
웹 서비스 등록 절차의 개선 방안

최유순* · 신현철* · 박종구**

Improved Web Service Publishing Method

Yue-Soon Choi* · Hyun-Cheol Shin* · Jong-Goo Park**

이 논문은 2002년도 원광대학교 연구비를 지원받았음

요 약

현재의 웹 서비스 구조는 제공자가 UDDI에 서비스에 대한 정보를 등록하는 일로 시작한다. 사용자가 UDDI에서 서비스 목록을 검색하면 서비스의 위치를 전송받아 제공자에게 WSDL을 요청한다. 제공자는 WSDL을 전달하면, 사용자는 WSDL에 정의 되어 있는 프로시저와 파라미터 데이터 타입에 맞추어 서비스를 요청한다.

본 논문에서는 이러한 웹 서비스 절차를 개선하였다. WSDL을 저장하기 위한 WSDL베이스를 이용하였다. WSDL 베이스는 WSDL 매니저에 의하여 관리된다. 서비스 제공자가 서비스를 등록할 때 서비스에 대한 정보를 UDDI 레지스트리에 저장하며, 이 때 WSDL을 같이 전송하도록 하였다.

ABSTRACT

In present web service structure, the service begins when supplier register the information of its service to UDDI(Universal Description Discovery and Integration). When user searches service directory in UDDI, the position of service is transmitted to it. And then, user requests WSDL(Web Service Description Language) to the supplier.

When WSDL is delivered to user, user requests the service according to WSDL. This thesis simplifies this web service process. This makes WSDL owned by supplier stored to WSDLBASE when supplier register its service. Also this makes it possible that user search the service and receives WSDL.

키워드

웹서비스 UDDI WSDL

1. 서 론

웹 서비스는 인터넷 상에서 표준기술을 통해서 이용될 수 있는 모든 비즈니스 처리를 의미한다. 웹 서비스는 웹 혁명의 다음 단계로서, 분산 컴퓨팅의 차세대 주자라고 할 수 있다. 웹 서비스는 웹 인터페이스를 통해서 기능을 수행한다. 기존의 클라이언트와 서버가 어떤 기술, 어떤 언어, 어떤 장

비라고 하더라도 실행이 가능하다. 개발자에게는 자신이 선호하는 기술을 선택할 수 있도록 해준다.

웹 서비스는 인터페이스 작성에 필요한 객체기술을 재사용할 수 있는 컴포넌트의 특성을 잘 살리고 있다^[1]. 컴포넌트 기술에 대한 관심이 고조되기 시작할 무렵에 개발자들은 간단한 코드만 재사용했다. 이것은 일반적으로 사용하는 함수들

* 백석대학 컴퓨터학부

** 원광대학교 전기전자정보공학부

과 서브루틴으로 구성되어 있으며 많은 문제들도 가지고 있다. 만약 다른 프로젝트에서도 사용될 수 있도록 주의 깊게 만들어지지 않는다면 별로 쓸모없게 되고 다시 만들거나 재구현해야 했다.

웹 서비스는 기본적으로 분산 솔루션이며, HTTP나 XML 같은 인터넷 표준에 의존하면서 발전하고 있다^[2]. 웹 서비스 이전에는 새로운 COM 컴포넌트를 만들면 필요로 하는 회사들이 구매하여 해당하는 기능이 필요한 서버나 솔루션에 설치해야 했다. 웹 서비스에서는 이러한 방법을 사용하지 않는다. 다시 말하면, 개발회사들이 웹 서비스에 사전에 준비된 COM 컴포넌트들을 제공한다. 따라서 그것을 필요로 하는 회사들은 그 웹 서비스에 접속만 하면 되고 자신의 서버에 COM 컴포넌트들을 설치할 필요가 없다. 웹 서비스의 기능이 향상됨에 따라 이제는 접속만하면 된다. 업그레이드가 필요할 때마다 서버에 새로운 COM 컴포넌트들을 다시 설치하기보다는 웹 서비스에서 바꾸기만 하면 된다.

