제 18 회 한국정보처리학회 추계학술발표대회 논문집 제 9 권 제 2 호 (2002.11)

ebXML Registry 기반의 UDDI Registry Server 구현 [1)]

박재홍°*, 김상균*, 이규철*, 조현규**
*충남대학교 컴퓨터공학과, **한국전자통신연구원
{jhpark°, skkim, kclee}@ce.cnu.ac.kr{hkcho}@etri.re.kr

Implementation of the UDDI Registry Server on top of the ebXML Registry

Jae-Hong Park°*, Sang-Kyun Kim*, Kyu Chul Lee*, Hyun-Kyu Cho**
*Dept. of Computer Engineering, Chungnam National University,
**Electronics and Telecommunications Research Institute

요 약

최근 들어 B2B 기반의 전자상거래 프레임워크로 UN/CEFACT 과 OASIS 를 주축으로 하는 ebXML 과 Microsoft, IBM, Ariba 를 포함한 기업과 단체를 주축으로 하는 웹 서비스가 대두되고 있다. 이들은 모두 사용자 및 어플리케이션들이 정보를 저장하고 이를 공유할 수 있는 Registry 를 기반으로 서비스를 제공하고 있으며, 이를 위해 ebXML 은 ebXML Registry 를, 웹 서비스는 UDDI Registry 를 이용한다.

ebXML 과 웹 서비스는 서로 다른 Registry 를 사용하고 있지만, Registry 의 구조와 기능은 유사하거나 동일한 부분이 많다. 본 논문은 이에 착안하여 서로 유사하거나 동일한 역할을 가진 ebXML RIM 의 구조와 UDDI 의 데이터구조를 매핑함으로써, ebXML RIM 상에 UDDI 데이터구조를 표현하는 규칙을 찾고, 이를 이용하여 UDDI Registry Client 의 서비스 요구를 ebXML Registry 를 이용하여 처리하는 UDDI Registry Server 를 설계 한다. 이는 ebXML Registry 에 별도의 변경을 가하지 않고도 UDDI Registry Client 가 ebXML Registry 를 사용할 수 있게 한다.

1. 서 론

최근 들어, 각 표준 단체나 기업들은 인터넷 기반의 B2B 전자 상거래를 위한 다양한 프레임워크를 연구, 개발하고 있으며, 제안된 프레임워크들은 각 기업들의 비즈니스에 관련된 기업정보 및 서비스 정보등을 저장하여 여러 사용자 및 어플리케이션들이 공유할 수 있는 Registry 를 기반으로 다양한 서비스를 제공하고 있다.

현재까지 많은 프레임워크들이 제안되었지만 그 중에서도 ebXML(electronic business eXtensible Markup Language)[1]과 UDDI(Universal Description, Discovery and Integration)[2]로 대표되는 웹 서비스가 기업들의 관심대상이 되고있다. 현재까지 많은 프레임워크들이 제안되었지만 그 중에서도 ebXML 과 UDDI 로 대표되는 웹 서비스가 기업들의 관심대상이 되고있다.

ebXML 은 "Creating A Single Global Market"이라는 기치 아래 그 동안 국제 EDI 표준을 추진해 왔던 UN/CEFACT 과 OASIS 가 주축이 되어, XML 을 이용하여 인터넷 기반의 e-business 가 가능하도록 제정한 전자 상거래 표준이다. ebXML 표준은 2001 년 5 월에 완성되었으며, 그 후에도 웹 서비스(Web Service)와 같은 다양한 표준을 ebXML 에서 수용하는 등 표준의

1) 본 논문은 한국전자통신연구원의 위탁연구과제인 'UDDI 레지스트리 서버 및 클라이언트 도구 개발'의 일부로 수행된 결과임.

가치를 높이기 위한 연구가 진행중이다. 특히 ebXML RIM(Registry Information Model)[3]과 RS(Registry Service)[4]는 버전 2.0 Draft 가 2001 년 12 월에 나온 상태이다. ebXML Registry 2.0 스펙은 기존의 1.0 을 수용하면서 웹 서비스의 기능을 수행할 수 있도록 만들어졌다. 이를 위해 ebXML Registry 모델에 서비스와 관련된 모듈을 추가하였으며, 거래 상대자들 사이에 메시지를 교환할 때 ebXML MS(Messaging Service)[5]만을 이용하는 것이 아니라 SOAP(Simple Object Access Protocol)[6]을 이용한 방법도 지원하고 있다.

