

사용자 측면의 웹서비스의 품질데이터를 제공하기 위한 UDDI의 설계 및 구현

Design and Implementation of UDDI to Provide the User-Side Quality of Web Service

조 풍 연* 이 남 용** 이 철 기***
(Poong-Youn Cho) (Nam-Yong Lee) (Choul-Ki Lee)

요 약

웹 서비스가 다양해지면서 웹 서비스의 품질은 웹서비스의 선택에 있어 중요한 중요대상이 되었다. 그러나 웹 서비스의 발견 및 관련 정보의 제공 등을 담당하고 있는 기존의 UDDI 아키텍처는 웹 서비스에 대한 품질 정보를 제공하지 않아 적절한 품질수준을 갖춘 적절한 웹 서비스를 선택하는 것이 어려운 실정이다. 이러한 필요에 따라 본 논문에서는 웹 서비스에 대한 품질데이터를 자동으로 수집하여 제공하기 위해 확장된 UDDI 를 제시한다. 이를 통해 특정 웹 서비스에 대해 자동화된 테스트를 바탕으로 응답시간, 처리량, 이용가능성, 신뢰성, 접근성에 대한 정보를 수집하여 사용자에게 제공함으로써 사용자가 웹 서비스를 선택하는 시점에 해당 웹서비스에 대한 품질정보를 고려할 수 있도록 하였다. 본 논문에서 제안한 UDDI를 활용할 경우 웹 서비스 사용자는 동일한 기능을 가진 웹서비스들 중에서 고품질의 웹서비스를 선택할 수 있으며, 이에 따라 웹서비스를 기반으로 한 애플리케이션의 신뢰성 확보 및 품질향상에 크게 기여할 수 있을 것으로 기대된다.

Abstract

Quality of the web service is one of the most important elements as various web services provide diverse functionality, and therefore, it is more and more difficult to satisfy customer's needs. Since the existing UDDI registry provides the basic information such as name and URL of a web service, users are having hard times to choose and customize their web service. In this paper, we propose an extended UDDI architecture for providing quality data of a web service. And in this architecture, we used an automated collection technique of data for testing web service which provides quality information to the users, including response time, throughput, availability, reliability and accessibility. With this new architecture, users and web developers can benefit from web services that customize information for the users and this ensures reliability of web-based applications.

Key words: Web-service, architecture, UDDI, XML, SOAP

† 본 연구는 숭실대학교 교내연구비 지원으로 이루어졌음.

* 주저자 : 숭실대학교 컴퓨터학과 대학원 박사과정

** 공저자 : 숭실대학교 컴퓨터학부 교수

*** 공저자 : 아주대학교 ITS대학원 특임교수

† 논문접수일 : 2009년 8월 5일

† 논문심사일 : 2009년 8월 22일

† 게재확정일 : 2009년 8월 24일

I. 서 론

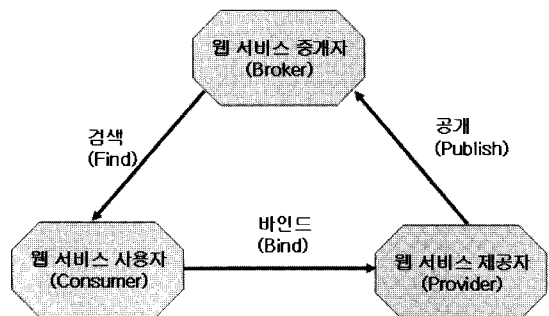
컴퓨터의 보급과 인터넷의 발달로 인하여 기업 간의 전자상거래를 위한 분산 컴퓨팅환경이 매우 빠르게 발전함에 따라, 기업 간의 서로 다른 시스템을 적은 비용으로 통합하려는 새로운 컴포넌트기반의 컴퓨팅 기술이 요구되고 있다. 이에 대한 하나의 해결책으로 제시된 것이 바로 웹 서비스(Web Service)기술이다 [1-3]. 웹서비스는 SOAP(Simple Object Access Protocol)이나 WSDL(Web Service Description Language), UDDI(Universal Description, Discovery, and Integration) 등의 표준 기술을 사용하여 네트워크에 연결된 다른 컴퓨터 간의 분산 컴퓨팅을 지원하는 기술이다 [4]. 이 중 UDDI는 다양한 종류의 서비스를 동적으로 발견하고 결합하는 기능과, 개방형 아키텍처를 기반으로 장소나 시간 그리고 디바이스의 종류에 구애 받지 않고 통합 서비스의 환경을 제공하는 기능을 제공한다. 그러나 웹 서비스상의 UDDI는 레지스트리라는 특성 때문에 웹 서비스의 회사이름, 주소, 전화번호, 회사에 대한 간단한 설명, WSDL 문서의 URL, 웹 서비스 종점 URL에 대한 정보만을 제공할 수 있고 웹 서비스의 품질을 식별하기 위한 정보를 제공할 수 없기 때문에 동일한 기능을 가지는 웹 서비스에 대해 사용자 또는 개발자가 고품질의 웹서비스를 선택하고 이를 바탕으로 신뢰성 있는 애플리케이션을 개발 및 통합을 수행하는 데 있어 많은 어려움이 있다 [5]. 이러한 문제에 따라 본 논문에서는 한국정보통신기술협회(TTA)에서 제시한 웹 서비스의 품질을 측정하기 위한 웹 서비스 품질모델과 웹 서비스 품질 테스트 가이드라인을 바탕으로 특정 웹 서비스 품질정보 제공을 위한 UDDI를 설계하고 구현하고 사례연구와 비교평가를 수행한다 [3, 6, 7]. 본 논문을 통해 제시되는 UDDI는 특정 웹 서비스를 사용하는 사용자로부터 품질 데이터를 수집하고 이를 바탕으로 해당 웹 서비스를 선택 시 적절한 품질정보를 제공함으로써 사용자 또는 개발자가 고품질의 웹 서비스를 제공받을 수 있게 한다. 또한 웹 서비스를 기반으로 한 애플리케이션의 신뢰성을 보장하고 품질을 향상시킬 수 있다.

