

분산 UDDI 상에서의 신뢰성 보장을 위한 웹 서비스 정보관리 시스템

배명훈^o 국윤규 김운용 정계동 최영근
광운대학교 컴퓨터과학과
mhsh1006@kw.ac.kr^o

A Study on Web Service Information Management System for a Reliability Guarantee on Distributed UDDI

Myeong-Hun Bae^o Youn-Gyou Kook, Woon-Yong Kim, Kye-dong Jung, Young-Keun Choi
Department of Computer Science, KwangWoon University

요 약

인터넷을 기반으로 하는 비즈니스의 활성화로 웹 서비스의 필요성이 증가함에 따라 웹 서비스의 등록 및 검색을 지원하는 UDDI의 중요성이 대두되고 있다. 현재 범용 UDDI 레지스트리(UBR, UDDI Business Registry)를 운영하고 있는 IBM, Microsoft, Ariba, SAP등은 상호간에 UDDI의 복제(Replication)를 지원하며 하나의 UDDI 레지스트리에 웹 서비스를 등록하면 다른 UDDI에도 자동으로 웹 서비스가 등록된다. 하지만, 이러한 복제는 최소 24시간을 주기로 복제되며, 또한 업데이트에 대해서는 복제작업이 명확히 이루어지지 않는다. 이는 환율 정보 서비스, 예약 서비스, 주식 정보 서비스등 실시간적인 동기화가 필요한 서비스들의 사용자에게 많은 문제점을 안겨준다. 이에 본 논문에서는 모니터링 및 UDDI 동기화 에이전트를 이용하여 웹 서비스의 등록, 삭제 및 업데이트 발생시 등록되어있는 모든 UDDI의 동기화를 수행하여 서비스 사용자에게 항상 신뢰할 수 있는 서비스를 제공 할 수 있도록 한다.

1. 서론

1990년대 월드와이드웹의 사용이 급격히 증가함에 따라 많은 기업들이 인터넷을 기반으로 비즈니스를 수행하기 시작하였다. 전자상거래를 촉진시키기 위해서 IBM, Microsoft, SAP, Ariba는 UDDI(Universal Description, Discovery and Integration)을 개발하였으며, UDDI는 공개적으로 접속이 가능한 UDDI 비즈니스 레지스트리(UBR, UDDI Business Registry)에 사용된다.

UDDI를 이용하여 각 업체의 웹 서비스들은 그들의 정보를 쉽게 공개할 수 있으며, 누구든지 이를 검색하여 원하는 정보를 찾아 이용할 수 있다[1, 4]. 또한, UDDI는 “단 한번 등록, 모든 곳에 배포” 원칙을 지키기 때문에 하나의 UDDI 레지스트리에 포함되어 있는 정보는 다른 레지스트리들로 복제(Replication)되게 된다[1].

하지만, 이러한 복제는 최소 24시간마다 수행되기 때문에 즉시 복제되는 것은 아니며 서비스의 등록 및 삭제시에는 명확하게 반영되지만 서비스의 업데이트시에는 동일한 서비스로 인식되어 명확한 복제를 수행하지 않는다[7]. 그로인해 일부 UDDI 레지스트리들의 경우 데이터의 신뢰성(Reliability)을 보장할 수 없게 된다. 서비스 사용자는 UDDI에 등록되어 있는 데이터가 얼마나 자주 업데이트 되는지 알 수 없기 때문에 서비스에 대한 설명과 하이퍼링크 등의 데이터가 가장 최신의 정보를 반영하고 있

는지에 대해 신뢰성을 가질 수 없게 되며, 최악의 경우 각각의 UDDI에서 서로 다른 웹 서비스를 제공하는 경우가 발생한다[5].

이에 본 논문에서는 서비스를 제공하는 서비스 제공자(Service Provider)가 제공하는 웹 서비스의 신뢰성을 보장할 수 있도록 이동 에이전트를 이용한 웹 서비스 정보 관리 시스템을 설계하였다.

웹 서비스 정보 관리 시스템은 현재의 복제 방식과는 달리 서비스 제공자의 시스템을 에이전트를 사용하여 모니터링함으로써 웹 서비스의 등록 및 삭제, 업데이트가 발생할 경우 지정된 UDDI들로 웹 서비스의 등록 및 동기화를 수행한다. 따라서, 환율 정보 제공 서비스나 항공사 예약 서비스, 주식 정보 제공 서비스등 서비스의 업데이트 발생시 실시간적인 동기화가 요구되는 웹 서비스의 이용자는 어느 곳의 UDDI를 사용하더라도 서비스 제공자가 제공하는 웹 서비스의 최신 버전을 사용할 수 있으며 다수의 UDDI를 효율적으로 관리할 수 있다.

