웹 서비스 기반의 개방형서비스 게이트웨이를 위한 UDDI 구조

이동헌° 김화성
광운대학교 전자통신공학과
{sherpa733@°, hwkim@daisy}kw.ac.kr

An UDDI registry architecture for Parlay X Gateway

Dongheon Lee°, Hwasung Kim
Dept. Electronic and Communications Engineering, Kwangwoon Univ.

요 약

최근 정보통신 환경은 통신, 방송, 인터넷이 통합되는 디지털 융합 서비스 제공 형태로 급속히 변화하고 있다. 이에 따라 통신환경이 개방형 네트워크 형태의 광대역통합망으로 변화하였다. 이런 광대역 통합망에서는 Open API(Application Program Interface)에 의하여 third-party 애플리케이션의 제공을 가능하게 한다. Open API에는 대표적으로 Parlay API가 있으며 Parlay API와 웹 서비스를 접목시키고 한단계 더 추상화한 Parlay X API가 있다. 웹 서비스의 요소중의 하나인 UDDI는 웹 서비스를 쉽게 검색, 공유 및 재사용 할 수 있도록 하는 환경을 가져오게 된다. 본 논문에서는 웹서비스기반의 개방형서비스 게이트웨이를 위한 효율적인 UDDI 레지스트리 구축방안에 대해 제안하였다.

1. 서 론

최근 정보통신환경은 모든 정보통신 기기가 하나의 네트워크에 접속되는 개방형 광대역통합망 (BcN) 기반의 네트워크로 진화할 전망이다.

이런 광대역 통합망에서는 Open API(Application Program Interface)에 의하여 third-party 애플리케이션의 제공을 가능하게 한다. Open API에는 대표적으로 Parlay API가 있으며 최근 Parlay 그룹은 Parlay API와 웹 서비스를 접목시키고 한단계 더 추상화한 Parlay X API 규격을 발표하였다. 웹 서비스(web service)는 " 표준화된 XML 메시지를 통해 네트워크상에서 접근 가능한 연산들의 집합을 기술하는 인터페이스" 로 웹을 통한 시스템 통합을 용이하게 하며, 서로 다른 벤더간의 상호운용이 이점을 가지고 있다. 웹 서비스는 기존의 분산 컴퓨팅 시스템을 대체할 수 있는 새로운 기술로써 주목을 받고 있다.

UDDI(Universal Description Discovery and Integration)는 웹 서비스에 대한 디렉토리 서비스를 지원하기 위해 개발된 표준으로써 웹 서비스를 등록하고 검색/발견하기 위한 메커니즘을 제공한다. 이러한 UDDI 서비스는 웹 서비스 인터페이스와 호환되는 응용 프로그램을 위한 기술 정보, 그리고 실행 시 그 인터페이스에서 성공적으로 바인딩 하는데 필요한 정보가 저장 되어 있으므로 향후 UDDI 서비스의 구현은 개발자들이 응용 프로그램 개발환경에서 웹 서비스를 쉽게 검색, 공유 및 재사용 할 수 있도록 하는 환경을 가져오게 된다.

본 논문에서는 웹 서비스와 Parlay X API에 대하여 알아 보고 개방형 서비스 게이트웨이를 위한 효율적인UDDI 레지스트리 구축방안을 제안하고자 한다.

2. 웹 서비스

웹 서비스는 애플리케이션 개발에서는 최신 기술이며, XML 기술을 기반으로 기존의 웹 환경을 이용한 분산 컴퓨팅을 가능케 함으로써 웹을 통한 시스템 통합을 용이하게 하며 웹 서비스의 모든 데이터 정의는 XML(eXtensible Markup Language)로 표현되며, SOAP(Simple Object Access Protocol)을 이용하여 서비스간 통신을 하게 된다. 또한, 각각의 서비스는 WSDL(Web Service Description Language)을 통하여 표준화된 포맷으로 정의되며, 서비스에 대한 검색은 UDDI(Universal Description Discovery & Interaction)를 통하여 수행된다.[1][2]

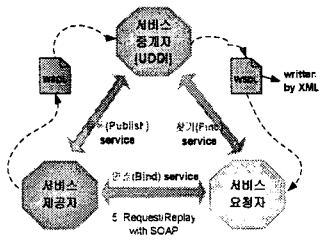

그림1. 웹 서비스의 구조

웹 서비스에는 크게 세 가지 역할이 존재한다.

