

Grid 서비스를 위한 UDDI 기반의 Repository 시스템에 대한 연구 및 설계

송인혁[†] 이정훈[†] 임효상[†] 황규영[†] 박형우^{*}

[†]한국과학기술원 전산학과/첨단정보기술연구센터, ^{*}한국과학기술정보연구원
 (mailto:iksong, handol, lorien, kywhang}@mozart.kaist.ac.kr, mailto:hwpark@kisti.re.kr

A Study and Design of a 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

컴퓨터 기술의 발전과 인터넷의 보급에 따라 분산된 컴퓨팅 자원과 정보를 효과적으로 활용하려는 다양한 연구가 진행되고 있다. 이러한 노력의 일환으로 최근 W3C 에서는 분산환경의 표준 프로그래밍 모델로 웹서비스라는 개념을 제시하였다. UDDI 는 이러한 환경에서 웹서비스에 대한 메타데이터를 저장하고 이를 검색할 수 있도록 기능을 제공하는 Repository 시스템이다. 웹서비스를 이용하고자 하는 응용프로그램은 UDDI 가 저장한 정보를 검색하여 원하는 웹서비스를 발견하고 이를 활용할 수 있다. 이와 같이 UDDI 는 웹서비스 기반의 분산 응용에서 필요한 디렉토리 서비스기능을 제공하여 분산된 웹서비스의 통합을 지원하는 기반 시스템으로 널리 활용될 것으로 기대된다. 본 논문에서는 W3C 에서 제정한 UDDI 3.0 Specification 을 분석하여 Repository 시스템에서 관리해야 할 데이터 구조를 설계하고, 제공하는 API 들을 기능별로 분류한다. 그리고 이러한 분석을 토대로 Repository 시스템의 아키텍처를 설계한다.

1. 서론

컴퓨터 기술의 발전과 인터넷의 보급에 따라 분산된 컴퓨팅 자원과 정보를 효과적으로 활용하기 위하여 네트워크 상에 존재하는 여러 대의 컴퓨터를 사용하여 서비스를 제공하는 분산 시스템 환경이 널리 쓰이고 있다. 이에 따라 응용프로그램도 분산 시스템을 활용하는 분산 응용프로그램 형태로 발전하고 있다[1]. 웹서비스(Web Service)는 이러한 분산환경에서 응용 프로그램 개발을 지원하기 위해 W3C 에서 제정한 분산 환경에 대한 표준 프로그래밍 모델이다[2].

본 논문은 웹서비스 기반의 분산응용에서 웹서비스 정보에 대한 저장 및 검색 기능을 제공하는 시스템인 UDDI(Universal Description, Discovery and Integration)의 구조와 기능을 분석하고 이를 기반으로 UDDI 를 따르는 Repository 시스템을 설계하고자 한다. 본 연구를 통해 설계된 Repository 시스템은 웹서비스 정보에 대한 효과적인 저장과 다양한 검색 기능을 제공하여, 분산된 웹서비스들을 통합하여 응용프로그램을 작성하고 수행할 수 있도록 한다.

본 논문에서는 W3C 에서 제정한 UDDI 3.0 명세(Specification)를 연구하여 Repository 시스템에서 관리해야 할 데이터의 구조를 분석하고, 이러한 데이터를 DBMS 에 저장할 수 있도록 데이터베이스 스키마를 설계한다. 그리고 웹서비스 정보의 저장 및 저장된 정보에 대한 다양한 검색을 제공하도록 하기 위하여, 설계된 데이터베이스 스키마에 대하여 효율적인 수행이 가능한 SQL 질의문을 설계한다. 또한 UDDI 3.0 명세에서 규정하고 있는 API 들을 바탕으로 Repository 시스템이 제공할 기능들을 분석하고 시스템의 아키텍처를 설계한다.

