웹서비스에서 비 기능적 요소를 고려한 시스템 아키텍쳐에 관한 연구

김철웅, 송영재
*경희대학교 컴퓨터공학과
e-mail:googoos@freechal.com

A study on System Architecture for Considering Non-Functional Properties in Web Service

Kim Chul Ung, Song Young Jae
Graduate school, Kyung Hee University
Dept. of computer engineering, Software Engineering Lab.

요 약

현재 웹상에는 수많은 서비스들이 존재하고 이러한 서비스들은 WSDL, SOAP, UDDI을 기본구조로 하여 서비스를 제공하고 있다. 그러나 급격하게 증가하는 서비스제공자들과 서비스 요청자 사이에서의 최적의 서비스를 제공하는 것에는 많은 어려움이 있었다. 이러한 문제를 해결하기 위해 UDDI는 자신의 Repository에 서비스 제공자에 대한 정보를 등록함으로써 사용자로 하여금 좀 더 용이하게 서비스에 대해 접근할 수 있는 방법을 제시하였다. 그러나, UDDI에는 비즈니스에 대한 기능적인 명세만 있을 뿐 비 기능적 특성에 대한 명세는 제공하지 않았기 때문에 사용자에게 비 기능적인 서비스에 대한 보장을 하지 못했다. 본 논문에서 제안한 QWSDL을 통해 사용자에게 비 기능적인 부분, 즉 웹서비스 성능에 대한 보장을 하였고, 이러한 비 기능적 정보를 Local Database에 서비스 성능평가 기준에 따라 Sorting 알고리즘을 사용하여 저장한다. 사용자는 다중질의를 통해 기능적, 비 기능적 질의를 동시에 하게 되고, 비 기능적인 면의 서비스 성능에 따라 Sorting된 서비스 중 최적의 서비스를 선택함으로써 사용자는 최상의 서비스를 제공받게 된다.

1. 서 론

웹 서비스의 등장으로 인해 웹 서비스 제공자와 사용자는 웹 서비스가 인터넷 상에서 비즈니스 서비스들을 연동시키고, 이기종 코드를 유용한 프로그램으로 통합시키는 세계를 열어줄 것으로 믿고 있다. 그러나 이러한 꿈이 실현되려면 사용자는 광대한 공용 네트워크에 있는 서비스들을 검색(Discovery)하고 발견(Find)할 수 있어야 한다. 이를 위해 서비스를 제공하는 제공자는 웹 서비스를 공용 혹은 사설 UDDI(Universal description, discovery and integration) 레지스트리에 공개를 한다[2]. UDDI 구조는 서비스 요청자로 하여금 간단한 형태의 검색을 가능하게 하며, 또한 서비스 제공자들로 하여금 자사에 대한 데이터를 공개하고 웹 서비스를 광고하도록 함으로써 자발적으로 분류 데이터를 제공하도록 한다.

UDDI 레지스트리는 UDDI 레지스트리가 제공하는 웹 서비스를 포함한 비즈니스를 기술한 정보를 받아들이고, 관심 있는 당사자들이 온라인 검색을 하여 해당 데이터를 다운로드하게 한다.

주문하기 위한 상담을 위해, 거래에 대한 정보-주소, 전화번호, 웹 싸이트, 웹 URL-를 찾는 방법이 필요하다. 거대 대리점, 명함, 직접 기록한 노트나 e-mail로부터 직접 정보를 얻을 수 있다. 또한 전화번호부에서 거래처 이름을 찾아 주소와 전화번호를 알 수도 있다.

웹상에서 누군가의 컴퓨터에 동작하는 프로그램에

애기하기 위해 자신의 컴퓨터에서 동작하는 프로그램에 필요한 정보는 공개되어야 한다. UDDI는 웹 서비스에 대한 개인별 전화번호부나 직업별 전화번호부와 같다.