II. 관련연구

1. 웹서비스

웹 서비스란 표준화된 메시지를 통하여 접근 가능한 네트워크 명령들의 집합을 기술한 인터페이스이다 [4]. 하지만 기업에서 개발되어진 웹 서비스는 플랫폼, 어플리케이션, 프로그래밍 언어가 모두 다르기 때문에 서로 상호 운용하기 위한 방법이 필요하다. 이에 여러 업체들과 표준화 기구에서는 UDDI, SOAP, WSDL을 기술표준으로 제정하여 각 웹 서비스가 서로 상호 운용이 가능하도록 하였다. UDDI는 웹 서비스의 공개(Publish)와 검색(Find)을 위한 XML 레지스트리(XML저장소)의 구현과 사용 방법을 표준화한 것이며, WSDL은 표준화된 방식으로 웹 서비스의 인터페이스를 기술하는 XML기술이다. 마지막으로 SOAP은 원격 응용프로그램 간의 정보를 교환하는데 필요한 구조를 표준화한 것이다. <그림 1>은 웹 서비스의 구조를 나타낸다.

현재 민간 표준단체인 OASIS에 의해 개발되고 사용되고 있는 UDDI표준은 크게 데이터구조, API스펙, 복제스펙, 오퍼레이션스펙으로 구성되어 있다. UDDI는 모든 정보를 XML을 이용해서 표현하고 저장하며, 이들 정보는 화이트 페이지, 옐로우 페이지, 그린 페이지로 분류되어 정보를 관리한다. 화이트 페이지는 회사의 이름, 주소, 전화번호 그리고 회사에 관한 설명이다. 옐로우 페이지는 산업계의 분류 체계별, 생산물과 웹 서비스의 분류체계별, 그리고 지역별 목록으로 구분되며, 그린 페이지는 각 회사에서 제



<그림 1> 웹서비스 구조
<Fig. 1> Structure of web-service

공하는 웹 서비스에 대한 기술적 정보를 말한다. 예를 들어 웹 서비스 시스템의 종점(Endpoint) URL이나, WSDL 문서의 URL 등이 여기에 속한다.

그러나 OASIS의 UDDI표준은 이와 같은 웹 서비스에 대한 단순 정보만을 제공할 뿐, 애플리케이션 개발과정에서 고려되어야 하는 웹서비스 품질에 관한 정보를 수집 및 제공하지 않는다. 이것은 웹서비스 기반의 애플리케이션 개발과정에서 품질을 고려한 신뢰성 있는 애플리케이션의 개발을 매우 어렵게 하며, 개발된 애플리케이션의 성능 및 안전성을 저하시키는 원인이 된다. 하지만 대부분의 기업이나 단체 등에서는 호환성 및 상호운용성 보장을 위해 OASIS의 UDDI 표준을 따르고 있으므로, 웹서비스 품질정보의 제공을 위해 UDDI의 표준 자체를 변경하는 것은 적절하지 않다. 따라서 웹서비스 기반 애플리케이션의 품질 문제를 해결하기 위해서는 표준 UDDI 구조를 준수하면서 웹서비스의 품질정보를 적절하게 수집하고 관리하기 위한 방안이 필요하다.

2. 웹서비스 품질

웹 서비스의 기술이 발달하면서 웹 서비스의 품질을 평가하기 위한 방법에 대하여 많은 연구가 이루어졌다[8]. IBM에서는 2001년 WSLA(Web Service Language Agreement)로 웹 서비스의 성능 및 안전성 측면 품질의 관리에 대한 연구를 수행하였으며 [5]. HP에서는 Open View Internet Service를 통해 웹서비스 품질에 대한 중요성을 언급하였다 [8]. 독일의 대학인 프레이아대학(Freie Universitat)에서도 웹 서비스 QoS 프레임워크를 개발하여 웹서비스 품질향상을 도모하였으며 [9], 국내에서는 한국정보통신기술협회(TTA)의 웹 서비스 품질모델과 웹 서비스 품질평가 가이드라인을 이용하여 많은 업체들이 이 표준을 기반으로 품질보증 체계를 구축하고 있다 [3, 6]. 웹 서비스의 품질은 ISO/IEC 9126에서 제시하는 소프트웨어 품질의 관점에서 바라보면 2가지로 품질로 분류된다. 이 중 하나는 웹 서비스를 하나의 응용소프트웨어 즉, 제품으로 보았을 때 고려되는 제품 자체의 품질이며, 다른 하나는 웹 서비스를 사용자가 이용할 때 느끼게 되는 사용

상의 품질이다. 또한 이러한 품질들은 사용시의 관점에 따라 사용자의 관점, 상호운용관점, 관리 및 보안 관점으로 크게 나누어 생각할 수도 있다 [3, 6, 7]. 하지만 웹 서비스의 서비스 지향적 특징을 고려할 때 웹 서비스의 품질은 이러한 두 가지 관점에서 평가하기 보다는 사용상의 품질 중심으로 그 모델을 정립될 필요가 있다. 이것은 웹 서비스가 웹을 통해 제공되는 하나의 제품 독립적 서비스이기 때문이다. 즉, 사용자는 웹 서비스의 구현은 고려하지 않으며, 웹 서비스가 제공하는 기능과 그 성능에 따라 특정 웹 서비스를 선택하기 때문이다 [10, 11]

따라서 본 논문에서는 사용자가 웹 서비스를 선택하는 과정에서 이러한 사용상의 품질요소들을 획득할 수 있도록 하는 UDDI를 제안한다. 이것은 전적으로 사용자의 관점에서 웹서비스를 테스트할 수 있도록 지원하며, 사용자의 환경에서 특정 웹서비스의 여러 가지 품질 요소들에 대한 정보를 획득할 수 있도록 한다. 사용자들은 이러한 정보를 바탕으로 고품질의 웹서비스를 선택 및 이용할 수 있으며, 이를 통해 웹서비스기반 애플리케이션의 신뢰성을 향상시키고 안정성을 보장할 수 있다.

