

# Web Service 에서의 QoS 지원방안에 관한 연구

\*이홍준, \*박상원, \*정성호, \*\*이강찬, \*\*전중홍  
\*한국의국어대학교 컴퓨터및정보통신공학부  
\*\*한국전자통신연구원

\*{donudory, swpark, shjeong}@hufs.ac.kr, \*\*{chan, hollobit}@etri.re.kr

## A Study on QoS Support for Web Services

\*Hongjun Lee, \*Sangwon Park, \*Seong-Ho Jeong, \*\*Kangchan Lee, \*\*Jonghong Jeon  
\*Dept. of Computer Science & Information Communications Engineering, HUFS  
\*\*Protocol Eng. Center, Electronics and Telecommunications Research Institute

### 요 약

웹 서비스 기술은 계속 발전하고 있으나, 현재의 인터넷 환경에서는 웹 서비스의 QoS 를 보장하기 어렵다. 본 논문에서는 기본 웹 서비스 구조를 분석하고, 웹 서비스에서 QoS 를 보장할 수 있도록 기존의 웹 서비스 구조를 확장한 웹 서비스 구조를 제시하며, 웹 서비스의 주요 기술들인 WSDL, UDDI, SOAP 의 확장된 모델을 제안한다. 아울러 이 모델을 이용한 응용으로서 SLA 정보를 이용한 웹 서비스에서의 QoS 지원 방안을 제안한다.

Key Words: Web Services, Quality of Services, WSDL, UDDI, SOAP, SLA

### 1. 서론

웹 서비스의 표준화를 진행하고 있는 W3C(World Wide Web Consortium)[1]의 표준문서에 따르면 웹 서비스는 URI(Uniform Resource Identifier)에 의해 식별되는 소프트웨어 애플리케이션으로, XML 에 의해 애플리케이션 인터페이스 및 바인딩이 정의되고 서술될 수 있으며, 다른 소프트웨어 시스템들에 의해 검색될 수 있다. 또한 인터넷 기반의 프로토콜을 경유한 XML 기반의 메시지를 사용하는 다른 소프트웨어 애플리케이션과 직접적인 상호운용이 가능한 시스템을 웹 서비스라고 정의하고 있다. 이러한 웹 서비스는 기존에 존재하던 분산객체기술들을 대체할 새로운 통합 기술로 주목받아 개발되고 있으나, 아직 상호운용성, 보안, QoS(Quality of Service)등의 문제들을 해결하지 못한 상태이며, 계속 연구 중에 있다. 그 중 본 논문에서 다루고자 하는 웹 서비스에서의 QoS 보장은 앞으로 많은 연구를 필요로 하는 주요 분야이다. 특히, 웹 서비스는 현재 QoS 가 보장되지 않고 Best-Effort 서비스만을 제공하는 인터넷에서 제공되어야 하기 때문에 QoS 의 보장이 필요한 여러 서비스들에 대한 QoS 제공 방안을 마련하는 것이 시급한 실정이다.

본 논문에서는 웹 서비스에서 QoS 를 제공하기 위해 확장된 웹 서비스 구조를 제시하고, SLA(Service Level Agreement)를 이용하여 웹 서비스의 주요 기술들인 WSDL, UDDI, SOAP 의 기능을 확장한 QoS 제공 방안을 제시한다.

### 2. 웹 서비스 구조

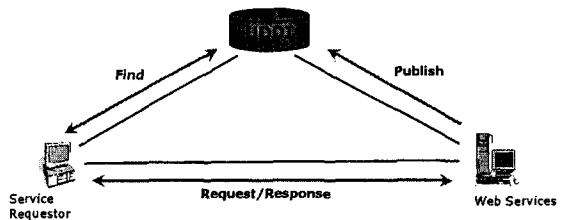


그림 1. 기본 웹 서비스 모델

위의 그림 1 은 서비스 요청자, 서비스 제공자, UDDI 등의 요소들로 구성된 기본적인 웹 서비스 모델을 보여주고 있다[1].