본 논문의 구성은 다음과 같다. 제 2 장에서는 본 논문의 연구 배경인 웹서비스의 구조와 웹서비스에서 UDDI 의 역할에 대해 설명한다. 제 3 장에서는 UDDI 3.0 명세를 분석하여 UDDI 에서 관리해야 할 데이터의 구조를 분석하고, 기능별로 API 들을 분류한다. 제 4 장에서는 제 3 장 분석결과를 토대로 Repository 시스템의 데이터베이스 스키마를

설계하고, 이에 대한 검색 및 저장을 수행하기 위해 필요한 SQL 질의식 설계에 대해 설명한다. 제 5 장에서는 Repository 시스템의 목표 아키텍처를 제안하고 마지막으로 제 6 장에서는 향후 연구를 기술하고 결론을 내린다.

2. 웹서비스 소개

본 장에서는 웹서비스의 구조와 웹서비스에서 UDDI 의 역할에 대해 설명한다. 먼저 제 2.1 절에서는 웹서비스 구조에 대해 설명하고 제 2.2 절에서는 웹서비스에서 UDDI 의 역할에 대해 설명한다.

2.1. 웹서비스 구조

웹서비스(Web Service)는 HTTP 와 XML 기반으로 W3C 에서 제정한 분산 응용 프로그램 개발 표준 모델이다[2]. 그림 1 은 웹서비스의 구조를 개념적으로 표현한 것이다[1].

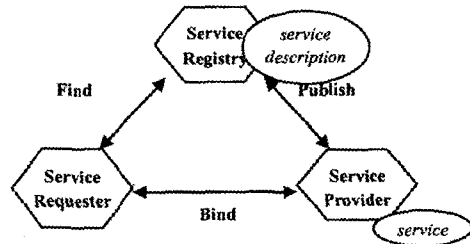


그림 1. 웹서비스의 구조

웹서비스는 서비스를 제공하는 Service Provider 와 서비스에 대한 메타 정보를 저장하고 관리하는 Server Registry, 그리고 서비스를 사용

본 연구는 첨단정보기술연구센터를 통하여 한국과학재단의 지원을 일부 받았음

본 연구는 한국과학기술정보연구원의 "그리드 서비스를 위한 Repository 시스템 개발" 과제에서 일부 지원 받았음

하는 Service Requester 로 이루어진다. Service Provider 는 자신이 제공하는 서비스에 대한 메타 정보인 *service description* 을 Service Registry 에 등록(Publish)한다. Service Registry 는 *service description* 을 저장하고 관리하면서 검색 기능을 제공한다. Service Requester 는 자신이 원하는 서비스를 Service Registry 에서 제공하는 검색 기능을 사용하여 찾고(Find), 해당 서비스의 *service description* 을 얻어온다. Service Requester 는 이렇게 얻어온 *service description* 을 사용하여 해당 서비스를 제공하는 Service Provider 로부터 원하는 서비스를 제공받는다(Bind). 이러한 과정을 통하여 Service Requester 는 분산되어 있는 서비스들을 통합하여 원하는 작업을 수행하게 된다.

2.2. UDDI 의 역할

UDDI 는 웹서비스 환경에서 *service description* 의 저장, 검색 및 관리를 담당하는 Service Registry 의 기능을 수행하는 Repository 시스템에 대하여 W3C 에서 제정한 표준이다. UDDI 표준을 따르는 Repository 시스템은 그림 1 에서와 같이 Service Provider 가 제공한 (Publish) *service description* 정보를 저장하고, 서비스를 사용하고 자 하는 Service Requester 의 검색 요청(Find)을 수행하는 기능을 제공한다. W3C 는 UDDI 명세를 통해 UDDI 가 제공해야 하는 API 들과 데이터 구조에 대해 설명하고 있다[4]. 명세에 따르면 UDDI 는 웹서비스에 대한 *service description* 정보로서 웹서비스에 접근(Bind)하기 위한 정보와 웹서비스 제공자에 대한 정보등을 저장한다. UDDI Repository 시스템은 웹서비스를 이용하려는 응용프로그램에게 이러한 정보를 제공하여 분산된 서비스들을 통합하여 원하는 작업을 수행할 수 있도록 함으로서 분산 응용프로그램의 기반 시스템으로 활용될 수 있다.