본 논문은 2장에서 관련연구에 대하여 살펴보고, 3장에서 본 논문에서 제안한 웹 서비스 정보 관리 시스템에 대하여 살펴보고, 마지막으로 결론으로 구성된다.

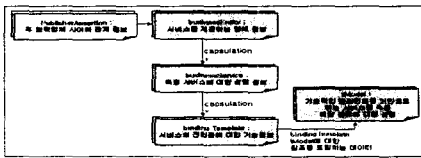
2. 관련 연구

2.1 UDDI

UDDI는 수많은 기업들의 주도 아래 추진되고 있는 프로젝트로, 웹 기반의 정보 표현과 검색, 통합에 대한 통일 규약을 정의한다[1].

UDDI는 웹 서비스의 구성 요소 기술 중 한 가지로서, 서비스 제공자 측면의 기업은 자사가 제공하는 전자상거래 및 기업정보 등을 UDDI 비즈니스 레지스트리(UBR)에 등록(Publish)하고, 서비스 이용자 측면의 기업은 UBR에 공개된 서비스들중에서 이용할 수 있는 서비스를 검색(Find)하며, 제공된 서비스의 기술사양이 일치할 경우, 사용자측과 제공자측이 상호 접속(Bind)한다[2].

UDDI 버전 2.0 데이터 구조 레퍼런스는 비즈니스 트랜잭션을 위해 클라이언트측이 알아야 하는 서비스 제공자의 정보를 명시하고 있다[3]. UDDI 정보 모델이라고 알려진 이 정보는 비즈니스 정보, 비즈니스 및 서비스 정보, 바인딩 정보, 서비스 및 명세 정보, 발행자-어썬션(Publisher-Assertion) 정보로 구성되어 있다. (그림 1)



(그림 1) UDDI 2.0 Data Structure

2.3 UDDI4J v2.0 API

UDDI4J는 UDDI 클라이언트 측에서 레지스트리와 상호 작용을 할 수 있도록 API를 제공하는 자바 클래스 라이브러리이다. UDDI4J는 그 기능에 따라 아래와 같이 5개의 패키지로 구분되어진다.

- ① org.uddi4j.client.xx : 질의와 응답에 관련된 메서드 호출을 담당하는 proxy 클래스의 집합
- ② org.uddi4j.datatype.xx : UDDI 자료구조에 해당하는 오브젝트를 정의한 클래스들의 집합
- ③ org.uddi4j.request.xx : UDDI 레지스트리에 대한 inquiry와 publish에 대한 클래스 집합
- ④ org.uddi4j.response.xx : 서버로부터 오는 다양한 응답 형태의 메시지를 나타내는 클래스의 집합
- ⑤ org.uddi4j.util.xx : UDDI 레지스트리를 검색하는 키 값에 해당하는 클래스들의 집합

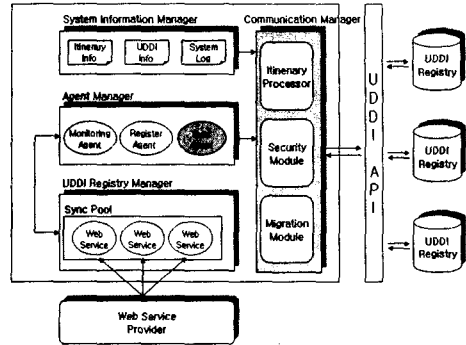
3. 웹 서비스 정보 관리 시스템

본 논문에서 제안하는 시스템은 UDDI의 동기화를 위해 필요한 정보들을 관리하는 시스템 정보 매니저, UDDI 동기화 및 등록을 수행하는 에이전트 매니저, 서비스 제공자가 제공한 웹서비스를 관리 및 모니터링하는 UDDI 레지스트리 매니저, 에이전트의 전송 및 이주보장을 담당하는 통신 관리자로 구성되어 있다. (그림 2)

서비스 개발자(Service Provider)는 배포하려는 웹 서비스를 관리하기 위해 Sync Pool에 서비스를 추가하게 되며, 모니터링 에이전트는 Sync Pool에 등록된 서비스 리스트에 대해 지속적인 모니터링 작업을 수행한다.