Ⅲ. 웹 서비스의 품질요구사항

일반적으로 웹 서비스의 품질을 측정하기 위한 요소는 실행가격, 실행시간, 명성, 신뢰성, 사용가능성, 응답시간, 그리고 처리량 등이다 [12, 13]. 본 논문에서는 한국정보통신기술협회(TTA)의 웹 서비스 품질모델의 품질요소인 응답시간(Response Time), 처리량(Throughput), 이용가능성(Availability), 신뢰성(Reliability), 접근성(Accessibility)의 품질요소를 기준으로 삼았다 [6, 7].

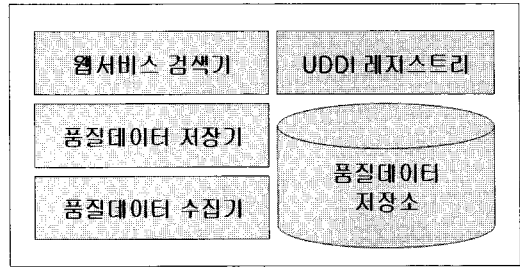
IV. 웹서비스의 품질데이터를 제공하기 위한 UDDI의 설계 및 구현

1. 웹 서비스의 품질데이터를 제공하기 위한 UDDI의 설계

본 장에서는 기존 웹 서비스 UDDI에서 제공하는

<표 1> 웹서비스의 품질요구사항
 <Table 1> Requirements of web-service

품질요소	정의	측정방법
이용가능성	웹 서비스가 존재하는지 혹은 즉시 사용가능한 상태인지의 측면에서의 품질[9], 즉 단위 시간동안 중에 웹 서비스가 다운되지 않고 있는 시간의 비율을 의미한다. 본 논문에서는 WSDL을 분석하여 웹 서비스의 메소드를 호출하는 방식으로 측정하였다.	이용가능성 = $1 - (\text{다운시간} / \text{단위시간})$
처리량	처리량은 웹 서비스가 단위 시간동안에 처리할 수 있는 서비스의 요청 수를 의미한다. 이러한 처리량은 웹 서비스 제공자에 대한 성능 지표로 사용될 수 있다. 처리량은 인터넷이라는 환경에서 얼마나 많은 사용자를 처리할 수 있는 지 나타낸다.	처리량 = $\frac{\text{완료된 요청 수}}{\text{단위시간}}$
응답시간	응답시간은 웹 서비스에 요청(Request)을 보내고 응답(Response)을 받는데 걸리는 시간을 의미한다. 이러한 응답시간은 실제 웹 서비스 호출에서 시간을 측정하며, 웹 서비스의 실행시간과 관련이 있다. 대개의 경우 응답시간은 일정 시간동안의 평균값으로 구한다	응답시간 = $\frac{\text{시스템응답 완료시간} - \text{사용자 요청시간}}$
신뢰성	주어진 시간 동안 약속된 서비스를 제대로 수행한 정도를 의미한다. 신뢰성 품질을 측정하기 위해서는 단위 시간동안에서 시스템다운 뿐만 아니라 서비스 요청이 실패에 대한 로그 분석을 통해서 에러간의 시간을 통하여 구해진다. 에러간 평균시간은 발생한 에러간의 평균 시간을 의미한다.	신뢰성 = $1 - \frac{\text{에러간 평균시간}}{\text{단위시간}}$
접근성	이용가능성이 보장되는 상황에서 서비스의 접근 가능한 정도를 나타낸다. 즉 한 시점에서 하나 이상의 서비스 요청에 대해서 성공적으로 응답할 수 있는지에 대한 정도를 의미한다. 어떤 경우에는 웹 서비스의 대해서 사용가능은 하지만 접근은 되지 않는 경우가 있다.	접근성 = $\frac{\text{응답 메시지 수}}{\text{요청 메시지 수}}$



<그림 2> 웹서비스의 품질데이터를 제공하기 위한 UDDI 아키텍처

<Fig. 2> UDDI architecture to provide quality information of web-service

웹 서비스의 기본적인 정보와 웹 서비스의 품질정보를 제공하기 위한 웹 서비스 품질 정보 제공을 위한 UDDI 아키텍처를 제시한다. <그림 2>는 해당 UDDI 아키텍처를 나타내고 있다. 웹 서비스 품질 정보 제공을 위한 UDDI 아키텍처는 UDDI레지스트리, 웹 서비스 검색기, 품질데이터 저장소, 데이터자동수집기, 품질데이터 저장기의 5가지 모듈로 구성된다. 웹 서비스의 품질정보를 측정하여 사용자에게 품질정보를 제공하고 저장하기 위한 구성모듈들의 역할은 <표 2>와 같다.

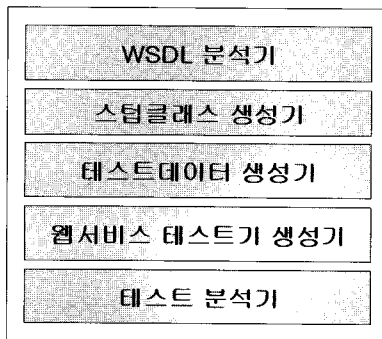
2. 품질데이터수집기 아키텍처

웹 서비스의 품질정보를 얻기 위해서는 품질데이터수집기를 이용하여야 한다. 웹 서비스의 품질을 측정하기 위한 품질데이터수집기는 <그림 3>과 같이 WSDL분석기, 스텝클래스 생성기, 테스트 데이터 생성기, 테스트 실행기, 테스트 분석기의 5가지 모듈로 구성되며, 각 모듈의 역할은 다음과 같다.