웹 서비스는 웹 상에서 이기종 시스템들 간의 어플리케이션들이 서로 접근 가능하도록 지원하는 소프트웨어 컴포넌트 모델이다. 웹 서비스는 XML 기술을 기반으로 한 개방형 표준들로 구성되어 있기 때문에 기존의 분산 컴퓨팅 모델에 비해 단순하고 이기종 시스템 간의 상호 운용성을 극대화하였다. 웹 서비스는 UDDI Registry 를 기반으로 웹 서비스의 등록, 검색을 위한 기반 기술을 제공하며, 웹 서비스의 요청 및 응답에서 사용되어지는 메시지 형식 위해 SOAP 을 정의하고, 웹 서비스 접근에 이용되는 SOAP 메시지, 프로토콜, 웹 서비스에 대한 인터넷 위치 정보 등을 기술하기 위해 WSDL(Web Service Description Language)[7]을 지원한다.

이와 같이 B2B 기반의 전자상거래 프레임워크로 ebXML 과 UDDI 로 대표되는 웹 서비스가 대두되고 있지만 현재로서는 기업에서 이 표준들 중 어느 하나를 도입하거나 두 가지를 모두 익힌다는 것은 어려운 실정이다. ebXML 은 웹 서비스의 기능을 수용하는 등 일반적이고 표준화된 프레임워크로 자리잡아가고 있지만 일반 거래 당사자들이 이용하기에는 아직 시간이 필요하며 웹 서비스는 Microsoft, IBM, , Ariba 를 포함한 기업과 단체가 주축이 되어 UDDI Registry 를 구축하고, 웹 서비스를 위한 도구들을 개발하여 쉽게 거래를 할 수 있도록 하고 있지만 일반적이지 못하기 때문이다. 또한 한 기업에서 두 프레임워크에 대한 솔루션을 모두 가지기에는 필요한 비용과 노력이 부담이 된다.

현재는 ebXML 보다 웹 서비스에 대한 솔루션이 많이 개발되어 있고, 도입 비용도 저렴하므로, 전자상거래를 하는 대부분의 기업은 웹 서비스를 많이 이용하고 있다. 이러한 상황에서 ebXML 이 현재 증가하고 있는 웹 서비스에 대한 요구를 수용하기 위해서는 UDDI Registry Client 가 UDDI Registry 뿐만 아니라 ebXML Registry 에도 서비스를 요구할 수 있도록 지원해야 하며 이를 위해서는 UDDI Registry Client 의 서비스 요구를 ebXML Registry 의 서비스 요구로 변환할 수 있는 UDDI Registry 서버가 필요하다.

이를 위해 본 논문은 웹 서비스의 UDDI Registry Client 가 UDDI Registry 에 서비스를 요구하는 것처럼 ebXML Registry 에도 서비스를 요구 할 수 있는 UDDI Registry Server 를 설계하였으며, 이는 UDDI Registry Client 의 서비스 요구를 ebXML Registry 의 서비스 요구로 바꾸어 처리 함으로써 ebXML Registry 에 별도의 변경을 가하지 않고도 UDDI Registry Client 가 ebXML Registry 를 사용할 수 있게 한다.

2. UDDI Registry Server 의 시스템 구조

2.1 시스템 구조

그림 1 은 본 논문의 전체 시스템 구조이다. 이 그림에서 "UDDI Registry Server"와 "UDDI Extension of ebXML"이 본 논문에서 설계한 부분이며, 이를 통해 UDDI Registry Client 는 UDDI Registry Server 를 이용하여 ebXML Registry 에도 UDDI Registry 서비스를 요구할 수 있다.

"UDDI Registry Server"는 UDDI 서비스 요구를 ebXML Registry 서비스로 바꾸어 처리하는 모듈이며, 이 모듈은 서비스 변환 시 "UDDI Extension of ebXML"에 정의되어 있는 데이터 매핑정보를 이용하다.

