

UDDI 3.0 레지스트리 서버의 설계 및 구현[§]

김영선*, 유수진*, 박송희*, 이경하*, 이규철**

A Design and Implementation of the UDDI 3.0 Registry Server[§]

Young-Sun Kim, Su-Jin Ryu, Song-Hee Park, Kyong-Ha Lee, Kyu-Chul Lee

Abstract

UDDI is the open standard for distributed Web-based registry standard which supports registrations and discoveries of Web services. Recently, UDDI version 3.0 specifications were opened by OASIS UDDI Spec. technical committee. UDDI version 3.0 supports many useful functions, which are not supported in version 2.0 specification, such as nested query, enhanced find qualifiers, digital signatures subscription, etc. Despite of its merits, complexity of the standard and backward-compatibility support makes it difficult to adopt this standard in real application domains. In this paper, we develop the Registry server which support UDDI version 3.0 standard. This server is based on common RDBMS and it supports all functions of UDDI version 3.0 including backward-compatibility with version 2.0 API, XML digital signature and URI-based key schemes.

Keyword : e-Business, UDDI, Web Services

[§] 본 연구는 정보통신부의 대학 IT연구센터(ITRC) 지원을 받아 수행되었습니다.

** 충남대학교 대학원 컴퓨터공학과

*** 충남대학교 공과대학 컴퓨터공학과 교수

1. 서론

UDDI(Universal Description, Discovery and Integration Service)[1]는 웹 서비스에 대한 디렉토리 서비스를 지원하기 위해 개발된 표준으로 웹 서비스를 등록, 검색하기 위한 메커니즘을 제공한다. 또한 UDDI 는 각 레지스트리 노드 간의 데이터 복제를 통하여 여러 레지스트리 노드들이 서로 비즈니스 정보를 공유할 수 있도록 하는 분산 레지스트리 개념을 도입함으로써 글로벌 비즈니스와 서비스에 대한 공통된 접근 방법을 제공할 수 있는 이점을 가진다. 최근에는 OASIS 의 UDDI 표준 기술 위원회를 통해 UDDI 버전 2.0 이 OASIS 표준으로 등재되었으며, 버전 3.0 이 제정, 공개되어 있다.

본 논문에서는 최근에 공개된 UDDI 3.0 표준을 지원하는 레지스트리 서버의 개발에 대하여 설명한다.

2. 관련 연구

2.1 UDDI 3.0 표준

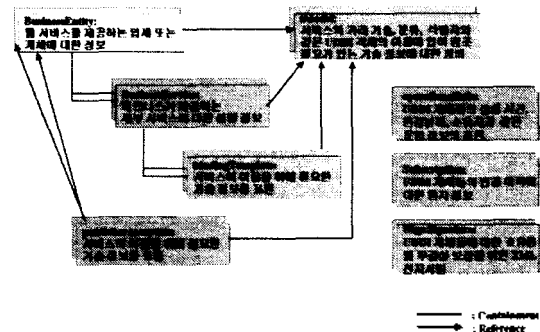
UDDI 관련 표준안[2]은 초기에 IBM, Microsoft, Ariba 등이 주도한 UDDI.org 에서 제정되었으며, 최근에는 관련 작업이 OASIS 의 UDDI Specification TC 에 이관되어 지속적인 표준화 작업이 진행 중에 있다. 현 시점에서 UDDI 2.0 은 OASIS 표준으로 표준화가 완료되었으며, 3.0 은 Committee specification 상태로 표준에 대한 추가적인 보완 작업이 진행 중에 있다. 또한 UDDI TC 에서는 3.0 의 표준화 작업에 맞추어 UDDI 4.0 에 대한 요구사항을 제

출함으로써 4.0 에 대한 개발 작업을 진행 중에 있다. 본 논문에서 제안하는 레지스트리는 현재 3.0 을 지원하도록 개발 중에 있으며 버전 2.0 과의 호환성을 유지하도록 설계되었다.