본 논문은 웹 서비스를 효율적으로 지원하기 위하여 연구되었다. 현재 웹 서비스를 지원하기 위한 여러 가지 표준이 지정되었다. 서비스 제공자는 요구되는 기능을 연관시키는 웹 서비스들을 보관하기 위해서 UDDI(Universal Description, Discovery and Integration) 레지스트리를 사용한다. UDDI 레지스트리는 개발자들이 정확하게 자신의 일을 처리하는데 도움을 주고 있다.

UDDI는 비즈니스 정보와 서비스의 내용을 저장하는데 사용되는 레지스트리이다. 웹 서비스 사용자는 UDDI로 등록된 비즈니스를 통해서 웹 서비스를 검색한다. UDDI 레지스트리 자체는 여러 웹 서비스를 외부로 드러내어 어떤 클라이언트라도 표준 SOAP 메시지를 사용하는 레지스트리를 찾을 수 있다.

WSDL(Web Service Description Language)은 웹 서비스가 표현하는 메소드와 각각의 메소드에 사용되는 파라미터들이 무엇인지 정의한다. WSDL 파일은 어떤 SOAP 메시지가 오고가는지 계약을 가한다. SOAP 메시지는 그 자체로서 XML로 인코딩되고, 서버에 요청되는 메소드와 클라이언트에 반환되는 데이터를 가지고 있다.

기존의 웹 서비스 구조에서는 서비스 제공자가 WSDL을 보유하고 있다. 서비스 사용자가 UDDI에서 서비스에 대한 위치정보를 확인한 뒤에 서비스를 이용하는데 필요한 정보를 얻기 위하여 서비스 제공자에게 WSDL을 요청하고 있다.

본 연구는 서비스를 UDDI에 등록할 때, 등록된 서비스의 위치정보와 WSDL을 관리하는 별도의

베이스를 두고자 한다. WSDL 베이스는 WSDL 문서를 보관하고 있다. 웹 서비스 검색 요청이 있을 때 UDDI 매니저는 서비스 위치 정보를, WSDL 매니저는 WSDL 문서를 전송하게 된다. 이는 WSDL 매니저를 통하여 WSDL 문서를 검색하고 요청된 서비스의 이용 절차를 간소화하기 위한 방법을 제시하는데 주된 목표를 두고 있다.

등록되는 서비스는 XML 형태로서 어떠한 환경에서도 포맷이 일정하다. 또한 전송은 HTTP를 통하기 때문에 방화벽에도 통과할 수 있는 장점이 있다.

본 논문의 구성은 2장에서 웹 서비스에 대한 전반적인 고찰을 한다. 그러므로 웹 서비스를 이용하는 방법과 현재 사용되고 있는 웹 서비스 기술을 알아본다. 3장에서는 본 논문에서 제시하는 WSDL 매니저의 개념을 정의하고 구조를 설계한다. WSDL 매니저를 이용한 서비스 구조를 나타내어 서비스 방식을 개선하기 위한 것이다. 4장에서는 3장에서 설계한 WSDL 매니저를 구현하고, 문서를 검색하기 위한 웹 서비스를 정의하여 구현한다. 그리고 WSDL 매니저를 기반으로 웹 서비스를 실행시켰을 때의 결과를 제시한다. 마지막으로 5장에서는 본 연구의 결과 및 웹 서비스에 대한 전망과 향후 연구과제에 대해 기술한다.

II. 관련 연구

본 장에서는 웹 서비스의 정의와 장점에 대하여 기술하고, 웹 서비스를 이용하는 절차 및 방법에 대해 고찰한다. 그리고 웹 서비스 제작시 사용되는 프로토콜인 SOAP의 개념에 대하여 설명한다.

2.1. 웹서비스 개요

인터넷에서 전자상거래 뿐만 아니라 다양한 웹 서비스를 지원하기 위해서는 자료전송을 담당하는 미들웨어로 CORBA보다 훨씬 간편한 SOAP을 이용하는 추세이다. SOAP을 이용하여 자료를 전송하고 서비스를 제공하는 측은 WSDL로 UDDI에 SOAP을 통하여 서비스를 등록한다[3,4,5,6,7].

그림 1은 기본 웹 서비스 구조를 보여주고 있다. 이 구조에서는 자동적으로 자료 전송이 가능하게 하고 있다.