서비스를 검색하고 요청하는 서비스 요청자, 서비

스를 제공하고 레지스트리에 등록하는 서비스 제공자, 서비스를 등록해 주고 서비스 요청자의 검색을 가능하게 하는 UDDI 가 존재하여 기본적인 웹 서비스를 수행할 수 있게 한다. 그러나 현재 이러한 기본적인 웹 서비스 모델에서는 QoS 를 제공하기 위한 어떠한 방안도 마련되어 있지 않아 서비스 요청자가 요구하는 QoS 를 보장하기는 어렵다.

아래의 그림 2 는 기본 웹 서비스 모델에서 QoS 를 제공하기 위해 확장된 웹 서비스 모델을 나타내고 있다[2].

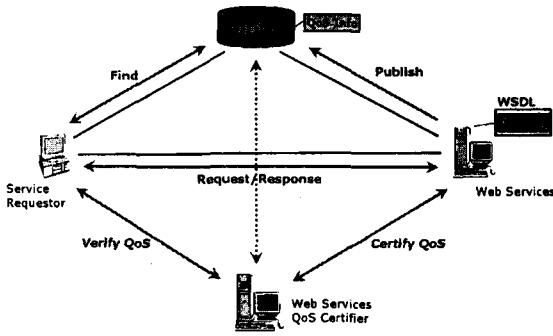


그림 2. 확장 웹 서비스 모델

확장된 웹서비스 모델에서는 기존 웹 서비스 모델에 새로운 요소인 QoS Certifier 와 확장된 UDDI 가 포함된다. QoS Certifier 는 서비스 제공자가 제공해 줄 수 있는 QoS 를 검증해 주는 역할을 하는데, 서비스 제공자는 서비스에 관한 명세를 담은 WSDL 을 UDDI 에 등록하기 전에 서비스에서 제공해 줄 수 있는 QoS 정보를 먼저 QoS Certifier 를 통해 검증 받고 그 결과로 고유한 식별키를 받게 된다. 그 후 서비스 제공자는 UDDI 레지스트리에 WSDL 과 QoS 정보를 등록하게 된다. 검증 받은 서비스에 대한 QoS 정보와 식별키는 QoS Certifier 의 repository 에 저장된다.

UDDI 에서도 서비스 제공자가 제시한 QoS 정보가 맞는 지를 QoS Certifier 에 문의한 후 최종적으로 UDDI 레지스트리에 저장하게 된다. 그러나, 현재의 UDDI 명세는 QoS 를 고려하지 않고 설계되어 있기 때문에 QoS 정보를 관리할 수 있도록 기능의 확장이 필요하다. 다음 절에서는 UDDI 에서 QoS 를 지원하기 위해 어떻게 UDDI 를 확장할 것인지를 논의하고, 그 방안을 제시한다.

### 3. UDDI 의 확장

UDDI 는 OASIS(Organization for the Advancement of Structured Information Standards)에서 표준화를 진행하고 있는데, 서비스 제공자와 서비스에 대한 정보만을 저장하는 현재의 명세로는 2 절에서 설명했던 QoS 지원을 위한 확장된 웹 서비스 모델을 지원할 수 없기 때문에 UDDI 의 기능 확장이 필요하다. 기존의 UDDI 명세에 있는 데이터 구조는 다음과 같다[3].

- businessEntity: 서비스에 대한 정보를 등록한 서비스 제공자에 대한 정보를 관리한다.
- businessService: 서비스에 대한 비 기술적인 정보를 관리한다.
- bindingTemplate: 서비스에 대한 기술적인 정보를 관리한다.
- tModel: 쉽게 정보에 접근할 수 있게 만든 기술적인 모델정보.