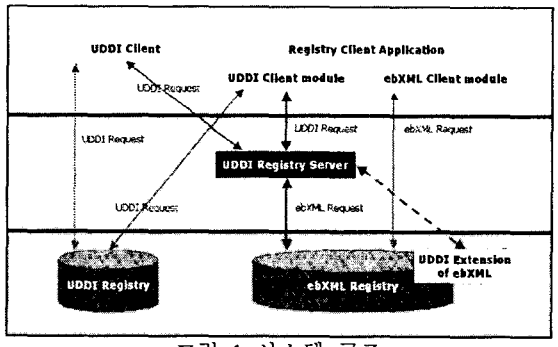

그림 1 시스템 구조

"UDDI Registry Server"의 자세한 동작은 4 장에서 설명된다.

3. ebXML-UDDI 서비스 제공자

ebXML RIM 과 UDDI 의 데이터구조에 의해 정의된 구조상 ebXML RIM 은 UDDI 데이터구조의 Super Set 으로 볼 수 있으며, 이에 착안하여 서로 동일하거나 비슷한 역할을 가진 ebXML RIM 의 구조와 UDDI 의 데이터구조를 매핑함으로써, ebXML RIM 상에 UDDI 데이터구조를 표현하는 규칙을 찾을 수 있다.

3.1 데이터구조 매핑

3.1.1 BusinessService

BusinessServices

UDDI				ebXML		
Field Name	Type	Length		Field Name	Type	Length
businessKey	UUID	41	v	Organization.id	string	64
v serviceKey	UUID	41	v	Service.id	string	64
v name	string	255		Name.value	string	256
description	string	255		Description.value	string	256
v bindingTemplate	struct			ServiceBinding Service Description		
v categoryBag	struct			Classification Name		

그림 2 BusinessService 의 데이터구조 매핑 테이블

그림 2 는 UDDI 데이터구조를 ebXML RIM 에 매핑 시킨 결과 중 BusinessService 에 해당되는 내용 이며. 좌측은 UDDI 데이터구조, 우측은 ebXML RIM 이다. 'v' 표시는 Mandatory Field 를 나타낸다.

그림 2 에서 보는 바와 같이 UDDI 데이터구조의 BusinessServices 는 ebXML RIM 의 Service 에 매핑 된다. BusinessServices 는 bindingTemplate 과 CategoryBag 을 Sub Structure 로 가지며, 이는 각각 ebXML RIM 의 ServiceBinding 과 Classification 에 매핑 된다.

UDDI 의 데이터구조와 ebXML RIM 을 매핑하는 과정에서 위의 그림 2 에서도 볼 수 있듯이 서로의 데이터 구조가 완전히 일치하지 않기 때문에 3.1.2 와 3.1.3 의 Mismatch 가 발생한다.

3.1.2 Mandatory/Option Mismatch

ebXML RIM 에서는 Mandatory 이고 UDDI 데이터구조에는 Option 인 Filed 를 서로 매핑할 경우, 또는 그 반대의 경우에 발생하는 문제이며 3.1.2.1 과 3.1.2.2 의 두 경우가 존재한다.

3.1.2.1 UDDI:Mandatory, ebXML:Option

UDDI 데이터구조의 Mandatory Field 를 ebXML RIM 의 Option Field 에 매핑할 경우, ebXML Registry Client 의 Publishing 과정에서 Option Field 를 저장하지 않으면 UDDI Registry Client 의 검색 결과 중 Mandatory Field 에 NULL 이 존재할 수 있다.

3.1.2.2 UDDI:Option, ebXML:Mandatory

UDDI 데이터구조의 Option Field 를 ebXML RIM 의 Mandatory Field 에 매핑할 경우, UDDI Registry Client 의 Publishing 과정 중 Option Field 에 값이 없으면 가상 데이터를 추가하여 저장해야 한다.

3.1.3 Data Type Mismatch

UDDI 데이터구조의 Field 를 ebXML RIM 의 Field 에 매핑할 경우, ebXML RIM 과 UDDI 데이터구조의 Data Type 이 다르면 발생하는 문제이며, 3.2.3.1 과 3.2.3.2 의 경우가 존재한다.