2.2 UDDI 데이터 구조

UDDI 레지스트리에서 이용되는 데이터 구조는 XML 스키마로 정의된다. XML 은 플랫폼에 중립적인 데이터 구조를 제공하며 계층적 관계를 표현하기에 적합한 형식을 가지고 있다. UDDI 표준에서는 스키마에서 표현된 데이터 구조를 기반으로 UDDI 정보를 기술하고 검증할 수 있도록 할 뿐만 아니라 다양한 데이터 타입을 제공한다는 장점으로 인하여 XML 스키마를 XML 메시지에 대한 스키마 정의 언어로 사용한다.

UDDI 에서의 주요 자료 구조는 비즈니스 정보, 서비스 정보, 바인딩 정보, 서비스를 이용하기 위해 필요한 기술 표준에 대한 정보를 표현한다. 이외에 비즈니스 간의 관계성을 표현하기 위한 추가적인 데이터 구조를 가진다. 각 자료 구조들이 가지는 데이터와 구조 간 계층 관계는 <그림 1>과 같다.



<그림 1> UDDI 3.0 데이터 구조

BusinessEntity 데이터 구조는 특정 서비스를 레지스트리에 등록하고자 하는 비즈니스 개체에 대한 정보를 표현한다. 이 BusinessEntityt 는 비즈니스 개체에 대한 정보를 표현하기 위한 최상위 구조이며, 비즈니스 개체가 제공하는 서비스는 이 BusinessEntity 의 하위에 존재한다. BusinessService 는 BusinessEntity 의 하위 데이터 구조로서 등록되는 서비스에 대한 논리적 정보를 표현하는데 사용된다. 하나의 BusinessEntity 에는 여러 개의 BusinessService 가 존재할 수 있으며, BusinessEntity 와는 별도의 키를 가진다. 하나의 BusinessEntity 내에 등록 가능한 서비스의 개수는 레지스트리 노드의 정책에 따라 제한할 수 있다. BindingTemplate 은 웹 서비스에 대한 기술적 정보를 표현하기 위해 사용되는 데이터 구조로써, 실제 웹 서비스의 실행에 필요한 기술 정보들을 담고 있다. 또한 TModel 데이터 구조는 UDDI 에서 표현되는 모든 객체들에서 사용될 수 있는 메타 데이터의 기술을 위해 사용된다. PublisherAssertion 데이터 구조는 비즈니스 간의 관계성을 표현하기 위해 2.0에서부터 추가된 데이터 구조이다.

버전 3.0에서는 OperationalInfo, Subscription 등과 같은 추가적인 데이터 구조들로 각각의 UDDI 개체들에 대한 생성시간과 같은 운영 정보의 유지와 사용자들의 UDDI 개체들에 대한 변경 이력에 대한 통지를 지원한다. 또한 XML Digital Signature 를 도입하여 각 사용자가 자신이 발행하는 비즈니스, 서비스 정보에 대해 전자 서명을 지원함으로써 사용자 권한과 보안 기능을 강화하였다.

2.3 UDDI API

UDDI API 는 서비스 요청자, 서비스 제공자가 레지스트리를 이용하는데 있어 필요한 API 와 레지스트리 운영 및 레지스트리간의 운영에 필요한 API 들로 크게 6 가지로 나눌 수 있다. 각각은 Inquiry, Publication, Security Policy, Custody and Ownership Transfer, Subscription, Value Set API 존재한다.

Replication, Custody & Ownership API 는 레지스트리 노드 간의 데이터 복제를 위해 사용되는 API 로 특정 노드와 토폴로지를 구성한 노드 또는 레지스트리들 간의 데이터 복제를 위해 사용된다. 이를 통해 UDDI 레지스트리는 어떠한 레지스트리에 접근해서도 동일한 비즈니스, 서비스 정보를 얻을 수 있다. Subscription API 는 사용자가 관심 있어 하는 비즈니스, 서비스 정보에 대한 변경 사항이 발생할 경우 이를 사용자에게 통지하기 위해 사용된다. Value Set API 는 비즈니스, 서비스 정보의 등록 시 같이 입력될 수 있는 분류, 식별 코드의 값이 유효한 값인지를 검증하기 위해 외부 서비스를 이용하는데 사용된다. 예를 들어 국내의 경우 어떠한 비즈니스의 등록 시 사업자등록번호나 주민등록번호를 식별코드로 이용할 경우 이 코드값이 실제 존재하는 사업자나 개인에 해당하는지를 인증하기 위해 사용된다.

