|
| I. 서 론            | IV. 웹서비스 구축사례 분석                         |
| II. SOA와 웹서비스 개요  | 1. 출판유통정보시스템                             |
| 1. 서비스 지향 아키텍처    | 2. NCSU Catalog Availability Web Service |
| 2. 웹서비스           | V. Library 2.0 적용방안                      |
| III. 웹서비스 프로토콜 분석 | 1. 국내의 Library 2.0 적용사례                  |
| 1. SOAP           | 2. 메타검색 시스템 구축 및 활용방안                    |
| 2. REST           | VI. 결 론                                  |
| 3. SRW/U          |  |
| 4. OAI-PMH        |  |

### 초 록

본 연구에서는 SOA와 웹서비스의 개념을 살펴보고, SOA를 구현하는 웹서비스의 주요 프로토콜인 SOAP, REST, SRW/U, OAI-PMH에 대한 기술을 분석하였으며, 국내의 도서관에서 SOAP과 REST 기반 웹서비스 구축사례를 각각 제시하였다. 또한 국내의 도서관에서 Web 2.0 기술을 활용한 Library 2.0 적용사례와 SOA 기반 웹서비스 프로토콜의 Library 2.0 적용방법을 제시하였다. 마지막으로 REST 기반 Open API를 이용하여 도서관 OPAC, 온라인서점, 웹 포털, 종합목록 등을 결합한 메타검색 시스템의 구축과 활용을 통한 Library 2.0 적용방안을 제안하였다.

주제어: 서비스 지향 아키텍처, 웹서비스, 웹 2.0, 도서관 2.0, 온라인열람목록

### ABSTRACT

This study examined the main concepts of SOA and Web Services, and conducted technical analysis of major Web Services protocols such as SOAP, REST, SRW/U, and OAI-PMH that realize SOA, and suggested examples of SOAP and REST based Web Services establishments in foreign and domestic libraries respectively. In addition, it introduced application examples of Library 2.0 using Web 2.0 technology in foreign and domestic libraries and suggested how to apply SOA based Web Services protocol in Library 2.0. Finally, it proposed how to put Library 2.0 into practice applying meta-searching system integrating library OPAC, online bookstores, web portal sites and library union catalogs using REST based Open API.

Key Words: Service-Oriented Architecture, SOA, Web Services, Web 2.0, Library 2.0, OPAC, Open API

\* 한국기초과학지원연구원 연구정책부 팀장(jekoo@kbsi.re.kr) (제1저자)

\*\* 충남대학교 사회과학대학 문헌정보학과 부교수(eblee@cnu.ac.kr) (공동저자)

• 접수일: 2007년 8월 25일 • 최초심사일: 2007년 8월 3일 • 최종심사일: 2007년 9월 21일

## I. 서론

1989년 Tim Berners-Lee에 의해 개발된 웹은 HTTP, HTML, URL 등의 표준 기술을 사용하여 분산환경에서 정보자원들의 공유와 호환을 가능하게 해주었다. 현재 웹은 종전에 시스템과 사용자를 연결하는 단순한 도구가 아니라 '개방, 공유, 참여'의 특징을 갖고 있는 웹 2.0 시대를 맞고 있다. 또한 끊임없이 새로운 표준과 기술이 개발되면서 지능형의 시맨틱 웹(Semantic Web), 6A 기반의 유비쿼터스 웹(Ubiquitous Web), 모바일 웹(Mobile Web) 등으로 급속히 진화해가고 있다.

1996년 4월 세계적인 IT 컨설팅업체인 Gartner에 의해 처음 소개된 서비스 지향 아키텍처(Service-Oriented Architecture, 이하 SOA)는 서비스를 공유하고 재사용할 수 있도록 정보시스템을 구축하는 소프트웨어 설계 방법론이다.

SOA는 이미 서비스 지향의 접근 방법인 CORBA, DCOM, RMI 등의 분산 객체 기술에서도 그 기본 개념이 사용되었다. 그러나 이들 기술은 기술적인 미성숙 및 공개 표준의 부재와 주요 소프트웨어 벤더들 간의 협력 부재로 인해 그리 많은 주목을 받지 못하였다.<sup>1)</sup>

최근 SOA는 1999년에 공개된 표준인 XML 기반 웹서비스(Web Services) 기술의 등장과 발전에 힘입어 서비스의 관점에서 소프트웨어 아키텍처를 설계하는 방법론으로 많은 각광을 받고 있다. 실제로 SOA를 구현하는 기술인 웹서비스는 XML, HTTP, SOAP(Simple Object Access Protocol), WSDL(Web Services Description Language), UDDI(Universal Description, Discovery and Integration) 등의 표준 메시지 포맷과 프로토콜을 사용하는 서비스 중심의 기술이다.

일반적으로 웹서비스를 구현하는 방식에는 SOAP, REST(Representational State Transfer), XML-RPC(Remote Procedure Call) 등이 있으나, 이 중에서 HTTP를 통해 XML 데이터를 제공하는 단순하고 가벼운(lightweight) 프로그래밍 모델인 REST 방식을 가장 많이 사용하고 있다.

아마존은 토이즈러스(ToysRUs) 등과 같은 소매 파트너를 묶는 복잡한 고가치(high value)의 B2B 연결에서는 SOAP을 사용하고 있지만, 전체 웹서비스의 95%는 단순한 REST 인터페이스를 사용하고 있다.<sup>2)</sup> 아마존이 제공하는 웹서비스는 REST가 SOAP와 비교하면 20%의 노력으로 SOAP의 80%에 해당하는 이점을 얻을 수 있다고 한다.<sup>3)</sup> 구글의 경우에도 2006년 12월초에 SOAP 기반 Search API 웹서비스를 중단하고, REST 기반의 AJAX Search API를 대안으로 채택하였다.<sup>4)</sup>

1) 이경하, 이규철, "SOA(Service-Oriented Architecture)와 웹 서비스," 정보과학회지, 제22권 제10호(2004.10), p.5.

2) Tim O'Reilly, *What Is Web 2.0*, 2005.  
〈<http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html>〉 [cited 2007. 4. 15].

3) Tim Anderson, *WS-\* vs the REST*, 2006.  
〈[http://www.regdeveloper.co.uk/2006/04/29/oreilly\\_amazon/](http://www.regdeveloper.co.uk/2006/04/29/oreilly_amazon/)〉 [cited 2007. 4. 15].

디지털도서관에서는 분산자원에 대한 접근과 검색을 위해 HTTP, SOAP, REST, SRW/U (Search and Retrieve Web Service/URL Service), OAI-PMH 등 다양한 표준 프로토콜을 사용하여 SOA 기반의 웹서비스를 제공하는 사례가 점차 늘어나고 있다. 특히 국외에서는 디지털도서관 구축 시 ZING(Z39.50-International: Next Generation) Initiative를 통해 개발된 SRW/U 프로토콜을 이용한 웹서비스가 널리 구현되고 있다. 반면 아직 국내 도서관에서는 디지털도서관 시스템 구축 시 SOA와 웹서비스의 적용과 활용은 초보적인 단계에 머무르고 있다.

한국정보산업연합회의 「2007년 IT산업 이슈 및 전망」에 따르면 시스템 통합(SI) 및 SW 산업의 서비스·기술 분야 키워드 중 SOA(52.4%), 유비쿼터스 컴퓨팅(49.7%), 웹 2.0(49.7%) 등의 순서로 가장 중요하게 평가받고 있는 것으로 나타났다.<sup>5)</sup> 그리고 Gartner의 Hype Cycle<sup>6)</sup>에서 SOA는 현재 3단계인 '기술에 대한 환상 소멸(trough of disillusionment)' 시기에 접어들었으며, 향후 2~5년 내에 주요 기술 중의 하나로 보급될 것으로 예상되고 있다. 또한 SOA는 2007년 새로 개발되는 주요 업무 어플리케이션 및 비즈니스 프로세스 설계에 50% 이상 사용되고, 2010년경에는 80% 이상 사용될 것으로 전망되고 있다.<sup>7)</sup> 한편 Gartner는 웹 2.0 기술 중 AJAX와 매쉬업이 향후 10년간 기업에 가장 큰 영향을 준다고 전망하였다.<sup>8)</sup>

디지털도서관시스템은 이기종의 분산시스템들을 손쉽게 통합하고, 컴포넌트 기반으로 웹서비스를 개발하여 정보시스템 구축의 비용과 시간을 크게 절감하며, 디지털도서관 시스템간의 상호운용성과 정보검색의 효율성을 확보하여 Library 2.0을 구현하는 것이 필요하다.

본 연구는 SOA와 웹서비스의 개념을 살펴보고, SOA를 구현하는 웹서비스의 주요 프로토콜인 SOAP, REST, SRW/U, OAI-PMH에 대한 기술을 분석하였으며, 국내의 도서관에서 SOAP과 REST 기반 웹서비스 구축사례를 각각 제시하였다. 또한 국내의 도서관에서 Web 2.0 기술을 활용한 Library 2.0 적용사례와 SOA 기반 웹서비스 프로토콜의 Library 2.0 적용방법을 제시하였다. 마지막으로 REST 기반 Open API를 이용하여 도서관 OPAC, 온라인서점, 웹 포털, 종합목록 등을 결합한 메타검색 시스템의 구축과 활용을 통한 Library 2.0 적용방안을 제안하였다.

4) Google, *Google AJAX Search API*, 2007. <<http://code.google.com/apis/ajaxsearch/>> [cited 2007. 4. 15].

5) 한국정보산업연합회 홈페이지, <<http://www.fkii.or.kr/>> [인용 2007. 4. 15].

6) Hype Cycle: Gartner가 가트너가 새로운 기술의 진화 트렌드를 설명하기 위해 개발한 것으로 시간의 경과에 따라 신기술의 성숙도(maturity)와 업계에 회자되는 가시성(visibility)을 이용해 기술의 진화를 5단계(초기, 피크, 하강, 점진적 상승, 안정기)로 설명하고 있다.

7) Gartner, *Predicts 2007: SOA Advances*, 2006.