- (1) WSDL분석기 : 웹 서비스의 모든 정보를 담고 있는 WSDL에서 클래스명, 인터페이스, 메소드명 등을 XML DOM 파서를 이용하여 추출한다.
- (2) 스텝클래스 생성기 : 스텝클래스 생성기는 WSDL의 정보를 이용하여 스텝클래스를 만든다. 만들어진 스텝클래스는 웹 서비스와 메시

<표 2> 품질정보 제공을 위한 UDDI의 구성모듈 및 역할
 <Table 2> List of UDDI modules & roles to provide quality information

구성모듈	역할
UDDI 레지스트리	기존의 웹 서비스 UDDI레지스트리와 동일한 역할을 수행하며, 웹 서비스의 WSDL의 위치와 웹서비스에 관련된 각종 정보들을 저장, 검색할 수 있도록 지원한다.
웹서비스 검색기	사용자가 웹 서비스를 검색할 경우, 사용자가 입력한 정보를 바탕으로 UDDI레지스트리에서 관련 웹 서비스에 대한 정보를 회수함과 동시에 웹서비스 품질데이터 저장기에서 해당 웹 서비스에 대한 품질정보를 회수하여 사용자에게 제공하게 된다.
웹서비스 테스트 클라이언트 생성기	사용자가 사용자의 위치에서 웹서비스를 테스트할 수 있는 테스트 클라이언트를 생성하여 제공한다. 웹서비스 테스트 클라이언트는 사용자가 웹서비스를 선택하는 과정에서 필요한 품질정보를 획득하기 위해 사용되며, 웹서비스 테스트 클라이언트의 수행결과는 사용자에게 제공됨과 동시에 UDDI에 전송되어 저장된다.
웹서비스 품질정보 저장기	사용자가 웹서비스를 검색하고 선택하는 과정에서 사용한 웹 서비스 테스트 클라이언트로부터 해당 웹서비스의 테스트 결과와 사용자 관련 정보를 수집하여 품질데이터 저장소에 저장한다. 이것은 향후 다른 사용자들이 해당 웹서비스를 검색할 경우에 관련 정보의 하나로 제공된다.
웹서비스 품질정보 저장소	웹서비스 테스트 클라이언트로부터 전송된 결과를 저장하는 저장소로 UDDI 레지스트리와는 별도로 관리되며, DB 와 같은 일련의 데이터 저장소이다.



<그림 3> 품질데이터수집기의 아키텍처

<Fig. 3> Architecture of quality information collector

지를 주고받는데 쓰인다.

- (3) 테스트데이터 생성기 : 웹 서비스의 품질정보를 수집을 위한 데이터를 자동으로 생성한다.
- (4) 웹서비스 테스트기 생성기: 입력으로 들어온 WSDL 과 생성된 테스트 데이터를 바탕으로 웹서비스를 테스트하기 위한 테스트기를 생성한다. 이러한 테스트기는 사용자에게 제공되며, 사용자는 이를 사용하여 해당 웹서비스에 대한 실질적인 품질정보를 얻을 수 있다.
- (5) 테스트 데이터 수집기 : 사용자에게 제공된 테스트기는 사용자의 환경에서 웹서비스를 테스트하여 사용자에게 그 결과를 제공함과 동시에 UDDI 상의 테스트 데이터 수집기에 전송하게 된다. 테스트 데이터 수집기는 이러한 데이터를 받아 품질데이터 저장기에 전달한다. 이러한 정보는 품질데이터 저장소에 저장되어 있다가 다른 사용자가 해당 웹서비스를 검색할 경우에 해당 사용자에게 제공된다.

3. 품질데이터수집기의 수집 알고리즘

품질데이터수집기의 생성알고리즘은 다음과 같이 6단계로 구성되어 있다.

- (1) UDDI레지스트리에서 검색된 웹 서비스의 WSDL을 추출한다.
- (2) 웹 서비스의 WSDL을 XML DOM 파서를 이용하여 클래스명, 인터페이스명, 메소드명, 리턴형 등을 구분하여 저장한다.
- (3) 웹 서비스의 WSDL을 이용하여 스텝클래스를 만들기 위한 config.xml파일을 생성한다.
- (4) config.xml과 wscompile을 이용하여 웹 서비스와 메시지를 주고 받기 위한 스텝클래스를 생성하여 jar형식으로 압축을 한다.
- (5) XML DOM 파서를 이용하여 분석된 것 중에서 클래스, 인터페이스, 메소드를 이용하여 웹 서비스와 자동으로 메시지를 주고받으면서 품질테스트를 하기 위한 자바코드를 생성하고 컴파일한다. 다음 <그림 4>는 WSDL분석기에

```

public static WSDLData[] processWithDOM (String urlString) throws Exception {
    Document outputDoc;
    Element body;
    HashMap authors = new HashMap ();
    Vector v = new Vector();
    String Mname=""; //메소드 네임
    WSDLData[] data = null;

    DocumentBuilderFactory dbf=DocumentBuilderFactory.newInstance();
    DocumentBuilder db = dbf.newDocumentBuilder ();
    Document doc = db.parse (urlString);
    Element elem = doc.getDocumentElement();

    // 메소드 추출하기
    NodeList nl = elem.getElementsByTagName ("operation");
    data = new WSDLData[nl.getLength()/2];
    for (int n=0; n< nl.getLength (); n++) {
        Element author = (Element)nl.item (n);
        Mname = author.getAttribute ("name");
        String parameter = author.getAttribute ("parameter:Order");
        if (parameter == "") {
            break;
        }

        data[n] = new WSDLData();
        data[n].targetName = target;
        data[n].serviceName = serviceName;
        data[n].portName = portName;
        data[n].setMethodName(Mname);
        data[n].setParameters(parameter);
    }
}
    
```

<그림 4> WSDL 분석기에서 웹서비스 메소드 추출을 위한 코드

<Fig. 4> Code to extract web-service method from the WSDL analyzer

서 메소드와 인자를 추출하기 위한 핵심 코드를 보여주고 있다.

