SOAP 기반 웹서비스와 RESTful 웹서비스 기술 비교

SOAP-based Web Services vs. RESTful Web Services

박유미 (Y.M. Park) 서비스융합연구팀 책임연구원
문애경 (A.K. Moon) 서비스융합연구팀 선임연구원
유현경 (H.K. Yoo) 서비스융합연구팀 선임연구원
정유철 (Y.C. Jung) 서비스융합연구팀 연구원
김상기 (S.K. Kim) 서비스융합연구팀 책임연구원

목 차

* 본 논문은 지식경제부 및 정보통신산업진흥원의 차세대 통신네트워크 산업원천기술 개발과제(2009-F-048-01)로 수행되었음.

최근 들어 웹서비스는 그 발전 과정에서 SOAP 기반의 웹서비스와 RESTful 기반의 웹서비스로 양분화되어 제공되고 있다. 서비스 인터페이스를 구현 로직으로부터 분리하는 웹서비스의 기본 개념을 따르면서 SOAP 기반의 웹서비스는 비즈니스 플로 처리를 위한 서비스 상호 연동에 주로 이용되고, RESTful 웹서비스는 데이터 접근에 주로 이용된다. 본 고에서는 SOAP 기반 웹서비스와 RESTful 웹서비스를 더 이상 구분하지 않고 통합하여 검색하거나 조합하고자 하는 시도의 첫 걸음으로 서비스 구조, 구현 기술, 사용 방법, 시맨틱 서비스화 연구의 관점에서 두 웹서비스를 비교하고자 한다.

Ⅰ. 서론

SOA 기반의 웹서비스 개념이 등장한 이후, 2000년대 초반부터 비즈니스 도메인의 서비스를 웹서비스로 개방하여 상호 연동하고자 하는 시도가, 최근에는 비즈니스 도메인 뿐 아니라 통신 및 인터넷 서비스 전 분야의 개방에 큰 파급 효과를 미치고 있다. SOA 기반의 웹서비스는 W3C의 WS-*로 일컬어지는 웹서비스 표준들을 통해 실현되고 있으며, WS-* 표준들은 SOAP을 이용하여 메시지를 전달하고 있다. 그러나 웹서비스 전달 프로토콜인 SOAP은 HTTP 응용 프로토콜로서 SOAP 헤더와 바디로 구성되어 있고, 메시지 송수신시 헤더와 바디의 인코딩/디코딩 과정이 필수이다. 따라서 기본 HTTP로 메시지를 전달하던 인터넷 서비스 분야에서는 원하는 기능에 비해 SOAP 프로토콜 처리의 오버헤드가 문제가 되었다. 이런 SOAP의 단점을 보완하고자 등장한 구현 기술이 RESTful 웹서비스이다. RESTful 웹서비스는 REST 기반의 웹서비스를 의미하고 HTTP의 기본 기능만으로 원격 정보에 접근하는 웹 응용 기술이다[1]. 이에 본 고에서는 두 기술의 웹서비스를 간략히 소개하고, SOAP 기반 웹서비스와 RESTful 웹서비스를 서비스 구조, 구현 기술, 사용, 시맨틱 서비스화 연구의 관점에서 비교하고자 한다. 이는 두 웹서비스를 더 이상 구분하지 않고 통합하여 검색하거나 조합하고자 하는 시도의 첫 걸음이 될 것이다. 본 고의 구성은 다음과 같다. Ⅱ장과 Ⅲ장에서는 두 웹서비스 기술을 간략히 소개하고, Ⅳ장에서는 다양한 관점에서 두 기술을 비교 분석한다. 마지막으로 Ⅴ장에서 두 기술의 발전 방향을 예측하며 결론을 맺는다.

Ⅱ. SOAP 기반 웹서비스

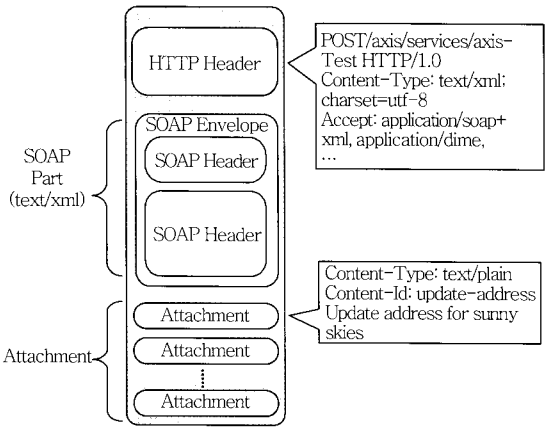
SOAP 기반 웹서비스와 RESTful 웹서비스의 비교에 앞서 본 장에서는 SOAP 기반 웹서비스에 대해 요약하여 설명한다.