- 1) Service Provider는 서비스를 제공하고, WSDL을 통해서 서비스 브로커에게 서비스를 전달한다.
- 2) Service Broker는 UDDI 레지스트리에 접근한다. Service Provider가 레지스트리로 알리

게 되면, Service Requestor는 레지스트리로 요청된 서비스를 갖고 있거나, 검색하게 된다.

- 3) Service Requestor는 UDDI를 통해 Service Broker에서 서비스를 검색하고, SOAP을 통해서 Service Requestor에게 서비스를 바인딩한다.

UDDI 레지스트리는 비즈니스 이름과 바인딩 정보, 어플리케이션 인터페이스 지점 또는 WSDL을 포함하고 있다[8,9,10]. CORBA나 RMI 같은 분산된 객체구조와는 달리, SOAP은 단지 Header와 Body를 가지고 있는 프로토콜이다. Service Requestor가 적당한 웹 서비스를 찾게 되면, Service Requestor로 SOAP 바인딩을 효율적으로 하기 위해서 Service Broker의 레지스트리에서 정보를 사용한다. WSDL은 웹 브라우저와 웹사이트 사이에서 서로 활동을 묘사하기 위해서 HTTP GET이나 POST 방식을 사용한다.

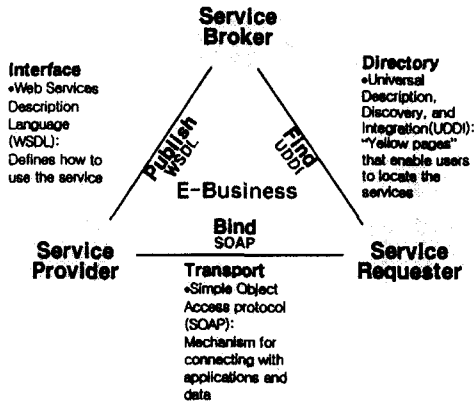


그림 1. SOAP과 UDDI, WSDL의 관계
Fig 1. SOAP, UDDI and WSDL Relation

2.2. SOAP

SOAP은 플랫폼과 무관하게 서비스, 객체 그리고 서버를 접근하기 위한 XML과 HTML의 사용 방법에 대하여 정의하고 있다. 달리 말하면, SOAP은 이질적인 소프트웨어 컴포넌트간의 접촉제 역할을 하는 프로토콜이다[11,12].

SOAP은 표준 분산 환경인 CORBA의 복잡한 자체 내의 구조와 사용법의 어려움을 해결하기 위한 방안으로 개발되었다. CORBA가 아무리 복잡한 구조를 가지고 있더라도 기본적인 개념은 서버에서 구현해 둔 서비스를 호출해서 처리된 결과 값만 받아서 이용하는 것이기 때문에 SOAP에서 호출방법

또한 XML을 기반으로 객체의 개념을 도입하여 호출/응답 서비스가 상호 독립적이면서 상호 운용성이 뛰어나다는 장점을 가지고 있다[13].

SOAP에서는 HTTP 혹은 SMTP(Simple Mail Transfer Protocol) 상에서 단순히 서비스 요청/응답에 대하여 표준 XML 문서 구조를 정의하여 서비스 제공업자와 클라이언트 간에 상호 교환하도록 구성되어 있다. 서비스를 요청하는 문서에는 사용할 서비스의 정보와 입력 값을 채워서 보내주면 서비스 제공업체는 이를 분석하여 해당 서비스를 수행한 후 그 결과 값을 응답 문서에 넣어서 돌려주는 방식이다.

SOAP은 SOAP envelope와 body를 가지며 선택적으로 header를 가질 수 있다. 그림 2에 SOAP의 구조를 나타내고 있다. SOAP은 무상태성이므로 어플리케이션이 저장해야할 것을 다룬다면 문제가 발생한다. 이때 세션ID를 이용해서 RPC 호출이 끝

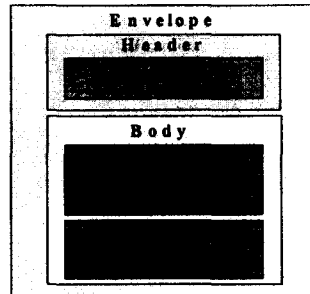


그림 2. SOAP의 기본 구조
Fig 2. Basic Architecture of SOAP

날 때까지 프로그램 상태를 기억하도록 한다. 그림 3은 클라이언트와 서버간의 SOAP을 통한 메시지 교환을 나타냈다.