- (6) UDDI에서 스텝클래스를 압축한 jar파일과 웹서비스의 품질정보를 테스트하기 위한 .class파일을 제공한다. 이렇게 만들어진 데이터수집기는 <그림 5>와 같이 각 품질요소별로 다음과 같은 로직을 이용하여 자동으로 테스트데이터를 수집하게 된다.

```

Input Data : Parsing Result of WSDL
if( Check Availability == Success ) {
for( Each Method in Web Service ) {
// Check Accessibility of Web Service
if( Method Call == Success ) {
Check Response Time (Average of 10 Times)
Check Throughput for 1 Minute
Check Reliability for 1 Minute
}
else {
Send Error Message
}
}
else {
Send Error Message
}
}
    
```

<그림 5> 품질데이터 수집 알고리즘
<Fig. 5> Quality information collecting algorithm

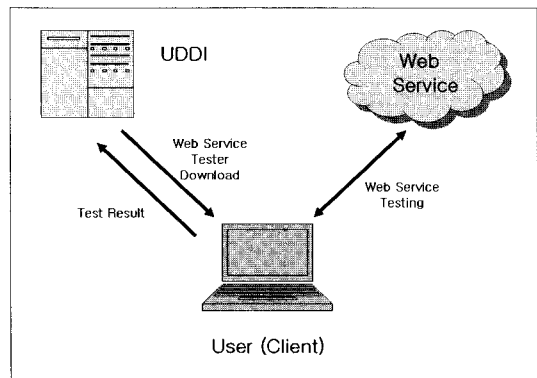
4. 웹서비스 품질데이터의 수집 및 제공

1) 품질데이터의 수집과정

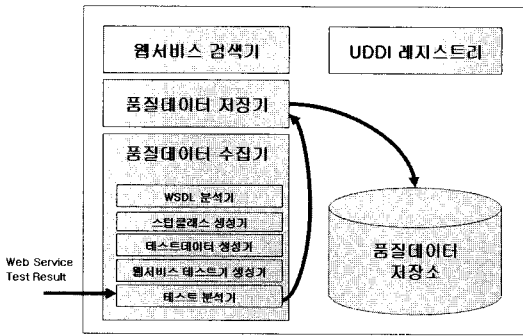
웹서비스의 품질데이터의 수집은 UDDI 에서 자동 생성되어 제공되는 웹서비스 테스트기를 통해서 이루어진다. 웹서비스 테스트기는 UDDI 에서 WSDL을 기반으로 자동생성되어 제공되며, 이를 위해 UDDI 는 내부적으로 웹서비스들의WSDL을 파싱하고 분석된 클래스명, 인터페이스명, 메소드명 등을 바탕으로 테스트데이터 생성기에서 생성된 테스트데이터를 결합하게 된다. 이렇게 생성된 웹서비스테스트기는 웹서비스를 사용하고자 하는 사용자가 필요시에 다운로드하여 실행할 수 있으며, 웹 서비스와 메시지를 주고받으면서 테스트데이터를 수집해 UDDI와 사용자에게 제공하게 된다.

웹서비스 테스트기로 부터 UDDI 로 전송되는 테스트 정보는 품질데이터 수집기의 테스트 정보 분석기에 의해 송신되며, 테스트 정보 분석기는 이 정보를 바탕으로 해당 웹서비스에 대한 품질 정보를 추출, 품질정보 저장기에 전달하게 된다. <그림 7>은 이러한 과정을 보여준다.

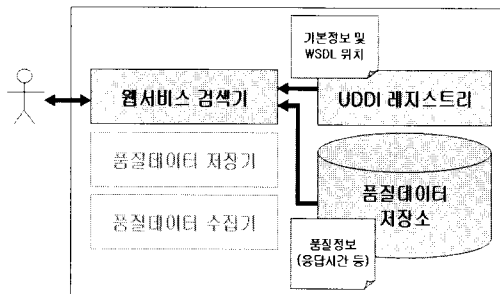
그리고 이러한 웹서비스의 품질정보를 받은 품질정보 저장기는 웹 서비스의 품질정보를 품질정보 저장소에 저장하여 추후에 이러한 품질정보를 해당 웹서비스를 사용하고자 하는 사용자에게 제공하게 된다.



<그림 6> 품질데이터의 수집과정
<Fig. 6> Process to collect quality information



<그림 7> 품질데이터의 저장과정
 <Fig. 7> Process to archive quality information



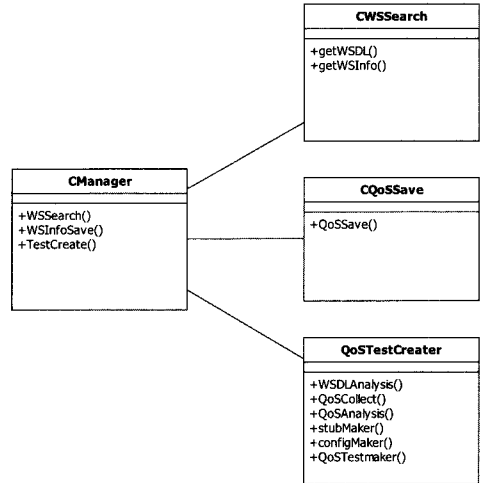
<그림 8> 품질데이터를 포함한 웹서비스의 검색과정
 <Fig. 8> Process to search web-service including quality information

2) 품질데이터의 제공과정

다음 <그림 8>은 사용자가 웹 서비스를 검색하여 웹 서비스의 WSDL과 품질정보를 제공하는 과정을 보여준다. 사용자가 웹 서비스에 대한 정보를 얻기 위해서 검색을 하면, 웹 서비스 검색기는 UDDI 레지스트리에서 웹 서비스의 기본정보와 WSDL의 위치를 얻어오고 품질정보 저장소에서 품질정보를 얻어와 이를 사용자에게 제공하게 된다.

V. 설계 및 구현

본 장에서는 본 논문에서 제시한 웹서비스 품질정보제공을 위한 UDDI를 설계하고 구현하였다. 해당 UDDI는 웹 서비스 사용자가 원하는 웹 서비스의 검색기능을 지원하며, 또한 웹 서비스에 대한 품질정



<그림 9> 웹서비스 품질정보 제공을 위한 UDDI의 클래스 다이어그램
 <Fig. 9> UDDI Class diagram to provide quality information about web-service

보를 제공한다. 사실UDDI의 설계는 UML을 이용하여 분석 및 설계하였다. 다음 <그림 9>는 사실UDDI를 구현하기 위한 UML의 클래스 다이어그램이다.

