

# UDDI 레지스트리에 기반한 웹 서비스 응용에 관한 연구

김태희\*, 강신재\*\*, 김선경\*\*  
\*대구대학교 컴퓨터정보공학과  
\*\*대구대학교 컴퓨터·IT공학부 교수  
e-mail:kth@webmail.daegu.ac.kr

## A Study on Web Service Application based on UDDI Registry

Tae Hee Kim\*, Sin Jae Kang, Sun Kyung Kim\*\*

\*Dept. of Computer and Information Engineering, Daegu University

\*\*School of Computer and Information Technology, Daegu University

### 요 약

최근의 웹 환경은 개별적인 웹 어플리케이션을 통합하는 웹 서비스의 방식을 원하게 되었다. UDDI는 이러한 요구를 수용하여 웹 서비스 시스템에 대한 정보를 한 곳에 분류하여 쉽게 검색할 수 있게 해주므로 특정한 웹 서비스를 찾기 위한 노력을 줄일 수 있다. 이러한 UDDI의 기본적인 검색 기능을 기반으로 웹 서비스 공개자는 다양한 형태로 응용된 정보를 클라이언트에게 제공할 수 있다. 본 논문에서는 UDDI 레지스트리에 정보를 올리고 원하는 형태로 정보를 가져오는 과정을 구현하였다.

### 1. 서론

인터넷상에는 여러 형태의 프로토콜이 사용되고 있다. 이러한 프로토콜들은 연결 지향적이어서 자원의 낭비, 보안의 취약성에 대한 문제가 있고 프로토콜 자체의 복잡성과 서로간의 호환성이 없어 시스템간의 연동이 필수적인 웹 환경에서는 새로운 분산 컴포넌트 기반의 컴퓨팅 기술이 요구된다. 그 결과, 인터넷상의 서로 다른 시스템과 프로토콜을 통합시키기 위한 웹 서비스 기술이 탄생하여 웹 이용자의 요구를 수용할 수 있게 되었다. 웹 서비스는 표준화된 XML 메시지를 통해 네트워크상에서 접근 가능한 연산들을 기술하는 인터페이스로 정의되며[1], 웹상의 이기종간의 컴퓨터 시스템 프로그램을 결합시키는 표준화된 소프트웨어 기술을 이용해 다양한 온라인 사업을 가능케 한다. 웹 서비스 시스템은 분산 환경의 표준 프로그래밍 모델로, 원격 클라이언트에 의해 호출될 수 있는 원격 프로시저의 집합 형태의 서버측 응용 프로그램이라고 할 수 있다.

UDDI(Universal Description, Discovery and Intergration)는 웹 서비스에 대한 메타 데이터를 저장하고 이에 대한 검색 기능을 제공하는 방법을 표

준화한 것이다. 인터넷에 분산되어 있는 웹 서비스를 효율적으로 검색하여 원하는 서비스를 얻을 수 있으며 여러 분야로 응용이 가능하다. UDDI 레지스트리는 웹 서비스에 대한 정보가 한 곳에 잘 분류된 것이다. 특정한 웹 서비스 시스템을 찾기 위하여 무작위로 검색하지 않아도 잘 분류된 정보가 있다면 매우 효율적인 환경이 될 것이다. 이런 역할을 통하여 지역별 업종별로 회사 정보를 분류하여 특정 회사를 쉽게 찾을 수 있도록 도와주는 정보 저장소의 역할을 한다. 각 UDDI 레지스트리 간에 복제 설정이 되어 있으면 복제 관계에 있는 모든 UDDI 레지스트리에 복제되어 저장되므로 웹 서비스 정보는 보편화되어 사용될 수 있다. UDDI는 많은 기업들이 추진하고 있는 프로젝트로 웹 기반의 정보 표현과 검색, 통합에 대한 통일 규약을 정의하고 있다.