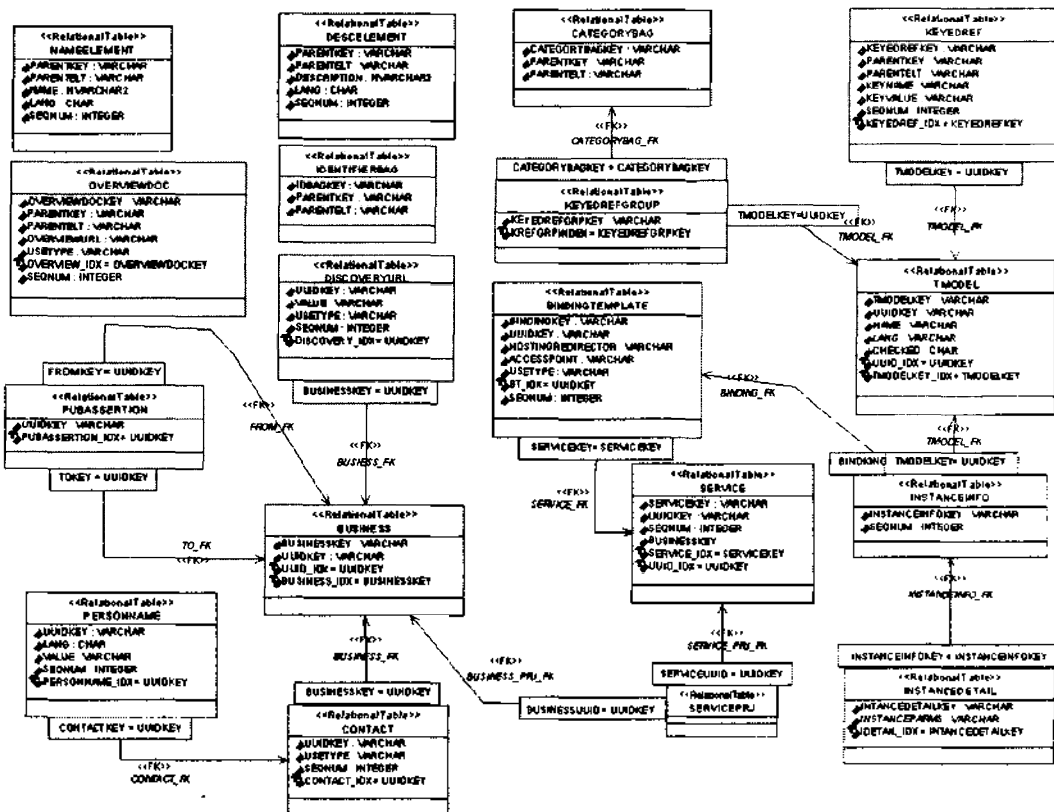
Inquiry 와 Publication API 는 비즈니스, 서비스정보의 검색과 등록을 위해서 이용된다. UDDI 버전 3.0에서는 중첩 질의의 지원과 함께 추가적인 검색 조건들을 제공함으로써 보다 향상된 검색 기능을 제공하고

있으며, 등록에 있어서도 트랜잭션과 전자 서명의 지원을 통하여 UDDI 버전 2.0 에서의 여러 문제들을 해결토록 하였다.