CManager클래스는 웹 서비스 검색 및 품질정보 저장과 품질정보 자동수집기와 같은 행위를 하기 위한 관리클래스이다. CWSSearch클래스는 UDDI에서 웹 서비스의 기본정보와 WSDL의 정보를 얻고, 품질정보저장소에서 웹 서비스의 품질정보를 얻는 역할을 하는 클래스이다. CQoSSave는 품질정보 자동수집기에서 보내온 웹 서비스의 품질정보를 품질정보저장소에 저장하는 역할을 하는 클래스이다. QoSTestCreator클래스는 테스트기를 만들기 위한 클래스로서, XML DOM 파서를 이용하여 분석하는 메소드, 스텝클래스를 만들기 위한 메소드, config.xml을 만들기 위한 메소드 등을 이용하여 품질정보 자동수집기를 생성한다.

웹 서비스는 여러 가지 언어를 이용하여 구현하였는데, EJB기반의 웹 서비스, .NET기반의 웹 서비스 여러 플랫폼으로 웹 서비스를 구현하였다. 구현된 웹 서비스는 Java WSDP Registry Server를 웹 서비스의 UDDI레지스트리로 사용하였다. 다음 <표 3>은 해당 UDDI를 구현한 환경을 설명한다.

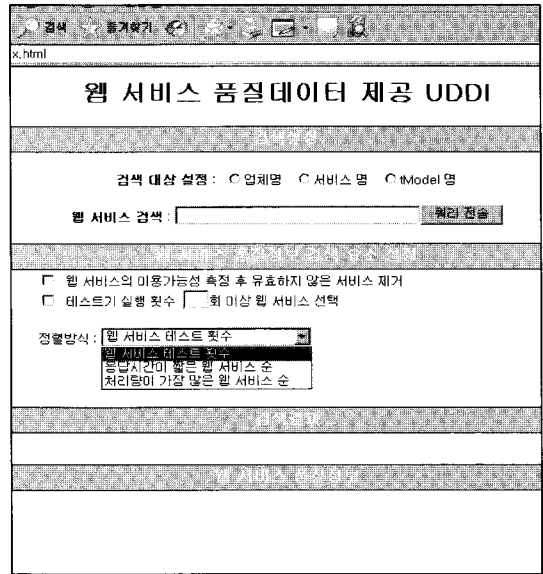
웹 서비스의 정보를 검색하여 사용하고자 하는 사

<표 3> 웹서비스 품질정보제공을 위한 UDDI의 구현환경
 <Table 3> UDDI system environment to provide quality information about web-service

운영환경	사양
OS	Windows 2000 Server
Web Server	Java(TM) Web Service Developer Pack 1.1의 Tomcat
JDK	JDK 1.6.0_02
개발언어 및 도구	XML, WSDL 1.1, SOAP, JSP, UDDI4J, JAVA Java(TM) Web Service Developer Pack 1.1
데이터베이스	MySql 5.0.24

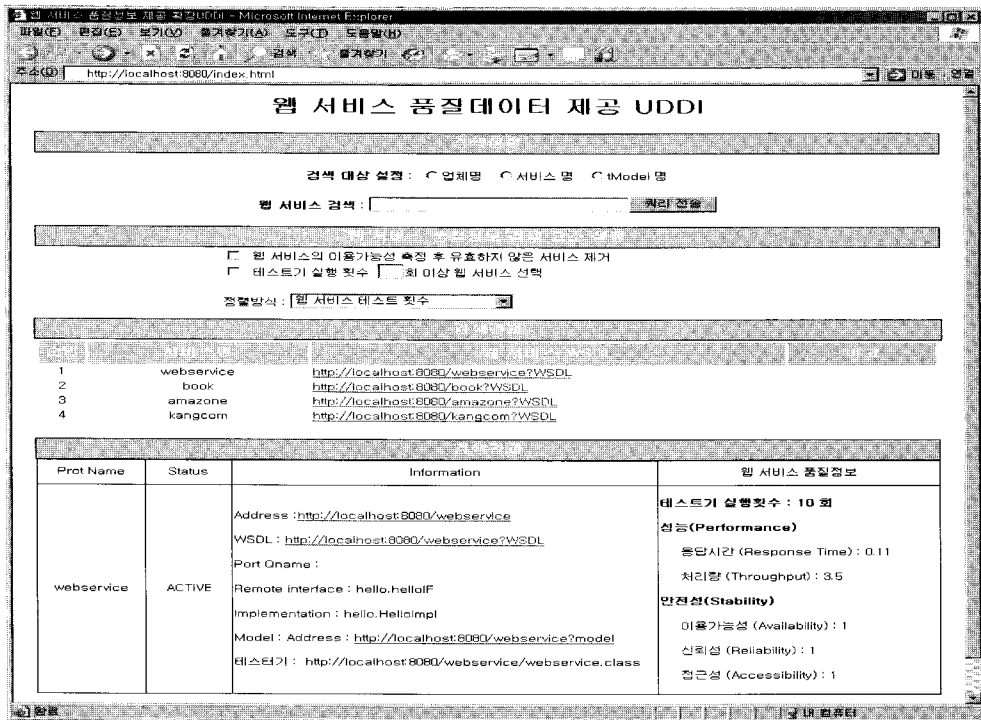
용자는 인터넷 익스플로러나 네스케이프 등의 브라우저를 통하여 해당 UDDI 에 접근하여 원하는 웹서비스를 검색할 수 있다. 다음 <그림 10>은 제안된 아키텍처를 바탕으로 구현된 UDDI의 메인화면이다.