<<http://www.gartner.com/DisplayDocument?id=498449>> [cited 2007. 4. 15].

8) Gartner, *Gartner's 2006 Emerging Technologies Hype Cycle Highlights Key Technology Themes*, 2006. <<http://www.gartner.com/>> [cited 2007. 4. 15].

## II. SOA와 웹서비스 개요

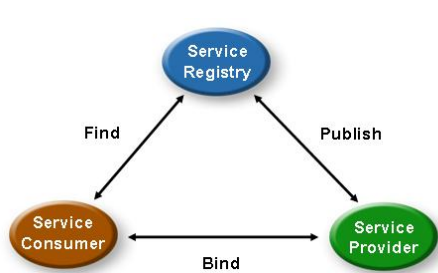
### 1. 서비스 지향 아키텍처

W3C는 SOA를 “호출(involve)이 가능한 컴포넌트의 집합”으로 정의하고 있으며, 여기서 컴포넌트는 “인터페이스(interface)의 정의가 공개(publish), 발견(discovery)이 가능한 것”을 의미하고 있다.<sup>9)</sup> Gartner는 SOA를 “잘 정의된 인터페이스를 가진, 재사용이 가능한 일련의 컴포넌트들로 구축되는 기술구조 방식”이라고 정의하고 있다.<sup>10)</sup>

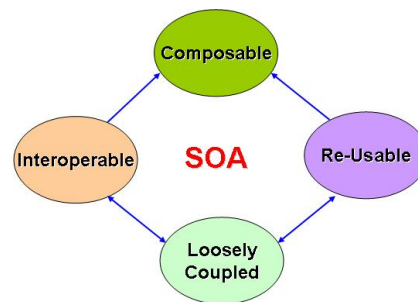
SOA에 대한 정의는 서로 다르지만 SOA는 ‘공유할 수 있고, 재사용할 수 있는 서비스라는 컴포넌트를 기반으로 정보시스템을 구축하는 아키텍처 모델’이라고 할 수 있다. SOA의 관점에서 서비스는 독립적(self-contained)이고, 인터페이스를 통해 자신이 가진 비즈니스 프로세스를 처리할 수 있는 컴포넌트로 정의되며, 비즈니스 프로세스와 서비스는 시간이 흐르면서 변하는 특성을 갖는다.

SOA는 <그림 1>과 같이 서비스 사용자(Service Consumer), 서비스 제공자(Service Provider), 서비스 중계자(Service Registry)로 구성되어 있다. SOA의 기본 아키텍처는 XML, HTTP, SOAP, WSDL, UDDI 등 표준 기술을 기반으로 하는 웹서비스의 아키텍처와 일치한다.

서비스 사용자는 서비스 제공자에 의해 제공되고 있는 하나 이상의 서비스를 사용한다. 서비스 제공자는 서비스 사용자가 호출 시 입력하는 값을 가공하여, 그에 해당하는 결과를 제공한다. 경우에 따라서 서비스 제공자는 다른 서비스 제공자의 서비스를 사용하는 서비스 사용자가 될 수도 있다. 서비스 중계자는 서비스에 대한 설명 정보를 저장하고 있다. 서비스 제공자는 자신이 제공하고 있는 서비스를 등록하고, 서비스 사용자는 자신의 원하는 서비스를 발견하여 사용한다.



<그림 1> SOA 아키텍처



<그림 2> SOA가 추구하는 네 가지 목표<sup>11)</sup>

9) W3C. *Web Services Glossary*, 2004. <<http://www.w3.org/TR/ws-gloss/>> [cited 2007. 4. 15].

10) Gartner Home page, <<http://www.gartner.com/>> [cited 2007. 4. 15].

11) 이상민, “비즈니스 신속성, SOA가 책임진다,” Oracle Korea Magazine,(Summer 2005), p.34.

〈그림 2〉와 같이 SOA에서 추구하는 목표는 플랫폼간의 상호운용성, 느슨한 연결, 기존 시스템의 재사용성, 기존 서비스를 이용하여 새로운 서비스를 생성할 수 있는 능력을 갖추는데 있다. 그리고 SOA에 가장 중요한 특징인 느슨한 연결(loosely coupled)은 ‘잘 정의된 인터페이스들을 사용해 서비스들을 연결하는 것’을 말한다. 특히 SOA는 플랫폼에 독립적이고, 프로그래밍 언어에 중립적이며, 표준 기술을 준수하여 상호운용성과 재사용성을 증가시켜 주고 있다.

한편 SOA는 〈표 1〉과 같이 웹 기반 표준 기술인 웹서비스를 활용하여 새로운 비즈니스를 창출한다는 측면에서 웹 2.0과 매우 유사한 특징을 갖고 있으며, 궁극적으로 웹 2.0이 글로벌 차원의 SOA를 실현할 수 있을 것으로 전망되고 있다.<sup>12)</sup>

〈표 1〉 Web 2.0과 SOA간 개념비교<sup>13)</sup>

구 분	Web 2.0	SOA
서비스 모델	· 웹서비스	· 웹서비스
선호하는 서비스 표준	· HTTP, XML, RSS, REST	· WSDL, UDDI, SOAP, BPEL, WS
재사용성	· 매우 높음	· 약간 높음
유연성 및 순응성	· 매우 높음 · 단순한 데이터 포맷 · 가벼운 프로그래밍 모델	· 높음(보다 더 공식적) · 조합과 통합(Composition and Orchestration) · 서비스 지향의 원칙
비즈니스 모델	· 롱테일(Long Tail) 효과 · 네트워크 효과 · 집단지능 활용 · 고객 셀프서비스	· BPM(Business Process Management) · 자산 통합(Asset Integration) · 데이터 퓨전(Data Fusion) · 레거시 자산의 생명주기 연장 · 비즈니스 활동 모니터링 · 비즈니스 지능(Biz Intelligence) 활용
설계 플랫폼	· AJAX · 신디케이션(Syndication) · 멀티 디바이스 소프트웨어	· Service Layer · Service Bus · Unit of Work
핵심역량	· 서비스로서의 SW(SaaS) · 데이터 소스에 대한 통제 · 공동개발자로서 사용자 신뢰 · 집단지능 이용 · 롱테일 효과 · 단일 디바이스(PC 플랫폼)를 넘어선 소프트웨어 · 가벼운(Lightweight) UI, 개발 모델, 비즈니스모델 채용	· 기능의 재정비 · 자산(Asset)으로서 데이터 · 접근가능성 · 시스템/데이터 통합 · 비용절감 · 비즈니스 기민성(Agility) · B2B 셀프서비스 · 오픈 스탠다드(Open Standard) · 온톨로지(Ontologies) · 오퍼레이션의 투명성 · 소비자 중심의 비즈니스 프로세스

\* 출처: SOA Web Service Journal, 2006.

12) 지은희, “SOA가 바뀌놓을 세상,” SW Insight 정책리포트, 제10호(2006. 5), p.77.

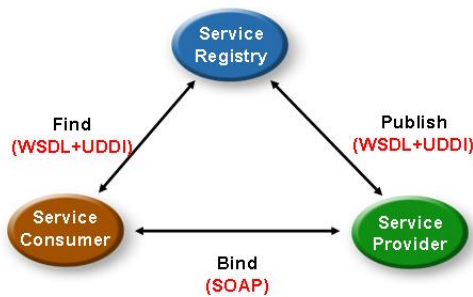
13) 지은희, 2006. 전개논문. 재인용.

## 2. 웹서비스

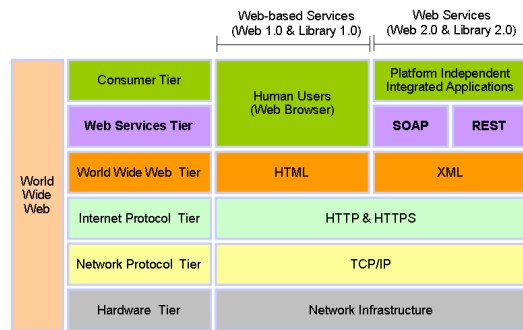
IBM은 웹서비스를 “표준화된 XML 메시지를 사용하여 네트워크를 통해 접근할 수 있는 프로그램의 모음을 기술하는 인터페이스”라고 정의하고 있다.<sup>14)</sup> Gartner는 웹서비스를 “분산 컴퓨팅을 위해 SOAP, WSDL, UDDI 등의 웹서비스 표준 기술 중 한 가지 이상을 적용한 소프트웨어 컴포넌트”라고 정의하고 있다.<sup>15)</sup> W3C에 따르면 웹서비스는 “네트워크 상에서 기계간에 상호작용을 지원하기 위하여 고안된 소프트웨어 시스템이고, 웹서비스는 기계가 처리 가능한 형태로 기술된 인터페이스(WSDL)를 가지며, 다른 시스템들은 주로 HTTP 프로토콜 상에서 SOAP 메시지를 사용하여 인터페이스에 명시된 방식으로 그 웹서비스와 상호작용”하게 된다.<sup>16)</sup>

웹서비스의 기본 아키텍처는 <그림 3>과 같이 플랫폼에 독립적으로 어플리케이션간 통신을 할 수 있도록 검색(find), 연결(bind), 등록(publish)의 세 가지 오퍼레이션을 정의하고 있다.

- 검색(find): 서비스 요청자들은 UDDI 레지스트리에서 필요한 웹서비스를 검색한다.
- 연결(bind): 서비스 제공자들은 자신들의 내용을 요청자들에게 SOAP이라는 방식을 통해 바인드해 준다.
- 등록(publish): WSDL을 비롯한 웹서비스의 상세 내용을 UDDI 레지스트리에 등록한다.



