

웹 서비스에서 UDDI 접근 방식의 개선 방안

최유순* · 소경영** · 박종구*

Access Method Improve Study for UDDI on Web Service

Yue-Soon Choi* · Kyung-Young So** · Jong-Goo Park*

요 약

본 연구는 UDDI에서 검색된 정보를 서비스 사용자에게 전달하여 서비스 사용자가 제공자에게 사양서를 요구하던 절차를 간소화하기 위해서 진행되었다. UDDI로부터 바로 제공자에게 WSDL문서를 요구하도록 하였다. 이를 위해서 UDDI에서 제공자에게 XML형태의 메시지를 생성하여 전달하도록 하였다. 이 메시지에는 서비스 사용자의 위치정보를 포함하고 있어 UDDI에서 요구한 문서가 서비스 사용자에게 전달된다. 메시지는 XML형태의 SOAP 프로토콜을 이용하였다.

ABSTRACT

Information searched in UDDI transfer web service user. Then web service user require interface to web service publisher. In this paper, we abbreviate procedure it. UDDI require WSDL document to web service publisher. For this reason, UDDI transfer XML form message to publisher. Message include user's location information and UDDI required document transfer from publisher to user. Message using SOAP protocol of XML format.

키워드

UDDI, Web Service, SOAP

1. 서 론

나날이 발전하는 인터넷 기술은 인류 역사상 가장 빠른 속도를 보이고 있다. 인터넷은 전세계 컴퓨터 통신망을 상호 연결하는 세계 최대의 통신망이다. TCP/IP를 통신 규약으로 하고, 전용 회선과 라우터로 Point-To-Point 방식으로 연결된다. 오늘날 다양한 분야에서 인터넷을 기반으로 하는 영역과 중요성은 점차 확대되고 있다. 그 중에서 웹 기술은 단순하고 편리한 사용자 인터페이스로 인하여 인터넷상에 분산되어 있는 정보와 서비스를 접근하는 매체로 폭넓게 보급되었다.

웹 서비스는 분산된 컴포넌트로부터 소프트웨어 시스템을 구축할 강력한 새로운 방법이다. 웹 서비스는 인터넷상에서 기능적인 어플리케이션을 인터페이

스로 이용할 수 있게 한다. 즉, 어플리케이션이 HTTP, XML, SMTP, 또는 Jabber와 같은 프로토콜을 이용하여 액세스 된다면 그것은 웹 서비스이다. 웹 서비스는 고차원적이긴 하지만 매우 단순하다.^[1]

웹 서비스는 전혀 새로운 것이 아니다. 수 년동안 인터넷을 통해서 전개되어 왔기 때문이다. 오늘날 대부분 인터넷을 통해 전개되고 있는 웹 서비스는 HTML 웹 사이트를 말한다. 이는 콘텐츠를 출판하고, 관리하고, 탐색하고, 검출하는 기술인 어플리케이션 서비스를 표준 프로토콜인 HTTP와 표준 데이터 형태인 HTML을 사용하므로서 이용할 수 있다. 클라이언트는 책을 사거나, 축하카드를 보내고, 뉴스를 읽는 등의 작업을 어플리케이션과 상호작용하면서 수행한다.^[2]

차세대 웹 서비스는 XML을 기본 데이터로 표시

*원광대학교 컴퓨터공학과

**익산대학 컴퓨터공학과

하며, 표준 프로토콜인 SOAP을 이용하므로서 RPC 보다 한차원 나은 형태의 메시지 형식을 정의하고 있다. 또한 이러한 웹 서비스를 기술하기 위해 WSDL을 사용하며, 데이터 공개 및 탐색을 위한 일종의 디렉토리 서비스로 UDDI를 활용하고 있다.

