

그리드 서비스를 위한 UDDI Repository 시스템의 저장 및 검색 기능의 설계 및 구현

송인혁[†] 이정훈[†] 임효상[†] 황규영[†] 박형우^{*}

[†]한국과학기술원 전산학과/첨단정보기술연구센터, ^{*}한국과학기술정보연구원

{mailto:iksong.handol, lorien, kywhang}@mozart.kaist.ac.kr, mailto:hwpark@kisti.re.kr

Design and Implementation of Saving and Searching Functions of a UDDI Repository System Based on UDDI for Grid Services

In-Hyuk Song[†], Jeong-Hoon Lee[†], Hyo-Sang Lim[†], Kyu-Young Whang[†], Hyoung-Woo Park^{*}

[†]Department of Electronic Engineering & Computer Science

Division of Computer Science and

Advanced Information Technology Research Center(AITrc)

Korea Advanced Institute of Science and Technology(KAIST),

^{*}Korea Institute of Science and Technology Information(KISTI)

요 약

UDDI는 인터넷에 분산되어 있는 웹서비스에 대한 메타 정보를 저장하고 이에 대한 다양한 검색 기능을 제공함으로써 서비스 요청자가 원하는 서비스를 손쉽게 발견할 수 있도록 하는 Repository 시스템이다[1]. 본 논문에서는 Repository 시스템의 데이터 구조 및 아키텍처 설계에 관한 선행 연구를 바탕으로 UDDI 3.0 표준 명세를 따르는 Repository 시스템의 저장 및 검색 기능을 설계하고 구현한다. 본 논문은 이를 위해 XML 형태의 UDDI 데이터를 데이터베이스에 효율적으로 저장하기 위한 저장 구조 설계를 설명하고 이를 통한 효율적인 UDDI 저장 기능의 설계와 구현에 관하여 설명한다. 또한 표준에서 명시하는 다양한 검색 방법을 분석하고 저장된 웹서비스 정보를 효율적으로 발견할 수 있도록 하는 검색 기능의 설계와 구현에 관해서 설명한다.

1. 서 론

웹서비스는 분산된 컴퓨팅 자원과 정보를 효과적으로 활용하려는 노력의 일환으로 W3C 에서 제정한 분산환경의 '표준 프로그래밍 모델'이다[2]. UDDI는 웹서비스에 대한 메타 데이터를 저장하고 이에 대한 검색 기능을 제공하는 Repository 시스템이다[1][8]. 웹서비스가 최근 분산 응용프로그램의 개발 및 서비스 제공 표준으로 널리 자리잡아 감에 따라 인터넷에 분산되어 있는 웹서비스를 효율적으로 검색하여 원하는 서비스를 발견할 수 있도록 하는 UDDI의 필요성이 높아지고 있다[3].

본 연구실에서는 이미 선행 연구를 통해 UDDI 3.0 표준 명세를 분석하여 Repository 시스템에서 관리해야 할 데이터 구조를 분석하고 이를 DBMS 에 저장하기 위한 데이터베이스 스키마를 설계하였다. 그리고 UDDI 에서 규정하고 있는 API 들을 기능별로 분류하고, 각각의 API 에서 데이터베이스에 접근하기 위해 사용하는 SQL 질의문을 설계하였다. 또한 SOAP 을 통하여 표준화된 방법으로 API 에 접근할 수 있고 범용 DBMS 를 저장시스템으로 활용할 수 있는 Repository 시스템의 아키텍처를 설계하였다[1].

본 논문은 이러한 선행 연구를 바탕으로 W3C 에서 제정한 UDDI 3.0 명세를 따르는 UDDI Repository 시스템의 저장 및 검색 기능을 설계하고 구현하는 것을 연구목표로 한다. 본 논문에서 설계하고 구현한 Repository 시스템은 웹서비스에 관한 정보를 효과적으로 데이터베이스에 저장하고 다양한 검색 기능을 제공함으로써 서비스 요청자가 원하는 서비스를 손쉽게 발견할 수 있도록 한다. 또한 W3C 에서 제정한 UDDI 3.0 표준 명세를 따르고 있어 웹서비스 표준에 따르는 모든 UDDI 환경에서 사용이 가능하며 웹서비스 표준에 따라 개발된 어떤 시스템과도 호환이 가능하다. 또한 시스템을 플랫폼 독립적인 Java 언어로 구현하고 데이터베이스와의 통신에 JDBC 표준

인터페이스를 사용함으로써 범용성 및 유연성을 제공한다.