본 논문에서는 UDDI 레지스트리를 기반으로 각 정보 제공자가 정보를 게시할 때 XML 데이터를 관계형 DBMS에 저장하는 방법을 연구하고, 제공된 정보를 이용하여 웹 서비스 공개자가 웹 서비스 소비자에게 원하는 형태로 정보를 검색할 수 있게 하는 웹 서비스 시스템을 구현하였다.

2. 관련 연구

2.1 UDDI의 구조

UDDI는 XML 기반의 자료 저장소이다. SOAP 메시지를 사용하여 클라이언트의 환경에 독립적으로 사용할 수 있고 XML 기반의 장치이므로 UDDI 레지스트리간의 데이터 교환이 가능하다.

UDDI API 엘리먼트는 정보의 생성과 변경, 삭제를 위한 XML 엘리먼트를 정의하고 있으며 이들 엘리먼트들은 정보를 생성하고 수정, 삭제하는 Publishing API 엘리먼트와 UDDI 레지스트리에서 비즈니스 정보를 찾기 위해 사용되는 Inquiry API 엘리먼트로 구분된다. UDDI는 데이터베이스 시스템과 기능이 유사하지만 서로 간에는 큰 차이가 있다. 데이터베이스 시스템은 클라이언트가 정보를 저장하고 접근하기 위해서는 JDBC 드라이버를 클라이언트 시스템에 따로 설치해야 한다. 각각의 JDBC 드라이버는 별도의 프로토콜을 사용하기 때문이다. UDDI 레지스트리에 접근하기 위해서는 SOAP 메시지를 사용하므로 프로바이더를 클라이언트 각각에 따로 설치하지 않아도 된다. 비즈니스 정보를 검색하고 게시하기 위해서는 JAXR(Java API for XML Registries) 클라이언트의 공통적인 객체를 생성시켜야 한다. 다음과 같은 순서대로 객체를 생성시킨다.

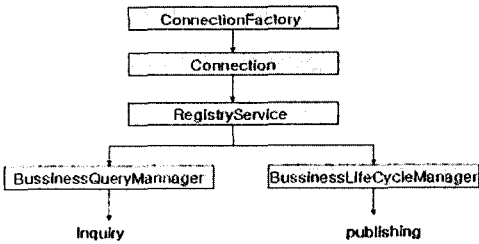


그림 1. 객체 생성 순서

- . ConnectionFactory 객체는 Connection 객체를 만들기 위한 Factory 객체다.
- . Connection 객체는 JAXR 클라이언트와 UDDI 레지스트리의 연결을 하기 위한 객체이다.
- . RegistryService 객체는 JAXR 프로바이더에 의해 구현된 BusinessQueryManager 객체와 Business-lifeCycleManager 객체를 얻을 수 있는 메소드를 제공하는 객체다.
- . BusinessQueryManager 객체는 JAXR 클라이언트가 UDDI 레지스트리에서 비즈니스 정보를 inquiry 하기 위해 사용하는 객체다.

. BussinessLifeCycleManager 객체는 JAXR 클라이언트가 UDDI 레지스트리에서 비즈니스 정보를 publishing 하기 위해 사용된다.

필요한 UDDI 레지스트리 데이터 구조는 등록할 서비스와 비즈니스 정보를 표현하기 위한 구조로 구성되어야 한다. 등록 및 검색을 지원하기 위해 비즈니스 정보를 이루는 엘리먼트는 다음과 같다.[2]

- . businessEntity
- . businessService
- . buindingTemplate
- . tModel

tMModel은 UDDI의 모든 개체들이 참조할 수 있는 메타 데이터의 기술을 위해 사용된다. UDDI 레지스트리는 tModel 정의를 보유하고 사용자가 추가적으로 등록할 수도 있다. 네임 스페이스를 제공하기 위해서 사용되기도 한다.

엘리먼트는 웹 서비스에 대한 메소드의 종류 및 인자의 데이터 타입이 정의된 WSDL 문서의 URL 경로가 기술된 <tModel> 엘리먼트를 참조한다.