클라이언트는 필요에 의해서 http를 경유, 객체를 생성하고 이의 메소드를 사용할 수 있다. 이런 과정이 복잡한 RPC를 거치지 않고 단지 http 상의 코멘트에 의해서 구동된다. SOAP을 쓰게 되면 단지 서버의 주소와 객체의 이름 그리고 전달되는 파라미터들만 명기하면 클라이언트의 운영체제나 환경에 상관없이 언제나라도 객체를 사용할 수 있다는 장점이 있다.

XML이 어플리케이션간에 전달되는 데이터를 표준화 했다면 SOAP은 서버의 객체를 호출하는 RPC 규약을 표준화함으로써 XML을 데이터 전달 도구로 사용할 때와 유사한 이점을 개발자들에게 줄 수 있다[11,13].

SOAP의 장점은 HTTP로 보내진다는 것이다. 대

부분의 방화벽은 인터넷을 통한 HTTP 트래픽을 최종 사용자(EndUser)에게 전송하는 것을 허용한다. 웹 서비스는 방화벽에서 같은 포트를 사용하면 작동하며 여러 기능들이 사용될 수 있도록 하면서 서버 어플리케이션들이 보호될 수 있도록 해야 한다.

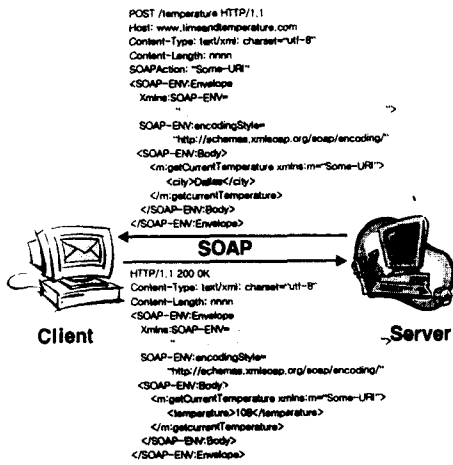


그림 3. SOAP의 메시지 전달 형태
Fig 3. Message Transfer Form of SOAP

III. 설계

현재의 웹 서비스는 UDDI의 UBR(UDDI Business Registry)가 서비스의 위치정보를 가지고 있다. 현재의 추세라면 앞으로 서비스의 양은 많아질 것으로 예측된다. 따라서 UBR은 현재보다 더욱 확대되어야 한다. 본 논문은 UBR의 기능을 확대시킬 수 있는 방법으로서 WSDL 문서를 저장하고 운영할 수 있는 서비스 구조를 제안하기 위한 연구이다. 그림 4는 본 연구에서 제안하는 웹서비스 구조도이다.

서비스 제공자는 UDDI 매니저에 서비스를 등록하면서 WSDL 파일도 같이 전송한다. UDDI 매니저는 UDDI 레지스트리에 서비스에 관한 정보를 저장하고, 서비스 사양서인 WSDL 파일은 WSDL 매니저에 의하여 WSDL 베이스에 저장된다. UDDI 매니저는 WSDL 파일의 정보를 WSDL 매니저에 저장하도록 하고 있다. 본 구성도에서 이용한 웹 서비스 아키텍처는 서비스 제공자가 서비스를 등록하면서 WSDL도 같이 등록하도록 하고 있다. 그림 5는 WSDL 매니저 기반의 웹 서비스 구조를 나타내고 있다. 여기에는 서비스 제공자가 등