본 논문의 구성은 다음과 같다. 제 2 장에서는 본 논문에서 설계하고 구현하는 Repository 시스템의 아키텍처를 소개한다. 제 3 장에서는 이러한 아키텍처를 바탕으로 Repository 시스템의 저장 구조에 대하여 분석하고 이러한 구조를 바탕으로 저장 기능의 설계 및 구현에 관해서 설명한다. 제 4 장에서는 UDDI 가 제공하도록 명시하고 있는 다양한 검색 기능들에 관해 설명하고 검색 기능의 설계 및 구현에 관하여 설명한다. 마지막으로 결론에서는 향후 연구를 기술하고 결론을 내린다.

2. 시스템 아키텍처

본 논문에서 설계하고 구현하는 Repository 시스템의 아키텍처는 그림 1 과 같다. UDDI Repository 시스템은 UDDI 데이터를 저장하고 관리하는 DBMS 와 UDDI API 기능을 수행하는 응용프로그램 로직, 그리고 UDDI API 기능을 웹서비스 형태로 제공하기 위한 SOAP 기반 응용프로그램 디스패처 이렇게 세 부분으로 구성된다.

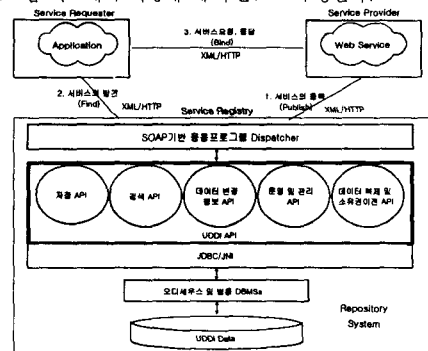


그림 1. Repository 시스템의 아키텍처

* 본 연구는 첨단정보기술연구센터를 통하여 한국과학기술원의 지원을 일부 받았음.
* 본 연구는 한국과학기술정보연구원 '그리드 서비스를 위한 Repository 시스템 개발' 과제에서 일부 지원 받았음.

DBMS 부분은 XML 형태의 UDDI 데이터를 선행 연구에서 설계한 데이터베이스 스키마에 따라 데이터베이스 테이블에 분할된 형태로 저장하고 DBMS 에서 제공하는 질의 기능을 통하여 검색한다[1]. 본 논문에서는 데이터베이스에 접근하기 위하여 표준 데이터베이스 인터페이스인 JDBC(Java Database Connectivity)[9]를 사용하여 표준 SQL 질의문을 수행함으로써 특정 DBMS 에 종속되지 않는 범용성을 제공한다. 또한 저장된 정보에 대한 보다 빠른 검색을 제공하기 위해 본 연구실에서 개발한 정보검색용 객체관계형 DBMS 인 오디세우스[4]의 정보검색 (Information Retrieval)[5] 기능이나 오라클의 컨택스트 업선과 같은 텍스트 인덱스 기능[6]을 사용할 수 있도록 설계하였다.

UDDI API 기능들은 응용프로그램 로직들의 집합으로 이루어진다. 각각의 응용프로그램 로직들은 사용자로부터의 수행 요청을 받아 JDBC 를 사용하여 DBMS 에 접근함으로써 해당 API 의 기능을 수행하고 결과를 돌려주는 역할을 한다. 본 논문에서 설계한 UDDI Repository 시스템에서는 UDDI API 기능을 수행하는 응용프로그램 로직을 Java 로 작성하여 수행 플랫폼에 독립적으로 동작할 수 있도록 하였다.

UDDI Repository 시스템은 사용자에게 웹서비스의 형태로 API 기능을 제공한다. 따라서 HTTP 를 통한 SOAP 기반의 인터페이스를 통해 API 수행 요청을 받고 결과를 돌려주는 구조를 가지고 있다[10]. SOAP 기반 응용프로그램 디스패처는 이러한 SOAP 형태의 UDDI API 수행 요청을 받아 해당 응용프로그램 로직을 수행하고 결과를 다시 SOAP 형태의 XML 문서로 돌려주는 역할을 수행한다. 본 논문에서 설계한 UDDI Repository 시스템은 SOAP 기반 응용프로그램 디스패처의 종류에 상관없이 웹서비스 형태의 UDDI API 수행요청을 처리할 수 있는 범용성을 가지도록 설계하였다.