2.2 XML 문서의 관계형 스키마로의 매핑

웹상의 문서는 데이터 중심의 표준 언어인 XML 문서로 표현된다. XML 문서는 트리 형태의 그래프로 나타낼 수 있으므로 그래프 구조로 변환되어 처리를 위한 스키마를 추출할 수 있다. 그래프에서 각 엘리먼트는 노드로 표현 되고 두 노드 사이에는 간선이 존재하고 각 간선의 노드는 식별자를 가지고 방향성을 갖는다.[3] 데이터베이스적 측면에서 XML은 내용을 표현하기 보다는 문서의 구조나 정보를 표현하기에 적합한 언어로, XML 데이터를 범용적이고 우수한 성능의 관계형 데이터베이스와 연동하여 이용하기 위해서는 두 구조 사이의 매핑 과정이 필요하다. XML문서의 구조 정보를 관계 데이터베이스의 테이블 형태로 구성하여 저장하는데 있어 XML의 계층적 구조와 데이터베이스의 평면적인 구조의 차이에 대한 처리과정이 필요한 것이다. 엄격한 스키마 구조를 가진 기존의 관계 데이터베이스 시스템을 이용하기 위해 XML의 구조 정보를 관계 데이터베이스의 테이블 형태로 매핑시키는 방법으로 구조 매핑 방법[4]이 있다. 구조 매핑 방법은, 구조 정보를 나타내는 엘리먼트를 분석하여 관계 스키마를 정의하여 XML 문서를 저장한다. 이를 위해 스키마는 DTD에 독립적인형태로 변환하고, 엘리먼트의 추가와 삭제, 검색의 효율성을 위해 노드간의 위

치 정보와 함께 독립적인 ID를 부여한 테이블을 작성한다. XML 문서의 구조 정보를 관계 데이터베이스에 저장하는 방법으로 분할 저장 기법과 가상 분할 저장 기법을 사용할 수 있다.[5] 분할 저장 기법은 XML문서를 엘리먼트의 단위로 나누어 저장하여 검색 시 구조 정보를 해당 엘리먼트나 하위 엘리먼트의 조합을 생성하여 처리한다. 엘리먼트의 추가, 수정, 삭제 시 관계되는 엘리먼트만 수정하여 편집과 관리 면에서 쉽고 노드를 공유할 수 있는 장점이 있지만, 검색 시에 각 엘리먼트의 내용을 조합하여 결과를 구성해야 하므로 검색 시간이 많이 걸리고 DTD 구조 정보가 변하면 관련된 데이터베이스 테이블을 모두 재구성해야 하는 단점이 있다.

가상 분할 기법으로 XML 문서를 관계 데이터베이스에 저장할 때는 각 단말 노드에 대하여 실제 문서에서 가지는 시작 오프셋과 종료 오프셋을 저장하는 방식으로, 전체 문서를 저장하는 분할 방식과는 달리 통합 과정이 필요하지 않아 노드 검색 시 위치 정보를 이용한 구조 검색이 가능하고 검색 효율이 상대적으로 우수하다. 이러한 장점과 함께 엘리먼트의 추가, 삭제시에 다른 엘리먼트들의 위치 정보도 수정해야 하므로 이에 대한 오버 헤드와 데이터베이스의 일관성 유지에 대한 문제가 있다. 따라서 주로 검색을 위주로 하는 시스템에 적합하다.

### 3. 시스템의 구현

구현 환경으로는 윈도우 XP 운영체제에서 JWSPD 기반으로 웹로직 서버를 사용하였고 저장 매체로는 MS-SQL Server를 사용하였다. 구현 언어는 JAVA를 사용하였다. UDDI 레지스트리는 IBM 테스트 레지스트리를 사용하였다.

#### 3.1 시스템의 구성

JWSPD 기반의 JAX-RPC로 구현된 웹 서비스 시스템의 구조는 <그림 3>과 같으며 시스템의 구성 요소들의 역할은 다음과 같다.