3. UDDI 저장 구조 및 API 분석

본 장에서는 UDDI 를 따르는 Repository 시스템을 효과적으로 설계하기 위하여, UDDI 3.0 명세에서 정하고 있는 데이터 저장구조와 API 에 대해 분석한다. 이를 위해 제 3.1 절에서는 UDDI 에서 관리해야 할 데이터의 저장구조를 분석한다. 그리고 제 3.2 절에서는 UDDI API 들을 기능에 따라 분류한다.

3.1. UDDI 저장 구조의 분석

본 절에서는 웹서비스 정보를 저장하기 위해 UDDI 에서 필요로 하는 저장구조와 저장구조를 구성하는 요소들에 대해 설명한다.

UDDI 3.0 명세에서는 UDDI API 에서 필요로 하는 여러 가지 데이터 구조[4]를 제시하고 있다. 본 논문에서는 이들을 사용 목적에 따라 주요 데이터 구조(Core Data Structure)와 추가 데이터 구조(Additional Data Structure)로 분류한다. 주요 데이터 구조는 웹서비스를 이용하기 위해 응용프로그램이 필요로 하는 웹서비스 위치정보나 웹서비스 제공자 정보 등을 저장하기 위해 사용된다. 주요 데이터 구조를 구성하는 세부 데이터 구조는 표 1 과 같다.

표 1. 주요 데이터 구조

분류	설명
businessEntity	서비스 제공자에 관한 정보
businessService	제공하는 서비스에 관한 정보
bindingTemplate	웹서비스의 위치 및 접근에 관한 정보
tModel	분류 및 식별 체계를 제공하는 정보

businessEntity 는 서비스를 제공하는 기업이나 기관에 대한 정보를 저장하는 데이터 구조이다. 그리고 businessService 는 이러한 기업이나 기관이 제공하는 서비스에 대한 이름이나 설명정보를 저장하기 위해 사용된다. bindingTemplate 는 businessService 를 구성하는 웹서비스에 대한 위치정보나 웹서비스를 호출하는데 사용하는 통신방법과 파라미터를 포함한다. 마지막으로 tModel 은 주요 데이터 구조를 구성하고 있는 businessEntity, businessService, bindingTemplate, tModel 을 분류하기 위해 사용되는 분류시스템에 대한 정보를 나타낸다.

추가 데이터 구조는 UDDI 시스템의 운영에 관한 정보를 포함한다. 이러한 추가 데이터 구조를 구성하는 세부 데이터 구조에는 businessEntity 간의 관계를 나타내는 정보를 저장하는 publisherAssertion, UDDI 간의 데이터 복제를 위해 사용되는 저장 구

조인 Replication, UDDI 주요 데이터에 대한 변화를 사용자에게 통보해 주기 위해 사용되는 저장 구조인 Subscription, 주요 데이터에 대해 발생한 생성,수정,삭제에 관한 기록 정보를 저장하는 operationalInfo, 마지막으로 UDDI 시스템 및 데이터에 대한 접근 제어 정보를 저장하는 securityPolicy가 있다.

제 3.2 절 에서는 UDDI 가 제공하는 API 를 기능별로 분류하고, 사용자에게 UDDI 표준에 따라 API 를 제공할 수 있는 방법에 대해 설명한다.

3.2. UDDI API 분류

UDDI 명세에는 Repository 시스템이 제공해야 할 기능들을 총 40 개의 API 형태로 제시하고 있다. 이러한 API 들을 기능에 따라 분류하면 표 3 과 같다.