서비스 제공자, 서비스 요청자, 서비스 중개자 이다. 그림1은 서비스 제공자, 서비스 요청자, 서비스 중개자가 가지는 각각의 오퍼레이션과 그들 사이의 관계를 보여준다. 서비스 제공자는 웹 서비스 형태로 사용자에게 비즈니스 기능을 제공하며 서비스 요청자는 웹 서비스를 사용하는 애플리케이션을 의미한다. 서비스 중개자는 서비스 제공자가 자신의 웹 서비스 상세 내역을 등록하고 서비스 요청자가 원하는 서비스를 발견할 수 있도록 해주는 중앙 저장소를 의미한다. 서비스 제공자는 등록(Publish) 오퍼레이션 과정을 통하여 서비스 중개자의 레지스트리에 자신이 제공할 수 있는 비즈니스 서비스의 명세를 공개 하

게 된다. 서비스 요청자는 찾기(Find) 오퍼레이션을 통하여 서비스 중개자, 즉 UDDI 레지스트리에 등록 되어 있는 서비스 제공자의 명세를 검색 후 원하는 정보를 얻게 된다. 이후에 서비스 요청자는 연결(Bind) 오퍼레이션을 통하여 서비스 제공자와 연결되며 서비스를 이용하게 된다.

3. UDDI 3.0

UDDI 버전 3 이 기존 버전들과 구조상으로 가장 큰 차이를 보이는 것은 개방형 UDDI 와 사설 UDDI 의 광범위한 상호작용을 지원한다는 점이다. 즉, 기업 내 혹은 기업간 수많은 소프트웨어 애플리케이션들의 설계와 운영을 지원하는 웹 서비스 기술의 보다 큰 집합의 중요한 요소로서 UDDI 등장하게 되었다.

UDDI 버전 3 은 다양하고 새로운 분류체계를 구현하기 위해 기존 버전에서 제공되던 기능들에 대한 변경은 최소화하면서 등록키(registry key)생성 및 관리, 등록 관찰 API 집합, XML 디지털 식별자(Digital signature)와 같은 주요 기능들이 추가 되었다.[3]

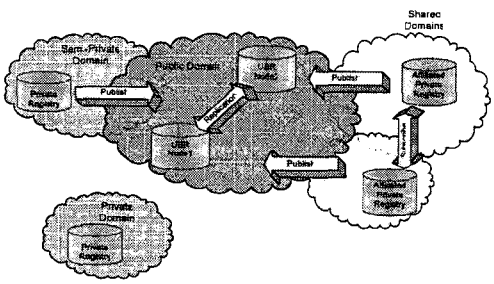

그림 2. UDDI 레지스트리 상호 작용

4. Parlay X API

4.1 Parlay X API

Parlay X API는 텔레포니 애플리케이션의 개발을 가능하게 하기 위하여 설계되었다. Parlay X API는 80%의 애플리케이션이 망 자원의 20%만 이용한다는 80/20 원칙과, 복잡한 Parlay API를 보다 단순하게 정의하자는 KISS(Keep it Simple, Stupid) 원칙에 의해 정의되었다. Parlay X는 현재의 Parlay API의 제공 능력을 블록화 하여 최상위로 상한 API형태를 제공하며, 추상화된 메소드들을 통하여 애플리케이션을 보다 손쉽게 개발할 수 있도록 해준다.[4][5]

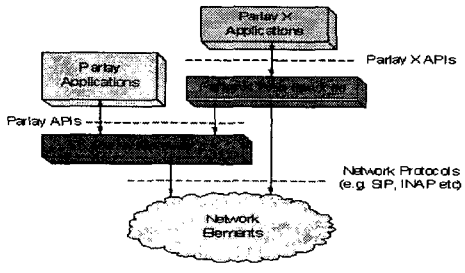

그림 3. Parlay X APIs and Parlay APIs

그림 3에서 Parlay X Web 서비스와 Parlay X API, Parlay API와의 관계를 보여 주고 있다. Parlay X 애플리케이션은 Parlay X API 인터페이스를 통해서 Parlay X Web서비스를 이용할 수 있다. Parlay X Web서비스는 기존의 Parlay API Interface를 이용하여 Parlay Gateway를 거쳐 네트워크에 접속하는 것도 가능하도록 되어 있고, 아니면 직접 네트워크에 직접 접속할 수 있게 되어 있다.