XML을 지원하는 서버에서는 서로 주고 받는 데이터를 읽을 수 있는 형태의 텍스트로 전달하고, 이 텍스트는 정의된 태그에 의해서 해석될 수 있으므로 개발과 디버깅에 드는 시간을 대폭 단축할 수 있다. 더불어 방화벽 안쪽에 설치된 서버라고 하더라도 대개 웹서비스를 위한 포트는 열어놓으므로 데이터를 요청해서 가져오는 것이 가능하다. XML을 지원하는 어플리케이션들이 많아지고, 서로가 XML 포맷으로 데이터를 주고 받으면서 최종 사용자가 받는 서비스의 질을 점점 높여 나가고 있다. 특히 개발자들이 현재의 웹 인프라를 이용해 다른 서비스에 접근해 데이터를 받아올 수도 있다.^{[3][4]}

SOAP은 플랫폼과 무관하게 서비스, 객체, 그리고 서버를 접근하기 위한 XML과 HTML의 사용 방법에 대하여 정의한 표준이다. 달리 말하면, SOAP은 이질적인 소프트웨어 컴포넌트간의 접착제 역할을 하는 프로토콜이다. 인터넷에서 구매를 위해 구매합수가 제공된 경우, 이 함수의 호출을 SOAP으로 표현한다. SOAP은 POST 방식으로 URL에 전달되지만, SOAP에서는 객체를 어떻게 실행하는지에 대해서는 언급을 하지 않기 때문에, 객체를 실행하고 함수를 호출하는 방법은 URL 내부의 코드에서 결정해 준다. 함수 호출에 대한 응답은 MessageType 헤더의 값으로 확인할 수 있다.] 응답 내용은 함수 호출 관련 엘리먼트 대신에 응답 엘리먼트에 기술된다. SOAP으로 호출과 응답이 이루어지는 웹 서비스 지원을 위해 UDDI 레지스트리를 검색하도록 한다. UDDI 레지스트리에는 다양한 서버에서 지원되는 서비스를 등록해놓은 저장소로서 사용자의 작업을 정확하게 수행하도록 하고 있다. UDDI 레지스트리에 저장된 정보를 활용하기 위하여 WSDL이 필요하다.^[5]

SOAP과 UDDI로 대변되는 웹서비스는 현재 상당한 정도의 표준화가 진척된 상태이며 다른 플랫폼에서 응용하고 있는 현실화된 방법이라는 것과 또한 개별적인 구축을 통해 서비스가 가능하다는 점에서

이점이 있다.

현재의 웹 서비스 지원 방식은 서비스 사용자가 UDDI에 등록된 서비스 정보를 요구하면, UDDI 관리자는 UDDI 레지스트리를 검색하여 해당 서비스에 대한 위치 정보를 사용자에게 알려준다. 사용할 서비스에 대한 위치를 알게된 서비스 사용자는 해당 위치로 서비스를 요청한다. 웹 서비스를 지원받기 위한 이러한 절차를 한단계 줄이기 위해 본 연구는 진행되었다.

UDDI 관리자는 서비스 사용자에게서 전송된 메시지를 받고 레지스트리를 검색한다. 레지스트리에 등록된 서비스가 검색되면 서비스 사용자에게 위치 정보를 전송하는 것이 아니라, UDDI에서 곧바로 위치 정보를 이용하여 서비스 제공자에게 메시지를 전송한다. 서비스 제공자는 UDDI에서 넘어온 요청을 받고 해당 응답을 서비스 사용자에게 전송한다. 본 연구를 위해서 UDDI를 만들어 사용하였다. UDDI는 Microsoft .NET Platform 기반에서 구축하였고, C#, ASP.NET, ADO.NET, XML, MS-SQL 등의 기술을 사용하였다.

본 논문의 구성은 1장은 서론으로서 전반적인 개요를 소개하였고, 2장은 관련 연구로서 웹서비스, SOAP, UDDI, WSDL에 관한 세부적인 설명하였다. 또한 현재 웹 서비스를 진행하고 있는 UDDI 시스템을 분석하였다. 3장에서는 본 연구에서 제안하는 웹서비스의 구성도를 설계하였고, 4장에서는 설계된 웹서비스 구성도를 기반으로 구현 과정을 나타내었으며, 5장에서 본 연구의 결과와 향후 연구 방향을 제시하므로써 본 논문을 맺는다.