3. 저장 기능의 설계 및 구현

본 장에서는 XML 형태의 UDDI 데이터를 효율적으로 저장하기 위한 저장구조 설계를 설명하고 본 논문에서 설계하고 구현한 UDDI 저장 기능에 관해 설명한다.

3.1. 저장 구조의 설계

UDDI 데이터는 XML 형태의 트리 구조로서 표현되지만 관계형 데이터베이스는 데이터를 풀랫한 테이블 구조로 표현되기 때문에 UDDI 데이터는 직접적으로 관계형 데이터베이스에 저장될 수 없다. 따라서 선 행 연구를 통해 이러한 XML 형태의 UDDI 데이터를 범용 관계형 데이터베이스에 효과적으로 저장하고 효율적으로 검색할 수 있도록 UDDI 데이터를 계층 구조에 따라서 관계형 테이블로 분할하여 저장하도록 스키마를 설계하였다[7].

XML 데이터를 관계형 데이터베이스 테이블에 분할하여 저장하게 되면 XML 데이터의 계층 정보가 손실되므로, 각각의 테이블에는 integer 타입의 고유 키를 저장하는 주 키 컬럼을 생성하고, 하위 계층 관계의 테이블에는 이러한 키를 참조하는 외래 키 컬럼을 생성함으로써 XML 형태의 데이터를 관계형 테이블에 분할할 때 발생하는 계층 정보 손실을 보호하도록 하였다. 이러한 방법으로 본 논문에서는 UDDI 데이터를 총 62 개의 관계형 테이블에 저장하도록 스키마를 설계하였다.

3.2 저장 기능의 설계 및 구현

UDDI 저장 기능 API 는 UDDI 주요 데이터의 등록, 수정 및 삭제 기능을 제공하는 API 그룹으로 총 14 개의 API 로 이루어져 있다. 저장 기능 API 는 API 이름에 따라 크게 UDDI 데이터의 저장과 삭제를 수행하는 API 와 서비스 제공자와 관련된 정보들을 관리하는 API 로 구분된다. 이 중 Save 형태의 API 는 UDDI 데이터의 신규 저장 및 이미 저장된 데이터의 수정에 사용된다. Delete 형태의 API 는 UDDI 데이터를 삭제하는데 사용된다. Add 형태의 API 는 서비스 제공자간의 연관성을 표현하는데 사용되며, Get 형태의 API 는 서비스 제공자가 등록한 주요 데이터의 요약 정보를 얻기 위해 사용된다. 본 논문에서는 그림 2 와 같은 구조를 통해 UDDI 에서 명시하고 있는 이러한 저장 기능들을 제공하도록 시스템을 설계하고 구현하였다.

본 논문에서는 먼저 사용자로부터 XML 형태로 들어온 데이터를 분석한다. 이는 SOAP 기반 응용 프로그램 디스패처를 통해 해당 XML 데이터를 UDDI 에서 명시하고 있는 표준 데이터 타입으로 변환하도록

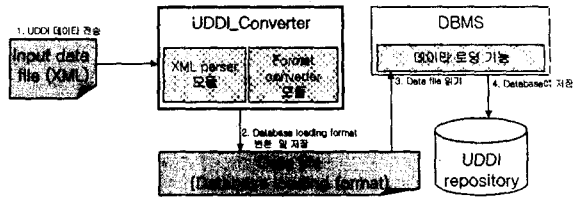


그림 2. 구현된 저장기능의 구조

하거나 XML 문자열 형태로 받아들여 UDDI Converter 에게 전달한다.

UDDI Converter 는 UDDI 데이터에 대한 저장 기능을 수행하는 모듈이다. UDDI Converter 는 XML 형식의 UDDI 데이터를 읽어 들여 저장하고자 하는 데이터가 UDDI 에서 명시하고 있는 규칙을 준수하고 있는지, 분류나 식별 정보 등이 올바른 값인지를 검사한다. 그런 다음 관계형 데이터베이스의 테이블 구조에 적합하도록 분할한 뒤, 분할된 정보를 데이터베이스가 읽어 들여 저장할 수 있는 형식인 베타 로딩 포맷으로 변형하여 파일로 저장하는 기능을 수행한다. 파일로 저장된 데이터들은 표준 SQL 로 변환하여 범용 관계형 DBMS 에 손쉽게 저장할 수 있는 구조를 취하고 있을 뿐만 아니라 시스템의 고장이나 데이터베이스의 파손, 새로운 시스템의 설치 및 구동시 일괄적으로 데이터들을 시스템에 읽어 들여 안정성 있고 효율적인 시스템 운용을 가능하도록 한다.