3. UDDI V3.0 레지스트리의 개발

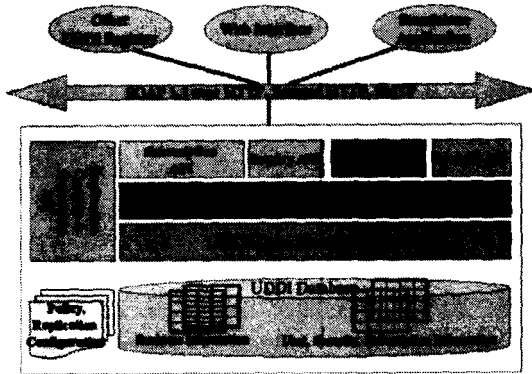
본 논문에서 개발하는 UDDI 3.0 레지스트리는 UDDI 의 각 개체들의 저장을 위하여 관계형 데이터베이스를 이용하였으며, 이를 위하여 관계형 데이터베이스와 XML 스키마 간의 사상을 정의하였다. UDDI 의 XML 스키마 정의에서 계층구조로 정의된 자료 구조들은 서로 다른 별도의 테이블로 정의되며 각 개체들의 키를 통해 참조하도록 설정된다. 각 개체들의 키는 3.0 표준에 따라 URI(Uniform Resource Identifier)키

로 식별이 되며, 2.0 과의 호환성을 위해 내부적으로 UUID(Universally Unique Identifier)키 값을 갖도록 하여 실질적인 참조 키로는 UUID 키 값으로 참조가 된다. 다음의 <그림 2>는 UDDI 3.0 데이터 구조를 UML 클래스 다이어그램으로 작성한 것이다. 이 다이어그램은 서로 다른 기능을 가진 기본적인 데이터 엘리먼트들 사이의 전체적인 관계(relationship)를 표현하고 있다. UDDI 레지스트리는 이러한 데이터베이스 테이블을 기반으로 여러 레지스트리 연산들의 수행을 위한 다양한 API 들을 제공한다. 이러한 API 에 대한 접근은 SOAP 을 통해 이루어지며, 전송되는 메시지는 UDDI XML 스키마 정의에 따른 XML 문서 형식을 취한다.



<그림 2> UDDI 3.0 데이터 구조의 UML 클래스 다이어그램

<그림 3>은 UDDI 3.0 레지스트리의 시스템 구조를 보이고 있다.



<그림 3> UDDI 3.0 레지스트리 구조도

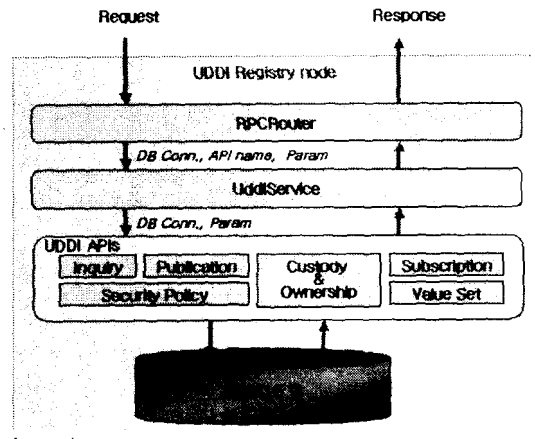
본 논문에서의 UDDI 3.0 레지스트리에서는 XML 형식의 메시지와 관계형 데이터베이스 간의 사상을 위해 UDDI 자료 구조에 대한 바인딩 모듈이 존재하며 이를 통해 API 기능을 구현하였다. 또한 레지스트리 운영에 있어 필요한 정책이나 데이터 복제 관련 설정들은 별도의 XML 파일로써 레지스트리에 유지토록 하며 이를 참조하여 보안과 정책 설정을 관리할 모듈을 개발토록 하였다.

3.1 전체 시스템 구조

레지스트리 서버 시스템의 전체 구조는 <그림 4>와 같다. 서버의 전체적인 동작과정을 살펴보면 UDDI 와 전송 프로토콜인 SOAP 을 사용하여 임의의 클라이언트 또는 다른 레지스트리 노드에서 수행하고자 하는 일련의 작업에 관련된 Request 메시지를 XML 을 사용하여 서버에 요청을 하게 된다.

이때 서버에서 첫 번째로 서블릿인 RPCRouter 가 요청 메시지를 받아 처리하

게 되는데, 이 RPCRouter 의 기능은 SOAP 메시지를 파싱해 SOAP 의 Body 블록 내에 있는 주 메시지를 분리해 내어 파라미터 정보를 저장을 하고, 최상위 엘리먼트인 API 이름을 추출하는 기능을 갖는다.



