

# BPEL4WS를 이용한 통합 관점의 웹서비스 품질 관리 시스템

정승준\*, 국윤규\*, 김운용\*, 최영근\*

\*광운대학교 컴퓨터과학과

e-mail:excepter@kw.ac.kr

## Quality Management System on Integrated Web Services using BPEL4WS

Rudy Jung\*, Yoon-Gyu Kook\*, Woon-Yong Kim\*,  
Young-Keun Choi\*

\*Dept of Computer Science, Kwang-Woon University

### 요 약

웹 서비스는 SOAP, WSDL, UDDI와 같은 표준을 기반으로 인터넷상의 분산된 여러 서비스들을 통합하는 가장 효과적인 방법으로 알려져 있다. 이러한 통합을 위하여 웹 서비스 표준 프로세스 통합모델 언어인 BPEL4WS가 제시되었다. 이와 같은 분산된 서비스 통합 시에 서비스 신뢰성 확보를 위해서는 각각의 웹서비스들의 품질 관리가 필요하다. 이에 본 논문에서는 각 웹 서비스들의 품질을 관리 할 수 있고, 품질을 분석하고, 분석 후에는 저 품질의 서비스는 원활한 서비스 제공을 위해 대체 하는 시스템을 제안한다. 또한 본 시스템에서 사용되는 품질 보증 정책과 저 품질 웹 서비스 교체 방법 및 전체 시스템의 프레임워크를 제안한다.

### 1. 서론

웹의 등장과 급속한 팽창으로 인해서, 각각의 서비스들은 좀 더 다양하고 복잡한 서비스를 제공하게 되었다. 이에 따라서 기존의 하나의 서버가 모든 일을 하는 시스템보다는 전문화된 여러 서비스들을 통합하여서 하나의 복잡한 서비스를 제공하는 시스템의 구성의 필요성이 생겨났다. 이를 구현하기 위한 표준으로써 웹 서비스가 제시되었다.[1] 이기종 간의 데이터 통합을 위한 XML과 전송 통합을 위한 HTTP 기반 위에서, 새로이 만들어진 SOAP[2], WSDL[3] 그리고 UDDI[4] 표준으로써 웹 서비스가 분산된 다 기종의 효율적인 통합을 가능하게 되었다.

그리고 이러한 웹 서비스들의 보다 표준화 되고 효과적인 통합을 위해서, 표준 프로세스 통합모델 언어인 BPEL4WS (Business Process Execution Language for Web Service)[5]가 제시되었다. BPEL4WS는 이미 구축되어있는 여러 웹 서비스들을 표준적으로 어떻게 통합을 하여서, 하나의 새로운 서비스를 만들 것인가를 기술한 모델 언어로써, 웹 서비스 통합 표준 언

어이다.

이러한 통합된 서비스의 가장 큰 문제점 중 하나가 신뢰성의 확보이다. 사용자 입장에서의 신뢰성은 서비스가 일정 이상의 품질 보증이 되어야지만, 확보 되는 것이다. 이러한 신뢰성 확보를 위해서는 해당 서비스 뿐만 아니라 서비스에 포함된 개개의 서비스들까지도 품질을 보증 해주어야한다.

통합 환경의 웹 서비스의 품질 보증 방법으로는 대표적으로 SLA(Service Level Agreement)를 이용한 방법이 있었다.[8] 이 방법은 사용자와 서비스 제공자 사이의 정해놓은 지표에 의한 보증 방법으로써는 효과적이지만, 웹 서비스의 품질이 저하 되었을 경우에 대응할 마땅한 방법이 없었고, SLA 협정을 맺지 않은 경우는 품질을 관리 하는 부분이 미흡하였다.

이에 본 논문에서는 웹 서비스의 품질을 측정하고 기록하여서, 그 결과를 통계 내고 분석해서 제공 해 주는 시스템을 제안한다. 본 시스템에서 쓰이는 품질 측정 기준이 담긴 문서를 PoQ(Policy of Quality) 문서라고 규정하였다. 또한 측정된 값을 분석하여서, 전체

서비스의 품질을 떨어뜨릴 소지가 있는 웹 서비스를 찾아내고, 대안 서비스로 대체를 해서 전체 서비스의 품질 관리를 해준다.

본 논문은 2장에서 통합 웹 서비스와 웹 서비스 품질에 관한 연구를 보이고, 3장에서 품질을 보증하는 기법을 제시한다. 그리고 4장에서는 이를 구현한 프레임워크를 모듈 별로 설명을 하고, 마지막 5장에서는 결론과 향후 과제에 대해 이야기 한다.