II. 관련연구

본 장에서는 최신의 웹 서비스 기술에 관한 전반적인 특징을 기술한다. 이를 위해 웹 서비스의 개념을 정립하고, 웹 서비스에 관련된 요청메시지, 응답 메시지를 구성하는 프로토콜인 SOAP에 대하여 기술한다. SOAP으로 구성된 메시지가 서비스 목록을 검색하는 UDDI, 서비스 기술서인 WSDL에 대하여 고찰한다.

2.1 웹 서비스

1989년 CERN(Conseil European pour la Recherche Nucleaire, 유럽 입자물리학 연구소)에서 일하고 있던 팀 버스너리 박사에 의해 웹 개념이 처음 사용되었다. 실험에 대한 정보들을 손실없이 관리할 수 있도록 하기 위해 '하이퍼텍스트'방법을 제안하였고, 팀은 여러 문서들을 링크로 연결하여 원하는 정보를 빠르게 검색할 수 있었다. 초기의 웹 서비스는 건조한 텍스트와 최소화된 이미지를 바탕으로 시작되었다.^[6]

현재의 웹 서비스는 XML을 기본 데이터로 표시하며, 데이터 인코딩 표준인 SOAP이라는 지금의 RPC보다 한차원 나은 형태의 메시지 형식을 정의하는 프로토콜을 사용하고 있다. 이러한 웹 서비스를 기술하는 언어는 WSDL이며, 데이터 공개 및 탐색을 위한 일종의 디렉토리 서비스로 UDDI를 활용하고 있다.

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 접착제 역할을 하는 프로토콜이다. HTTP와 XML의 동시 사용을 원하는 경우, 표준적인 방법으로 경쟁적인 기술들을 연계하기 위한 매커니즘을 SOAP은 제공한다. 이것은 HTTP와 XML을 단일 솔루션으로 결합하여 전혀 새로운 수준의 상호 운용성을 제공한다. 예를 들면, 윈도우 환경에서 비주얼 베이직으로 작성된 단말 프로그램으로부터 UNIX 환경에서 작동하는 CORBA 서비스를 손쉽게 작동시킬 수 있으며, 자바 스크립트로 작성된 단말 프로그램으로부터 대형 컴퓨터에서 작동 중인 서비스를 작동시킬 수도 있다.^[7]

SOAP의 인터페이스는 XML로 표현되며 HTTP와 이메일과 같은 범용 프로토콜을 이용하여 인터넷으로 전송되기 때문에 싸고 간편하게 애플리케이션 소프트웨어들을 통합할 수 있으며 보다 간단한 컴퓨터 환경을 구축할 수 있다. SOAP을 이용하기 위해서는 약간의 프로그래밍 기술 능력이 요구되며 아직 해결해야 할 문제가 남아있는 것은 사실이지만

SOAP은 e비즈니스 통합을 위해 중요한 도구가 될 것으로 전망된다.

SOAP에서는 HTTP 혹은 SMTP 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 입력값을 채워서 보내주면 서비스 제공업체는 이를 분석하여 해당 서비스를 수행한 후 그 결과값을 응답 문서에 넣어서 돌려주는 방식이다.

구조를 보면, SOAP은 SOAP envelope와 body를 가지며 선택적으로 header를 가질 수 있다. SOAP은 무상태성이므로 어플리케이션이 저장해야할 것을 다룬다면 문제가 발생한다. 이때 세션ID를 이용해서 RPC 호출이 끝날때까지의 프로그램 상태를 기억하도록 한다. <그림 1>은 클라이언트와 서버간의 SOAP을 통한 메시지 교환을 나타내었다.^[8]

2.3 UDDI

UDDI는 마이크로소프트, IBM 그리고 Ariba에 의해서 제안된 프로젝트의 결과로서 기업들에게 어느 누구나 사용할 수 있는 Universal Business Registry에 그들이 제공하는 웹 서비스에 대한 정보