UDDI는 윈도우의 레지스트리와 마찬가지로 전역 고유식별자(GUID, UUID)를 사용해 검색 수행 기능을 제공하거나, 혹은 각각의 페이지(White pages, Yellow pages, Green pages)형태로 구분된 레지스트리에 분류되어 있는 비즈니스 이름, 주소, 연락처 정보, 지리 위치, 산업 타입, 비즈니스 ID, 비즈니스 프로세스 정의등에 의한 방법에 따라 검색 수행 기능을 제공한다[2].

그러나 사용자들의 요구사항이 단지 “검색”이라는 기준을 넘어서 “서비스의 품질”에 관련된 사항들에 대한 요구사항이 증가하게 되었다[4]. 그럼에도 불구하고 UDDI에서 제공하는 기능적 정보- 비즈니스 이름, 주소, 연락처 정보, 지리 위치, 산업 타입-에 의한 검색만으로는 사용자의 요구사항을 만족시켜주지 못하고 있는 것이 현실이다.

본 논문에서는 Q-WSDL을 통해 사용자의 비 기능적 요구사항-비용(Cost), 응답시간(Response Time), 안정성(Stability), 이용가능성(Available)등-에 대한 질의까지 사용자 다중질의를 통해 허용함으로써 사용자가 기존에 제공되던 기능적 명세뿐 아니라 비기능적 명세 또한 가능하게 했다. 이러한 다중질의를 통해 얻어진 정보를 Local Database에 각 기준에 따른 서비스 성능에 따라 Sorting 되어서 저장되게 된다. 사용자는 각각의 Sorting 된 정보를 기능적, 비기능적 정보를 통한 질의를 함으로써 기능적, 비기능적으로 사용자 질의에 가장 적합한 서비스를 제공받는것이 가능하게 하고자 한다.

2. 관련 연구

2.1 웹 서비스 아키텍쳐

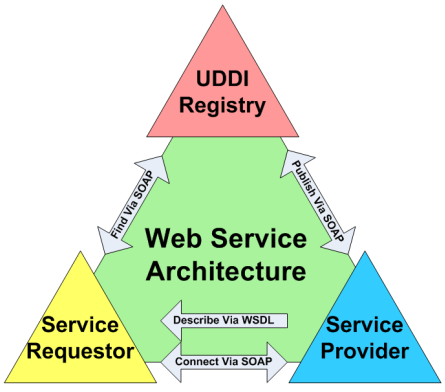

그림1. 웹 서비스 아키텍쳐

웹 서비스의 동작과정은 세 가지 역할과 그 역할간의 세 가지 상호작용으로 요약할 수 있다. 즉, 서비스 제공자는 서비스를 개발해 그 상세내역(description)과 함께 서비스를 공개(Publish)하고 서비스 요청자는 원하는 서비스를 검색(Find)한 후, 자신의 웹 서비스나 애플리케이션에 바인딩(binding)하는 과정을 거치게 된다.

2.2 웹 서비스 핵심 기술

XML(eXtensible markup Language)

웹 서비스에서 XML은 가장 중요한 기반 기술이다. XML을 이용하면 기업 내 서로 다른 시스템과 애플리케이션 간에 정보를 공유할 수 있고, 나아가 기업간의 각종 데이터의 교환이 가능하다. 즉, IT의 공통어와 같은 개념이다.

SOAP(Simple object Access Protocol)

SOAP는 일종의 XML 메시징을 지원하는 HTTP에 대한 확장이다. 브라우저에서 HTML 페이지를 다운로드하거나 표시되게 요청하기 위해 HTTP를 사용하지 않고, SOAP는 HTTP요청을 통해 XML 메시징을 보내고 HTTP 응답을 통해 응답을 받는다. SOAP는 웹 서비스 구현에 도달하기 위해 웹과 다른 타입의 네트워크상에 XML 문서를 전송한다.