일반적으로 '웹서비스'란 분산되어 있는 콘텐츠를 추상적인 서비스 형태로 개방하여 표준화된 형태로 공유하는 기술로서 SOA 개념을 실현하기 위한 대표적 기술이라 할 수 있다[2].

웹서비스 내의 모든 데이터는 XML로 표현되고, 그 데이터들과 이를 다룰 수 있는 오퍼레이션들이 WSDL로 정의되면 UDDI라는 전역적 서비스 저장소에 등록(publish)되어 누구라도 서비스를 찾을 수 있도록 공개된다. 공개된 웹서비스가 이용될 때, 서비스 요청자와 서비스 제공자 간에 SOAP을 이용하여 서비스를 호출하고 결과를 돌려받게 된다.

이러한 웹서비스 기술은 이기종 플랫폼 위에 구축된 응용들 간에 연동을 목적으로, 상호 이해 가능한 포맷의 메시지를 SOAP으로 송수신함으로써 원격지에 있는 서비스 객체나 API를 자유롭게 사용하고자 하는 기업의 요구에서 출발하였다. SOAP은 특정 분산 기술 또는 플랫폼에 의존하지 않고 분산 객체를 액세스 할 수 있는 프로토콜로서 HTTP 상에서 전송된다. 모든 SOAP 메시지는 (그림 1)과 같이 SOAP 봉투(envelope), SOAP 헤더(header), SOAP 바디(body)로 구성된 하나의 XML 문서로 표현되는데 이러한 복잡한 구성으로 인해 HTTP 상에서 전달되기 무겁고, 메시지 인코딩/디코딩 과정 등 웹서비스 개발의 난이도가 높아 개발 환경의 지원이 필요하다.

이러한 SOAP을 통해 전달하려는 대상은 웹서비스를 기술(description)한 WSDL 문서이다. WSDL



(그림 1) SOAP 구조

은 특정 비즈니스가 제공하는 서비스를 요청자들이 전자적으로 접근하여 이용할 수 있도록 표준 형태로 정의하는 XML 기반의 언어이다. 여기에는 웹서비스 이름과 주소(URI), 인터페이스, SOAP 메시지 인코딩 방식, SOAP 전달을 위한 전송 프로토콜 등이 기술된다. WSDL은 W3C에서 표준화되고 있으며 현재 산업계에서는 WSDL 1.1과 2.0을 지원하고 있다.

Ⅲ. RESTful 웹서비스

웹의 창시자 중 한 사람인 Roy Fielding이 그의 박사 학위 논문[1]에서, 현재의 웹 아키텍처가 웹의 본래 설계의 우수성을 활용하지 못하므로 웹의 장점을 최대한 활용할 수 있는 네트워크 기반의 아키텍처를 제안했는데 이것이 REST이다. REST는 부수적인 레이어나 세션 관리를 추가하지 않고도 HTTP 프로토콜로 데이터를 전달하는 프레임워크이다. 또한 클라이언트/서버 간의 구성요소를 엄격히 분리하여 구현은 단순화시키고 확장성과 성능은 높일 수 있는 아키텍처다. 최근 들어 REST는 웹에 개방된 리소스들을 원격에서 또는 지역적으로 쉽게 이용하려는 웹 응용에 정착하게 되었고, REST 아키텍처 스타일에 따라 정의되고 이용되는 서비스나 응용을 RESTful 웹서비스라 한다[3]-[5]. 여기서 리소스(resource)란 REST 아키텍처의 핵심 요소로서 웹사이트, 블로그, 이미지, 음악, 이용자, 지도, 검색 결과 등 웹에서 다른 이들과 공유하고자 개방된 모든 자원을 의미한다. REST 구조에서의 리소스는 그들의 고유한 URI를 가지며, HTTP의 기본 메소드인 GET/PUT/POST/DELETE만으로 접근할 수 있다.

〈표 1〉 HTTP 요청의 의미

HTTP 요청 메소드		CRUD 오퍼레이션
POST	create a new resource	CREATE
GET	retrieve a representation of a resource	READ
PUT	modify or overwrite an existing resource	UPDATE
DELETE	delete an existing resource	DELETE

리소스에 접근하기 위한 HTTP 요청은 <표 1>과 같이 일반 CRUD 오퍼레이션에 대응한다.

HTTP 기본 메소드로 전달되는 리소스는 다양한 방식으로 표현되는데, XML, JSON, HTML, 텍스트, 이미지 등이 가능하며 클라이언트에서 원하는 형식으로 표현하면 서버에서 이를 처리하게 된다. 리소스의 다양한 표현 방식은 HTTP의 accept 헤더 값 또는 URI 파라미터로 지정하면 된다.