표 3. UDDI 가 규정하는 API 의 기능에 따른 분류

분류	세부 분류	설명
저장	Publication	UDDI Core Data 의 등록/수정/삭제를 제공하는 API 그룹
검색	Inquiry	UDDI Core Data 에 대한 검색을 제공하는 API 그룹
데이터 변경	Subscription	UDDI Core Data 의 변화를 사용자에게 알리는 기능을 제공하는 API 그룹
운영 및 관리	Security Policy	UDDI Data 에 대한 사용자의 접근권한을 부여하기 위한 API 그룹
	ValueSet	UDDI Core Data 의 분류 및 식별을 위해 사용한 값이 유효한 값인지의 검증용 위한 API 그룹
데이터 복제 및 소유권 이전	Custody and Ownership	UDDI Data 의 소유권 이전을 지원하는 API 그룹
	Replication	UDDI 간의 데이터 복제를 할 수 있도록 제공되는 API 그룹

UDDI3.0 명세에서는 UDDI 시스템이 응용 프로그램이나 사용자에게 제공하는 API 들을 기능에 따라 웹서비스 정보에 대한 저장기능을 제공하는 API 들, 저장된 웹서비스 정보에 대한 검색기능을 제공하는 API 들, 주요 데이터의 변경 정보를 요청자에게 통보하는 기능을 제공하는 API 들, UDDI 의 저장시스템인 데이터베이스에 저장된 웹서비스 정보를 다른 UDDI 에 복제하는 기능을 제공하는 API 들, UDDI 시스템의 운영 및 관리기능을 제공하는 API 들로 나누고 있다. UDDI 가 제공하는 이러한 API 들을 사용하고자 하는 응용프로그램이나 사용자는 분산환경의 객체를 호출하기 위해 W3C 에서 제정한 표준 프로토콜인 SOAP(Simple Object Access Protocol)을 사용하여 원하는 웹서비스를 호출할 수 있다. SOAP 은 HTTP 와 XML 을 사용하여 표준화된 방법으로 분산된 객체를 호출하고 응답을 받을 수 있는 특징을 가지므로 응용 프로그램의 개발 환경이나 수행 환경에 상관없이 동일한 방법으로 응용 프로그램에 대한 수행요청을 할 수 있다는 장점을 가지고 있다. 본 논문에서는 API 호출 방법으로 UDDI 3.0 명세에서 제시하는 SOAP 과 API 의 연동을 고려하여 API 들을 설계하였으며 기능에 따른 API 의 분류를 기반으로 Repository 시스템의 아키텍처를 설계하였다.

4. 데이터베이스 스키마 및 질의식 설계

본 장에서는 제 3 장에서 언급한 UDDI 에서 관리하는 데이터 구조의 분석내용을 토대로 Repository 시스템의 데이터베이스 스키마를 설계한다. 그리고 설계한 데이터베이스 스키마에 대한 검색 및 저장을 수행하기 위해 사용하는 SQL 질의식 설계에 대해 설명한다. 제 4.1 절에서는 Repository 시스템에서 관리하는 데이터베이스 스키마를 설계에 대하여 설명하고 제 4.2 절에서는 이러한 스키마에 대한 SQL 검색 질의식 설계에 대해 설명한다.

4.1. 데이터베이스 스키마 설계

본 절에서는 Repository 시스템의 데이터베이스 스키마 설계에 대해 설명한다. 이미 제 3.1 절에서 언급한 바와 같이 UDDI 를 구성하는 데이터 구조들은 이를 활용하는 목적에 따라 주요 데이터 구조와 추가 데이터 구조로 나누어진다. 주요 데이터 구조는 4 개의 세부 데이터 구조로 나뉘며 각 세부 데이터 구조는 대응하는 여러 개의 데이터베이스 테이블들로 구성된다. UDDI 에 저장될 웹서비스 정보는 XML Document 형태로 입력되며 대응하는 테이블들의 구조에 맞게 분할된 후 저장된다. 주요 데이터에 대한 개념적인 구조는 그림 2 와 같다.