WSDL(Web Service Description Language)

WSDL은 기업에서 UDDI에 등록된 웹 서비스를 어떻게 찾고 알아볼 수 있는지에 관한 표준 XML vocabulary다. 웹 서비스를 기술하는 스크립트인 WSDL은 XML포맷으로 구성되고 HTTP를 통해 전달될 수 있으며 인터페이스를 정의하는 IDL에 해당한다. WSDL을 이용해 웹 서비스 제공자는 사용자에게 해당 웹 서비스의 정확한 인터페이스와 사용되는 데이터 타입, 전송 프로토콜에 대한 상세 정보를 전달할 수 있다.

UDDI(Universal Description, Discovery and Integration)

UDDI는 인터넷 상의 전 세계 비즈니스 목록에 자신을 등재하기 위한 XML 기반의 레지스트리로써 각 기업들이 웹 상에서 서로를 찾을 수 있도록 함으로써 온라인 트랜잭션을 간략하게 하여 웹 서비스 상에서의 정보검색에 대한 정확성과 효율성을 높이려는 궁극적인 목표를 가지고 있다.

2.3 AUSE(Advanced UDDI Search Engine)

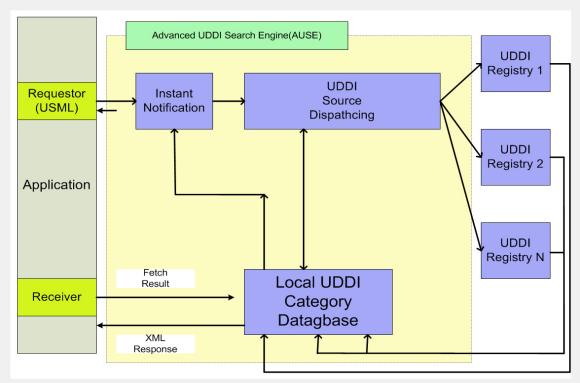

그림2. AUSE의 시스템 아키텍쳐

AUSE(Advanced UDDI Search Engine)는 UDDI 2.0에서 제안된 모델로써 사용자는 USML(UDDI Search Markup Language)을 통하여 서비스 요청을 하게 되고, AUSE는 각각의 멀티 UDDI레지스트리를 검색하여 해당 정보를 가져온 후, Information Aggregation Broker를 통해 정보를 통합 결과값을 리턴받게 된다[5]. 이때, UDDI의 저장위치 정보는 Local UDDI Database에 저장되게 된다. 이 Database는 검색요청시마다 업데이트되며, 이 업데이트 된 정보를 검색 요청시마다 사용하게 된다[1].

3. Q(Quality)-WSDL을 이용한 사용자 다중질의

3.1 사용자 다중질의

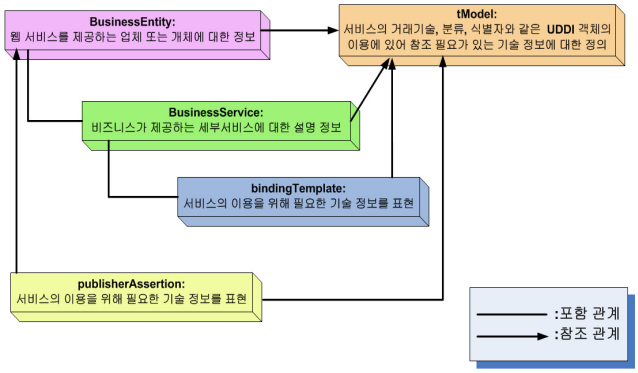

그림3. UDDI 데이터 구조

그림3에서 보듯이, UDDI 데이터 구조는 위와 같은 구조를 이루고 있고, 이중 최상위에 속하는 BusinessEntity는 비즈니스 및 서비스에 대한 특성을 나타내는 데이터들을 총칭한 것으로, 아래와 같이 분류할 수 있다.