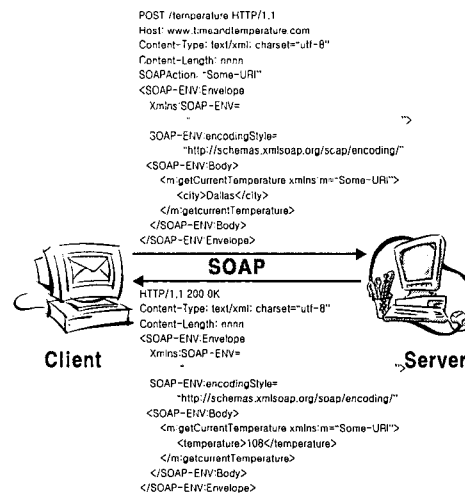


그림 1. SOAP의 메시지 전달 형태

를 알리는 것을 가능하게 하고 있다.^[9] 만일 UDDI가 방대한 지지를 얻는다면 선두 기업들은 이것이 모든

규모의 회사들이 전세계적인 기회를 디지털 혁명의 의해서 제공받게 될 B2B 온라인 상거래의 성장을 가속화 시키는 것에 앞장설 것이라고 말하고 있다.

UDDI Business Registry를 사용하면 협력업체에 의해 제공된 프로그래밍이 가능한 웹 서비스의 정보를 찾을 수 있다. 또한 이 레지스트리를 사용하면 모든 조직에서 제공할 웹 서비스에 대한 정보를 게시할 수 있다. 프로그래머의 인터페이스는 XML, SOAP 및 HTTP에 능숙한 프로그래머가 프로그램에서 직접 레지스트리와 상호 작용할 수 있게 하는 UDDI 레지스트리 사양의 일부로 정의되어 있다.^{[10][11]}

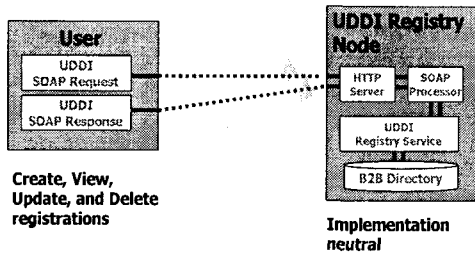


그림 2. UDDI와 SOAP

<그림 2>는 간단한 구성 요소 집합과 UDDI에 대한 이들의 관계를 보여준다. 프로그램에서의 요청 서식 설정, 요청 관리자로서의 요청 전송 및 응답 수신을 가능하게 하는 세 개의 기본 개체가 나타나 있다. 요청 관리자는 모든 XML 및 SOAP 세부 정보, 인증 및 오류 관리를 전부 숨긴다.^[12]

UDDI는 businessEntity, businessService, bindingTemplate, 그리고 tModel이 있다. businessEntity는 비즈니스에 관한 정보를 제공하면서 하나 또는 여러개의 businessService를 포함한다. 비즈니스는 서비스 제공자를 말한다. 웹 서비스에 대한 기술적이고 비즈니스적인 정의는 businessService와 businessTemplates에서 정의된다. bindingTemplate는 한 개 이상의 tModel을 포함한다. tModel은 서비스를 위한 기술적 사양을 정의한다.

2.4 WSDL

WSDL은 새로운 형식 정의 언어를 도입하는 것

은 아니고, 메시지 형식을 설명하는 다양한 형식 시스템의 필요성을 인식하여 정식의 형식 시스템으로 XML 스키마 규격(XSD)을 지원한다. 그러나 현재의 메시지 형식을 포함하여 앞으로 사용하게 될 모든 메시지 형식을 설명하는 데 있어 단일 형식 시스템 문법을 기대하기 어렵기 때문에, WSDL은 확장성을 통해 다른 형식의 정의 언어를 사용할 수 있도록 하고 있다.^[13] 또한 WSDL은 일반적인 바인딩 메커니즘을 정의한다. 이 메커니즘은 특정 프로토콜이나 데이터 형식 또는 구조를 추상 메시지, 작업 또는 종점에 연결하는 데 사용되고 추상 정의를 재사용할 수 있도록 한다. UDDI에 정의되어 있는 WSDL 서비스를 검색하고, 이용하는데, WSDL 문서는 service interface와 service implementation 두 부분으로 구분된다. <그림 3>에 그 구조를 보이고 있다. Service interface 문서는 하나 또는 그 이상의 서비스를 구현하는데 사용될 WSDL 서비스를 정의하고, service interface provider에 의해 개발되고 지원된다. Service implementation 문서는 서비스 인터페이스를 구현할 서비스를 정의하고, service provider에 의해 생성되고 사용된다.^{[14][15][16]}

service interface는 UDDI의 tModel에 제공되므로 서비스 재사용할 수 있도록 정의한다. service implementation은 서비스의 인스턴스를 정의한다. 각 인스턴스는 WSDL 서비스 엘리먼트에 정의되어 있고, service implementation 문서에 있는 각 서비스 엘리먼트는 UDDI의 BusinessService에서