리소스와 그를 지칭하는 URI, HTTP로 단일화된 인터페이스, 다양한 리소스 표현 등과 더불어 빼놓을 수 없는 RESTful 웹서비스의 특성이 스테이트리스(statelessness)이다. HTTP의 특성을 상속하여 REST 역시 스테이트리스 특성을 가지게 되는데 스테이트리스란 웹서비스 제공 서버 측에서 클라이언트의 상태 정보를 저장, 관리하지 않는 것을 의미한다. 그러나 RESTful 웹서비스 제공시 응용의 상태와 리소스의 상태 관리가 필요할 수 있으며, 이를 위해서는 서버와 클라이언트 간에 이들의 상태를 명시적으로 전달하는 방법을 이용해야 한다. 이는 REST라는 이름이 함축하고 있는 특성이기도 하다.

이와 같이 RESTful 웹서비스는 리소스 중심의 표현, 전달, 접근 방식의 특성으로 인해 리소스 기반 아키텍처(ROA)라고 한다. ROA는 서비스 중심의 SOA에 대응되는 개념으로 일컬어지고 있다. 즉, RESTful 웹서비스는 리소스 URI를 알면 웹서버와 웹클라이언트의 종류에 상관없이 HTTP 프로토콜만으로 접근 가능한 서비스라 할 수 있다. 이러한 단순 명료한 접근 방식 때문에 구글, 야후, 트위터 등 대부분의 웹 2.0 API가 RESTful 웹서비스로 제공되고 있으며, 위젯을 이용한 서비스 매시업(mashup)을 활성화시킨 원동력이기도 하다. 또한 기존에 제공되던 SOAP 기반의 웹서비스 조차도 RESTful 웹서비스화 되어 동시에 제공되는 추세이다.

Ⅳ. 기술 비교

두 기술의 소개를 통해 SOAP 기반의 웹서비스는 서비스의 기술문서(WSDL)를 매개로 서비스 능

력을 개방하고자 한 기술이고, RESTful 웹서비스는 URI을 기반으로 리소스를 개방하고자 하는 기술임을 설명하였다. 본 장에서는 서비스 구조, 구현 기술, 사용의 관점에서 두 기술을 비교 분석하고자 한다. 또한 최근 웹서비스 계의 화두가 되고 있는 시맨틱 서비스화와 관련된 두 기술의 접근 현황에 대해서 설명한다.

1. 서비스 구조

(그림 2)는 SOAP 기반 웹서비스 기술(a)과 RESTful 웹서비스 기술(b)을 서비스 아키텍처 측면에서 비교한 그림이다. SOAP 기반 웹서비스가 SOA 구조에 따라 UDDI 레지스트리를 통해 웹서비스를 등록하고(publish), 탐색하고(find), 바인딩하여(bind) 이용한다면, RESTful 웹서비스는 리소스를 등록하고 저장해두는 중간 매체 없이 리소스 제공자가 직접 리소스 요청자에게 제공하는 방법을 따르고 있다.

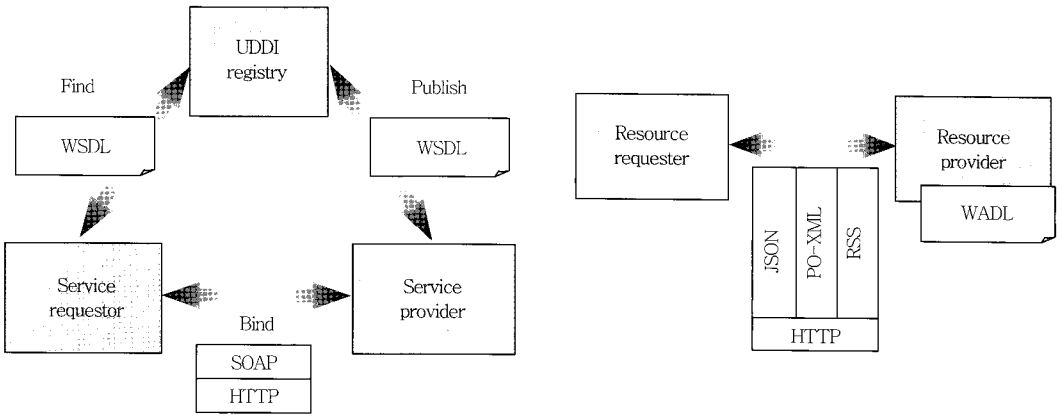
서비스 실행 관점에서, SOAP 기반 웹서비스에서는 서비스 제공자와 요청자 간에 SOAP 프로토콜로 메시지를 주고 받는 방식으로 서비스를 이용한다. 즉, 서비스 요청자가 웹서비스 요청을 SOAP으로 인코딩하여 서비스 제공자에게 전달하면, 서비스 제공자는 이를 디코딩하여 적절한 서비스 로직을 통하여 결과를 얻고, 그 결과를 다시 SOAP 인코딩하여 서비스 요청자에게 반환한다. 한편, RESTful 웹서비스는 기본 HTTP 프로토콜의 메소드 GET/PUT/POST/DELETE를 이용하여 다양한 형태로 표현된(JSON, XML, RSS 등) 리소스를 직접 실어 나른다.