4. 2 Parlay X 게이트웨이 서비스

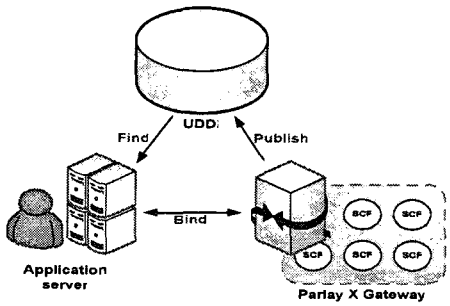

그림 4. Parlay X 게이트웨이 서비스

그림 4는 Parlay X 게이트웨이 서비스를 나타내고 있다. Parlay X 게이트웨이의 서비스라함은 Parlay X의 Open API를 구현한 각각의 SCF(Service Capability Features)들이 제공하는 기능들이며 일반 적인 웹 서비스와는 다르게 웹 서비스 제공자는 웹 서비스 애플리케이션 서버 역할을 하는 게이트웨이를 구현하고, 그 게이트웨이에 Parlay 웹 서비스에서 사용하게 될 인터페이스를 정의한다. 제공자는 웹 서비스 중개자에게 WSDL 파일을 등록하고 웹 서비스 요청자인 Parlay 애플리케이션은 제공자가 등록한 WSDI 파일을 중개자로부터 찾아오게 된다. 마지막으로 Parlay 애플리케이션을 연결하게 되면 서비스 게이트웨이의 프레임워크를 통해 게이트웨이의 인터페이스를 사용할 수가 있게 되어 서비스가 이루어지게 된다.

그러나 그림 4에서와 같이 구축된 환경에서는 개발자가 SCF를 이용해 애플리케이션을 개발하여 실행하던 중에 SCF의 주소변경 및 오류 발생으로 인해 응답이 없을 때 애플리케이션은 다시 정확하게 동작하는 SCF의 주소를 UDDI에 검색 API를 통해 질의하고 얻어진 정보를 통해 다시 애플리케이션을 실행하여 서비스를 제공하게 된다. SCF 동작에 문제가 생겼을 경우, 다시 정확하게 동작하는 SCF를 찾아 애플리케이션을 재 실행하기 전 까지 그 기간 동안은 애플리케이션을 사용하고 있는 사용자들에게 서비스를 제공하지 못하게 되는 문제점이 있다.

5. UAS를 이용한 개방형 서비스 UDDI 레지스트리 구축 모델

위에서 언급한 문제점을 해결하기 위해 UDDI의 기능을 보조할 수 있는 UAS(User Agent Server)를 추가하여 개방형 서비스 게이트웨이를 위한 효율적인UDDI 레지스트리 구축방안을 제안하고자 한다. UAS는 애플리케이션 서버가 사용하고 있는 SCF(Service Capability Features)의 종류 및 서버의 관심 목록 등의 정보를 지니고 있으며 UDDI에 새롭게 공개된 정보 중에 자신이 관리 하고 있는 애플리케이션이 등록한 정보와 관련 있는 정보는 API를 통해 UDDI로부터 그 정보를 Update한다.

또한 자신의 정보를 효율적으로 관리 하기 위하여 데이터 저장소를 두어 정보를 관리한다.