<그림 3> 웹서비스 아키텍처



<그림 4> 웹서비스 스택<sup>17)</sup>

14) K. Gottschalk et al., "Introduction to Web services architecture.", *IBM Systems Journal*, Vol.41, No.2(2002), pp.170-177.  
 15) Seeley, Rich. *Gartner offers another Web services definition*, 2003. <<http://www.adtmag.com/article.aspx?id=7675>> [cited 2007. 4. 15].  
 16) W3C. *Web Services Glossary*, 2004. <<http://www.w3.org/TR/ws-gloss/>> [cited 2007. 4. 15].  
 17) Talis. *Talis Keystone Technology Brief*, 2007. <[http://www.talis.com/applications/products/assets/keystone\\_technology\\_brief.pdf](http://www.talis.com/applications/products/assets/keystone_technology_brief.pdf)> [cited 2007. 4. 15].  
 채수정.

〈그림 4〉와 같이 웹서비스는 기존 HTML 기반의 웹 서비스와는 달리 XML 기반의 표준화된 언어와 프로토콜에 의해 사용자가 시간, 장소, 단말기에 구애받지 않고 원하는 기능 및 서비스를 이용할 수 있는 유비쿼터스 환경의 핵심 요소가 될 것으로 보인다.

SOA를 구현하기 위한 웹서비스의 기본 표준은 다음과 같다.

- XML: 웹서비스에서 가장 중요한 기반 기술이고, 데이터의 정의, 공유 및 교환에 사용되며, 웹서비스의 기능 호출 및 결과 등이 모두 XML 문서로 전송된다.
- HTTP: 웹서비스의 전송 프로토콜로 웹서비스에서 사용되는 XML 메시지를 전달하는 역할을 한다.
- SOAP: 서로 다른 소프트웨어 컴포넌트들을 연결하는 역할을 하는 프로토콜로 XML과 HTTP를 사용해 플랫폼과는 상관없이 서버와 서비스에 접근할 수 있다.
- WSDL: 특정 비즈니스가 제공하는 서비스를 설명하고, 개인이나 다른 회사들이 그러한 서비스에 전자적으로 접근할 수 있는 방법을 제공하기 위해 사용되는 XML 기반의 언어이다.
- UDDI: 웹서비스와 비즈니스 프로세스 등을 등록하고 실시간으로 조회할 수 있도록 해주는 공용 디렉토리이면서 프로토콜의 집합체이다.

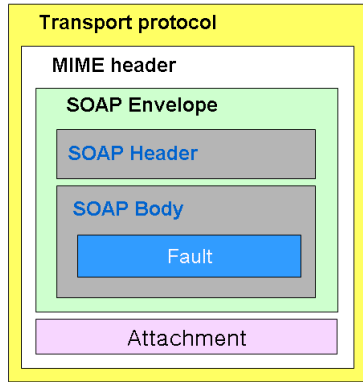
한편 웹서비스 관련 기술들은 크게 W3C, OASIS, WS-I(Web Service Interoperability) 등에 의해 표준화되고 있다.

### Ⅲ. 웹서비스 프로토콜 분석

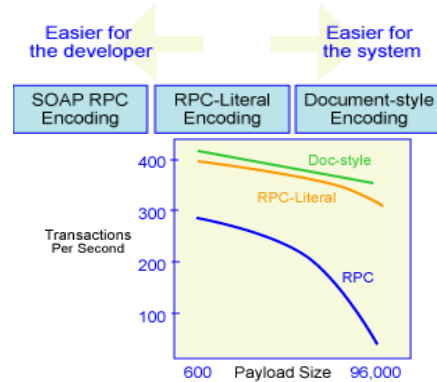
#### 1. SOAP

SOAP은 분산환경에서 정보를 교환하기 위한 가벼운 표준 XML 프로토콜이다. SOAP은 XML-RPC로 1998년에 시작되었으며, RPC는 네트워크 상의 다른 컴퓨터의 프로그램에게 서비스를 요청할 때 사용하는 프로토콜이다. SOAP은 웹서비스에서 응용 프로그램 사이에 공유되는 데이터 형식을 정의하는 프로토콜로 사용된다.

SOAP 메시지의 구조는 〈그림 5〉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. 아울러 SOAP 메시지는 이메일과 같이 MIME 타임을 사용하여 하나 이상의 첨부파일을 보낼 수 있다.<sup>18)</sup>



<그림 5> SOAP 기본 구조



<그림 6> SOAP 인코딩 스타일

SOAP의 기본적인 기능은 다음과 같다.

- SOAP Envelop(필수): SOAP에서 사용하는 요소를 위한 네임스페이스 선언과 메시지에 쓰이는 데이터를 위한 인코딩 스타일 속성을 정의한다. 여기서 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 호출 내용을 XML로 인코딩하거나 메시지 안에 오류 혹은 상태 정보(fault)를 전달하는데 사용한다.

<그림 6>과 같이 SOAP 인코딩 스타일은 크게 RPC 인코딩, RPC-literal 인코딩, Document-style 인코딩의 세 가지 방식이 있으며, Document-style이 늘어나는 페이로드(payload) 사이즈에도 비교적 안정적인 성능을 보여주고 있다. 따라서 SOAP 인코딩 스타일은 시스템 확장성과 신뢰성에 큰 영향을 미칠 수 있고 SOAP 인코딩 스타일에 따라 성능에 큰 차이를 보이기 때문에 웹서비스 개발 시 어떤 인코딩 방식을 사용할 것인지 신중히 선택해야 한다.<sup>19)</sup>

## 2. REST

REST는 Roy Fielding<sup>20)</sup>이 2000년에 네트워크 시스템의 아키텍처 스타일을 설명하기 위해 “네

18) W3C, *SOAP Tutorial*, <<http://www.w3schools.com/soap/default.asp>> [cited 2007. 4. 15].

19) Cohen, Frank, *Understanding SOAP Encoding Impact on Web Service Performance in WebLogic Workshop*, 2003. <<http://www.dev2dev.co.kr/pub/a/2003/03/Cohen.jsp>> [cited 2007. 4. 15].

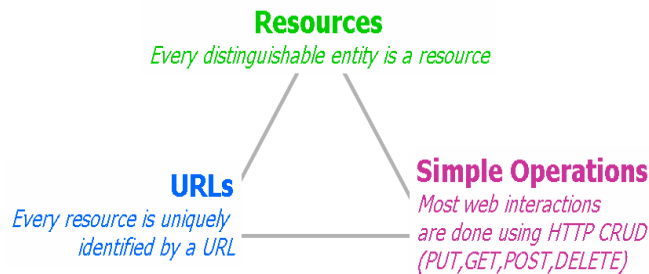
20) Fielding, Roy Thomas, *Architectural Styles and the Design of Network-based Software Architectures*, Doctorial



트위크 기반 소프트웨어 설계의 아키텍처 스타일과 디자인”이라는 박사학위 논문에서 최초로 제안한 용어이다. REST는 기존의 웹 어플리케이션(machine to human) 뿐만 아니라 웹서비스(machine to machine)를 위한 아키텍처 스타일이다.

REST는 그 자체가 표준 또는 규격이 아니라, 현재의 웹과 같이 '분산된 하이퍼미디어 시스템을 위한 소프트웨어 아키텍처 스타일'을 말한다. 분산컴퓨팅 모델인 REST는 인터넷의 표준인 URI, HTTP, XML, RDF 등을 사용하고 있어 매우 큰 확장성을 갖고 있다.

〈그림 7〉과 같이 REST 스타일의 웹서비스는 모든 서비스 대상을 '리소스'라는 개념으로 묶어 연결하는 리소스 지향(resource-oriented)의 서비스이다. REST에서 리소스의 식별은 URL을 사용하고, 리소스의 전달은 HTTP를 통해 이루어지며, 리소스의 내용은 XML로 기술된다. 또한 리소스에 대한 오퍼레이션은 HTTP 메소드(GET, POST, PUT, DELETE)를 사용한다. 또 REST에서 가장 강조되고 있는 리소스는 정보이다. 즉 리소스는 인터넷 식별자인 URI를 가지며, 리소스의 상태(state)는 시간이나 조건에 따라 변화될 가능성이 있다. 그러나 리소스의 의미는 시간이나 조건이 변화되어도 변함이 없다. 또한 리소스간의 관계는 하이퍼링크로 서로 연결되고, 그 기능 확장에는 XML 네임스페이스를 사용한다.



〈그림 7〉 REST 디자인 패턴<sup>21)</sup>

REST는 웹상의 모든 정보는 URL로 표현되고, 해당 URL이 표현하고 있는 상태에서 다른 URL의 상태로 이동하게 되며, 잘 정의된 URL을 통해 웹 어플리케이션을 실행시켜 그 결과(상태)를 전달받아 처리한다.

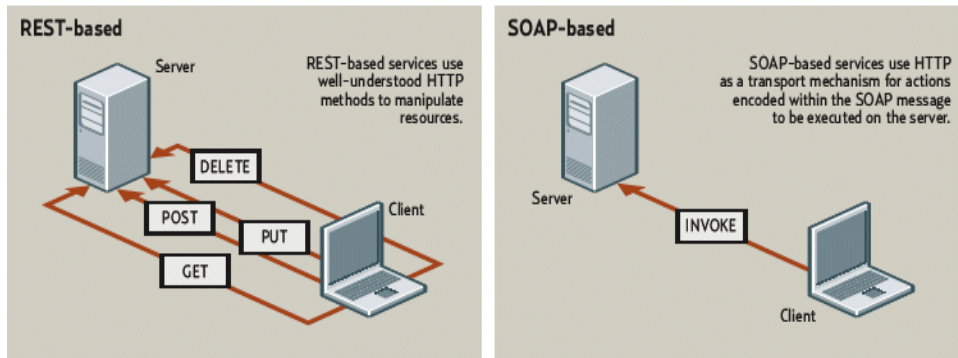
〈그림 8〉은 REST와 SOAP 기반의 검색 기법을 도식화한 것이고, 〈표 2〉는 출판유통진흥원과 아마존에서 제공하는 SOAP 기반 웹서비스 명세서인 WSDL과 아마존에서 도서 검색을 위해 제

Dissertation, University of California, Irvine, 2000.

21) Jaesun Han, *Introduction and Analysis of Web 2.0 Technologies*, 2006.

〈<http://www.web2hub.com/blog/jshan/attachment/1149006439.ppt>〉 [인용 2007. 4. 15].