Sync Pool에 새로운 데이터가 추가되었을 경우, 모니터링 에이전트는 최초 서비스 등록으로 인식하여 등록 에이

전트에게 웹 서비스 등록을 요청하게 되며 이후 업데이트에 대해 지속적인 모니터링을 수행한다. 모니터링 에이전트가 업데이트를 감지하면 에이전트 매니저에게 서비스 동기화를 요청하게 되며 동기화 에이전트는 서비스가 등록되어 있는 모든 UDDI의 동기화 작업을 수행한다.



(그림 2) 웹 서비스 정보 관리 시스템

3.1 시스템 정보 관리 모듈

시스템 정보 매니저에서는 서비스 제공자가 개발한 웹 서비스를 등록시키고자 하는 범용 UDDI 및 사설 UDDI에 대한 정보를 포함하는 UDDI Info 테이블(표 1)과 각 UDDI의 경로 및 수행 순서를 포함하는 Itinerary Info, 마지막으로 각 웹 서비스의 접근 및 수정 등의 기록을 유지하는 시스템 로그로 구성되어진다.

UDDI Name	UDDI URL	UDDI Type	Access Profile
IBM	ibm-3.com/uddi/services/uddi/	public	#0001
Microsoft	uddi.rte.microsoft.com	public	#0002
WASP	sysintec.com/uddi/web/	private	#0003
DOCOM	pllab.kw.ac.kr/uddi/web/	private	#0004

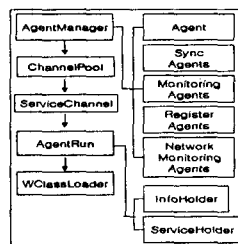
(표 1) UDDI Info 구성 항목

UDDI 동기화 에이전트는 각 UDDI별 ID와 Password등으로 구성되어지는 Access Profile을 참조하여 UDDI에 웹 서비스의 등록 및 업데이트 작업을 수행하게 된다.

Id	Pwsd	Publish	Modify	Detail Info
mhsh1006	0000	04/1/20	04/1/25/16	return type:String

(표 2) Access Profile 구성 항목

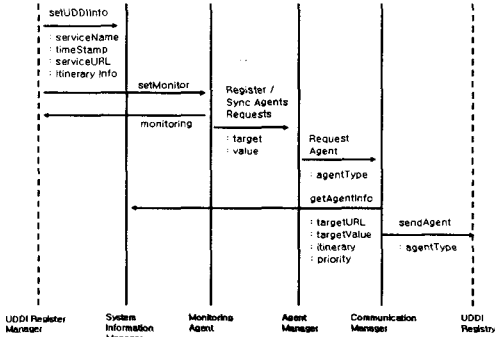
3.2 에이전트 관리 모듈



(그림 3) 에이전트 관리 모듈

에이전트 매니저에서는 Sync Pool의 웹 서비스 추가 및 수정을 모니터링하는 모니터링 에이전트, 개발된 웹 서비스의 등록을 수행하는 등록 에이전트, 웹 서비스의 변경 시 웹 서비스가 등록되어 있는 모든 UDDI의 동기화를 담당하는 UDDI 동기화 에이전트로 구성되어진다.(그림 3)

(그림 4)는 웹 서비스 정보 관리 시스템의 동기화 흐름을 나타낸다. (그림 5)는 모니터링 에이전트가 Sync Pool에서 서비스의 변경이 발생하였을 경우 에이전트 매니저에게 UDDI의 동기화 작업을 요청하는 부분이다.



(그림 4) UDDI 관리 시스템 흐름도

```
private class MonitoringAgents implements Agent, Runnable {
    private Vector serviceList;
    private SyncPool pool;
    private ServiceHolder holder;
    private AgentManager manager;
    private WebServiceGUI serviceGUI;
    .....
    public MonitoringAgents( SyncPool pool, AgentManager manager ) {
        this.pool = pool;
        this.manager = manager;
    }
    .....
    public void run() {
        serviceList = pool.getServiceList();
        serviceGUI = pool.getServiceGUI();

        while( true ) {
            .....
            // Checking for Web Service List contained Sync Pool
            for( int i=0; i<serviceList.size(); i++) {
                // Check what service was modified
                #((ServiceHolder)serviceList.get(i)).isModified() {
                    holder = (ServiceHolder)serviceList.get(i);
                    // Web Service Sync processing
                    #! (manager.request( AgentManager.SYNC, holder )) {
                        serviceGUI.setResult( "Sync processing successful" +
                            "modify >>>" + holder.getLastModifyList() );
                    } else {
                        manager.request( AgentManager.SYNC_FAIL, holder );
                        .....
                    }
                }
            }
        }
    }
    .....
}
```

(그림 5) 모니터링 에이전트

3.3 UDDI 관리 모듈

UDDI 등록 관리자에서는 서비스 제공자가 개발한 웹 서비스의 등록 및 관리를 담당하며, Sync Pool에 등록된 웹 서비스는 등록 에이전트에 의해 각각의 UDDI로 등록되어진 후 모니터링 에이전트에 의해서 감시된다. 만약 Sync Pool에 등록된 웹 서비스에 변경사항이 발생할 경우 에이전트 관리자에게 UDDI의 동기화 작업 수행을 요청하게되며 에이전트 관리자는 동기화 에이전트에게 작업 수행을 요청한다.