2. 서비스 기술

<표 2>는 SOAP 기반 웹서비스 기술과 RESTful 웹서비스 기술을 비교하여 요약한 표이다.

SOAP 기반 웹서비스의 요구가 기업의 비즈니스 환경에서 응용 서비스 간 상호 운용을 위해 시작된 데 비해, RESTful 웹서비스는 인터넷 서비스 업체들이 응용 개발자들에게 손쉬운 데이터 제공을 목적으로 출발했다. 따라서 SOAP 기반 웹서비스는 서비스를 제공하고 이용하는 프로그램들이(기계) 잘 이해할 수 있도록 엄격한 문법에 따라 개발되었고, 때문에 개발자들에게는 웹서비스 기본 스펙을 알아야 하는 비교적 고난이도의 프로그래밍 능력이 요구되었다. 따라서 SOAP 기반 웹서비스 개발의 편의성을 도모하기 위해 다양한 개발 환경들이 지원되고 있다.

반면 RESTful 웹서비스는 기계보다는 사람이 이해하기 쉽도록 인터넷 기본(HTTP와 XML) 이외에 별도의 개발/실행 환경을 필요로 하지 않는다. 특히



(a) SOA에 근거한 SOAP 기반 웹서비스

(b) ROA에 근거한 RESTful 웹서비스

(그림 2) 서비스 구조 비교

〈표 2〉 서비스 기술 비교

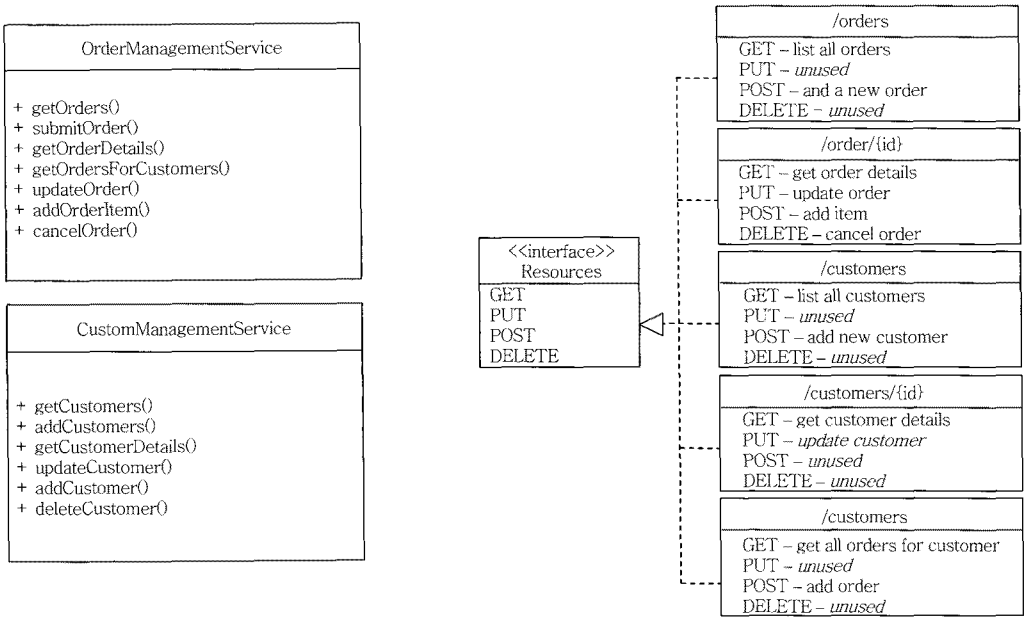
		SOAP 기반 웹서비스	RESTful 웹서비스
배경 및 현황		• 기업을 위한 비즈니스 응용에서부터 출발 • IBM, BEA(현재 IBM으로 통합), Oracle 등을 선두로 하는 웹서버 벤더에서 주창 • SOA의 서비스는 대부분이 비즈니스 컴포넌트로서의 의미를 가짐	• WEB 2.0은 서비스 애플리케이션에서부터 시작 • 구글, 아마존, 야후와 같은 인터넷 서비스 기업에 의해서 주창 • 맵이나 뉴스, 가젯 등과 같이 UI 성격을 갖는 서비스가 대다수임
특징		• The Machine-Readable Web: 사람보다는 기계가 해석할 수 있는 웹 • Stateful: 오퍼레이션 중 서비스 상태가 일관되게 유지, 관리되어야 함 • 엄격한 문법 검사, 서비스 계약에 충실 • 웹 서버 등 웹서비스 개발 환경이 지원되어야 함	• The Human-Readable Web: 사람이 해석할 수 있는 웹 • Stateless: 오퍼레이션 중 서비스/리소스의 상태를 관리하지 않음(HTTP의 기본 메커니즘), 필요한 경우에 직접 관리해야 함 • 기본 XML만으로 서비스 개발 가능 • 별도의 개발 환경 지원이 필요 없음
적용기술	전달 메커니즘	Remote Procedure Call	Publish/Syndicate Pattern
	전달 프로토콜	SOAP/HTTP, SMTP	HTTP
	서비스 명세	WSDL	WADL, XML, JSON, hREST(시맨틱 REST) 등
	서비스 레지스트리	UDDI	없음
	필요 스택	W3C의 WS-*스택(WS-addressing, WS-security 등)	없음
주요 적용 분야		트랜잭션 프로세싱	데이터와 UI(User Interface) 프로세싱
현재의 문제점		어려운 사용법, 무거운 프로토콜	표준의 부재, 관리가 어려움