##### 1) EJB

정보를 데이터베이스에 저장하고 검색하는 로직을 수행한다. 데이터베이스는 JDBC를 사용한다.

##### 2) 웹 서비스 시스템과 WSDL

클라이언트로부터 검색 요청을 담은 SOAP 메시지를 받아 해당되는 EJB 원격 메소드를 호출하고 실행 결과를 다시 SOAP 메시지로 구성하여 클라이언트로 전송한다.

##### 3) UDDI 레지스트리

웹 서비스의 시스템 정보를 게시하여 클라이언트들의 사용을 돕는다.

##### 4) JSP

실질적인 클라이언트가 각 비즈니스 업체가 제공하는 정보를 얻는 원격 프로시저를 호출한다.

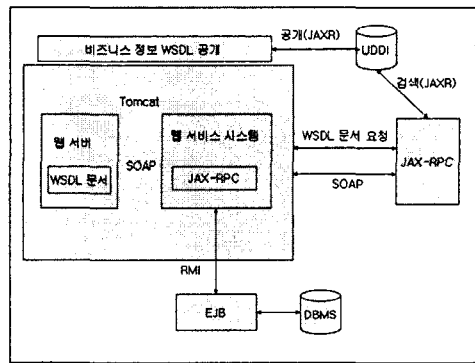


그림 2. 웹 서비스 시스템 구조

#### 3.2 XML 데이터의 처리

관계형 데이터베이스에서 구현하기 위해서는, XML 문서가 미리 정의된 스키마가 없고, 문서 자체에 데이터와 데이터 구조를 갖고 있기 때문에 먼저 XML에 대해 질의 처리를 위한 스키마를 추출하고, DTD에 포함된 엘리먼트와 구조 정보를 얻는다.

```

<!-- ===== Start Entity Declaration ===== -->
<!-- ===== End Entity Declaration ===== -->

<!-- ===== Start Element Declaration ===== -->
<ELEMENT books (#PCDATA | book)*>
<ELEMENT book (#PCDATA | 가격 | 저자 | 출판사 | 도서코드 | 분류 | 제목)*>
<ELEMENT 가격 (#PCDATA)>
<ELEMENT 저자 (#PCDATA)>
<ELEMENT 출판사 (발행정보)>
<ELEMENT 발행정보 (#PCDATA | 초판일자 | 재판일자)*>
<ELEMENT 초판일자 (#PCDATA)>
<ELEMENT 재판일자 ANY>
<ELEMENT 도서코드 (#PCDATA)>
<ELEMENT 분류 (#PCDATA)>
<ELEMENT 제목 (#PCDATA)>
<!-- ===== End Element Declaration ===== -->
    
```

그림 3. 엘리먼트의 구조

XML 스키마는 DTD에 독립적인 형태를 사용하고, 구조적 매핑을 위해 구조 정보를 나타내는 엘리먼트를 분석하여 관계 스키마를 정의한다.

관계형 데이터베이스의 테이블을 생성하기 위해서는 주어진 DTD등을 분석하여 경로 정보 테이블을 구성한다. XML 문서를 왼쪽 자식, 오른쪽 형제의 구조를 갖는 이진 트리로 표현하고 각 엘리먼트에 임의의 고유한 식별자인 ID를 부여하고 각 엘리먼트에

대해 엔트리로 매핑 한다.

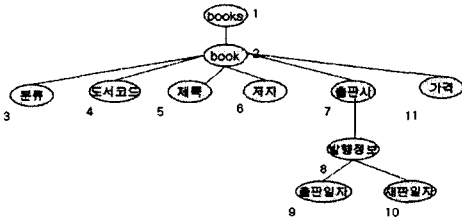


그림 5. XML 데이터 그래프

사용된 테이블은 XML 문서를 실제적으로 저장하기 위한 테이블, XML문서의 세부 정보를 나타내기 위한 테이블로 순서를 나타내는 pid, 데이터 타입을 나타내는 type, 그리고 엘리먼트와 문서 내용을 가지고 있는 content 컬럼으로 구성된다.