UDDI Converter 는 이러한 기능을 수행하기 위해 XML Parser 모듈과 Format converter 모듈로 구성된다. 먼저 XML parser 모듈은 입력된 XML 형식의 UDDI 데이터의 구조를 분석하고 이를 Format converter 모듈에게 전달하는 역할을 수행한다. Format converter 모듈은 XML parser 모듈로부터 수신한 UDDI 데이터 구조 정보를 데이터베이스가 제공하는 데이터 로딩 기능을 사용하여 데이터베이스 테이블에 저장할 수 있도록 데이터 로딩 포맷으로 변형하여 파일로 저장한다.

4. 검색 기능의 설계 및 구현

UDDI 3.0 명세에서는 사용자가 원하는 서비스를 손쉽게 발견할 수 있도록 하기 위해 여러 가지 형태의 검색 방법을 제공하도록 명시하고 있으며 총 10 개의 API 를 제공한다. 본 장에서는 UDDI 에서 제공하도록 명시하고 있는 검색 방법들을 분석하고 본 논문에서 설계하고 구현한 검색 기능에 관해서 설명한다.

4.1. 문자열 정보를 검색 조건으로 하는 검색 기능

UDDI 명세에서는 Repository 에 저장하고 있는 UDDI 주요 데이터 중 특정 이름이나 설명 정보를 가지는 데이터가 존재하는지 검색하고 그 데이터의 개략적인 정보를 검색하는 형태의 검색 방법 제공하도록 명시하고 있다. 이러한 검색기능을 제공하는 API 를 Find 계열의 API 라 하며, Find 계열의 API 들은 다양한 문자열 검색 방법을 통해 저장된 데이터에 대한 검색 기능을 제공하고 있다. Find 형태의 API 는 검색어와 검색 방법, 그리고 분류 및 식별 정보를 입력 받아 찾고자 하는 UDDI 데이터들의 요약정보를 돌려주는 기능을 제공한다. 이 API 를 통해 사용자는 관심의 대상이 되는 데이터가 존재하는지 알 수 있으며 요약정보에 포함된 데이터의 키 값을 통해 자세한 데이터 정보를 열람할 수 있다.

UDDI 명세에서는 사용자가 손쉽게 원하는 서비스를 발견할 수 있도록 하기 위해 다양한 검색방법(FindQualifier)을 지원하도록 명시하고 있는데 대표적인 검색방법으로는 검색 조건으로 주어진 문자열과 동일한 문자열 가지는 주요 데이터를 찾는 Exact match 방법, 검색 조건으로 주어진 문자열의 일부와 동일한 문자열 정보를 저장하고 있는 주요 데이터를 찾는 Approximate match(혹은 Partial match)방법, 검색 조건으로 주어진 문자열의 대소문자를 구분하지 않고 검색하는 방법인 Case-insensitive match 방법, 검색 조건으로 주어진 문자열의 대소문자를 구별하여 검색하는 방법인 Case-sensitive match 방법 등이 있다. 본 논문에서 설계하고 구현한 시스템은 UDDI 에서 명시하고 있는 이러한 검색 방법들을 제공함으로써 사용자가 원하는 서비스를 손쉽게 발견할 수 있도록 하였다.

4.2. Key 정보를 검색 조건으로 하는 검색 기능

UDDI 주요 데이터의 구성요소는 상호간의 식별을 위해 유일한 key 정보를 가진다. 그리고 Repository 시스템은 이러한 key 정보를 검색 조건으로 활용하여 저장된 데이터를 검색할 수 있도록 응용프로그램 개발자나 사용자에게 key 정보를 이용한 검색 기능을 제공한다. 검색 기능 중 이러한 기능을 담당하는 API 들은 Get 계열의 API 들로 이들은 Find API 의 검색 결과로 받은 요약 정보로부터 얻은 키 값을 입력 받아 계층적인 데이터 구조를 가지는 UDDI 주요 데이터의 상세 정보를 순차적인 drill-down 형태로 열람하는데 사용된다. 본 논문에서는 서비스 요청자로부터 넘어온 키를 받아 여러 개의 테이블에 분할되어 있는 데이터를 수집하여 사용자에게 상세 정보를 반환하도록 검색 기능을 설계하고 구현하였다.