구 분	내 용
White pages	이름, 주소, 전화번호, 메일 등 entity 의 일반정보
Yellow pages	분류 및 식별체계에 대한 정보
Green pages	서비스 호출방법, 즉, binding에 대한 정보

표1. BusinessEntity의 데이터 분류

그러나, 이러한 데이터 분류기준을 토대로 UDDI를 검색할시 문제점을 갖게 되는데, 이는 서비스 사용자에게 비용, 응답시간, 안정성, 이용가능성 등 비기능적인면의 서비스에 대한 정보를 제공하지 못한다는 단점을 가지고 있다[3].

```
<!ELEMENT Source (#PCDATA)>
<!ELEMENT SourceURL (#PCDATA)>
<!ELEMENT BusinessName (#PCDATA)>
<!ELEMENT Identifier (#PCDATA)>
<!ATTLIST Identifier type (D-U-N-S|ThomasRegister) #REQUIRED>
<!ELEMENT Category (#PCDATA)>
<!ATTLIST Category type (NAICS|UNSPSC|GEO|UDDITYPE|SIC|types) #REQUIRED>
<!ELEMENT ServiceName (#PCDATA)>
<!ELEMENT ServiceTypeName (#PCDATA)>
<!ELEMENT DiscoveryURL (#PCDATA)>
```

그림4. 기능적 요소만을 제공하는 USML DTD

이러한 문제점을 해결하고자 사용자 다중질의를 제안하였다. 사용자 다중질의는 사용자로 하여금 현재 제공되고 있는 기능적인 서비스 검색 외에 UDDI에 Q-WSDL을 제안함으로써 사용자에게 비 기능적인 면까지 고려하여 서비스를 제공하는 것을 가능하게 한다.

```
<!ELEMENT SourceURL (#PCDATA)>
<!ELEMENT BusinessName (#PCDATA)>
<!ELEMENT Identifier (#PCDATA)>
<!ATTLIST Identifier type (D-U-N-S|ThomasRegister) #REQUIRED>
<!ELEMENT Category (#PCDATA)>
<!ELEMENT Cost (#PCDATA)>
<!ELEMENT Response Time (#PCDATA)>
<!ELEMENT Stability (#PCDATA)>
<!ELEMENT Available (#PCDATA)>
<!ELEMENT ServiceName (#PCDATA)>
<!ELEMENT ServiceTypeName (#PCDATA)>
<!ELEMENT DiscoveryURL (#PCDATA)>
```

그림5. 비 기능적 요소를 추가한 USML DTD

이때 사용자에게 제공되는 비 기능적인 정보는 서비스 성능에 따라 사용자에게 최상의 서비스부터 제공하게 된다.

3.2 서비스 성능에 따른 Sorting

서비스 요청자는 어플리케이션을 이용해 기능적인 면과 비 기능적인 면을 동시에 질의할 수 있다고 가정한다. 본 논문에서는 사용자가 "WEB"라는 비즈니스 이름을 갖고 "Cost"와 "Response Time"측면에서 최상의 값을 갖는 서비스를 검색한다고 가정한다.

```
<Query>
<Source>IBM_UDDIv2</Source>
<SourceURL>http://www-3.ibm.com/services</SourceURL>
<BusinessName>WEB/BusinessName>
<Category type="NAICS">33461</Category>
<Identifier type="D-U-N-S">00-136-8083</Identifier>
<Cost> ... </Cost>
<Response Time> ... </Response Time>
<DiscoveryURL>http://www.ibm.com/</DiscoveryURL>
</Query>
<AggOperator>OR</AggOperator>
```

그림6. 비 기능적 속성에 대한 질의

그 결과는 아래와 같다.