공하는 REST 기반의 간단한 검색 요청 사례를 제시한 것이다.



〈그림 8〉 검색 기법: REST vs SOAP<sup>22)</sup>

〈표 2〉 SOAP과 REST 기반 웹서비스 적용사례

구분	SOAP	REST
출판유통 진흥원	http://220.121.151.5/Kpd3rd.Facade/KpdWebServices.asmx?WSDL	-
아마존	http://soap.amazon.com/schemas2/AmazonWebServices.wsdl	http://webservices.amazon.com/onca/xml?Service = A WSECommerceService & AWSAccessKeyId = 061X24C38W0FZ10KD482 & Operation = ItemSearch & SearchIndex = Books & Keywords = SOA

### 3. SRW/U

ANSI/NISO Z39.50은 이기종 컴퓨터간의 표준화된 정보검색 프로토콜로써 도서관의 문헌정보 검색서비스 정의 및 프로토콜 설정에 관한 국제표준이다.

Z39.50은 분산형 클라이언트/서버 구조를 기반으로 하고 있으며, ISO에서 제정한 OSI(Open Systems Interconnection) 참조모델의 응용계층에 해당되고, 네트워크 상에서 정보시스템간 호환성 및 상호운용성을 지원하고 있다. Z39.50의 주요 기능은 접속, 질의, 정보검색, 결과세트 삭제, 브라우저, 정렬, 접근제어, 자원관리, 설명, 확장 서비스, 해제 등이다.

22) MacVittie, Lori, *REST(Representational State Transfer): Taking a REST From SOAP*, 2006.  
 〈http://www.networkcomputing.com/channels/enterpriseapps/showArticle.jhtml?articleID=193005691〉  
 [cited 2007. 4. 15].

2000년대에 들어서면서 Z39.50은 ZING(Z39.50- International Next Generation) Initiative를 통해서 개발된 SRW/U 프로토콜을 적용한 디지털도서관 구축사례가 급격히 늘어나고 있다. 차세대 Z39.50인 SRW/U는 전통적인 Z39.50의 개념과 특징을 유지하면서 웹 환경에 적합한 표준 기술(XML, HTTP, SOAP, REST 등)을 사용하여 개발한 웹서비스 프로토콜이다.

SRW/U의 주요한 특징은 클라이언트/서버의 연결이 모듈성, 확장성, 신뢰성 등이 뛰어난 무상태 연결(stateless connection)이고, 질의어는 사람이 이해하기 쉬운 CQL(Common Query Language)을 사용하고 있으며, 레코드 구문은 XML만을 사용하고 있다. SRW/U는 Explain, Scan, SearchRetrieve 등의 세 가지 기능을 제공하고 있다.<sup>23)</sup>

〈표 3〉은 전통적인 Z39.50과 SRW/U 웹서비스 프로토콜의 차이점을 비교한 것이다.

〈표 3〉 Z39.50과 SRW/U 비교분석

구분	Z39.50	SRW/U
연결 상태	Connections, Sessions, Stateful	Connectionless, Stateless
정보 전송	Sockets	HTTP
레코드 구문	복수(MARC, CCF, GRS-1 등)	단일(XML)
질의 방식	속성 벡터	단순 색인
질의 언어	RPN(Reverse Polish Notation)	CQL(Common Query Language)
검색 대상	데이터베이스	서버
정보 표현	ASN-1	XML/WSDL
인코딩 기술	BER(Basic Encoding Rules)	XML

SRW는 SOAP 기반의 웹서비스 프로토콜로써 클라이언트/서버간에 SOAP 메시지의 요청과 응답은 주로 HTTP POST 메소드를 사용하여 전송되며, 데이터 인코딩은 XML을 사용한다. 반면에 SRU는 REST 기반의 웹서비스 프로토콜로써 URL 구문에 name/value 형식의 파라미터가 인코딩 되어 HTTP GET 메소드만을 사용하여 요청과 응답이 이루어지고, 응답 내용은 SOAP 메시지의 XML 형식만을 전달한다.

〈표 4〉는 SRW/U의 데이터 요청에 의해 XML로 처리된 데이터 응답 예시이다.

23) Library of Congress, *SRU/SRW(Search and Retrieve URL/Web Service)*,  
 〈<http://www.loc.gov/standards/sru/>〉 [cited 2007, 4. 15].

〈표 4〉 SRW/U 프로토콜의 응답(response) 예시

SRW	<pre> &lt;SOAP:Envelope xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/"&gt; &lt;SOAP:Body&gt; &lt;SRW:searchRetrieveResponse xmlns:SRW="http://www.loc.gov/zing/srw/"   xmlns:DIAG="http://www.loc.gov/zing/srw/diagnostics/"&gt; &lt;SRW:version&gt;1.1&lt;/SRW:version&gt; &lt;SRW:numberOfRecords&gt;1&lt;/SRW:numberOfRecords&gt;   &lt;SRW:records&gt;     &lt;SRW:record&gt;       &lt;dc:title&gt;Law and Technology Journal&lt;/dc:title&gt;       &lt;dc:identifier&gt;http://www.example.com/journal/&lt;/dc:identifier&gt;       &lt;! - record content -&gt;     &lt;/SRW:record&gt;   &lt;/SRW:records&gt; &lt;/SRW:searchRetrieveResponse&gt; &lt;/SOAP:Body&gt; &lt;/SOAP:Envelope&gt; </pre>
SRU	<pre> &lt;SRW:searchRetrieveResponse xmlns:SRW="http://www.loc.gov/zing/srw/"   xmlns:DIAG="http://www.loc.gov/zing/srw/diagnostics/"&gt; &lt;SRW:version&gt;1.1&lt;/SRW:version&gt; &lt;SRW:numberOfRecords&gt;1&lt;/SRW:numberOfRecords&gt;   &lt;SRW:records&gt;     &lt;SRW:record&gt;       &lt;dc:title&gt;Law and Technology Journal&lt;/dc:title&gt;       &lt;dc:identifier&gt;http://www.example.com/journal/&lt;/dc:identifier&gt;       &lt;! - record content -&gt;     &lt;/SRW:record&gt;   &lt;/SRW:records&gt; &lt;/SRW:searchRetrieveResponse&gt; </pre>

#### 4. OAI-PMH

OAI-PMH는 웹 2.0의 가벼운 프로그래밍 모델이면서 REST 기반 웹서비스 프로토콜이다. OAI-PMH는 디지털 콘텐츠의 저장소(repository)에서 메타데이터 수확(harvesting)을 통하여 메타통합검색(aggregated search)을 제공하여 디지털 콘텐츠의 효율적인 유통과 정보검색의 상호호환성을 지원하고 있다.

OAI-PMH는 표준 기술인 XML, HTTP, Dublin Core에 기반을 두고 있으며, 데이터 제공자(DP, Data Provider)와 서비스 제공자(SP, Service Provider)의 두 가지 기능으로 구성된다. DP는 디지털 콘텐츠의 저장소에서 메타데이터를 노출시켜 OAI-PMH를 지원하고, SP는 OAI-PMH를 통해 DP가 제공하는 메타데이터를 수확하고 부가서비스(검색, 브라우징, 원문제공 등)를 제공한다. OAI-PMH의 공통 메타데이터 표준은 기본적으로 Dublin Core이지만, 다양한 응용 영역에 적합한 메타데이터 표준을 사용할 수 있다.

OAI-PMH에서 사용하는 6개의 명령어(verbs)는 디지털 콘텐츠의 저장소에 관한 정보를 얻기 위한 명령어(Identify, ListSets, ListMetadataFormats)와 메타데이터의 수확을 위한 명령어(ListIdentifiers, GetRecord, ListRecords)로 구분된다. 모든 OAI-PMH 요청은 name/value 리스트로 구성된 단일 URL을 HTTP GET 또는 POST 메소드를 사용하여 전송되며, 모든 OAI-PMH 응답은 유니코드(unicode)인 UTF-8을 사용하여 XML 문서로 인코딩된다. 응답 레코드인 XML 문서는 헤더(header), 본문(body), 부가정보(about)의 세 가지로 구성된다. 특히 모든 레코드의 공통정보인 헤더는 메타데이터 형식에 독립적이며 저장소 내의 아이템에서 메타데이터를 추출하는 키(unique ID)와 아이템의 생성, 삭제, 수정일 등을 나타내는 형식으로 구성되어 있다.<sup>24)</sup>

## IV. 웹서비스 구축사례 분석

### 1. 출판유통정보시스템

출판유통진흥원은 지난 2001년부터 출판유통의 현대화를 위한 출판유통정보시스템 구축을 진행하고 있다. 출판물 유통에 필수적인 목록 메타데이터와 이미지 등 도서정보제공서비스를 신청방식에 따라 일괄제공 서비스와 웹서비스 방식으로 제공하고 있다.<sup>25)</sup>

웹서비스는 도서산업의 상품정보를 전자적인 형태로 표현하고 전달하기 위한 메타데이터인 ONIX (ONline Information eXchange) 기반 도서정보를 출판사, 서점, 유통사 등에서 각자의 홈페이지를 통해 서비스할 수 있도록 제공되고 있다. <그림 9>와 같이 SOAP 프로코토를 이용한 도서정보 조회 웹서비스를 통해 웹서비스 클라이언트에게 곧바로 전달된다.

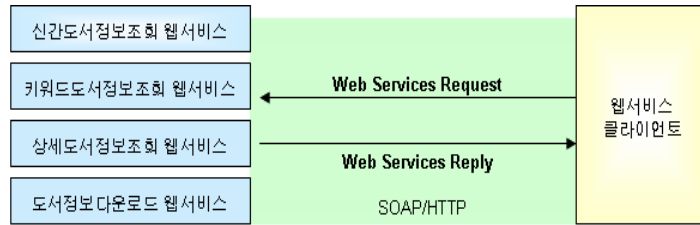
출판유통정보시스템 웹서비스를 활용한 웹서비스 클라이언트 개발 절차는 다음과 같다.