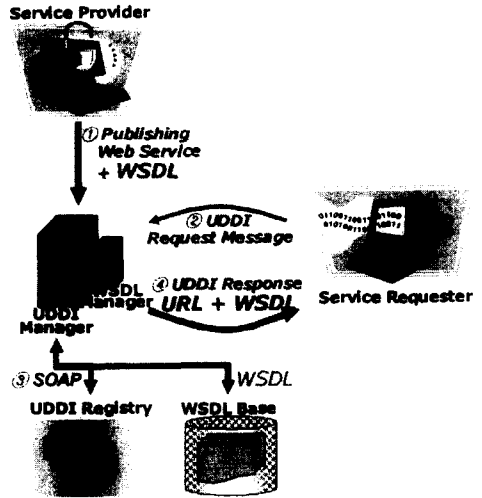


그림 4. 웹서비스 제안 구성도
Fig 4. Web Service proposal Diagram

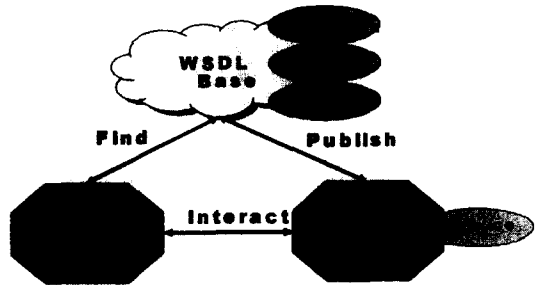


그림 5. 웹서비스의 역할
Fig 5. Act of Web Service

록한 WSDL이 WSDL 베이스에 포함되어 있다. 서비스 사용자가 서비스를 검색할 때, UDDI Manager는 서비스에 대한 정보와 서비스 사양서인 WSDL도 같이 전송한다. 그러므로 기존의 구조에서 서비스 사용자가 서비스 제공자에게 별도로 WSDL을 요청했던 절차를 한 단계 줄이게 된다. WSDL 베이스는 방대한 양의 자원을 저장할 수 있다. 또한 저장공간이 커진 베이스는 WSDL을 보관할 수 있도록 구성되었다. 이는 서비스 사용자가 요청한 서비스의 주소를 가지고 제공자에게 WSDL을 요청해야 하는 절차를 줄이도록 하기 위한 것이다. 따라서 서비스 사용자는 UDDI에서 검색된 서비스의 위치정보와 WSDL을 동시에 제공받는다.

IV. 구현

본 연구에서 제안한 WSDL 매니저를 기반으로 하는 웹 서비스 구조를 모델링 하기 위해서 객체지향 기법을 이용하였다. 유즈케이스 다이어그램에서는 Actor의 활동에 초점을 맞추고 있다. 여기에서 보여지는 Actor는 서비스 사용자와 서비스 제공자이다. 그림 6은 유즈케이스 다이어그램을 나타내고 있다.

서비스 제공자는 서비스 등록을 수행하고, 서비스가 등록될 때 WSDL도 같이 전송한다. 서비스 제공자가 등록한 서비스는 검색 디렉토리인 UDDI 레지스트리에, WSDL은 파일시스템으로 WSDL 베이스에 저장된다.

UDDI 레지스트리에는 서비스 제공자에 의하여 사용자의 정보와 서비스 정보가 저장된다. 서비스

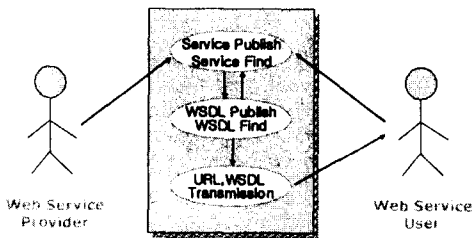


그림 6. 유즈케이스 다이어그램
Fig 6. Usecase Diagram

제공자는 서비스를 등록하기 위한 권한이 있어야 한다. 사용자 아이디와 암호, 이름, 전자우편 등 제공자에 관한 정보와 제공자가 속한 조직의 정보를 입력해야 한다. 입력된 제공자의 정보는 UDDI 레지스트리에 저장되어 관리되고, 서비스에 관한 정보와 서로 연결된다. 서비스 사용자에게 의하여 요청된 정보는 UDDI 매니저에 의하여 UDDI 레지스트리에서 검색을 하는데 검색요청이나, 검색결과 전송은 SOAP 형태로 이루어진다. 그림 7은 UDDI를 구성하는 테이블 구조이다.

UDDI에서 비즈니스 정보는 businessEntity 요소에 저장된다. name 요소는 회사명을 저장하고, description 요소는 비즈니스에 대한 텍스트 설명을 저장한다. contact 요소는 하위요소로 contact 요소를 저장하여 비즈니스 안에서 사람들의 접근 정보를 저장한다.