4.3. 분류 정보를 검색 조건으로 하는 검색 기능

UDDI 주요 데이터는 자신의 정보를 분류하고 식별할 수 있도록 분류 정보 및 식별 정보를 가질 수 있고 UDDI 3.0 명세에서는 이러한 정보를 이용하여 검색을 수행하는 기능을 사용자 및 응용프로그램 개발자에게 제공하도록 기술하고 있다. UDDI 주요 데이터는 자신을 분류하거나 식별하기 위해 CategoryBag 이나 IdentifierBag 을 구성요소로 포함하고 있으며, 검색 기능 API 중 Find 계열의 API 를 사용하여 저장된 데이터에 대한 간략한 정보를 얻고자 할 때 검색 조건으로 분류 정보 및 식별 정보를 활용할 수 있다.

4.4. 검색 기능의 설계 및 구현

본 논문에서는 앞에서 언급한 문자열 정보를 검색조건으로 이용한 주요 데이터 검색 기능과 key 정보를 통한 데이터 검색 기능, 그리고 분류 정보를 통한 데이터 검색 기능을 모두 제공하도록 시스템을 설계하고 구현하였다. 본 논문에서는 그림 3 과 같은 형태로 검색 기능을 설계하고 구현하였다.

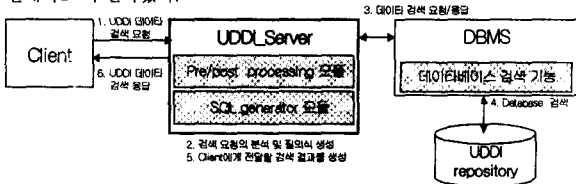


그림 3. 구현된 검색 기능의 구조

UDDI Server 는 웹이나 웹서비스 API 를 호출을 통해 Client 로부터 저장된 데이터에 대한 검색 요청을 받고 이를 분석한다. 이를 통해 검색 조건이나 검색 방법에 대한 요구 조건을 분석하고 검색 질의식 생성을 위해 필요한 정보를 추출한다. UDDI Server 는 이를 이용하여 검색을 위한 질의식을 생성하고 이를 DBMS 에 전달하여 검색 요청을 처리하게 된다. DBMS 는 UDDI Server 로부터 검색 질의문을 수신하여 저장된 데이터 중 검색 조건에 부합하는 서비스 데이터를 결과로 UDDI Server 에게 전송하게 되며 최종적으로 Client 에게 검색 결과를 HTML 이나 XML 형식의 문서로 응답하게 된다. 이러한 검색 절차를 효과적으로 수행하기 위해 UDDI Server 는 2개의 세부 모듈로 구성된다.

- Pre/Post Processing 모듈

Client 로부터 수신한 그리드 서비스 검색 요청 정보를 분석하여 검색에 필요한 정보를 추출한다. 그리고 추출한 검색 정보를 SQL generator 모듈에게 전달한다. 후에 검색 결과가 SQL generator 모듈을 통해 DBMS 로부터 전송되면 이를 Client 가 수신할 수 있는 형식의 데이터로 변형하여 Client 에게 전송하는 기능을 수행한다.

- SQL Generator 모듈

SQL generator 모듈은 Pre/post processing 모듈로부터 입력된 검색 조건을 이용하여 DBMS 에게 전송할 검색 질의문을 생성하고 이를 DBMS 에 전송하여 검색을 수행한다.

6. 결론

본 논문에서는 선행 연구를 통해 설계한 Repository 시스템의 아키텍처와 데이터베이스 스키마 설계를 바탕으로 UDDI Repository 시스템의 저장 및 검색 기능을 설계하고 구현하였다. 이를 위해 먼저 XML 형태의 UDDI 데이터를 데이터베이스에 효율적으로 저장할 수 있도록 저장구조를 설계하였으며 표준에서 명시하는 총 14 개의 저장 기능 API 를 효과적으로 설계하고 구현하였다. 저장 기능을 통해 저장되는 UDDI 데이터들은 표준 SQL 로 변환하여 범용 관계형 DBMS 에 손쉽게 저장할 수 있는 구조를 취하고 있어 여러가지 긴급 상황이나 새로운 UDDI 시스템의 구축시 안정성 있고 효율적인 시스템 운용을 가능하도록 한다. 본 논문에서는 또한 UDDI 표준에서 명시하고 있는 다양한 검색 기능들을 분석하고 이를 효과적으로 제공할 수 있도록 표준에서 명시하고 있는 총 10 개의 검색 기능을 설계하고 구현하여 서비스 요청자가 원하는 서비스를 효과적으로 발견할 수 있도록 하였다.