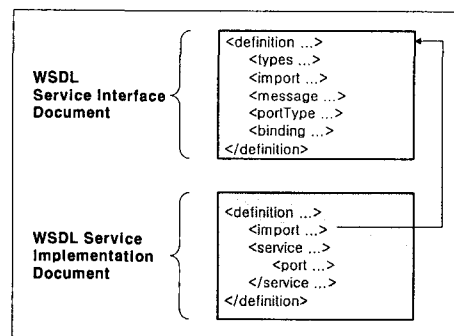


그림 3. WSDL 문서 형식

사용된다. WSDL 서비스가 정의될 때, service implementation이 BusinessService에 표현되기 전에

service interface는 tModel에 정의되어야 한다. <그림 4>는 WSDL과 UDDI의 맵핑을 나타낸 것이다.

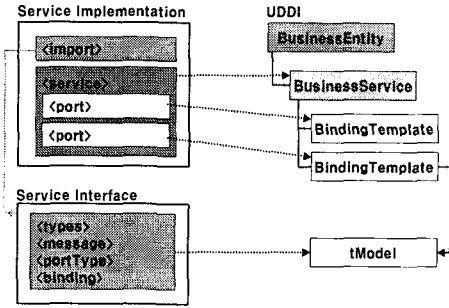


그림 4. WSDL과 UDDI의 mapping

III. 설 계

본 논문에서 제안하는 웹 서비스 지원 과정은 기존의 방법보다 절차를 간략화 시킨 것이 특징이다. 서비스 제공자는 UDDI 레지스트리에 서비스를 등록한다. 서비스 사용자가 UDDI 요청 관리자에게 UDDI를 요청하면, UDDI 레지스트리에서 위치정보를 검색하게 된다. 이 때, UDDI 요청 관리자는 사용자가 요청한 위치정보를 사용자에게 전달하는 것이 아니라, 바로 서비스 제공자에게 전송케 하므로서 통신상의 절차를 간단화 시킬뿐만 아니라 응답 시간도 단축시키고자 한다. <그림 5>는 본 논문에서 제안한 웹 서비스 구성도이다.

3.1 서비스 작성

서비스 제공자는 자신이 제공할 서비스를 작성한다. 서비스를 작성한 다음 사용자들이 언제라도 검색하고 이용할 수 있도록 UDDI 레지스트리에 등록하도록 한다.

3.2 서비스 등록

UDDI 레지스트리에 서비스를 등록하기 위해 레지스트리 웹 사이트에 방문하여 데이터를 관리하는데 사용할 특정 UDDI 운영자를 선택한다. 운영자를 선택하게 되면, 등록 자격 증명에 등록한다. UDDI 등

록 API를 사용하기 위해 UDDI 서버에 연결하려면 사용자 이름과 암호 쌍으로 이루어진 자격 증명이 필요하기 때문이다.

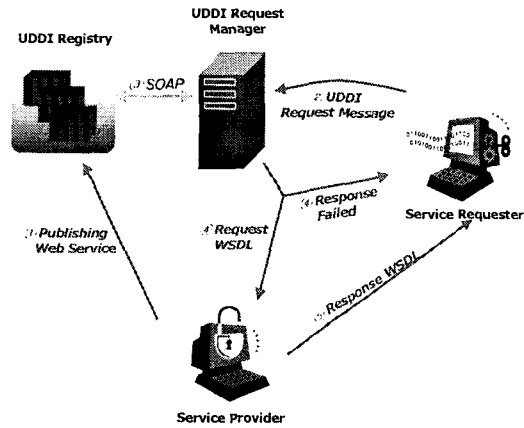


그림 5. 제안 시스템의 구성도

자격 등록이 끝나면 조직에 대한 정보를 등록한다. 조직 정보는 조직 이름과 설명 및 연락처(옵션)를 비롯한 업무 또는 조직에 대한 간단한 자료를 말한다. UDDI를 사용하는 주 목적은 다른 사용자들에게 서비스의 존재를 알리는 것이다. 웹 서비스를 제공하므로서 사용자는 업무와의 상호 작용에 사용할 수 있게 된다.