웹서비스는 하나 또는 두 개의 기술적인 웹 설명을 가지는데, 서비스 정보는 기술과 비즈니스 설명을 다룬다. 이것은 businessService 요소 안에 저장된다. 기술적인 설명은 서비스 사용자가 서비스

이름	데이터 형식	길이	Null 허용
userId	varchar	10	
pwd	char	10	
name	char	10	
email	varchar	30	
address	varchar	80	✓
phone	char	20	✓
comName	varchar	15	✓
comInfo	varchar	50	✓
comAddr	varchar	80	✓
comPhone	char	20	✓
comFax	char	20	✓

이름	데이터 형식	길이	Null 허용
serviceKey	char	50	
userId	varchar	10	
serviceName	varchar	50	✓
serviceDesc	varchar	70	✓
serviceURL	varchar	70	✓
urlName	char	20	

그림 7 UDDI 테이블 설계
Fig 7. UDDI Table design

와 통신하는 정보를 포함한다.

웹서비스를 호출하는 정보는 bindingTemplate에 저장된다. bindingTemplate는 기술과 비즈니스 데이터를 웹서비스에 연결한다. 따라서 웹서비스를 통합하는데 필요한 모든 기술적인 정보를 알아야 한다.

tModel 구조체는 기술적인 지문, 인터페이스, 메타데이터의 추상타입을 표현한다. 템플릿을 바인딩하는 것은 하나 EH는 그 이상의 tModel을 구현하는 것이다. bindingTemplate 내부에서 tModel의 특별한 곳에 접속 포인트를 등록한다. tModel은 그것을 참조하는 bindingTemplate으로부터 구분되어 공개된다.

본 연구에서는 서비스 정보 저장과 검색을 위하여 UDDIDiscover와 ServiceDiscover 컴포넌트를 이용하였다.

UDDIDiscover는 서비스 사용을 위한 컴포넌트이다. 서비스 사용자는 이용할 서비스가 어디에 있는지 검색하기 위해서 UDDI에 접속하여 위치정보를 요청한다. 이 때 UDDI 매니저는 서비스를 검색하여 서비스에 대한 정보와 위치 그리고 WSDL을 SOAP 형태로 사용자에게 전송한다.

ServiceDiscover는 서비스 사용자가 서비스를 사용하기 전에 검색기능을 제공한다. 사용자는 서비스가 존재하는 위치를 알아야 요청할 수 있으므로 UDDI 레지스트리에게 요청을 하게 된다. UDDI에서 검색된 서비스는 위치정보와 WSDL 파일이 함께 전송된다.

서비스 사용자가 의뢰한 서비스의 WSDL 파일을 전송하기 위하여 프록시로 접근한다. 그림 8은 WSDL 파일을 사용자에게 전송하는 프록시 코드이다. 프록시는 서비스 정보를 가지고 있는 UDDI에서 wsdlName 파일 이름을 가져온다. WSDL 파일은 파일 객체를 통해 스트림 객체를 취한 후, 바이트 배열안에 파일 내용을 기록한다.

```
# searching wsdl file name from database
SqlConnection dbConn = new SqlConnection(ConStr);
dbConn.Open();

string strSearchQuery = "SELECT * ";
strSearchQuery += "FROM tblServiceInfo ";
strSearchQuery += "WHERE serviceName = '"+rqtServiceName+"'";

//wsdl file transmission to service requestor
WsdFile wsdlFile = new WsdFile();
wsdlFile = tblServiceInfo.wsdlName;
string filePath = wsdlFile;

a.Close();
dbConn.Close();
```

그림 8 WSDL 전송 프록시
Fig 8. WSDL Transfer Proxy

V. 결론

웹 서비스는 인터페이스 작성에 필요한 객체 기술을 재사용할 수 있는 컴포넌트의 특성을 잘 살리고 있다. 서비스 제공자는 요구되는 기능을 연관시키는 웹 서비스들을 보관하기 위해서 UDDI 레지스트리를 사용한다.