본 논문에서 설계하고 구현한 시스템의 특징은 첫째, W3C 에서 제안하는 UDDI 3.0 표준 명세를 따르고 있어 CGSA 를 비롯한 웹서비스 표준을 따르는 모든 환경에서 사용이 가능하며 웹서비스 표준에 따라 개발된 어떤 시스템과도 호환이 가능하다. 둘째, 시스템 설계시 범용 데이터베이스와의 연동을 고려하여 데이터베이스 표준 프로토콜인 JDBC 를 사용하였다. 또한 시스템이 플랫폼에 독립적으로 사용 가능하도록 하기 위해 Java 언어를 사용하여 시스템을 설계하고 구현하여 특정 DBMS 나 플랫폼에 독립적으로 수행이 가능하므로 시스템 활용도가 높을 것으로 예상된다. 또한 SOAP 형태의 UDDI API 수행 요청을 처리하는 SOAP 기반 응용프로그램 디스패처를 자유롭게 선택하여 이용할 수 있는 구조를 취하도록 함으로써 디스패처의 종류에 상관없이 웹서비스 형태의 UDDI API 수행요청을 처리할 수 있는 범용성을 가지도록 하였다.

향후에는 Repository 시스템에 저장된 데이터를 보다 효율적으로 검색할 수 있도록 하기 위해 정보검색(Information Retrieval) 기능을 사용하는 방법에 대해 연구하고자 한다. 또한 그리드 서비스 정보에 확장된 저장 기능과 저장된 그리드 서비스 정보에 대한 확장된 검색 기능의 연구를 수행하고자 한다. 이밖에도 Repository 시스템이 제공해야 할 기능인 데이터 변경 통보 기능, 데이터 분류 정보 관리 기능, 데이터 보안 기능, 데이터 중복(Replication) 기능, 데이터 소유권 이전 기능에 관해서도 설계하고 구현하고자 한다.

참고 문헌

- [1] 송인혁, 이정훈, 임효상, 황규영, 박형우, "Grid 서비스를 위한 UDDI 기반의 Repository 시스템에 대한 연구 및 설계", 한국정보과학회 학술발표 논문집(II), pp.76-78, 2003년 10월.
- [2] Heather Kreger, "Web Service Conceptual Architecture(WSCA 1.0)", IBM Software Group, May 2001(available from http://www-3.ibm.com/software/solutions/webservices/pdf/WSCA.pdf).
- [3] 정부연, 웹서비스의 현황 및 비즈니스 모델의 변화, 정보통신정책 연구원 정보통신정책 제 14 권 15 호, 2002년 8월.
- [4] 한옥산, 이민재, 이재길, 박상영, 황규영 "오디세우스 객체관계형 멀티미디어 DBMS 의 아키텍처", 한국정보과학회 추계학술발표회 논문집, pp. 45-47, 2000년 10월.
- [5] 박병권, 정보 검색과 데이터베이스 관리 시스템의 밀접함을 위한 역색인 구조와 질의 최적화, 박사 학위 논문, KAIST 전산학과, 1998.
- [6] Oracle Technology Network, "Oracle Text," Oracle, Jan. 2004 (available from http://otn.oracle.com/products/text)
- [7] 이기훈, 오디세우스 객체관계형 DBMS 를 위한 XML 저장 시스템의 설계 및 구현, 석사학위논문, KAIST 전산학과, 2002.
- [8] Tom Bellwood, et al., "UDDI Version 3.0 Published Specification," OASIS, July 19, 2002.
- [9] Seth White, et al., "JDBC API Tutorial and Reference, Second Edition," Addison Wesley, Jun 1999.
- [10] Box, D. et al., "Simple Object Access Protocol(SOAP) V1.1", W3C, May 2000(available from http://www.w3.org/TR/2000/NOTE-SOAP-20000508).