3.1.3.1 ebXML > UDDI

UDDI 데이터구조 Filed 의 Data Type 크기가 ebXML RIM Filed 의 Data Type 크기보다 작을 경우, UDDI Registry Client 에서 검색 시 truncation 이 발생 할 수 있다.

3.1.3.2 ebXML < UDDI

UDDI 데이터구조 Filed 의 Data Type 크기가 ebXML RIM Filed 의 Data Type 크기보다 클 경우, UDDI Registry Client 에서 저장할 Data 에 truncation 이 발생 할 수 있다. 이 때는 truncation 될 Data 를 ebXML RIM 의 Slot 을 이용하여 분할저장하며, 후에 UDDI Registry Client 의 검색 시 Slot 에 나누어 저장된 내용을 merge 하여 돌려준다.

3.2 System Usage

본 논문에서 설계된 UDDI Registry Server 를 이용한 UDDI Registry Client 의 ebXML Registry 사용에 대해 살펴보면, ebXML Registry 를 UDDI Registry Client 와 ebXML Registry Client 가 공유하여 사용할 경우, UDDI Registry Client 는 3.1.2.1 과 3.2.3.1 에서 논한 바와 같이 검색결과 중 Mandatory Field 에 NULL 값이 존재하거나, Data 의 truncation 이 발생 할 수 있다. 그러나 ebXML Registry 를 UDDI Registry Client 용으로만 사용할 경우에는 3.1.2.2 와 3.1.3.2 의 Mismatch 에도 불구하고 각각에서 제시한 Mismatch 해결 방법인 가상데이터 추가와 분할저장기법을 이용하여 Data Integrity 를 보장한다.

4. UDDI API 변환 메커니즘

4.1 에서는 UDDI Registry Server 의 동작단계를 설명하며, 동작단계에서 핵심이 되는 UDDI API 변환 메커니즘을 4.2 에서 자세히 설명한다.

4.1 UDDI Registry Server 의 기본 동작

본 논문에서 설계된 UDDI Registry Server 는 기본적으로 A 부터 E 까지의 5 단계로 동작하며 아래에서 각각의 단계를 예를 들어 설명한다.

A. UDDI Registry Client 에게 전달 받은 API 메시지를 분석하여 어느 메시지 변환 알고리즘에 적용해야 할지를 알아낸다. 그림 3 은 UDDI API 메시지인 save_service 의 예이다.

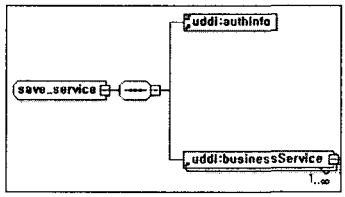

그림 3 save_service 의 예

B. 메시지 변환알고리즘을 적용하여 UDDI API 메시지를 ebXML Registry 메시지로 변환한다. 그림 4 는 UDDI API 인 save_service 메시지를 ebXML Registry 메시지로 변환한 결과 중의 일부이다.

```
- <SubmitObjectsRequest>
  - <rim:LeafRegistryObjectList>
    - <rim:Service id="serviceKey" status="Submitted" majorVersion="1" minorVersion="0">
      + <rim:Name>
      + <rim:Description>
      - <rim:ServiceBinding id="bindingKey" service="serviceKey" accessURI="accessPoint">
        + <rim:Description>
        + <rim:SpecificationLink specificationObject="externalLinkId2" id="tModelKey">
        </rim:ServiceBinding>
    </rim:Service>
                    :
```

그림 4 ebXML Registry Publishing Message

C. 변환된 메시지를 ebXML Registry Server 에 전달한다.

D. ebXML Registry Server 에게 Response 메시지를 받는다. 그림 5 는 Response 메시지이며, Success, Failure, Unavailable 중 하나의 값을 가질 수 있다.

```
<RegistryResponse status="Success" />
<RegistryResponse status="Failure" />
<RegistryResponse status="Unavailable" />
```