3.3 UDDI 요청 메시지

웹에서 제공하는 다양한 서비스를 이용하기 위해서 사용자는 서비스를 요구한다. 그러나 필요한 서비스가 어느 곳에 있는지에 대한 위치 정보를 알 수 없기 때문에 CORBA의 디렉토리 서비스에 해당하는 UDDI에게 요청 메시지를 보낸다.

3.4 UDDI 조회

UDDI 요청 관리자는 서비스 제공자가 제공하는 웹 서비스에 대한 정보를 찾기 위해 UDDI 레지스트리와 상호작용 한다. 요청된 서비스에 일치하는 정보를 선택하여 서비스 제공자에 대한 정보를 가져온다. UDDI 레지스트리에서 사용 가능한 정보는 회사 이름, 연락처 뿐만 아니라 설명 및 웹 서비스 기술 정보와 같은 업체 정보를 포함한다. 또한 웹 사이트 주

소, 웹 서비스 연락처, 웹 서비스 작성에 사용되는 기술 사양 및 서비스 계약을 설명하는 정보에 대한 포인터 등의 항목이 포함된다. XML 포맷으로 발견된 문서들을 이용하여 서비스 제공자에게 서비스 설명을 요청한다.

3.5 웹 서비스 호출

서비스 제공자는 UDDI 요청 관리자로부터 전달된 설명서 요청을 받고 서비스 사용자에게 XML 포맷의 WSDL 설명서를 전달한다. 서비스 사용자는 서비스 설명서를 사용하여 웹 서비스에 대한 요청을 만들어 서비스를 호출한다. 웹 서비스는 XML 포맷으로 응답을 반환한다.

IV. 구현

본 연구는 UDDI 게시용 서버로 Windows XP 에 서버를 설치하여 사용하였고, 서비스 제공자와 서비스 사용자는 Windows 2000 Server를 사용하였다. 윈도우는 가장 보편적인 컴퓨터 운영체제로 사용되고 있으며, 마이크로소프트사는 웹 서비스를 지원하는 다양한 툴을 지원하고 있기 때문에 연구 환경으로 적합하다. 서비스 개발은 .NET 환경에서 이루어졌고, UDDI 레지스트리를 가상으로 만들기 위해 UDDI SDK를 사용하였다. 또한 C#, ASP.NET, ADO.NET, XML, MS-SQL 등의 기술을 사용하였다.

웹 서비스 사용자는 웹 서비스 디렉토리인 UDDI 를 사용하여 서비스의 위치를 결정한다. UDDI는 웹 서비스의 디스커버리 도큐먼트에 링크를 가진다. XML 디스커버리 파일은 디스커버리 도큐먼트에 대한 요청에 의해 웹 서비스 제공자에게 전달된다. 디스커버리 파일은 WSDL 파일 링크를 가진다. WSDL 파일은 XML 포맷으로 서비스를 설명한다. 웹 서비스 사용자는 WSDL 파일을 사용하여 프록시 객체를 만들고 웹 서비스를 사용하기 시작한다.

UDDI에서 서비스 제공자에게 메시지를 전송하기 위하여 SOAP RPC를 이용하였다. UDDI를 클라이언트로 하고, 제공자를 서버로 정의하여 원격 메소드를 호출하였다. <그림 6>에 원격 메소드 호출 그림을 표현하였다.