<그림 4> 레지스트리 서버 시스템 구조

이 RPCRouter 의 기능은 SOAP 메시지를 파싱해 SOAP 의 Body 블록 내에 있는 주 메시지를 분리해 내어 파라미터 정보를 저장을 하고, 최상위 엘리먼트인 API 이름을 추출하는 기능을 갖는다. 이때 각각의 API 에서의 데이터베이스 작업처리를 위해 DB 커넥션을 생성하게 된다. 두 번째 작업으로는 이전 처리 결과인 DB 커넥션, API 이름, 그리고 파라미터 정보들을 UddiService 라는 서블릿에 다시 전달을 해주게 되는데 UddiService 에서는 전달 받은 API 이름으로 해당 서비스요청에 따른 API 를 호출한다. 이때 전달 받은 파라미터 정보와 데이터베이스 커넥션을 호출되는 API 에 인자로 넘겨주게 된다.

레지스트리의 구현 환경으로서 운영체제는 윈도우 2000 서버를 사용하였으며, 자바를 기반으로 개발 킷은 JDK 1.4, 웹 서버로

는 Tomcat 4.04 를 기반으로 구현하였다. 클라이언트와 서버간의 SOAP 메시지 통신에서 요청, 응답 메시지의 효율적인 처리를 위해 기존 UDDI4j 2.0 라이브러리를 확장하여 UDDI4j 3.0 버전을 개발, 이용하였다. 그 밖에 서비스와 레지스트리 운영에 필요한 데이터의 저장과 관리를 위해 관계형 데이터베이스는 오라클 9i 를 사용하였고, 전자서명 처리를 위해 Apache XML-Security 를 사용하여 개발하였다. 본 레지스트리의 구현 환경의 특징으로는 DBMS, 어플리케이션 서버에 종속적이지 않는 시스템을 개발함으로써 다양한 운영 환경을 지원토록 설계하였다.

4. 결론 및 전망

본 논문에서는 e-비즈니스 분야에서 적용 가능한 웹 서비스의 등록, 검색을 위해 요구되는 UDDI 3.0 레지스트리의 개발에 대해 소개하였다. UDDI 3.0 표준에서는 기존 UDDI 2.0 에서 지원하지 못하였던 subscription, XML 전자 서명의 지원 등과 같은 많은 기능이 추가되었으며, 기존 기능들에 대한 많은 보완이 이루어졌다. 뿐만 아니라 기존 UDDI 2.0 과의 호환성을 위해 2.0 API 의 이용 또한 가능하도록 정의하였으며, URI 기반의 키를 이용함으로써 레지스트리 간의 토폴로지 구성 및 복제가 가능하도록 하였다. 본 논문에서는 일반적인 관계형 DBMS 와 JSP 가 지원 가능한 웹 서버를 이용하여 구축이 가능한 UDDI 3.0 레지스트리 서버를 개발하였다.

본 논문에서 개발한 레지스트리 서버는 UDDI 3.0 의 모든 기능을 지원하는 세계

최초의 레지스트리로서 그 활용 가능성과 국내 전자 거래 환경에의 파급 효과가 매우 클 것으로 예측된다.

참고 문헌

- [1] 이경하, 이규철, “웹 서비스의 향후 발전 방향”, 한국정보처리학회지 7 월호, 2002 년
- [2] 한국전산원 “공공부문 전자거래를 위한 UDDI 등의 활용 방안 연구”, 최종연구 보고서 2002 년 11 월
- [3] OASIS UDDI Specification TC, “UDDI Version 2.03 Data Structure Reference”
<http://uddi.org/pubs/DataStructure-V2.03-Published-20020719.htm>, July 2002.
- [4] OASIS UDDI Specification TC, “UDDI Version 3.0 Technical Committee Specification”,
<http://uddi.org/pubs/uddi-v3.00-published-20020719.htm>, July 2002.
- [5] UDDI.org, “UDDI Technical White Paper”,http://www.uddi.org/pubs/Iru_UDDI_Technical_White_Paper.PDF, September 2000.
- [6] W3C.org, “Web Services Glossary” W3C Working Draft 14 November 2002, <http://www.w3.org/TR/ws-gloss/>