RESTful 웹서비스를 제공하는 인터넷 서비스 업체들은 무료로 이용할 수 있는 서비스 매시업 환경을 함께 제공하고 있어 개발자 폭이 널리 확대되고 있다.

그러나 RESTful 웹서비스의 인기를 주도한 이유였던 인터넷 기본 외 별도의 표준을 요구하지 않는다는 특성이, 개발 및 관리를 어렵게 만드는 문제를 파생하게 되었다. 데이터에 대한 의미 자체가 비즈니스의 필수 요건이 아니었기 때문에 서로 데이터를 전송할 수 있는 정도의 상호 이해 수준의 표준만이 요구되었을 뿐 산업계 표준이 필요하지 않았다. 현재 REST의 산업계 표준이 없다 보니 개발에 있어서 여러 가지 문제점이 드러나게 되었다. 표준이 없는 경우에는 자체 스펙을 정하거나 유사 표준을 따라야 하는데, 자체 스펙을 정하는 것이 쉽지 않을 뿐더러 REST에 대한 잘못된 이해로 인한 잘못된 유사 표준들이 등장하는 경우도 발생하였다. 따라서 표준과 개발 인프라가 잘 갖추어진 SOAP 기반 웹서비스에 비해, RESTful 서비스는 표준과 개발 인프라에 얽매이지 않는 대신 서비스와 시스템도 REST 개념에 맞는 설계가 필요하다.

3. 서비스 사례

(그림 3)을 통하여 SOAP 기반의 웹서비스(a)와 RESTful 웹서비스(b)가 정보나 서비스를 공개하는 방식을 비교할 수 있다[3]. 그림 (a)와 (b) 모두 고객의 주문을 관리하는 기능을 가지는 웹서비스이다. SOAP 기반의 웹서비스(a)에서는 고객의 정보, 주문 정보 등을 검색, 변경, 삭제하는 오퍼레이션이 각기 존재하여 필요한 기능을 수행할 때 해당 오퍼레이션을 호출하는 방식으로 일반적인 프로그래밍 개념과 동일하다. 반면에 RESTful 웹서비스(b)에서는 총주문(/order), id를 가지는 특정 주문(/order/{id}), 총 고객(/customers), 고객 한 명을(/customer/{id}) 모두 리소스로 정의하고, 각 리소스에 URI을 할당한 후 URI에 대해 HTTP GET/ PUT/POST/DELETE



(a) SOAP 기반 웹서비스 (b) RESTful 웹서비스

(그림 3) 서비스 사례 비교

오퍼레이션을 수행한다. 예를 들어, '주문번호 12의 주문 상세 내역을 가져오라'는 요청은 SOAP 기반 웹서비스라면 getOrderDetails(order_no=12)로 호출될 것이며, RESTful 웹서비스라면 http GET/order/12로 수행할 것이다. 즉 order 12가 리소스로서 GET 메소드로 관련 정보를 얻을 수 있다.