UDDI 레지스트리는 일종의 디렉토리 서비스로서 개발자들이 정확하게 자신의 일을 처리하는데 도움을 준다. WSDL은 웹 서비스가 표현하는 메소드와 각각의 메소드에 사용되는 파라미터들이 무엇인지를 정의한다. WSDL 파일은 어떤 SOAP 메시지가 오고가는지에 대해서 계약을 가한다. 본 논문에서는 이러한 웹 서비스 절차를 간소화시켰다. 제공자가 가지고 있는 WSDL을 서비스 등록 시에 WSDL 베이스에 저장하였다. 그래서 사용자는 서비스 검색과 동시에 WSDL을 전송받을 수 있게 되었다.

웹 서비스를 지원하는 절차를 줄여 시간을 단축 시킴으로써 사용의 편리성을 제고시켰다. 또한 WSDL을 효율적으로 관리하며 사용자에게는 사용의 편리성을 제공했다.

추후 연구과제로 WSDL을 효율적으로 관리할 수 있는 새로운 방법이 제시되어야 한다. 웹 서비스가 많아지면 UDDI에서 검색되는 서비스에 대한 한계가 예상되기 때문이다. 많은 양의 서비스에 대한 또 다른 검색방법의 연구도 지속되어야 할 과제이다.

참고문헌

[1] A. Kropp, Ch. Leue and R. Thompson (editors), *Web Service for Remote Portlets, Working draft 0.85 (OASIS, 26 November 2002)*, <http://www.oasisopen.org/committees/wsrp>

[2] B. Daum and U. Merten, *System architecture with XML*, Morgan Kaufmann Publishers, San Francisco, CA, 2003.

[3] Ch. Kaler (ed), *Web Services Security*, <http://www-106.ibm.com/developerworks/web-services/library/ws-secure>

[4] D. König, M. Kloppmann, F. Leymann, G. Pfau and D. Roller, *Web service invocation framework: A step towards virtualizing components*, Proc. XMIDX'2003 (Berlin, Germany, February 17-18, 2003).

[5] F. Leymann, D. Roller and M.-T. Schmidt, *Web services and business process management*, IBM Systems Journal 41(2)(2002).

[6] F. Leymann, and D. Roller, *Business processes in a Web services world*, IBM Corporation 2002,

[7] IBM and Microsoft, *Security in a Web Services World: A Proposed Architecture and Roadmap*, IBM Corporation and Microsoft Corporation (2002), msdn.microsoft.com/ws-security

[8] Jiang-jie Zhang, Tian Chao, Henry Chang, Jen-Yao Chung, *XML-Based Advanced UDDI Search mechanism for B2B Integration*, Electronic Commerce Research, 3, 2003. Kluwer Academic Publishers.

[9] M. Champion, Ch. Ferris, E. Newcomer and D. Orchard, *Web Services Architecture*, <http://www.w3.org/TR/ws-arch/>

[10] S. Burbeck, *The Tao of e-business services*, IBM Corporation, 2000, <http://www-4.ibm.com/software/developer/library/ws-tao/index.html>

[11] Y.-S. Tan, B. Topol, V. Vellanki and J. Xing, *Implementing service Grids with the service domain toolkit*, IBM Corporation, 2002

[12] St.Laurent, Johnston, *Programming Web Services with XML-RPC*, O'REILLY, 2001.

저자 소개

최유순(Yue-Soon Choi)



2004 원광대학교 컴퓨터공학박사
2004 백석대학 IT교수
2004 사이버테러정보전학회 이사
※관심분야 : 웹서비스, 임베디드 시스템, 모바일프로그래밍, 소프트웨어공학



신현철(Hyun-Cheol Shin)

2002 원광대학교 컴퓨터공학박사
1994 백석대학 컴퓨터학부 교수
2004 한국정보처리학회 이사

※관심분야 : 모바일 프로그래밍, 임베디드 시스템, 정보통신, 정보보안, 소프트웨어공학



박종구(Jong-Goo Park)

1999 동국대학교 통계학박사
1981 원광대학교 전기전자정보공학부 교수
1999 한국게임학회 이사

※관심분야 : 웹프로그래밍, 전문가시스템, 소프트웨어공학, 신뢰성공학, 모바일 프로그래밍