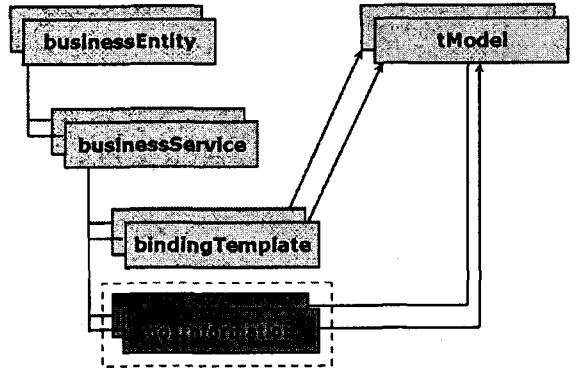


그림 3. 확장 UDDI 데이터 구조

그림 3 은 위에서 설명한 기본 UDDI 구조에 QoS 정보를 관리할 수 있도록 qosInformation 이라는 새로운 데이터 구조를 businessService 구조의 하부에 정의한 모습을 나타내고 있다. qosInformation 이 businessService 의 하부에 들어 간 이유는 각 서비스에서 제공할 수 있는 QoS 요소가 다르므로 서비스를 관리하는 businessService 에 들어가야 하기 때문이다.

qosInformation 구조에서는 서비스 제공자가 등록된 QoS 정보를 관리하는데, 아래 그림 4 에서 나타내고 있는 QoS 요소들이 포함되어 서비스 요청자에게 QoS 정보를 제공할 수 있다.

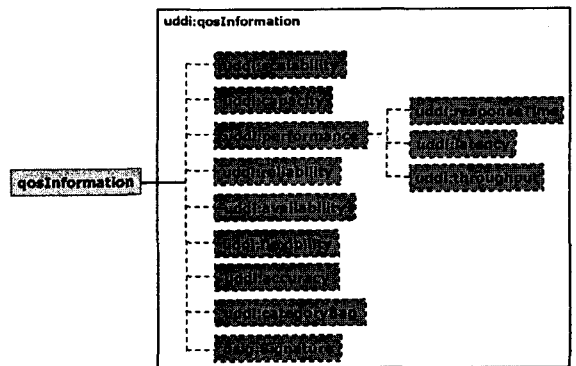


그림 4. qosInformation 데이터 구조

qosInformation 에 포함될 수 있는 QoS 요소들은 다음과 같은 것들이 있다[4].

- Scalability: 사용자의 요구에 맞추기 위해 크기나 용량을 변경해도, 그 기능이 계속하여 잘 동작할 수 있는 능력.
- Capacity: 동시에 처리할 수 있는 처리용량.
- Performance: 성능은 response time, latency, throughput 등으로 측정될 수 있는 측정 요소.
- Reliability: 신뢰성은 서비스와 서비스 품질을 유지하는 능력으로 서비스 요청자와 제공자가 주고 받은 메시지 전달의 정확도 등으로 평가.
- Availability: 가용성은 서비스를 사용할 수 있다는 가능성을 나타내는 요소.
- Flexibility: 잘못된 입력이나 오류가 존재해도 서비스가 올바른 기능을 수행할 수 있는 능력.
- Accuracy: 서비스 제공 시 발생하는 에러율.

위의 요소들 이외에도 Exception handling, 트랜잭션과 관련된 Integrity 등이 포함될 수 있다.

4. WSDL에서의 QoS 명세

WSDL(Web Services Description Language)은 웹 서비스로의 접근 방법, 서비스, 서비스타입 및 프로토콜, 데이터 타입들을 상세하게 정의하는 언어로 웹 서비스 표준기술이다. 다시 말해서 서비스에 대한 정보를 기술하는 언어인데, 서비스 제공자는 UDDI에 등록하기 위해 WSDL로 서비스에 관한 정보를 기술하게 된다[5].

서비스 제공자는 UDDI에 등록할 QoS 정보를 얻기 위해 XML Script Language로 작성된 Mapper를 이용하는데 Mapper는 아래 그림 5와 같이 SLA의 QoS 파라미터들로부터 QoS 정보를 추출하기 위해 사용된다.



그림 5. Mapper

SLA란 서비스 제공자와 서비스 요청자 사이에 체결된 계약을 의미하는데, Best-effort 서비스만을 제공하는 인터넷 환경에서 서비스 제공자가 서비스 요청자에게 QoS를 포함하여 어느 정도의 서비스를 제공할 수 있는지를 명시하여 이를 보장해 준다. 서비스 요청자는 원하는 서비스를 제공할 수 있는 서비스 제공자와 SLA를 맺음으로써 QoS를 보장받게 된다[6].