이와 같이 SOAP 기반의 웹서비스를 사용하고자 할 때에는 웹서비스의 위치(바인딩 주소)뿐 아니라 오퍼레이션을 알아야 하는 반면, RESTful 웹서비스를 사용하고자 할 때에는 대상 리소스의 URI만 파악하면 된다. 이것은 모든 RESTful 웹서비스가 HTTP 메소드라는 공통의 인터페이스를 이용하므로 가능한 일이라 하겠다.

4. 시맨틱 웹서비스화

시맨틱 웹서비스란 인터넷에 개발된 수많은 웹서비스를 보다 의미적으로 활용하기 위하여 시맨틱 기술을 도입한 웹서비스를 의미하며, 시맨틱 웹서비스 기술[8],[20]이란 웹서비스에 다양한 의미 정보를 부가하여 궁극적으로 의미적 검색을 통한 동적 서비스 조합을 실현하는 데 필요한 요소 기술이다. 머지않은 미래에 인터넷의 수많은 서비스 중 사용자의 의도와 목적에 꼭 맞는 서비스를 손쉽게 찾고 또는 즉시 새로 만들어 사용할 수 있는 서비스 환경이 도래할 것이며, 이때 시맨틱 웹서비스 기술이 견인차 역할을 하게 될 것이다. 이에 본 절에서는 현재 널리 사용되고 있는 SOAP 기반의 웹서비스와 RESTful 웹서비스에 시맨틱 기술이 도입되며 시맨틱 웹서비스로 발전하고 있는 현황을 간략히 소개한다.

<표 3>은 SOAP 기반의 웹서비스에 시맨틱 기술을 도입하고 있는 연구 프로젝트 현황을 정리한 표로 기본 SOAP 웹서비스와 비교하여 설명하고 있다. CMU의 OWL-S 관련 프로젝트[8], DERI의 WSMO 프로젝트[6]-[9], Georgia Tech의 METEOR-S 프로젝트[10]-[15] 등에서 SOAP 기반 웹서비스에 서비스 명세, 어노테이션, 검색, 조합, 중재 등 시맨틱 핵심 기술을 적용하는 연구를 수행하고 있다.

OWL-S는 표준 웹 온톨로지 언어인 OWL을 개발한 미국의 DAML 컨소시엄에 의해 W3C 표준으

〈표 3〉 SOAP 기반 웹서비스의 시맨틱 기술 도입 현황

비교 대상 기술	SOAP 기반 웹서비스	시맨틱 웹서비스 프로젝트		
		OWL-S	WSMO [16]-[20]	METEOR-S [10]-[15]
서비스 어노테이션 (Annotation)	기능 없음	OWL-S	WSMO	SAWSDL, MWSAF 등 [11]-[13]
온톨로지 명세	기능 없음	OWL	WSMO	OWL, RDF
서비스 검색 (Discovery)	UDDI	OWL-S4UDDI (Matchmaking, Broker, P2P)	WSMO Matchmaker	METEOR-S Web Service Discovery Infrastructure (WSDI)[14],[15]
서비스 조합 (Composition)	OASIS WS-BPEL, W3C WS-CDL 등	OWL-S4UDDI+ Planner+ OWL-S Reasoner+ OWM-S VM	WSMO Studio (Choreography Designer, BPMO Editor) WSMX (Choreography Engine) BPEL4SWS, WSMO BPEL Engine	BPEL4WS Web Service Composition Framework(MWSCF) [14],[15]
서비스 중재 (Mediation)	기능 없음	OWL-S Broker	WSMO Mediator	-
서비스 수행 (Execution)	웹 응용서버(예, IBM Websphere, Oracle Web Logic, MS .NET 등)	OWL-S VM(Axis and WSIF(Web Services Invocation Framework) 이용)	WSMX	BPEL4J/ActiveBPEL 이용

로 제안된 시맨틱 웹서비스이다[8]. OWL-S는 서비스 개요를 기술하는 서비스 프로파일(profile)과 서비스 실행과 연관된 프로세스 정보를 제공하는 프로세스 모델(process model), 서비스 매핑을 담당하는 서비스 그라운딩(grounding)으로 구성되어 있다. OWL-S를 이용한 시맨틱 웹서비스 검색(discovery)에 대한 연구로는 OWL-S/UDDI Matchmaker, OWL-S Broker, OWL-S for P2P 등이 있다. 서비스 조합은 OWL-S4UDDI, Planner, OWL-S Reasoner, OWM-S VM을 이용한 접근 방법들이 시도되고 있다. 또한 OWL-S Broker를 이용하여 서비스 중재(mediation) 기능을 제공할 수 있으며, 서비스 실행은 OWL-S Virtual Machine을 이용한 접근 방법이 있다.