- ① 웹서비스 사용자 등록
- ② 웹서비스 인증키 발급
- ③ 웹서비스 샘플 및 개발 가이드 다운로드 및 참조
- ④ 자체시스템에 맞춰 웹서비스 클라이언트 개발

24) Open Archives Initiative, *The Open Archives Initiative Protocol for Metadata Harvesting*, 2004.

<<http://www.openarchives.org/OAI/openarchivesprotocol.html>> [cited 2007. 4. 15].

25) 출판유통진흥원 홈페이지, <<http://www.booktrade.or.kr/>> [인용 2007. 4. 15].



〈그림 9〉 출판유통정보시스템에서 제공되는 웹서비스

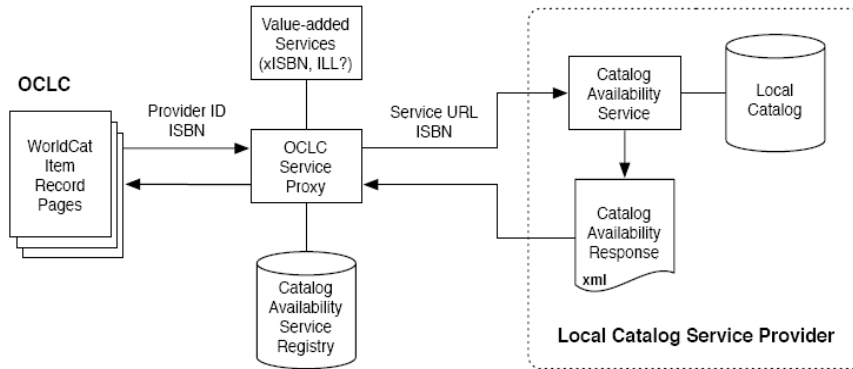
## 2. NCSU Catalog Availability Web Service

NCSU Libraries는 〈그림 10〉과 같이 Digital Library Initiative 프로젝트의 일환으로 REST 기반의 웹서비스인 'Catalog Availability Web Service'를 개발하였다.<sup>26)</sup>

Catalog Availability Web Service는 NCSU Libraries의 목록에서 특정 ISBN에 해당되는 개별 자료(item)의 이용가능한 정보를 제공한다. 여기서 개별자료의 이용가능한 정보는 'Availability for checkout, Available for in-library use only, On reserve'인 경우에 해당된다. ISBN이 알려져 있지 않거나 대출 또는 이용이 허락되지 않을 경우에는 OCLC의 xISBN 서비스를 이용하여 로컬 도서관의 OPAC에서 이용자가 질의한 ISBN과 관련된 다양한 출판 형태의 개별자료를 검색해 준다. 이를 통해 도서관 이용자들이 로컬도서관에서 현재 이용이 가능한 도서관자원을 파악하는데 도움을 제공하고 있다.

〈그림 11〉에서 제시된 특정 ISBN 번호(1875847634)는 호주 Melbourne의 Text Publishing사에서 1998년도에 출판한 Murray Bail의 "Eucalyptus" 도서이고, 연관 ISBN 번호(0374148570)는 특정 ISBN 검색으로도 이용가능성을 확인할 수 없어 OCLC의 xISBN 서비스를 이용하여 미국 New York의 Farrar Straus & Giroux사에서 1998년에 출판한 도서를 대신 검색해 준 것이다. 〈표 5〉와 같이 OCLC의 xISBN 서버에서 리턴된 XML 파일을 대상으로 동일한 저작의 ISBN 번호들을 추출하여 로컬도서관에 해당 자료가 소장되어 있는지 여부를 확인할 수 있다.

26) NCSU Libraries, *Catalog Availability Web Service*, 2006.  
 〈<http://www.lib.ncsu.edu/catalog/ws/oclc/>〉 [cited 2007. 4. 15].



<그림 10> Catalog Availability Web Services: Architecture

특정도서 ISBN: 1875847634	연관도서 ISBN: 0374148570
<p><b>ISBN: 1875847634</b></p> <p>This item is available at NORTH CAROLINA STATE UNIV <input type="button" value="Check availability"/></p> <ul style="list-style-type: none"> <li>■ <a href="#">Generate raw XML response</a> for this ISBN</li> <li>■ <a href="#">View AJAX code</a> used for this prototype</li> </ul> <p>1875847634 <input type="button" value="Change ISBN"/></p>	<p><b>ISBN: 1875847634</b></p> <p>This item is available at NORTH CAROLINA STATE UNIV <input type="button" value="Check availability"/> <a href="#">1 copy of a similar item available</a> -- D.H. Hill Library</p> <ul style="list-style-type: none"> <li>■ <a href="#">Generate raw XML response</a> for this ISBN</li> <li>■ <a href="#">View AJAX code</a> used for this prototype</li> </ul> <p>1875847634 <input type="button" value="Change ISBN"/></p>

<그림 11> Catalog Availability Web Services: Prototype

〈표 5〉 Catalog Availability Web Service의 XML 메시지 예시

NCSU Libraries	
Request	http://www.lib.ncsu.edu/catalogws/?service=availability&isbn=1875847634&output=xml
Response	<pre> &lt;?xml version="1.0" encoding="UTF-8" ?&gt; &lt;availabilityResponse xmlns="http://www.lib.ncsu.edu/catalogws/1.0"&gt;   &lt;item&gt;     &lt;isbn&gt;1875847634&lt;/isbn&gt;     &lt;totalAvailable&gt;1&lt;/totalAvailable&gt;     &lt;catalogLink&gt;http://catalog.lib.ncsu.edu&lt;/catalogLink&gt;   &lt;/item&gt;   &lt;altItem&gt;     &lt;isbn&gt;0374148570&lt;/isbn&gt;     &lt;totalAvailable&gt;1&lt;/totalAvailable&gt;     &lt;totalUnavailable&gt;0&lt;/totalUnavailable&gt;     &lt;availableLibraryName&gt;D.H. Hill Library&lt;/availableLibraryName&gt;     &lt;catalogLink&gt;http://www2.lib.ncsu.edu/catalog/?N=0&amp;Ntk=ISBN&amp;Ntt=0374148570     &lt;/catalogLink&gt;   &lt;/altItem&gt; &lt;/availabilityResponse&gt;                     </pre>
OCLC WorldCat xISBN Service	
Request	http://labs.oclc.org/xisbn/1875847634
Response	<pre> &lt;?xml version="1.0" encoding="UTF-8" ?&gt; &lt;idlist&gt;   &lt;isbn&gt;1875847634&lt;/isbn&gt; * Melbourne: Text Pub., 1998. Hardcover.   &lt;isbn&gt;0374148570&lt;/isbn&gt; * New York: Farrar Straus &amp; Giroux, 1998. Hardcover.   &lt;isbn&gt;0156007819&lt;/isbn&gt; * Melbourne: Text Pub., 1999. Paperback.   &lt;isbn&gt;1860464947&lt;/isbn&gt; * London: The Harvill Press, 1998. Paperback.   &lt;isbn&gt;1860464955&lt;/isbn&gt; * London: The Harvill Press, 1999. Paperback.   &lt;isbn&gt;1875847944&lt;/isbn&gt; * Melbourne: Text Pub., 2006. Paperback.   &lt;isbn&gt;9041402861&lt;/isbn&gt; * Amsterdam: Anthos, 1998.   &lt;isbn&gt;0676971776&lt;/isbn&gt; * Ontario: Random House of Canada, 1998. Hardcover.   &lt;isbn&gt;0676972373&lt;/isbn&gt; * Ontario: Random House of Canada, 1999. Paperback.   &lt;isbn&gt;9041405526&lt;/isbn&gt; * Amsterdam: Flamingo, 2001.   &lt;isbn&gt;963859313x&lt;/isbn&gt; * Budapest: Ulpius-ház, 1999. [Hungarian]   &lt;isbn&gt;2221087615&lt;/isbn&gt; * Ontario: Random House of Canada, 1999. Paperback.   &lt;isbn&gt;9532060065&lt;/isbn&gt; * Zagreb: Meandar, 1999. [Croatian]   &lt;isbn&gt;9657120055&lt;/isbn&gt; * Tel Aviv: Hargol, 2001. [Hebrew] &lt;/idlist&gt;                     </pre>

## V. Library 2.0 적용방안

### 1. 국내외 Library 2.0 적용사례

현재 도서관은 Web 2.0 기술을 적용하여 디지털 콘텐츠의 생산과 유통, 정보검색 및 브라우징 방식, 어플리케이션과 사용자 인터페이스, 서비스 제공방식 및 수준, 디바이스 등을 혁신할 수 있다.



즉 대표적인 Web 2.0 기술인 RSS, 블로그(Blog), 위키(Wiki), 포드캐스트(Podcast), 폭소노미(Folksonomy), 북마크릿(Bookmarklet), 툴바(Toolbar), 포드캐스트(Podcast), 오픈 API(Open API), 매쉬업(Mash-up), 아작스(AJAX), 플래쉬(Flash), 플리커(Flickr), 소셜북마크(Social Bookmark), 소셜네트워크(Social Network), 위젯(Widgets) 등을 도서관에 적용하여 Library 2.0을 구현하고 있다.

〈표 6〉은 국외 도서관에서 Web 2.0 기술의 유형별로 Library 2.0 적용사례를 제시한 것이고, 〈표 7〉은 국내 도서관의 Library 2.0 적용사례로 포항공과대학교 청암학술정보관과 성균관대학교 학술정보관이 다양한 유형의 실험적인 서비스를 제공하고 있다.

국내의 도서관에서 Web 2.0 기술을 도서관에 접목시킨 Library 2.0 적용사례와 가능한 서비스를 요약하면 다음과 같다.