웹 서비스 사용자는 UDDI의 메인화면에서 찾고자 하는 웹 서비스의 이름을 넣고 검색을 한다. 검색



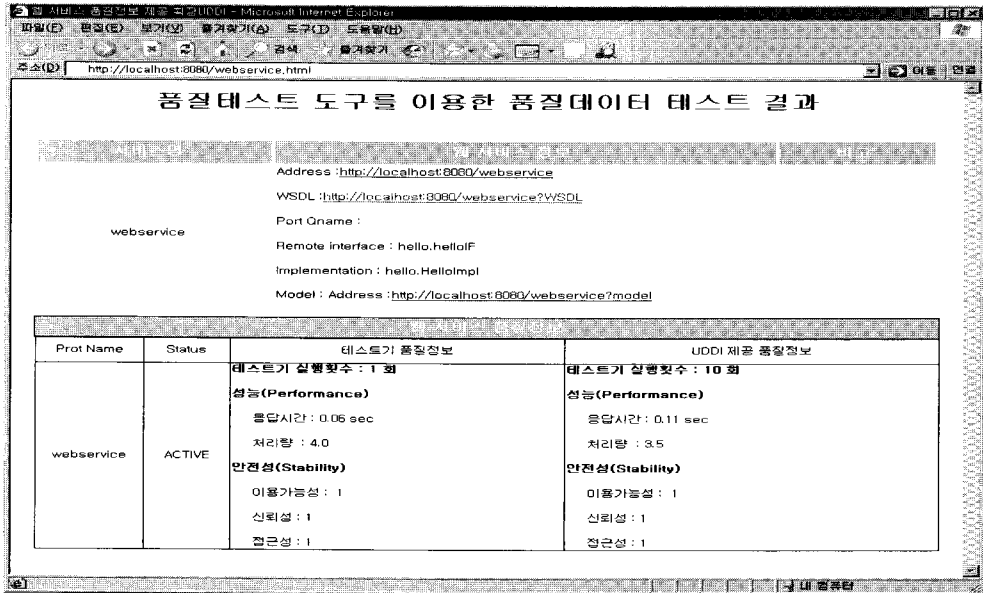
<그림 10> 웹서비스의 품질데이터 제공을 위한 UDDI의 메인화면

<Fig. 10> UDDI main UI to provide quality data of web Service



<그림 11 > UDDI의 검색 후 정보제공화면

<Fig. 11> Search result of UDDI



<그림 12> 데이터수집기 실행 후 결과화면
 <Fig. 12> Result after run of Data Collector

을 하게 되면 기존의 UDDI에서 제공하던 기본정보와 품질정보가 사용자에게 제공된다. 사용자는 제공된 정보를 보고 웹 서비스의 WSDL을 이용하여 웹 서비스를 구축할 수 있으며, 만약 웹 서비스의 품질 정보를 직접 테스트 하고 싶다면 웹서비스 테스트기를 다운받아서 웹 서비스의 품질정보를 직접 측정할 수 있다. 품질정보의 측정이 모두 끝이 나면, 품질정보는 사용자에게 제공됨과 동시에 UDDI에 저장되어 다른 사용자가 해당 웹 서비스를 검색할 시에 제공되게 된다.

<그림 11>은 UDDI의 검색기에서 검색 조건에 맞게 웹 서비스의 검색결과이다. 검색결과로는 웹 서비스명과 웹 서비스 WSDL을 제공하며, 그 밑에 웹 서비스의 품질정보를 자세히 보여주고 있다.

<그림 12>는 웹서비스 테스트기를 실행하여 품질 정보를 테스트한 결과를 보여주고 있다. 기존의 웹 서비스 품질정보와 웹서비스 테스트기를 이용하여 테스트한 품질정보를 서로 비교하여 볼 수 있다.

<표 4>에서는 기존의 UDDI와 본 논문에서 제안한 UDDI에서 제공하는 정보를 비교를 통해 본 논문에서 설계한 UDDI의 기능을 평가하였다. UDDI에서

는 회사이름, 주소, 전화번호, 회사에 대한 간단한 설명, WSDL 문서의 URL, 웹 서비스 종점 URL이 제공되고 있다. 본 논문에서 제안한 UDDI에서는 응답시간, 처리량, 이용가능성, 신뢰성, 접근성, 테스트횟수의 품질정보요소가 추가되었다.

<표 4> 기존 UDDI와 제안된 UDDI 간의 비교
 <Table 4> Comparing results from existing and proposed UDDI

	UDDI	웹서비스 품질정보 제공을 위한 UDDI
기본정보	회사의 이름	회사의 이름
	주소	주소
	전화번호	전화번호
	회사에 관한 설명	회사에 관한 설명
	WSDL 문서의 URL	WSDL 문서의 URL
	웹 서비스 종점 URL	웹 서비스 종점 URL
추가정보	없음	응답시간
	없음	처리량
	없음	이용가능성
	없음	신뢰성
	없음	접근성
	없음	테스트 횟수

VI. ITS 분야에서의 본 연구의 활용방안

본 논문에서 제시한 UDDI 아키텍처는 차량 애드혹 네트워크(VANET)[14]상에서 최적의 통신노드를 탐색을 위해 유용하게 활용될 수 있다. 차량 애드혹 네트워크의 경우, 이동하는 차량들 간의 유동적인 네트워크이므로 노드들 간의 연결이 빈번하게 끊기며 이로 인해 차량 애드혹 네트워크상의 정보의 전송의 신뢰성을 보장할 수 없다 [15].

그러나 차량 즉 노드상에서 연결가능한 이웃노드의 품질, 즉 가용성, 처리량 응답시간 등을 연결이전에 파악할 수 있다면, 정보교환을 위해 신뢰성 있는 노드를 선택하여 정보전송의 신뢰성을 높일 수 있다. 즉, 차량은 자신의 위치에서 연결 가능한 노드들의 품질정보를 유기적으로 파악 관리하기 위한 UDDI를 탑재하고, 이러한 UDDI는 사용자의 정보교환을 위해 연결 가능한 노드의 품질정보를 자동으로 수집함으로써, 정보송수신의 신뢰성을 높일 수 있을 것으로 판단된다.

또한, 이를 실현하기 위해서는 본 논문에서 제시하는 UDDI를 보다 개선하여, 연결 가능한 노드들 및 서비스들의 품질정도를 지능적이고, 유기적으로 수집할 수 있는 매커니즘의 추가구현이 필요하다. 따라서 이 부분은 향후 연구를 통해 보다 체

질정보는 제안된 UDDI 에서 WSDL 을 바탕으로 자동으로 생성 및 제공되는 웹서비스 테스트기 이용하여 사용자 환경에서 수집되며, 이것은 사용자 뿐 만 아니라 UDDI 의 품질정보 저장소에 저장되어 해당 웹서비스를 사용하고자 하는 모든 사용자들에게 최적 웹서비스를 선택할 수 있도록 한다.