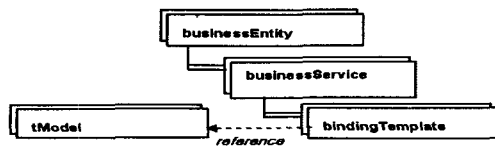


그림 2. UDDI 주요 데이터 구조

주요 데이터 구조의 구성요소인 businessEntity, businessService, bindingTemplate은 각각 1대 n의 포함관계를 가진다. 그리고 분류 정보를 나타내기 위해 tModel을 참조할 수 있다. 그림 2에서 보는 바와 같이 UDDI는 웹서비스에 대한 정보를 계층적인 구조로 저장하고 있으며 tModel을 통해 저장된 데이터를 다양한 분류시스템과 연관시킬 수 있다 따라서 businessEntry로부터 시작하여 businessService, bindingTemplate의 순서로 검색을 수행하는 drill-down 형태의 검색을 통한 웹서비스의 발견 및 다양한 검색 SQL 질의문을 통한 웹서비스의 발견을 모두 효과적으로 지원할 수 있다[4]. 추가 데이터 구조는 주요 데이터 구조와는 달리 데이터 구조간의 연관 관계가 거의 없다. 따라서 Repository 시스템의 각 기능에서 사용할 수 있는 독립적인 테이블 형태로 데이터베이스 스키마를 설계하였다.

이와 같은 설계를 바탕으로 생성한 데이터베이스 테이블의 수는 총 62개로 이들 중에 주요 데이터 구조는 33개의 테이블로, 추가 데이터 구조는 29개의 테이블로 설계가 되었다.

4.2. 저장 및 검색을 위한 SQL 질의식 설계

본 논문에서 설계한 Repository 시스템은 API를 통하여 제 4.1절에서 설계한 데이터베이스 테이블에 대한 저장 및 검색 기능을 제공한다. 이러한 기능을 제공하기 위하여 각 API의 기능에 따라 데이터베이스에 대한 저장 및 검색을 수행하는 SQL 질의식을 설계하였다. SQL 질의식은 각 API의 검색 특성과 설계된 데이터베이스 스키마를 고려하여 최적의 성능을 발휘할 수 있는 방향으로 설계되었다.

저장 및 검색을 위해 설계한 SQL 질의문은 Repository 시스템의 저장 시스템으로 범용 DBMS를 사용하는 것을 고려하여 표준 SQL 문법에 따라 설계하였다. 그리고 대용량의 웹서비스 정보를 효율적으로 검색할 수 있도록 설계하였다. 덧붙여서 웹서비스 정보의 검색을 위한 SQL 질의문 설계시 본 연구실에서 개발한 객체관계형 데이터베이스인 오디세우스의 IR(Information Retrieval)기능을 이용하여 입력된 keyword를 포함하는 텍스트 형태의 웹서비스 정보를 효율적으로 검색할 수 있도록 하였다.

5. 시스템 아키텍처

본 장에서는 본 논문에서 제안하는 Repository 시스템의 목표 아키텍처에 대해서 설명한다.