첫째, 성균관대학교 학술정보관과 서울대학교 중앙도서관은 OPAC 검색결과를 RSS 피드로 제공하고 있으며, 도서관 이용자는 RSS 리더기, 블로그, 개인화 홈페이지, 모바일 서비스 등에 RSS 피드 주소를 등록해 두면 이용자가 원하는 검색결과와 업데이트된 정보를 실시간으로 확인할 수 있다. UCSC Library는 RSS 기반의 확장 기술인 포드캐스팅(podcasting)을 통해 강의 내용이 담긴 오디오 파일을 MP3 플레이어에 다운로드하여 청취할 수 있도록 하고 있다. 장차 도서관 OPAC은 이용자가 입력한 키워드의 검색결과에 해당되는 RSS 피드를 생성하고 주소를 복사하여 사용할 수 있도록 RSS 피드 생성기를 추가하여 도서관 이용자 주도의 SDI 서비스를 제공할 수 있다.

둘째, 블로그는 도서관에서 마케팅과 커뮤니케이션 수단으로 서비스형 블로그와 설치형 블로그를 사용하고 있다. 미국의 Plymouth State University Library는 설치형 블로그인 WordPress 기반으로 WPopac을 제공하고 있다. 또 Ann Arbor District Library는 도서관 웹사이트 자체를 블로그 형태로 운영하고 있다. 블로그는 RSS 공개를 통해서 커뮤니티간에 정보를 공유할 수 있다.

셋째, API(Application Programming Interface)는 응용 프로그램간에 데이터를 주고받을 수 있도록 만든 표준화된 인터페이스를 외부에 공개한 것으로 대부분 웹서비스 형태로 제공되고 있다. 일반적으로 Open API는 주로 데이터 처리가 복잡한 SOAP 프로토콜 보다 가볍고 단순한 REST 프로토콜을 사용하고 있다. 실제로 NCSU Libraries는 REST 기반 웹서비스 프로토콜을 적용한 OCLC의 xISBN Web Service와 자관의 OPAC을 결합한 'Catalog Availability Web Service' 즉 '이 책의 다른 판' 서비스를 개발하여 도서관 이용자에게 OPAC의 검색성을 높여주고 있다. 이와 같이 Open API는 다양한 형태로 공개된 웹 플랫폼들을 유기적으로 묶어주고 있다.

넷째, 성균관대학교 학술정보관은 '도서관 책 검색 2.0'을 통해서 도서관 이용자가 온라인서점에서 49개 도서관의 OPAC를 검색할 수 있는 북마크릿을 제공하고 있다. 북마크릿은 도서관 OPAC에서 ISBN 검색이 가능해야 하고, 온라인서점은 도서 상세정보의 URL에 ISBN을 반드시 포함하고 있어야 한다. 한편 도서관 시스템 벤더마다 북마크릿을 자동으로 생성해 주는 어플리케이션도 필요하다.

〈표 6〉 국외 Library 2.0 적용사례<sup>27)</sup>

구분	도서관	적용 유형	URL
RSS	農林水産研究情報センター	신착도서·저널	http://ss.cc.affrc.go.jp/ric/home.html
	廈門大學圖書館	예약도서 도착통지	http://210.34.4.20/big5/
	台東大學圖書館	도서만남 기한알림	http://www.lib.nttu.edu.tw/
	NCSU Libraries	OPAC 검색결과	http://www.lib.ncsu.edu/catalog/
	SCOPUS	웹DB 검색결과	http://www.scopus.com/
블로그	University of Pennsylvania Library	업무활동	http://www.library.upenn.edu/blos/staffweb
	Plymouth State University Library	OPAC	http://lamson.wpopac.com/library/opac/
	Ann Arbor District Library	웹사이트	http://www.aadl.org/
위키	University of Connecticut Library	업무활동	http://wiki.lib.uconn.edu/wiki/Main_Page
	Oregon Library Instruction Wiki	이용지도	http://instructionwiki.org/Main_Page
	USC Aiken Gregg-Graniteville Library	웹사이트	http://library.usca.edu/Main/HomePage
	Open WorldCat (WikiD)	OPAC 서평	http://alcme.oclc.org/wikid/WikiD
아작스	Texas A&M Libraries	참고정보 서비스	http://tamulibraries.pbwiki.com/
	Georgia Tech.: Library and Information Center	검색어 자동완성	http://findit.library.gatech.edu/
플리커	Virginia Tech Libraries	검색건수 추정치	http://addison.vt.edu/
	Westmont Public Library	신착도서 안내	http://www.flickr.com/photos/westmontlibrary/
폭소노미	University of Winnipeg Library	도서관 투어	http://www.flickr.com/photos/uwinnipeglibrary/
	University of Pennsylvania Library	도서관 목록	http://tags.library.upenn.edu/
	Ann Arbor District Library	Social OPAC	http://www.aadl.org/catalog/
포드 캐스트	The Art Museum Social Tagging Project	Social Tagging	http://www.steve.museum/
	Curtin University Library	도서관 이용안내	http://library.curtin.edu.au/podcast/index.html
툴바	UCSC Library	교육·연구 지원	http://library.ucsc.edu/mt/synergyblog/
	Lansing Public Library	개별도서관	http://www.lansing.lib.il.us/L2.htm
북마크릿	OCLC WorldCat	서지유틸리티기관	http://www.oclc.org/toolbar/
	LibraryLookup Project	Explorer, Firefox	http://weblog.infoworld.com/udell/stories/2002/12/11/librarylookup.html
	Greasemonkey User Scripts	Firefox	http://www.greasespot.net/
Open API	OCLC WorldCat xISBN and FRBR	Explorer, Firefox	http://xisbn.worldcat.org/liblook/listbookmarks.htm
	Talis Prism	Bigfoot Store API	http://www.talis.com/tdn/
Mash-up	Talis Wilkorm	Libmap UK	http://research.talis.com/2005/google-maps/libmapuk2.html
	NCSU Libraries	REST	http://www.lib.ncsu.edu/catalog/ws/oclc/
웹서비스	Library of Congress	SRW/U	http://www.loc.gov/standards/sru/
	HKUST Library	OAI-PMH	http://repository.ust.hk/dspace/
소셜북마크	Lansing Public Library	del.icio.us	http://del.icio.us/lansingpubliclibrary
소셜 네트워크	Lansing Public Library	MySpace	http://www.lansing.lib.il.us/
	UIUC Library Library	Facebook	http://www.library.uiuc.edu/
개인화 서비스	Hennepin County Library	RSS	http://www.hclib.org/
	NCSU Libraries	iGoogle Widget	http://www.lib.ncsu.edu/dli/projects/catalogwsapps/
	OCLC WorldCat	iGoogle Gadget	http://worldcat.org/googlegadgets/worldcatsearch.xml

27) 구중익, 이응봉, “북마크릿을 활용한 LibraryLookup 서비스 제공방안에 관한 연구,” 정보관리학회지, 제23권, 제3호(2006. 9), p.53. 재수정.

〈표 7〉 국내 Library 2.0 적용사례<sup>28)</sup>

구분	도서관	적용 유형	URL
RSS	포항공과대학교 청암학술정보관	게시판(공지사항)	http://library.postech.ac.kr/
	성균관대학교 학술정보관	정보광장(공지사항, 건의사항, 서평), 소장자료 검색결과, 신착도서 안내	http://skkcl.skku.ac.kr/
	한국과학기술정보연구원 yesKISTI	맞춤정보, 글로벌동향브리핑(GTB), KISTI의 과학향기, 과학기술동영상, 특허정보, 주간기술동향지(TLD), OAJ, 부품소재, 지식질의 What is?, 바이오인포매틱스	http://www.yeskisti.net
	서울대학교 중앙도서관	게시판(공지사항, FAQ, 도서관 의견함, 홈페이지 건의사항, 자료기능 문의, 분실물 센터, e-Resource 공지사항), 신착도서 안내, 신착자료/단행본 검색결과	http://library.snu.ac.kr/
	한국과학기술원 과학도서관	게시판(공지사항, FAQ, Q&A, 건의사항, 신규 DB/새소식, DB가이드, 분실물센터), 신착도서 안내	http://library.kaist.ac.kr/
	경남대학교 중앙도서관	일반(공지사항, 신착도서 안내), 개인(공지사항, 대출현황, 희망도서 및 문헌복사 처리상태	http://library.kyungnam.ac.kr/
	한양대학교 백남학술정보관	게시판(일반 공지사항, DB 공지사항)	http://library.hanyang.ac.kr/
	덕성여자대학교 도서관	게시판(공지사항, 공지사항, 서평), 신착도서 안내	http://discover.duksung.ac.kr/
	환경부 환경종합디지털도서관	정보광장(원문정보, 신착자료), 주제분류	http://library.me.go.kr/
	법원도서관	신간도서 안내(판례자료, 연구자료), 신착도서 안내, 최신 대법원판례정보, 전국법원 주요판결	http://library.scourt.go.kr/
블로그	한국과학기술정보연구원 yesKISTI	이용자	http://www.yeskisti.net
	한국정보문화진흥원 국가지식포털	이용자	http://www.knowledge.go.kr/
	포항공과대학교 청암학술정보관	관리자	http://postechlibrary.tistory.com/
	한국기술교육대학교 다산정보관	관리자	http://dasan.tistory.com/
	동국대학교 중앙도서관	관리자	http://dgulibrary.tistory.com/
위키	포항공과대학교 청암학술정보관	사서용어사전2.0	http://libterm.springnote.com/
툴바	포항공과대학교 청암학술정보관	Firefox용 도서관 툴바 LibX	http://postechlibrary.tistory.com/27
Open API	경남대학교 중앙도서관	도서검색 API	http://library.kyungnam.ac.kr/
Mash-up	포항공과대학교 청암학술정보관	Amazon Web Service	http://library.postech.ac.kr/
폭소노미	서울대학교 중앙도서관	OPAC	http://library.snu.ac.kr/
소셜북마크	성균관대학교 학술정보관	SKKUBE	http://bee.skku.ac.kr/
북마크릿	성균관대학교 학술정보관	도서관 책검색 2.0(49개 기관)	http://lomofan.cafe24.com/zb5/?sid=31
	포항공과대학교 청암학술정보관	OCLC xISBN	http://xisbn.worldcat.org/liblook/index.htm
플리커	포항공과대학교 청암학술정보관	신착도서	http://www.flickr.com/photos/p_tech/
웹서비스	한국교육학술정보원 dCollection	OAI-PMH	http://www.riss4u.net/
	출판유통진흥원	SOAP	http://www.booktrade.or.kr/
개인화 서비스	포항공과대학교 청암학술정보관	iGoogle용 도서관목록 검색 가젯	http://library.postech.ac.kr/

28) 구중익, 이응봉, 2006, 전계논문. p.53. 재수정.