또, 엘리먼트 정보를 담은 테이블과, 전체 문서에 대한 일반적인 내용을 나타내는 테이블이 필요한데 경로 정보를 나타내는 path와 각 path를 구분하기 위한 pathid, 시작 엘리먼트를 나타내는 sid, 종료 엘리먼트를 나타내는 eid와 부모 엘리먼트를 나타내는 parid로 구성된다.

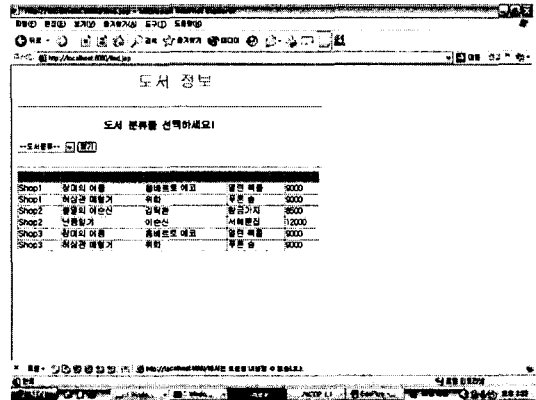
테이블을 생성한 다음과 같은 순서로 개발한다.

- 1) 원격 프로시저를 정의하는 원격 인터페이스 작성
- 2) 원격 인터페이스를 구현한 클래스 작성
- 3) 배치 기술자DD 작성

서블릿 컨테이너에 담아야 할 내용인 배치 기술자 web.xml을 작성하여야 한다. 기술되는 내용은 세션에 대한 정보와 URL 매핑에 관한 정보, 초기화에 대한 정보, 디폴트 문서에 대한 정보가 올 수 있으며 적용시킬 내용이 없더라도 반드시 존재하여야 한다. 또한 타이 클래스 및 WSDL 문서를 자동으로 생성시키기 위한 RML interface 배치 기술자인 jaxrpc-ri.xml 문서를 작성하여야 한다.

- 4) 타이 클래스 및 WSDL 문서 생성
- 5) WAR 및 배치

패키징 및 배치를 위해 타이 클래스와 WSDL 문서를 자동으로 생성시키고 WAR 파일에 추가시킨다. 웹 서비스 시스템의 최종 산물인 WAR 파일을 만들어 서블릿 컨테이너에 배치하여 클라이언트의 원격 프로시저 호출을 수행할 수 있게 한다.



#### 4. 결론

본 논문에서는 JAVA로 클라이언트측 응용 프로그램을 이용하여 UDDI 레지스트리에 정보를 게시하고 검색하는 과정을 보였다. 다양한 형태로 정보를 게시할 수 있고 웹상의 클라이언트가 원하는 형태로 정보의 응용이 가능하다. UDDI 3.0은 웹 서비스의 기술의 중요한 요소이나 표준의 복잡성으로 제반 기술들에 대한 응용이 개발 단계에 머물러 있다. 단순한 게시와 검색을 넘어 UDDI 레지스트리를 응용한 웹 서비스의 가능성을 알아보았다. 향후, 데이터베이스 연결 시 다양한 형태의 데이터 처리와 검색속도를 향상시키기 위한 연구가 과제로 남는다.

#### 참고문헌

- [1] Heather Kreger, "Web Services Conceptual Architecture(WSCA 1.0)", IBM Software Group, May 2001.
- [2] UDDI version 3.0 Published Specification, <http://uddi.org/pubs/uddi-v3.00-published-20020719>
- [3] Roy Goldman, Jennifer Widom, "Data Guide : Enabling Query Formulation and Optimization in Semistructured Databases", In Proceeding of VLDB, 1997.
- [4] P. Florescu and D. Kossman, "Storing and Querying XML Data Using an RDBMS", IEEE Data Engineering Bulletin 22(3), pp.27-34, 1999.
- [5] S. Malaika, "Using XML in Relational Database", 15th Conf. on Data Engineering, Australia, p.167, 1999.