그림 5 ebXML Registry Response Message

D. ebXML Registry Response 메시지를 UDDI Registry Response 메시지로 변환한다. 그림 6 은 ebXML Registry 의 Response 가 Success 일 경우에 반환되는 UDDI Registry Response 메시지의 일부이다.

```
- <businessService serviceKey="serviceKey">
    <name>Name</name>
    <description>Description</description>
  - <bindingTemplates>
    - <bindingTemplate bindingKey="bindingKey">
        <accessPoint URLType="mailto" />
```

그림 6 UDDI Response Message 예

E. 마지막으로, UDDI Registry 의 Response 메시지를 UDDI Registry Client 에게 전달한다.

4.2 UDDI API 변환 메커니즘

UDDI API 변환 메커니즘을 save_service 의 예를 들어 설명한다.

save_service 는 BusinessService 를 저장하는 API 이다. 그림 7 에서 보는 바와 같이 save_service 는 Registry 에 저장된 값들 중에서 현재 저장할 serviceKey 를 가진 BusinessService 가 있는지를 검색한 후, 검색 결과가 있으면 Update Business Service, 그렇지 않으면 Insert Business Service 를 수행한다.

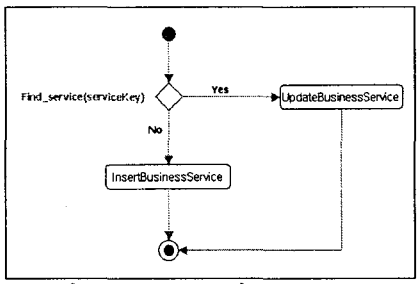

그림 7 save_service 의 Activity Diagram

save_business 가 Insert Business Service 로 처리될 경우에는 ebXML Registry 의 SubmitObjectRequest 로 변환된다. 된다.

그림 8 은 그림 7 의 InsertBusinessService 에 해당 되는 메시지 변환 Activity Diagram 이다. 그림 8 에서 각 단계별로 입력되는 값은 3 장의 그림 2 의 매핑정보에 맞추어 SubmitObjectRequest 메시지를 구성한다.

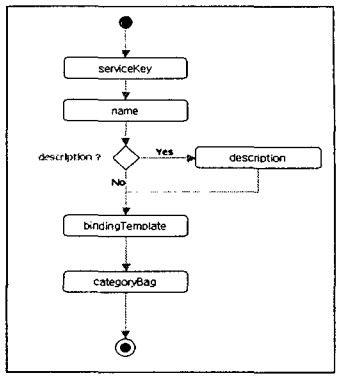

그림 8 save_business 의 InsertBusinessService Activity Diagram

5. 결론 및 향후 연구 방향

본 논문에서는 UDDI Registry Client 가 ebXML Registry 를 UDDI Registry 와 같이 사용할 수 있도록 하는 UDDI Registry Server 를 설계하였다. 이를 위해 UDDI Registry Server 의 시스템 구조를 분석하고, ebXML-UDDI 서비스 제공자와 UDDI API 변환 메커니즘을 설계하였다.

앞으로는 설계된 UDDI Registry Server 를 구현하여 ebXML Registry 와 UDDI Registry 를 통합할 수 있는 인터넷 환경의 B2B 전자상거래를 위한 새로운 솔루션으로 이용될 수 있도록 할 것이다.

참고문헌

[1] UN/CEFACT, OASIS, "electronic business eXtensible Markup Language), http://www.ebxml.org

[2] Microsoft, IBM, Ariba, "UDDI (Universal Description, Discovery and Integration) Version 2.0", http://www.uddi.org/specification.html

[3] UN/CEFACT, OSAIS, "Registry Information Model v2.1", http://www.oasis-open.org/committees/regrep/documents/2.1/specs/ebrim_v2.1.pdf

[4] UN/CEFACT, OSAIS, "Registry Services Specification v2.1", http://www.oasis-open.org/committees/regrep/documents/2.1/specs/ebrs.pdf

[5] UN/CEFACT, OSAIS, "Message Service Specification v2.0", http://www.ebxml.org/specs/ebMS2.pdf

[6] W3C notes, "Simple Object Access Protocol(SOAP) 1.1", http://www.w3.org/TR/SOAP/

[7] W3C notes, "Web Services Description Language (WSDL)1.1", http://www.w3.org/TR/SOAP/