3.4 통신 관리 모듈

통신 관리자에서는 등록 에이전트 및 동기화 에이전트의 수행 및 네트워크 측정 작업을 수행하며, 시스템 정보 매니저가 유지하는 UDDI Info, Itinerary Info등을 참조하여 서비스 제공자가 등록 및 동기화를 원하는 모든 UDDI의 동기화 작업을 수행한다. 통신 관리 모듈은 에이전트의 효율적인 수행을 위해 각 UDDI들의 네트워크

상태를 측정하는 네트워크 모니터링 에이전트를 이용하여 네트워크 상태를 측정하며 이를 바탕으로 동기화 에이전트의 작업 수행경로를 구성하게 된다. 통신 관리 모듈은 UDDI 리스트, 결합 여부 정보, 네트워크 상태 정보등을 유지하는데 UDDI 리스트는 서비스 개발자가 등록을 원하는 공용 UDDI 및 사설 UDDI에 대한 정보를 나타내며, 결합 여부 정보는 각 UDDI의 서비스 제공 가능 여부를 나타낸다. 네트워크 상태 정보는 네트워크 모니터링 에이전트에 의해 측정된 각 UDDI의 네트워크 속도를 나타내게 된다. 작업 수행 경로 프로세서는 위의 정보를 바탕으로 효율적인 동기화 에이전트의 작업 수행 경로를 생성하게 된다. 생성된 작업 수행 경로를 사용하여 등록 및 동기화 에이전트는 지정된 각 UDDI로 웹 서비스의 등록 및 동기화 작업을 수행하게 된다.

4. 결론

UDDI는 웹을 통해서 업체나 서비스의 정보를 쉽게 공개할 수 있으며, 누구든지 정보를 검색하여 이용할 수 있는 환경을 제공한다. 하지만, UDDI v2.0에서는 각 UDDI들 간의 데이터 동기화를 위해 최소 24시간 주기로 복제(Replicaiton)을 수행한다. 이는 서비스 제공자가 등록된 서비스의 수정 발생시 그 결과가 즉시 적용되지 못하며 서비스 제공자가 각각의 UDDI에 등록되어 있는 서비스를 모두 수정하여야 하는 불편함을 초래한다.

본 논문에서 제안된 UDDI 관리 시스템은 서비스 제공자가 등록된 웹 서비스를 수정할 경우 서비스가 등록되어 있는 모든 UDDI의 개수 및 종류에 상관없이 이동 에이전트를 이용하여 신속한 동기화를 수행함으로써 서비스 사용자에게 해당 웹 서비스의 신뢰성을 보장시킨다.

향후 연구로는 웹 서비스의 동기화 작업 수행 시 분산되어있는 에이전트 시스템들의 자원을 공유하여 동기화 작업을 수행함으로써 좀더 효율적인 동기화 작업의 처리가 필요하다.

5. 참고 문헌

- [1] UDDI Technical White Paper, http://www.uddi.org/pubs/UDDI_Technical_White_Paper.pdf
- [2] ECIF, <http://www.ecif.or.kr>
- [3] UDDI Version 2.0 Data Structure Reference, <http://www.uddi.org/pubs/DataStructure-v2.00-Open-200110608.pdf>
- [4] UDDI Executive White Paper, http://www.uddi.org/pubs/UDDI_Executive_White_Paper
- [5] A. Meyer, "UDDI Registries and Reuse", 28 January2002, http://www.e-serv.ebizq.net/wbs/meyer_1.html
- [6] 전문 개발자를 위한 자바 웹 서비스, Pearson Education, 2003
- [7] MSDN Technical Report, http://www.microsoft.com/korea/msdn/library/dn_voices_webservice/html/service10032001.asp
- [8] UDDI4J v2.0 API, <http://www.uddi4j.org>
- [9] 국윤규, 김운용, 정계동, 김영철, 최영근, "분산 DB 환경에서 동기화 이동에이전트", 한국정보처리학회 논문지A, Vol. 10-A, No 4, Oct. 2003.