그리고 Library 2.0 사례분석을 통해서 살펴본 바와 같이 SOA와 웹서비스는 데이터, 서비스 및 어플리케이션들을 외부에 노출시키고 통합해 주고 있다. 특히 웹서비스 프로토콜인 REST, SOAP, XML-RPC, SRW/U, OAI-PMH는 가볍고 단순한 프로그래밍과 느슨하게 연결된 형태로 플랫폼 기반의 디지털도서관 시스템을 구축하는데 최적의 대안이 되고 있다. 또한 Web 2.0 기술인 Open API, 매쉬업, AJAX 등은 디지털도서관 시스템의 개방성 및 접근성, 서비스 제공 수준과 정보검색의 브라우징 방식을 개선하는데 핵심 기술로 활용할 수 있다.

〈표 8〉은 SOA 기반 웹서비스 프로토콜과 Web 2.0 기술을 활용한 Library 2.0 적용방법을 제시한 것이다.

아울러 SOA와 웹서비스를 기반으로 메타검색 시스템 구축에 필요한 국내외 온라인서점, 웹 포털, 종합목록 등에서 제공하고 있는 Open API의 구체적인 사례를 살펴보면 다음과 같다.

〈표 8〉 SOA 기반 웹서비스 프로토콜의 Library 2.0 적용방안

웹서비스 프로토콜	Web 2.0 기술	Library 2.0 적용방안
SOAP	Open API	- 편목 시 도서분류표, 주제명표목표, 전자파일, 시소러스 등을 참조연계 - 종합목록 또는 온라인서점의 검색API를 이용하여 OPAC에 검색결과 추가 - 트랜잭션 처리(상호대차, 원문복사)
REST	XML, RSS	- 개인화홈페이지 또는 PC 데스크톱에서 공지사항, 신착자료 조회 - 개인화홈페이지에서 MyLibrary의 공지사항 조회
	XML	- 자관의 OPAC에서 ISBN 검색을 허용하는 Open URL 제공 - 종합목록에서 특정 ISBN에 해당하는 연관 저작의 검색결과 제공
	AJAX	- 개인정보 실시간 조회(대출 및 예약황, 상호대차 및 원문복사, 희망도서신청) - 검색어 자동완성
	Open API Mash-up AJAX JSON	- 종합목록, 회원정보, 지도 API를 결합하여 인접도서관 정보 제공 - 출판시 도서목록서비스에서 자관 OPAC에 책정보(책소개, 저자소개, 서문, 목차, 서평, 이미지) 추가 - 종합목록 또는 온라인서점의 검색API를 이용하여 OPAC에 검색결과 추가 - 웹포털(Google, 네이버 등)의 검색결과를 OPAC 검색결과에 추가 - 외부의 시소러스 사전을 참조하여 연관검색 기능 제공 - 하나의 검색어에 대한 연관검색어를 버블 형태로 시각화하여 제공 - 오픈소스 형태소 분석기를 사용하여 정확한 한글/한자 및 영어 처리 - 검색어의 한/영 자동변환 기능
XML-RPC	XML	- 온라인서점에서 블로그 또는 디지털도서관의 MyLibrary에 원격 글쓰기 - 종합목록 검색결과에서 곧바로 자관의 디지털도서관에 상호대차 신청
SRW/U	XML, RSS	- 분산검색(distributed searching), 전문검색(full-text)
OAI-PMH	XML, RSS	- 메타데이터 수확(metadata harvesting), 통합 메타데이터 DB구축, 메타검색

아마존은 Amazon E-Commerce Service(ECS 4.0)를 통해서 SOAP과 REST 기반 Open API를 제공하고 있고, 도서의 검색 목록을 제공하는 ItemSearch와 상세정보를 조회하는 ItemLookup 방식을 적용하여 개발자 및 사용자에게 WSDL과 XML 형태로 데이터를 제공하고 있다. 또 책이미지는 8개 문자로 사이즈를 지정하거나 ‘\_’기호로 옵션을 자유롭게 추가하여 사이즈 조정, 할인

을 표시 등을 표시할 수 있도록 하고 있다. 아마존의 웹서비스를 이용하기 위해서는 먼저 Web Services Account를 생성하고, Access Key와 다른 서비스 시 인증을 위한 Secret Access Key를 발급받아야 한다.<sup>29)</sup>

아마존의 검색엔진인 A9의 OpenSearch는 웹사이트의 검색결과를 공유할 수 있는 간단한 규격을 공개하였고, OpenSearch 규격에 맞게 검색기능을 제공하는 웹사이트에서 XML/RSS 형태로 검색결과를 제공하고 있다.<sup>30)</sup> 또 MS사의 IE7(Internet Explorer 7)은 RSS를 구독하는 기능뿐만 아니라 OpenSearch 1.1을 지원하고 있다.<sup>31)</sup> OpenSearch는 IE7외에도 Firefox에서도 사용할 수 있는 포맷이며, 도서관이 OpenSearch 규격에 맞게 OPAC을 공개할 경우 웹 브라우저의 OpenSearch 지원 기능을 통해서 도서관 이용자는 웹브라우저에서 직접 자관의 OPAC을 검색할 수 있다.

OCLC의 WorldCat은 15백만여 개의 ISBN과 12백만여 개의 FRBR 그룹을 포함하고 있는 xISBN DB를 보유하고 있으며, REST 기반 xISBN API를 통해서 FRBR 북마크릿 서비스를 제공하고 있고, Open API의 사용횟수가 500회 이상인 경우 별도의 과금정책을 운영하고 있다.<sup>32)</sup>

국내에서도 일부 온라인서점과 웹 포털에서 REST를 이용한 도서검색 Open API를 제공하기 시작하였다. Open API를 이용하기 위해서는 인증 키 또는 서명 키를 발급받아야 하고, 검색은 GET 또는 POST 방식으로 요청이 가능하며, 모두 1일 5,000회로 제한되어 있다. 네이버는 책 검색 API의 출력결과를 XML 형식만 제공하고 있고, 다음은 도서 검색 API의 출력결과를 XML, RSS, JSON 형식으로 전송하고 있다. 또 알라딘은 도서 검색 및 개별 출력 API의 출력결과를 XML, JSON(JSON-CallBack 방식 포함),<sup>33)</sup> 확장RSS(OpenSearch) 형식으로 제공하고 있다.<sup>34)</sup>

그러나 국내의 온라인서점과 웹 포털에서 제공하는 도서 검색 API의 이용과정에서 검색요청 변수, 출력결과 포맷과 항목이 서로 상이하어 상호호환성에 문제가 있고, 서비스의 안정성과 보안 이슈가 발생할 수 있으며, Open API의 사용횟수나 트래픽에 제약이 있다. 또 Open API를 이용한 매쉬업 서비스의 개발자와 사용자에게 충분한 예제 프로그램과 지원방식을 제공하지 못하고 있다.

29) Amazon. *Amazon Web Services*.

〈<http://www.amazon.com/gp/browse.html?node=3435361>〉 [cited 2007. 8. 17].

XWebServices, Inc. *SOA Hub - Service Oriented Architecture Portal*.

〈<http://www.soahub.com/>〉 [cited 2007. 8. 17].

30) A9. *OpenSearch*. 〈<http://www.opensearch.org/>〉 [cited 2007. 8. 17].

31) Mielke, Markus. Internet Explorer 7, 2006.

〈[http://download.microsoft.com/download/2/8/e/28e60dcc-123c-4b27-b397-1f6b2b6cb420/Part2\\_MM.pdf](http://download.microsoft.com/download/2/8/e/28e60dcc-123c-4b27-b397-1f6b2b6cb420/Part2_MM.pdf)〉 [cited 2007. 8. 17].

32) OCLC. *WorldCat Web service: xISBN*.

〈<http://xisbn.worldcat.org/xisbnadmin/index.htm>〉 [cited 2007. 8. 17].

33) JSON(JavaScript Object Notation): 속성(name)과 값(value)으로 구성된 간단한 데이터 교환 포맷임.

34) 네이버. 책 검색 API, 〈[http://openapi.naver.com/page.nhn?PageId=1\\_06](http://openapi.naver.com/page.nhn?PageId=1_06)〉 [cited 2007. 8. 17].

다음. 도서 검색 API, 〈<http://dna.daum.net/apis/search/book>〉 [cited 2007. 8. 17].

알라딘. 도서(상품) 검색 API, 〈<http://www.aladdin.co.kr/ttb/apiguide.aspx#searchapi>〉 [cited 2007. 8. 17].

## 2. 메타검색 시스템 구축 및 활용방안

본 연구에서는 국내의 온라인서점, 웹 포털, 종합목록에서 제공하는 Open API와 도서관 OPAC 등을 결합하여 새로운 형태의 메타검색 시스템 구축과 활용방안을 제안하고자 한다.

도서관은 업무와 서비스 종류에 따라 검색, 대출, 예약, 상호대차, 원문복사, 희망도서 등 다양한 Open API를 개발할 수 있으며, 시스템 관점에서 다음의 일곱 가지 사항을 고려해야 한다.

첫째, 시스템과 이용자 수준에서 Open API의 기능과 검색요청 변수, 상세 검색요청 변수 및 검색결과 출력필드를 정의한다.

둘째, Open API를 사용한 데이터 요청 시 사용자의 아이디와 패스워드를 사용하여 인증하는 대신 별도의 인증 키와 서명 키를 발급하여 관리한다.

셋째, 데이터 요청(request)은 SOAP, REST, XML-RPC, SRW/U, OAI-PMH 모두 사용이 가능하고, HTTP를 통해 GET 또는 POST 방식을 적용한다.