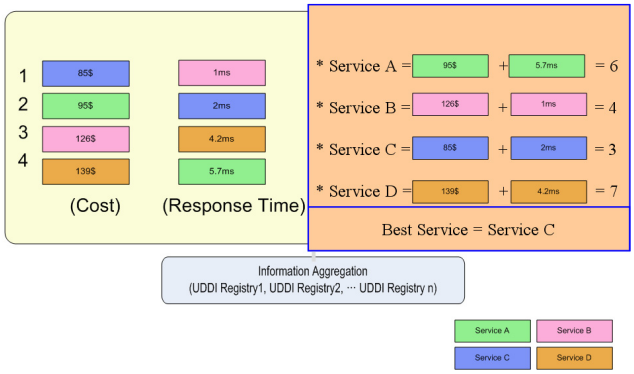

그림7. 비 기능적 속성 질의 결과

서비스 제공자는 위의 서비스질의 조건에 따라 "WEB"라는 비즈니스 네임을 갖는 서비스들은 검색하였으며, 검색된 각각의 서비스들은 각각의 서비스 성능 기준에 따라 분류, sorting 되었다. 그리고 서비스 요청 기준에 따라 가장 합당한 서비스는 "Service A"라 할 수 있겠다.

비 기능적인 분류에 따라 정렬된 서비스를 구현한 시스템 전체 아키텍쳐를 보면 아래와 같다.

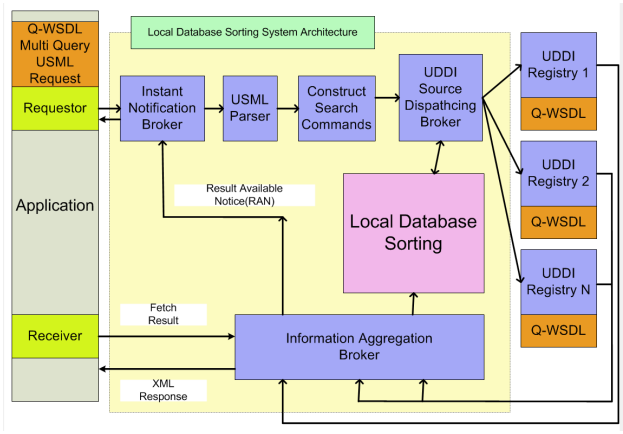

그림8. Local Database Sorting System Architecture

비 기능적 서비스는 UDDI 레지스트리에 Q-WSDL에 의하여 정의되었고, 요청자는 정의된 Q-WSDL을 사용해 기능적, 비 기능적 질의를 동시에 수행하는 다중질의를 수행하였다. 검색된 서비스는 Local Database에 정렬되어 저장되고, 검색된 서비스 중 요청자의 기준으로 검색한 사용가능한 서비스 중에서 가장 좋은 성능을 가진 서비스가 요청자에게 리턴 되었다.

4. 결론 및 향후 연구 과제

본 논문에서는 UDDI에 비 기능적 속성을 정의하는 Q-WSDL을 정의하였고, 이로 인해 사용자에게 기능적, 비 기능적 속성을 동시에 질의할 수 있는 다중질의를 가능하게 하였다. 이로 인해 요청자는 웹서비스 검색 시 요청자 기준에 따른 서비스 품질의 기준을 정하여 기준에 맞는 사용가능한 최상의 서비스를 제공받을 수 있을 것으로 기대된다.

그러나 본 논문에서 제시한 Q-WSDL에 관한 상세한 명세와 기술이 좀 더 필요한 상황이다. 또한, 서비스 성능 평가에 기준이 되는 요소들의 명확한 평가의 기준과 기준요소 선택의 결정기준이 요구된다. 마지막으로 비 기능적 요소들을 평가하는 Sorting 알고리즘에 대한 구체적인 설계를 구현하는 것이 향후의 과제라고 보여 진다.

참 고 문 헌

[1]Liang-Jie Ahang, Haifei Li, Henry Chang,"XML based Advanced UDDI Search Mechanism for B2B Integration", IBM T.J Watson Research Center, Route 134, Yorktown Heights, Ny 10598.

[2]"Web Services And UDDI", IBM Corporation.

[3]Liang-Jie Zhang, Haifei Li, Henry Chang, "Business Explorer for Web Services, downloadable software package on IBM", http://www.alphaworks.ibm.com/tech/be4ws, Dec, 2001.

[4]한국전산원,"Quality Models and Test Guidelines for Web Services Management", Dec, 2004

[5]Web Service Invocation Framework (WSIF), http://www.alphaWorks.ibm.com, 2001