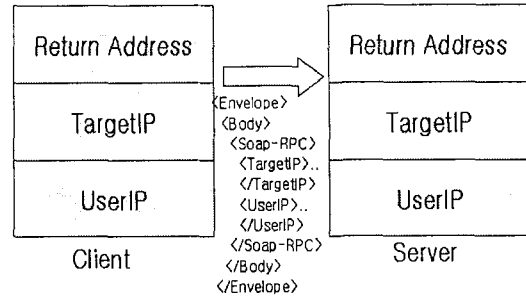


그림 6. UDDI에서 서비스 제공자에게 메소드 호출

```

namespace RequestUddiProxy{
using System;
using System.Web.Services;
using System.Xml.Serialization;
using System.Web.Services.Protocols;
using System.Diagnostics;

[System.Web.Services.WebServiceBindingAttribute(Name="RequestUddiSoap",
Namespace="http://www.mhps.or.kr/RequestUddiSoap")]

public class RequestUddi:
System.Web.Services.Protocols.SoapHttpClientProtocol {
[System.Diagnostics.DebuggerStepThroughAttribute()]
[System.Web.Services.Protocols.SoapDocumentMethodAttribute(
RequestNamespace="http://www.mhps.or.kr/RequestService",
RequestNamespaceUri="http://www.mhps.or.kr/RequestService",
ResponseNamespace="http://www.mhps.or.kr/RequestService",
Use=System.Web.Services.Description.SoapBindingUse.Literal,
ParameterStyle=System.Web.Services.Protocols.SoapParameterStyle.Wrapped)]
public string RequestService(string TargetIP, string UserIP){
object[] results = this.Invoke("RequestService", new object[] {
TargetIP,
UserIP});
}
}
    
```

그림 7. 메시지 호출을 위한 프록시 클래스

메소드는 프록시 클래스를 갖고 있으며 비동기 웹 메소드 RequestService를 가지고 있다. 프록시는 using으로 네임스페이스를 포함시켰다<그림 7>. 호출 메시지는 콜백 함수가 호출될 때까지 처리가 진행된다. 프로세싱이 완료되면 콜백 작업이 자동적으로 진행되어 서비스 제공자에게서 선택된 요구사항이 서비스 사용자에게 전달된다. <그림 8>은 비동기 콜백을 사용하여 메소드를 호출한 작업코드이다.

```

using System;
Using System.Runtime.Remoting.Messaging;
Using RequestUddiProxy;
class RequestCallBack
{
public static void Main()
{
RequestUddi ru = new RequestUddi();
AsyncCallback cb = new AsyncCallback(
RequestCallBack.RequestServiceCallback);
}
}
public static void RequestServiceCallback(IAsyncResult ar)
{
}
}
    
```

그림 8. 메소드 호출을 위한 콜백

V. 결 론

웹 서비스는 표준 인터넷 프로토콜을 사용하여 액세스할 수 있는 프로그램 가능한 응용 프로그램 로직이다. 웹 서비스의 필수조건은 표준 인터넷 프로토콜의 결합을 이용하여 메시지를 받고, 전달할 수 있어야 한다.

웹 서비스는 작동을 위해 서버 환경을 요구하지 않는다. 웹 서비스는 표준 인터넷 기술이 사용되어지는 곳은 어느곳에서도 배치될 수 있다. 이는 웹 서비스는 주된 역할을 하거나 서버를 통한 응용으로부터 PDA까지 어느 것에 의해서도 유용하게 사용된다.

웹 서비스는 서비스를 요구하는 어느 누구나 쉽게 찾고, 이용할 수 있어야 한다. 웹 서비스 구성에 있어서, 서비스 공급자는 서비스 레지스트리를 통해 제공되는 서비스 품목을 공개한다. 서비스 사용자는 그들에게 필요한 서비스를 찾기 위해 서비스 레지스트리를 검색한다. 서비스 사용자는 사람일 수도, 프로그램일 수도 있다.

기존의 서비스 제공 방식은 서비스 제공자가 레지스트리에 서비스 목록을 등록하면 서비스 사용자는 레지스트리를 검색하여 원하는 서비스의 위치를 받는다. 그러면 사용자는 위치정보를 보고 서비스 제공자에게 서비스에 대한 기술서를 요청한다. 서비스 제공자는 요청한 서비스에 대한 기술서를 WSDL 파일로 서비스 사용자에게 전송한다. 이러한 절차는 서비스 사용자가 UDDI에서 위치정보를 알려올 때까지 기다려야하는 번거로움이 있다.