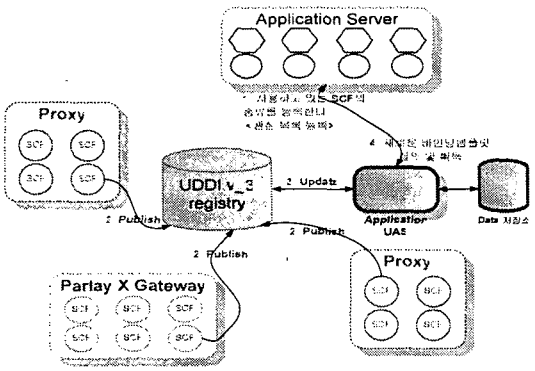

그림 5. UAS(User Agent Server)를 이용한 개방형 게이트웨이 서비스 구조

그림 5는 제안된 개방형 서비스 게이트웨이 상에서 애플리케이션이 실행되는 동작 과정을 나타내고 있다. 동작 절차는 다음과 같다. 먼저, 애플리케이션은 개발 후 자신이 실행되기 전에 자신의 SCF의 유형과 관심목록을 UAS에 등록시킨다. 이후 새로 생성된 SCF의 정보는 Parlay X Gateway를 통해 UDDI 에 등록된다. 이 구조의 UDDI는 UDDI 버전 3을 사용함으로써 개방형 UDDI와 사설 UDDI간의 광범위한 상호작용을 지원할 수 있게 되어 UDDI에 걸리는 과도한 부하를 줄일 수 있다. UDDI에 등록된 정보는 정기적으로 UDDI API를 통해 UAS에 Update되며 Update된 정보는 UAS의 데이터 저장소에 저장되게 된다. 그리고 애플리케이션이 서비스를 제공하던 중에 오류가 발생할 시에는 UAS에 질의를 하고 UDDI로부터 UAS에게 Update된 정보를 받는다. 이 정보를 이용하여 애플리케이션 서버는 서비스를 다시 정상적으로 실행 시킨다.

위의 방법을 사용할 경우 애플리케이션 설계 시 하나의 SCF에 여러 개의 바인딩템플릿이 존재할 때 여러 개의 바인딩템플릿을 자신의 메모리에 저장할 수 있도록 설계해야 한다는 번거로움을 제외하고는 애플리케이션의 오류 발생시 기존의 방법과 같이 UDDI 레지스트리에 질의 할 때보다 단순히 UAS에 준비되어 있는 목록을 사용함으로써 SCF의 바인딩템플릿 Update 정보를 더 짧은 시간 안에 획득할 수 있는 장점이 있다.

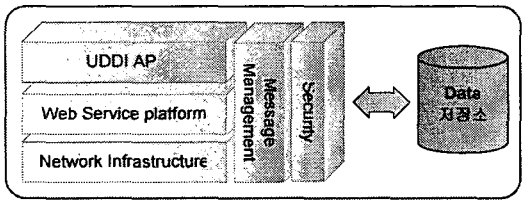

그림 6. UAS의 구조

그림 6은 애플리케이션 서비스가 사용된 후, 서비스 오류가 발생했을 경우에 접속하는 UAS의 구조를 보여 주고 있다. 기본적으로 Network Infrastructure와 Web Service platform을 지니고 있으며 UDDI와 UAS간에 Update 정보를 주고 받기 위해 UDDI API를 지원할 수 있어야 한다. 또한 애플리케이션에게 SCF의 바인딩템플릿 정보를 안정적으로 보내기 위하여 데이터 전송을 관리하는 Message Management를 지니고 있다. 그리고 애플리케이션이 등록했던 정보와 UDDI로부터 받은 Update된 SCF의 바인딩 템플릿 리스트를 저장하고 있어야 하므로 데이터 저장소가 존재한다.

6. 결론

광대역통합망(BcN) 기반에서, 서비스 계층을 통신망의 제어 및 전송 계층으로부터 분리하고, 이들 간에 표준화된 인터페이스(Open API)를 도입하여, 하부 통신망의 구조에 독립적으로 다양한 서비스가 개발될 수 있도록 하는 개방형 서비스 구조가 차세대 통신망의 기본적인 요구 사항으로 제시되고 있다.

Open API 환경에서의 웹 서비스 적용을 위해서는 기본적으로 Parlay X API나, 혹은 Parlay Web Service를 사용하여야 한다. 웹 서비스의 요소중의 하나인 Parlay X 게이트웨이 서비스 환경에서의 UDDI 구현은 웹 서비스를 쉽게 검색, 공유 및 재사용 할 수 있도록 하는 환경을 가져오게 된다.

UDDI를 사용한 망의 구조를 기반으로 애플리케이션 서비스가 제공되고 있을 경우 발생하는 오류를 빠르게 복구하는 방안에 대해 고려하였다. 그리고 이에 대한 해결책으로써 애플리케이션이 관심을 가지는 SCF의 리스트를 관리하고 애플리케이션이 제공하는 서비스 오류가 발생시 기존의 구조보다 빨리 대처할 수 있는 Parlay X 게이트웨이 서비스를 위한 UDDI 레지스트리 구축방안을 제안하였다.

7. 참고 문헌

1. W3C, Web Service Architecture, February 2004
2. W3C, Web Services Description Language (WSDL) Version 2.0 Part 1 : Core Language, March 2004
3. OASIS, UDDI Version 3.0.2 UDDI Spec Technical committee Draft, October 2004
4. Parlay X Web Service Specification, version 1.0, June 2004
5. Parlay Web Service - White Papers, version 1.0, June 2004