본 논문에서 제안한 UDDI 아키텍처를 활용할 경우, 웹 서비스 사용자는 원하는 동일한 기능을 제공하는 웹 서비스 중에서 서비스의 실행이 가장 빠른 웹 서비스를 선택할 수 있으며, 웹 서비스의 처리량을 알 수 있어 사용자가 구현하고자 하는 웹 서비스의 처리량에 맞는 웹 서비스를 선택할 수 있다. 또 웹 서비스를 즉시 사용할 수 있는지 확인할 수 있으며, 웹 서비스의 기능들이 잘 동작하는지 신뢰성 등의 정보를 얻을 수 있다. 또한 웹 서비스 제공자는 같은 종류의 웹 서비스 개발 시에 UDDI에서 웹 서비스를 검색하여 테스트환경에서 나온 품질정보를 비교 및 평가하는데 유용하게 사용할 수 있다.

향후 연구에는 본 논문에서 제안한 웹 서비스의 품질정보 외에 사용자가 웹 서비스의 품질을 평가하기 위한 요소로 비용(Cost), 미터링(metering), 평판(Reputation), 적합성(Suitability) 등에 대한 연구가 진행될 것이다. 또한 WSDL 을 기반으로 웹 서비스 개발자가 직접적으로 웹서비스 테스트를 위한 데이터나 로직 등을 기술할 수 있는 방법을 연구할 예정이다.

VII. 결 론

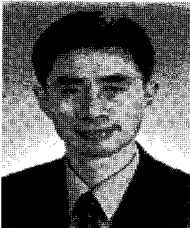
UDDI는 웹 서비스를 검색하고 등록할 수 있는 웹 서비스 레지스트리이다. 하지만 레지스트리라는 특성 때문에 웹 서비스의 기본적인 정보만을 제공할 수 있을 뿐 사용자가 웹서비스 선택 시 최적의 웹서비스를 선택하기 위해 필요한 품질정보를 제공할 수 없다는 문제점을 안고 있다. 이에 본 논문에서는 웹 서비스의 품질정보를 얻기 위한 방법으로써 기존의 UDDI레지스트리를 확장하여 제공자의 이름과 WSDL 의 위치와 같은 기본적인 웹서비스 정보 뿐 아니라, 응답시간, 처리량, 이용가능성, 신뢰성, 접근성과 같은 품질데이터를 제공하기 위한 UDDI 아키텍처를 설계하고 이를 구현하였다. 웹 서비스의 품

참 고 문 헌

- [1] 이강찬, 이승윤, “웹 서비스 표준 기술 동향과 전망,” *한국정보과학회지 웹서비스 특집*, 제22권, 제10호, pp. 11-18, 2004. 10.
- [2] 안무정, 이승윤, “웹 서비스 플랫폼 현황과 도입 전략,” *한국정보과학회지 웹서비스 특집*, 제22권, 제10호, pp. 41-47, 2004. 10.
- [3] *NCA- IV-RER-04052, 웹 서비스 품질 모델 및 테스트 가이드 라인연구*, 한국전산원, pp. 1-110, 2004.
- [4] *Web Service Architecture*, W3C Working Group Note, 2004.

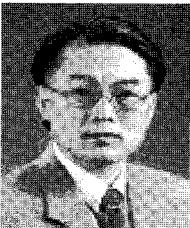
- [5] S. Patil and E. Newcomer, "ebXML and Web Services," *IEEE Internet Computing*, vol. 7, no. 3, pp. 74-82, May. 2003.
- [6] TTAS.KO-10.0204, 웹 서비스 품질 모델, 한국정보통신기술협회(TTA), pp. 1-36, 2006.
- [7] TTAS.KO-10.0204, 웹 서비스 품질 테스트 가이드 라인, 한국정보통신기술협회(TTA), pp. 1-41, 2006.
- [8] M. Tian, A. Gramm, T. Naumowicz, H. Ritter, and J. Schiller, "A concept for QoS integration in web service," *Proc. Int. Conf. Web Information Systems Engineering Workshop*, pp. 149-155, Dec. 2003.
- [9] A. Dan, A. R. Franck, A. Keller, R. King, and H. Ludwig, *Web Service Level Agreement (WSLA) Language Specification*, IBM, 2002.
- [10] A. Padovitz, S. Krishnaswamy, and S. W. Loke, "Towards Efficient Selection of Web Services," In *The 1st International Workshop on Web Services and Agent-based Engineering(WSABE2003)* Sydney, July 2003.
- [11] J. Day, "Selection the best web service," *Proc. Grad. Symp.*, CS Dept., University of Saskatchewan, pp. 7-8, 2004.
- [12] HP Open View Internet Services, <http://www.openview.hp.com/products/ovis/>.
- [13] D. A. Menasce, "QoS issue in web services," *IEEE Internet Computing*, vol. 6, no. 6, pp. 72-75, Nov. 2002.
- [14] V. Cherfaoui, T. Denoeux, and Z. L. Cherfi, "Distributed data fusion: application to confidence management in vehicular networks," *Proc. Int. Conf. Information Fusion*, pp. 1-8, July 2008.
- [15] 이상선, "VANET(Vehicle Ad-hoc Network) 환경에서의 라우팅 기술 및 서비스 개발 동향," *KIISE*, vol. 22, no. 1, pp. 62-80, 2008. 5.

저자소개



조 풍 연 (Cho, Poong-Youn)

1996년 : 동국대학교 정보산업대학원 전자계산과(공학석사)
2007년 : 숭실대학교 일반대학원 컴퓨터학과(박사수료)



이 남 용 (Lee, Nam-Yong)

1999년 : 숭실대학교 컴퓨터학부 교수
1990년 ~ 1993년 : 미국 미시시피주립대학교(MSU) 경영정보학(MIS)(경영학박사)
1980년 ~ 1983년 : 고려대학교 경영대학원 경영정보학(MIS)(경영학석사)



이 철 기 (Lee, Choul-Ki)

2006년 3월 ~ 현재 : 아주대학교 ITS대학원 특임교수
2004년 ~ 2006년 : 아주대학교 교통연구센터 부센터장
2000년 ~ 2004년 : 서울지방경찰청 교통개선기획실장
1998년 : 아주대학교 건설교통공학과 교통공학전공 졸업 (박사)