DERI 연구소에서 제안된 WSMO[9]는 시맨틱 웹서비스의 다양한 측면을 서술하기 위한 개념 모델로서, 4가지 시맨틱 웹서비스의 핵심 요소(ontologies, goals, web services, mediation)를 온톨로지 형태로 정의한 명세서이다. WSML은 WSMO를 표현하기 위한 언어이고, WSMO Studio에서 제공하는 choreography designer를 이용하여 서비스를 조합할 수 있으며, WSMX의 choreography engine[16]에서 실행한다. 또한 시맨틱 웹서비스를 실행하기 위한 BEPL 엔진에 대한 연구도 있다[17].

특히, Georgia Tech의 METEOR-S 프로젝트[10]에서는 WSDL(WSDL-S/SAWSDL), UDDI, BPEL4WS에 시맨틱을 추가하는 어노테이션(annotation)에 관한 연구를 집중적으로 수행하였으며[11]-[13], 이에 기반한 웹서비스 검색(discovery) (WSDI) 및 웹서비 조합(composition) (MWSCF) 등에 관련된 프로토타입을 구현하였다[14],[15]. 서비스 중재(mediation) 및 수행(execution)과 연계한 연구도 진행하였으나, 이들을 위한 독자적 프레임워크를 만들지는 않았으며 다양한 중재기법의 적용과 BPEL4J 엔진을 사용한 수행을 검증하였다. 현재 공개되어 있는 결과물로는 웹서비스 어노테이션 프레임워크인 "MWSAF",

WSDL-S/SAWSDL 어노테이션 도구인 "Radiant", WSDL 2.0 기반 SAWSDL 어노테이션 도구인 "Woden4SA-WSDL" 등이 있다.

이상과 같이 CMU의 OWL-S 프로젝트, DERI의 WSMO 프로젝트에서는 기존의 웹서비스를 시맨틱화 하기 보다 새로운 시맨틱 웹서비스 생성을 위한 시맨틱 컴퓨팅 기술에 중점을 두고 있으며, Georgia Tech의 METEOR-S 프로젝트에서는 기존의 웹서비스를 바탕으로 부분적으로 시맨틱 기술을 적응시키는 연구를 수행하고 있다. 따라서 기본 웹서비스로부터 시맨틱 웹서비스로 발전하는 단계에서 METEOR-S 프로젝트에서는 과도기적인 발전 방향을 추구하는 반면에, OWL-S 프로젝트와 WSMO 프로젝트는 혁신적인 서비스 개량을 시도하고 있다.

반면, RESTful 기반의 웹서비스에 시맨틱 기술이 도입되고 있는 현황[18],[19]으로는 hREST[13]의 경우 MicroWSMO[19]에 대한 연구가 있지만, SOAP 기반 웹서비스에서의 시맨틱 기술 도입 연구에 비해 연구 범위나 기술 표준화 상황이 미진한 상황이다. 전반적으로 기능적(functional), 비기능적(non functional), 행위적(behavioral), 정보(informational) 시맨틱 기반의 반자동 어노테이션 기법과 검색 기술들이 일부 연구되고 있으나, RESTful 웹서비스의 중재 및 컴포지션 기술에 대해서는 시맨틱 기술 도입의 관심이 필요한 시점이다. 이는 RESTful 웹서비스 구현에 관한 표준기술이 부재하여 구현기술의 제약이 없으므로 다양한 기술로 빠르게 인터넷에 전파되는 장점이 있지만, 반대로 구현 기술의 다양성이 RESTful 웹서비스에 시맨틱 기술을 적용하기 위한 본격적인 연구가 늦어지는 주요 원인으로 분석된다.

V. 결론

본 고에서는 산업계에서 널리 이용되고 있는 두 가지 대표적인 웹서비스 기술에 대해 서비스 구조, 기술, 이용, 시맨틱 서비스화의 관점에서 비교하였다.

현재는 두 기술이 본래의 목적에 맞게 각자의 분야에서 활발히 이용되고 있으나, SOAP-to-REST 또는 REST-to-SOAP 등 상호 변환 기술이 등장하고 있는 상황으로 보아 조만간 서비스와 리소스 개념의 경계가 허물어질 것으로 예상된다. 따라서 RESTful 웹서비스와 SOAP 기반 웹서비스의 구분없이 통합 검색 또는 조합이 필요할 것으로 보이며, 이를 위한 기술이 활발히 연구될 것으로 판단된다. 또한 시맨틱 서비스로의 발전 과정에서 SOAP 기반 웹서비스에 시맨틱 기술을 접목하려는 연구들이 시도되고 있으나 상용화 단계는 아니며, RESTful 웹서비스의 시맨틱 서비스 시도는 아직 미진한 상황이다. 그러나 곧 시맨틱이라는 요소가 두 기술의 공통 분모로서 서비스의 통합 검색 및 조합을 더욱 활발히 유도할 것으로 기대된다.