따라서 본 연구에서는 서비스 사용자가 UDDI에 게 서비스를 검색하도록 요청하면, UDDI 관리자는 검색된 서비스에 대하여 바로 서비스 제공자에게 메시지를 전달하도록 하고 있다. 서비스 사용자는 위치정보를 알기 위하여 지체할 필요가 없고, 기술서 요구를 위해 번거로운 절차를 피할 수 있다.

본 연구를 위해 UDDI 레지스트리를 설정하였고, 게시할 서비스를 위해 .NET 환경에서 비주얼 베이직을 이용하여 asmx 파일로 된 서비스를 정의하였다. ASP.NET에서는 정의한 서비스에 대하여 서비스 기술서인 WSDL 파일이 XML 형태로 자동 생성되는 편리함을 준다. 웹 서비스의 자료 형태로 사용할 수 있는 XML 형태로 기술서가 지원되기 때문에 서비스 사용자는 서비스 검색 이후에 바로 서비스 기술서를

사용하기만 하면 된다는 간편함이 있다.

참고문헌

- [1] Mark Jones, "Implementing Enterprise-class Web Service", <http://www.flamenconetworks.com/>.
- [2] Bradley L. Jones, "It's Coming : We Services"
- [3] <http://www.xml.org/>
- [4] <http://www.xml.com/>
- [5] Fran Cohen, "Testing .NET Services", Windows Developer, February 2002.
- [6] Graham Glass, "The Web Services (r)evolution", "<http://www-106.ibm.com/developerworks/library/ws-peer3/>
- [7] Skonnard, A., "SOAP:The Simple Object Access Protocol", <http://www.microsoft.com/mind/0100/soap/soap.asp/>, Microsoft Internet Developer, January 2000.
- [8] SOAP toolkit, "<http://msdn.microsoft.com/library/en-us/soap/htm/>
- [9] <http://uddi.microsoft.com/>
- [10] <http://www.ibm.com/>
- [11] <http://www.hp.com/>
- [12] Aaron Skonnard, "Publishing/Discovering Web Services via DISCO/UDDI", "<http://staff.develop.com/aarons/>.
- [13] Ajamu Wesley, "WSFL:No Web service is an island", "<http://www-106.ibm.com/developerworks/library/ws-wsfl2>".
- [14] Aaron Skonnard, "SOAP:The Simple Access Protocol", MSDN magazine, 2000.
- [15] Kennard Scriber and Mark C. Stiver, "Understanding SOAP", SAMS, 2000.
- [16] Scott Seely, "SOAP : Cross Platform Web Service Development Using XML", PHPTR, 2002.

저자소개



최유순(Yue-Soon Choi)

1986년 원광대학교 컴퓨터공학과
학사

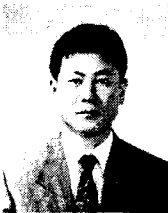
1990년 원광대학교 컴퓨터공학과
석사

2000년 원광대학교 컴퓨터공학과

박사과정 수료

2001~현재 원광대학교 BK21 겸임교수

※관심분야: 소프트웨어공학, 컴포넌트 소프트웨어,
웹프로그래밍, 웹서비스



소경영(Kyung-Young So)

1986년 원광대학교 전자공학과 학
사

1990년 원광대학교 컴퓨터공학과
석사

2001년 원광대학교 컴퓨터공학과

박사

1991년~현재 익산대학 컴퓨터학과 부교수

※관심분야: 소프트웨어공학, XML, 전문가시스템,
웹프로그래밍



박종구(Jong-Goo Park)

1969년 동국대학교 농업경제학과
학사

1975년 동국대학교 전자정보처리
학과 석사

1999년 동국대학교 통계학과 박
사

1981년~현재 원광대학교 컴퓨터공학과 교수

※관심분야: 전문가시스템, 소프트웨어공학, 소프트
웨어 신뢰성 공학, 모바일 프로그래밍