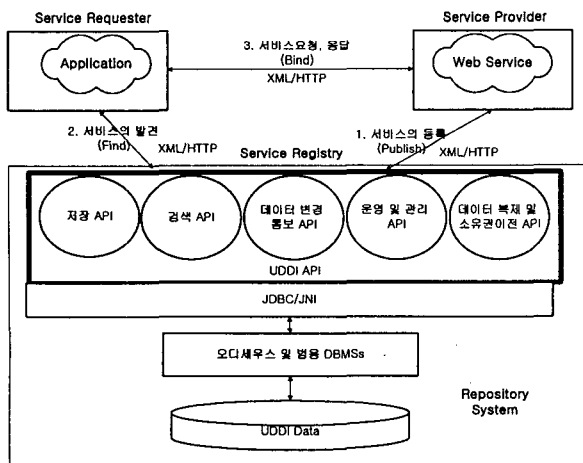


그림 3. Repository 시스템 아키텍처

그림 3은 그림 1에서 설명한 웹 서비스를 제공하는 ServiceProvider, 웹서비스를 이용하는 응용프로그램(Service Requestor), 그리고 웹서비스 정보를 저장하는 Repository 시스템이 어떤 연관관계를 가지고 연동되는지를 나타내고 있다. ServiceProvider는 Repository 시스템이 제공하는 저장 API를 사용하여 자신이 제공할 웹서비스에 대한 정보를 Repository 시스템에 등록한다. 웹서비스를 사용하고자 하는 응용프로그램은 Repository 시스템이 제공하는 검색 API를 사용하여 원하는 웹서비스에 대한 위치정보를 얻는다. 그리고 이를 통하여 웹서비스를 접근하고 활용할 수 있게 된다. 본 논문에서 제안하는 Repository 시스템은 응용프로그램이 이러한 방법으로 웹서비스를 활용하도록 지원하기 위해 웹서비스 정보의 저장, 검색, 데이터 변경통보, 복사 및 소유권이전, 운영 및 관리로 분류되는 API 그룹을 응용프로그램에게 제공하도록 설계되었다. 그리고 이러한 API들은 SOAP을 통하여 표준화된 방법으로 호출할 수 있도록 설계하였다. 또한 저장시스템으로써 범용 DBMS를 사용하는 것을 고려하여 각 API에서는 JDBC를 통하여 데이터베이스에 접근하도록 하여 확장성을 높였다 [5][6].

6. 결론

웹서비스는 분산된 컴퓨팅 자원과 정보를 효과적으로 활용하려는 노력의 일환으로 W3C에서 제정한 분산환경의 표준 프로그래밍 모델이다. UDDI는 이러한 환경에서 웹서비스에 대한 메타데이터를 저장하고 이를 검색할 수 있도록 기능을 제공하는 Repository 시스템이다. 웹서비스를 이용하고자 하는 응용프로그램은 UDDI가 저장한 정보를 검색하여 원하는 웹서비스를 발견하고 이를 활용할 수 있다. 이와 같이 UDDI는 웹서비스 기반의 분산 응용에서 필요한 디렉토리 서비스기능을 제공하여 분산된 웹서비스의 통합을 지원하는 기반 시스템으로 널리 활용될 것으로 기대된다.

본 논문에서는 W3C에서 제정한 UDDI 3.0 명세를 분석하여 Repository 시스템에서 관리해야할 데이터 구조를 분석하고 이를 DBMS에 저장하기 위한 데이터베이스 스키마를 설계하였다. 그리고 UDDI에서 규정하고 있는 API들을 기능별로 분류하고, 각각의 API에서 데이터베이스에 접근하기 위해 사용하는 SQL 질의문을 설계하였다. 또한 SOAP을 통하여 표준화된 방법으로 API에 접근할 수 있고 범용 DBMS를 저장시스템으로 활용할 수 있는 Repository 시스템의 목표 아키텍처를 설계하였다.

향후에는 본 논문에서 설계한 아키텍처를 기반으로 Repository 시스템의 기능을 단계적으로 구현할 예정이다. 또 요즘 분산 연구 분야에서 주목 받고 있는 웹서비스를 기반으로 하는 그리드 서비스에 활용할 수 있도록 개발된 시스템을 활용하기 위한 연구를 수행할 것이다.

참고 문헌

- [1] 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>).
- [2] Ewald, T., "Understanding Web Services", Microsoft Corporation Articles, Microsoft Co., Sept. 2002 (available from <http://msdn.microsoft.com/webservices/understanding/default.aspx>).
- [3] 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>).
- [4] Tom Bellwood, Luc Clement, David Ehnebuske, Andrew Hately, Marayann Hondo, "UDDI Version 3.0 Published Specification", OASIS, July 19, 2002.
- [5] Seth White, et al., "JDBC API Tutorial and Reference, Second Edition", Addison Wesley, Jun 1999.
- [6] JavaSoft, Java Native Interface Specification Release 1.1, SUN, May 1997 (available from <http://java.sun.com/j2se/1.4/docs/guide/jni/spec/intro.doc.html>)