그림 5의 과정을 통해 추출된 QoS 정보들은 아래 그림 6과 같이 UDDI에 등록하기 위해 QoS 정보를 표현할 수 있게 확장된 WSDL 스키마를 이용해 WSDL로 작성된다. 그림 7은 확장된 WSDL 스키마를 이용해 작성한 WSDL 파일의 예를 보여주는데 기존의 WSDL 파일과 달리 새로 qosInformation이 추가되어 QoS 정보들을 기술하고 있음을 알 수 있다.

```

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">
      QoS Schema for WSDL Extension
    </xsd:documentation>
  </xsd:annotation>
  <xsd:simpleType name="Scalability">
    <xsd:restriction base="xsd:string">
      <enumeration value="TRUE"/>
      <enumeration value="FALSE"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name="Capacity">
    <xsd:restriction base="xsd:int"/>
  </xsd:simpleType>
  <xsd:complexType name="Performance">
    <xsd:element name="ResponseTime" type="xsd:float"/>
    <xsd:element name="Latency" type="xsd:float"/>
    <xsd:element name="Throughput" type="xsd:float"/>
  </xsd:complexType>
  .
  .
  .
  
```

그림 6. 확장 WSDL 스키마

```

<?xml version="1.0" encoding="UTF-8" ?>
<definitions name="FooSample"
  targetNamespace="http://tempuri.org/wsd/"
  xmlns:wdslns="http://tempuri.org/wsd/"
  xmlns:typens="http://tempuri.org/xsd"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns="http://schemas.xmlsoap.org/wsd/"
  xmlns:wslQoS="http://cclab.hufs.ac.kr/wslQoS/">
  <wslQoS:qosInformation>
    <wslQoS:Capacity>1000</wslQoS:Capacity>
    <wslQoS:ResponseTime>2</wslQoS:ResponseTime>
    <wslQoS:Latency>0.5</wslQoS:Latency>
    <wslQoS:Throughput>1</wslQoS:Throughput>
    <wslQoS:Availability>98.9</wslQoS:Availability>
    <wslQoS:ExceptionHandling>TRUE</wslQoS:ExceptionHandling>
  </wslQoS:qosInformation>
  <types>
    .
    .
    .
  
```

그림 7. WSDL 예제

위의 그림 6에서는 QoS 정보를 WSDL에 기술할 수 있게 확장된 WSDL 스키마를 보여주고 있는데, 각 QoS 파라미터들을 기술할 수 있도록 앞의 3절에서 설명했던 각 QoS 요소들과 요소들의 데이터 타입 및 인자들을 XML 스키마로 정의하고 있다. 또, 위의 그림 7에서는 그림 6의 WSDL 스키마를 이용하여 실제로 서비스를 UDDI에 등록하기 위해 WSDL 파일을 만든 예를 보여주고 있는데, qosInformation 요소내의 인자로 capacity가 1000, responseTime이 2, Latency가 0.5, throughput이 1, availability가 98.9, ExceptionHandling은 TRUE라고 기술되어 있다. 이는 서비스 제공자측에서 제공할 수 있는 QoS의 정도를 나타내며, 사용된 각 QoS 관련 인자들은 위 그림 6의 확장된 WSDL 스키마 파일에 정의되어 있다.

5. 서비스 검색을 위한 SOAP 요청, 응답 메시지

서비스 제공자는 QoS Certifier로부터 검증 받은 QoS 정보를 4절의 그림 7과 같은 WSDL 파일로 만들어 UDDI에 등록한다. 이렇게 UDDI에 등록된 서비스

의 QoS 정보는 UDDI 레지스트리에 저장되어 있다가 서비스 요청자의 검색 요청에 대한 결과로 서비스 요청자에게 제공된다.