넷째, 데이터 응답(response)은 XML, RSS, JSON 형식으로 가능하나, 크로스 도메인(cross domain) 간의 제약사항과 인코딩 스타일에 따른 처리 속도와 시스템 부하 여부를 고려한다.

다섯째, 모든 데이터에 대한 인코딩은 유니코드인 UTF-8을 적용하는 것이 바람직하다.

여섯째, 데이터 요청 및 응답 시 발생할 수 있는 오류코드와 메시지를 제공한다.

일곱째, Open API의 사용권 및 제한사항 등을 포함하는 이용약관을 제공한다.

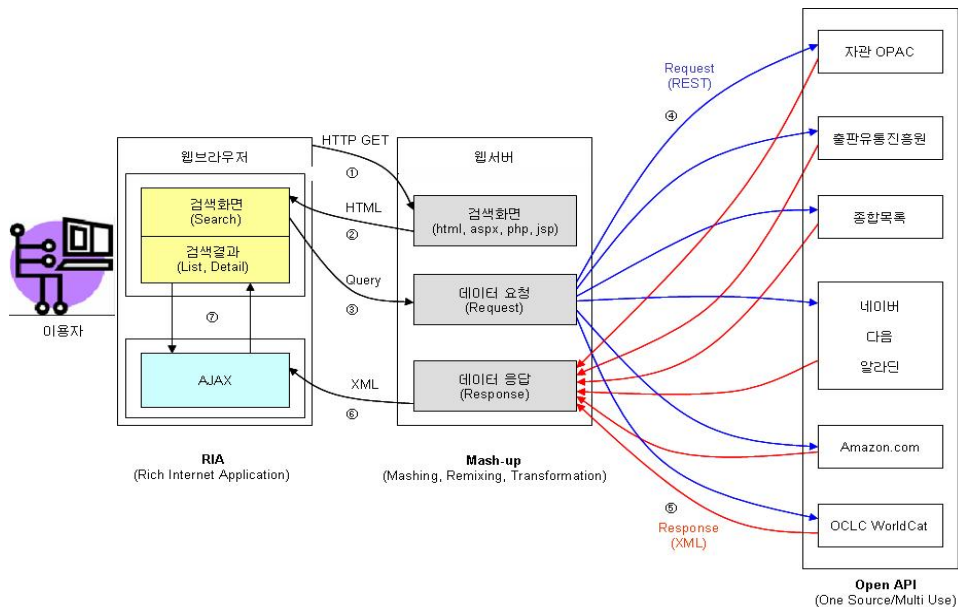
그리고 국내의 온라인서점, 웹 포털, 종합목록에서 제공하는 Open API와 도서관 OPAC 등을 결합한 메타검색 시스템을 활용하여 도서관과 이용자가 얻을 수 있는 주요 이점은 다음과 같다.

첫째, 도서관 이용자는 자관의 OPAC 검색결과와 온라인서점, 웹 포털, 종합목록에서 제공하는 서지정보를 한눈에 비교할 수 있어 대출 및 예약, 상호대차 및 원문복사, 희망도서신청, 강의교재 확인 등 정보검색에 소요되는 시간 및 비용을 절감할 수 있다.

둘째, 도서관은 출판유통진흥원의 도서정보 제공서비스와 네이버, 다음, 알라딘, 아마존 등의 도서검색에서 제공하는 Open API를 사용하면 별도의 DB구축 없이 적은 비용으로 국내외 도서의 책정보(책소개, 저자소개, 추천글, 목차, 출판사 및 미디어 서평, 독자 및 전문가 리뷰)와 책이미지를 자관의 OPAC 검색결과에 포함시켜 도서관 이용자에게 풍부한 서지정보를 제공해 줄 수 있다.

셋째, 도서관은 국립중앙도서관의 출판시 도서목록서비스(e-CIP)와 국가자료공동목록시스템(KOLIS-NET), 학술교육학술정보원의 전국대학소장자료(RISS), 한국과학기술정보연구원의 국가과학기술전자도서관(NDSL) 등에서 Open API를 제공할 경우 자관의 OPAC 검색결과에 타도서관의 소장정보를 포함시켜 이용자에게 도서관 정보자원의 이용기회를 크게 넓혀 줄 수 있다.

<그림 12>는 REST 기반 Open API를 이용한 메타검색 시스템의 아키텍처를 제안한 것이다.



〈그림 12〉 REST 기반 Open API를 이용한 메타검색 시스템 아키텍처

## VI. 결론

이용자 중심의 서비스는 도서관과 정보통신기술이 지향하는 핵심적인 요소이다. 현재 Web 2.0이 경제, 사회, 문화 등 전 영역에 걸쳐 변화를 주도하고 있고, 도서관 영역에서도 지식정보의 생산·유통·마케팅과 비즈니스 모델 및 프로세스 혁신 등 새로운 대응전략 모색이 필요한 시기이다.

SOA는 '잘 정의된 인터페이스들을 가진, 재사용이 가능한 일련의 컴포넌트를 기반으로 정보시스템을 구축하여 서비스 중심의 플랫폼을 구축하는데 필요한 아키텍처 방법론'을 제공하고 있다.

웹서비스는 XML, HTTP, SOAP, WSDL, UDDI의 표준 기술을 사용하여 언제, 어디서나, 어떤 기기에서나 원하는 정보, 응용 기능, 서비스 등을 사용하고 제공할 수 있는 서비스이다.

SOA를 구현하는 기술인 웹서비스는 이질적인 플랫폼과 기술에 종속되지 않고 느슨하게 연결된 프로세스(loosely coupled process)를 통해서 기존 컴포넌트의 재사용을 용이하게 하고, 시스템과 기술간의 연계·통합과 상호운용성을 확보해주는 글루(glue) 기술이 되고 있다.

본 연구에서는 최근 XML과 웹서비스가 효율적인 표준 기술로 평가를 받으면서 SOA가 다시 주목되고 있는 가운데 SOA와 웹서비스의 개념을 살펴보고, SOA를 구현하는 웹서비스의 주요 프로토콜인 SOAP, REST, SRW/U, OAI-PMH에 대한 기술을 분석하였다. 그리고 국내외 도서관에서 SOAP과 REST 기반 웹서비스 구축사례, Web 2.0 기술을 활용한 Library 2.0 적용사례 및

SOA 기반 웹서비스 프로토콜의 Library 2.0 적용방법을 제시하였다. 또한 REST 기반 Open API를 이용하여 도서관 OPAC, 온라인서점, 웹 포털, 종합목록 등을 결합한 메타검색 시스템 구축과 활용을 통하여 Library 2.0 적용방안을 제안하였다.

Library 2.0 구현을 가속화하는데 있어서 핵심 기술인 SOA 기반 웹서비스 프로토콜의 활용방안을 요약하면 다음과 같다.

첫째, 분산환경에서 정보를 교환하기 위한 XML 기반 웹서비스 프로토콜인 SOAP은 도서관에서 상호대차나 원문복사와 같이 복잡한 트랜잭션 처리에 적합하다. 특히 국가대표도서관이 제공하는 도서분류표, 주제명표목표, 전거파일, 시소러스 등을 SOAP, REST, SRW/U 등을 기반으로 디지털도서관에서 공유하면 도서관 편목업무의 효율성을 높이고 시간과 비용을 절감할 수 있다.

둘째, SOAP에 비해 단순하고 가벼운 웹서비스 프로토콜인 REST는 디지털도서관 시스템을 느슨하게 통합해 주고, Open API에 사용되어 기존의 데이터와 서비스를 재조합한 매쉬업 서비스 창출이 가능하다. REST 기반 도서 검색용 Open API의 경우 아마존 A9의 OpenSearch 스펙을 참고하여 개발할 수 있고, 검색결과는 XML, RSS, JSON 형식으로 제공할 수 있으며, AJAX 기술을 사용하면 풍부한 사용자 편의 기능을 제공할 수 있다. 한편 저작물 이용허락표시(CCL) 범위 내에서 종합목록, 웹DB, 전자저널, 지식저장소 등의 Open API도 시급히 제공되어야 할 것으로 사료된다.

셋째, XML-RPC는 다른 기종의 운영체제에서 인터넷을 통하여 다른 환경에서 수행되는 프로시저를 호출할 수 있도록 해준다. 도서관 웹사이트에서 XML-RPC와 Open API를 제공할 경우 도서관 이용자가 전자저널, 웹DB 등 웹사이트에서 직접 서지정보를 MyLibrary에 저장할 수 있다. 또 도서관 이용자가 온라인서점의 웹페이지에서 관심도서 목록을 디지털도서관의 MyLibrary에 저장할 수 있고, 반대로 도서관 이용자가 자관 OPAC의 검색결과를 자신의 블로그, 소셜 북마크, 소셜 네트워크 등의 개인화 홈페이지에 저장할 수 있다.

넷째, 디지털 콘텐츠의 개방형 지식저장소 구축 시 REST 기반의 웹서비스 프로토콜인 OAI-PMH를 사용하여 메타데이터를 수확(harvesting)하고, 정보검색을 위한 웹서비스 프로토콜인 SRW/U를 적용함으로써 시스템의 상호운용성, 정보공유 및 검색의 효율성을 크게 높일 수 있다.

향후 연구 과제로는 도서관 OPAC에서 REST 기반 Open API 제공에 필요한 메타데이터 요소를 추출한 후 전문가 및 이용자 집단의 평가 및 검토를 거쳐 온라인서점, 웹 포털, 종합목록 등의 Open API를 결합한 메타검색 시스템을 구현하고 검색의 효율성 및 사용성 평가도 필요한 것으로 보인다.

마지막으로 국내 도서관의 Library 2.0 적용사례에서 나타났듯이 Web 2.0 기술의 활용은 전반적으로 미흡한 실정이나, SOA와 웹서비스를 점진적으로 도입하여 특정 서비스 단위로 디지털도서관 시스템의 기능을 컴포넌트화 하면 Library 2.0이 신속하게 확산될 수 있을 것으로 기대된다.

〈참고문헌은 각주로 대신함〉