용 어 해 설

ROA(Resource Oriented Architecture): 인터넷의 모든 콘텐츠(예, HTML, 텍스트, 이미지, 이메일 등)를 리소스로 활용하기 위한 구조로서 REST가 지향하는 서비스 구조이며, 서비스 중심의 SOA(Service Oriented Architecture)와 대비되는 개념으로 주로 사용된다.

시맨틱 웹서비스(Semantic Web Services): 웹서비스에 시맨틱 웹을 접목시킨 기술로서, 온톨로지를 활용하여 웹서비스의 비기능적/기능적/정보적/행위적 특성을 부가적으로 명세(description)함으로써 궁극적으로 웹서비스 검색, 조합, 중재 과정을 자동화, 정교화하기 위한 기술이다.

약어 정리

API	Application Program Interface
DAML	DARPA Agent Markup Language
DERI	Digital Enterprise Research Institute
JSON	JavaScript Object Notation
OWL-S	OWL-based Web Service Ontology
PO-XML	Plain-Old XML
REST	Representational State Transfer
ROA	Resource Oriented Architecture
SAWSDL	Semantic Annotations for WSDL
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol

UDDI	Universal Description, Discovery and Integration
URI	Uniform Resource Identifier
WADL	Web Application Description Language
WSDL	Web Services Description Language
WSMO	Web Service Modeling Ontology
XML	eXtensible Markup Language

참 고 문 헌

[1] 김창환, "웹서비스 개념 및 응용 서비스 동향," 정보통신산업진흥원, 주간기술동향 1395호, 2009. 5., pp.13-26.

[2] Roy Fielding, "Architectural Styles and the Design of Network-based Software Architectures," Dissertation of Doctor of Philosophy in Information and Computer Science, University of California, IRVINE, 2000.

[3] Leonard Richardson and Sam Ruby, *RESTful Web Services*, O'Reilly Media, May 2007.

[4] Cesare Pautasso, "REST vs. SOAP: Making the Right Architectural Decision," WWW2008, Apr. 2008, pp.805-814. Available at http://www.pautasso.info.

[5] Greg Hines, "RESTful Web Services and Drupal," PingVision, 2008. Available at http://pingvision.com/files/restful_web_services_and_drupal.pdf.

[6] D. Fensel, M. Kerrigan, and M. Zaremba, *Implementing Semantic Web Services*, Springer, 2008.

[7] 문애경 외, "시맨틱 웹 서비스를 위한 시맨틱 어노테이션 기술 동향," 전자통신동향분석, 제25권 제2호, 2010. 4., pp.121-131.

[8] D. Martin et al., "OWL-S: Semantic Markup for Web Services," Nov. 2004. Available at http://www.w3.org/Submission/OWL-S/.

[9] Dumitru Roman et al., "D2v1.3. Web Service Modeling Ontology(WSMO)," Oct. 2006. Available at http://www.wsmo.org/TR/d2/v1.3/.

[10] http://lsdis.cs.uga.edu/projects/meteor-s/.

[11] K. Sivashanmugam et al., "Adding Semantics to Web Services Standards," ICWS'03, June 2003, pp.395-401.

[12] Nicole Oldham et al., "METEOR-S Web Service Annotation Framework with Machine Learning Classification," SWSWPC'04 Part of the 2nd ICWS'04, July 2004.

[13] R. Akkiraju et al., "Web Service Semantics - WSDL-S," A Joint UGA-IBM Technical Note, version 1.0, Apr. 2005.

[14] Kaarthik Sivashanmugam et al., "Framework for Semantic Web Process Composition," *Int'l J. of Electronic Commerce*, Vol.9, No.2, Winter 2004-5, pp.71-106.

[15] Kunal Verma et al., "METEOR-S WSDI: A Scalable Infrastructure of Registries for Semantic Publication and Discovery of Web Services," *J. of Information Technol. and Management, Special Issue on Universal Global Integration*, Vol.6, No.1, 2005, pp.17-39.

[16] http://www.wsmx.org/.

[17] Nitzsche et al., "BPEL for Semantic Web Services(BPEL4WS)," LNCS 4805, OTM 2007, pp.179-188.

[18] T. Vitvar, J. Kopecky, J, Viskova, and D. Fensel, "WSMO-Lite Annotations for Web Services," ESWC, 2008.

[19] D12V0.1, HRESTS & MICROWSMO, CMG WG Working Draft, 2009.

[20] http://www.wsmostudio.org/.