아래의 그림 8 과 9 는 각각 SOAP 을 이용한 서비스 요청자의 검색 요청 메시지와 UDDI 로부터의 결과 응답 메시지를 보여주고 있다.

```
<?xml version="1.0" encoding="UTF-8" ?>
<envelope xmlns="http://schemas.xmlsoap.org/soap/envelope/">
  <body>
    <find_service businessKey="*" generic="1.0" xmlns="urn:uddi-org:api">
      <name>Hotel Reservation</name>
      <qosInformation>
        <ResponseTime>2</ResponseTime>
        <Availability>98.9</Availability>
      </qosInformation>
    </find_service>
  </body>
</envelope>
```

그림 8. SOAP 요청 메시지

```
<?xml version="1.0" encoding="UTF-8" ?>
<envelope xmlns="http://schemas.xmlsoap.org/soap/envelope/">
  <body>
    <serviceList generic="1.0" xmlns="urn:uddi-org:api"
      operator="www.ibm.com/services/uddi" truncated="false">
      <serviceInfos>
        <serviceInfo serviceKey="9021cb6e-e8c9-4fe3-9ea8-3c99b1fa8bf3"
          businessKey="b42b5fef-85df-4bf1-b468-62a356089ea8">
          <name>Hotel Reservation</name>
          <qosInformation>
            <ResponseTime>2</ResponseTime>
            <Availability>99.9</Availability>
          </qosInformation>
        </serviceInfo>
        <serviceInfo serviceKey="74154900-f060-11d5-bca4-002035229c64"
          businessKey="b6cb1cf0-3aaf-11d5-80dc-002035229c64">
          <name>Hotel Reservation</name>
          <qosInformation>
            <ResponseTime>1</ResponseTime>
            <Availability>99.1</Availability>
          </qosInformation>
        </serviceInfo>
      </serviceInfos>
    </body>
  </envelope>
```

그림 9. SOAP 응답 메시지

그림 8 의 SOAP 요청 메시지에는 response time 이 2ms 이내이고, availability 가 98.9%이상인 Hotel reservation 서비스를 찾고 있음을 보여주고 있다.

그림 9 의 UDDI 로부터의 SOAP 응답 메시지에는 SOAP 요청 메시지에 기술되어 있는 QoS 등급을 만족하는 서비스들의 목록과 response time, availability 에 대한 정보, 제공자의 식별코드, 서비스 식별코드가 서비스 요청자에게 검색요청에 대한 응답으로 되돌아 온 것을 보이고 있다.

## 6. 결론

본 논문에서는 확장된 웹 서비스 모델을 제시하여 웹 서비스에서 QoS 가 보장될 수 있다는 것을 보여 주었다. 확장된 웹 서비스 모델을 적용하기 위해 기본 웹 서비스 모델에 QoS Certifier 를 포함하였으며, QoS 정보를 관리하기 위해 기존의 UDDI 명세를 확장하였다. 또한 서비스 제공자 측에서 제공해 줄 수 있는 QoS 정보를 기술하여 UDDI 에 등록하기 위해 기존의 WSDL 스키마를 확장하였고, 서비스 요청자 측에서는 확장된 WSDL 스키마를 이용해 QoS 등급에 맞는 서비스를 검색할 수 있게 하였다.

향후에는 UDDI 확장과 관련하여 데이터 구조의 확장뿐 아니라 QoS 정보를 검색하고 처리할 수 있도록 UDDI API 의 확장이 필요하다. 아울러 QoS Certifier 가 QoS 검증을 위해 서비스 요청자, 서비스 제공자, UDDI 등과 같은 웹 서비스 모델 내의 다른 요소들과의 통신방법에 대한 연구가 구체적으로 이루어져야 할 것이다.

## 참고문헌

- [1] W3C, "Web Services Architecture", W3C Working Draft, May 2003.
- [2] Shuping Ran, "A Model for Web Services Discovery With QoS", ACM SIGecom Exchanges, 4(1), Mar. 2003.
- [3] IBM, Microsoft, HP, Oracle, Intel, SAP, "UDDI Version 3.0 Published Specification", <http://www.uddi.org>, July, 2002.
- [4] A. Mani, A. Nagarajan, "Understanding Quality of Service for Web Services", IBM developerWorks, Jan. 2002.
- [5] W3C, "Web Services Description Language(WSDL) Version 1.2", W3C Working Draft, Mar. 2003.
- [6] Judith M. Myerson, "Guarantee your Web Service with an SLA", IBM developerWorks, April, 2002.