## 2. 관련 연구

### 2.1 BPEL4WS

BPEL4WS는 여러 웹 서비스들을 하나의 통합된 프로세서로 만드는 표준으로 채택된 언어이다. 웹 서비스의 통합 시에 순차적으로 나열하는 방식이 아닌, 표준 기술 방법을 통해서 분기, 순환, 연산 등이 필요한 복잡한 프로세스도 표현 할 수 있다.

프로세스는 메시지를 교환하는 주체가 되는 파트너 정보가 있고, 작업의 흐름은 활동으로 나타나진다. 활동은 주로 주고받는 메시지에 관련된 기본 활동과 순환이나 분기 등에 관련된 구조적 활동 그리고 소멸 등의 특수한 목적에 쓰이는 특별 활동으로 구성되어 있다. 각각의 활동은 실행 중, 실행대기 등의 상태정보를 가지고 있게 된다.[5]

### 2.2 웹서비스 상에서의 품질

웹서비스의 품질을 측정하고 분석하기 위해서는 먼저 명확한 지표를 정해야한다. 웹서비스 상에서의 품질 지표는 다음과 같이 몇 가지로 분류가 된다.

가용도(Availability)는 서비스가 운영되는 시간의 백분율 값이다. 보안(Security)은 악의적인 공격으로부터의 방어, 인증 메커니즘 등의 보안 요소이다. 응답 시간(Response Time)은 서비스 요청에 서비스가 응답하는 시간이고, 마지막으로 처리량(Throughput)은 서비스 요청을 처리하는 비율이다.[6]

## 3. BPEL4WS 상의 품질 보증 방법

### 3.1 품질 보증방법

BPEL4WS 상에서 웹서비스들의 품질 보증을 위해서는 먼저 서비스 제공자가 품질 보증에 관한 기본적인 설정을 해야 하고, 특정 서비스에 대한 품질 기준이 있을 경우 이를 정해 주어야한다. 그림 1이 이를 명시한 스키마이고, 본 논문에서는 PoQ 문서

라고 부른다.

품질 측정을 위한 범주으로써는 가용도, 응답시간, 처리량, 보안이 있는데, 본 논문에서는 보안을 제외한 세 가지 범주를 기준으로 정하였다. 즉 사용자가 각 범주에 대한 품질 기준 값을 PoQ 문서에 정해 놓으면, 시스템에서 이를 기준으로 품질 분석을 하고, 정하지 않은 경우에는 자체적인 기준으로 품질 분석을 하게 된다.

PoQ 문서에서 PoQ에는 전체 서비스의 품질 기준을 정할 수 있는 entireThreshold와, 각 웹 서비스에 대한 품질 정책을 설정 할 수 있는 policy가 있다. 또한 시스템에서 품질 정보를 분석하는 시간을 정할 수 있는 analyzeTime이 있다.

policy에는 품질 정책을 정할 대상과 품질 기준치, 교체 할 수 있는 다른 웹 서비스에 대한 정보가 있다. threshold에는 품질 측정 지표 정보인 category와 실제 기준 값에 대한 내용인 threshold와 unit이 있다. alternative는 대체 웹 서비스의 관한 정보가 들어가 있다.

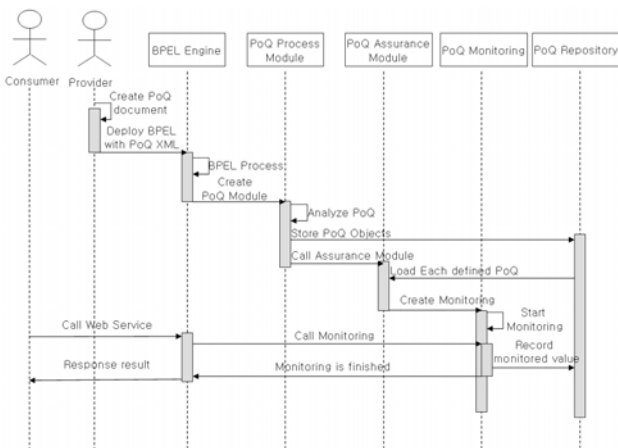
```
<?xml version="1.0" encoding="UTF-8" ?>
- <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="PoQ" type="PoQType" />
  <xsd:complexType name="PoQType">
    <xsd:sequence>
      <xsd:element name="entireThreshold" type="thresholdType" minOccurs="0" maxOccurs="unbounded" />
      <xsd:element name="policy" type="policyType" minOccurs="0" maxOccurs="unbounded" />
      <xsd:element name="analyzeTime" type="analyzeTimeType" minOccurs="1" maxOccurs="1" />
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="policyType">
    <xsd:attribute name="policyId" type="int" />
    <xsd:attribute name="partnerLink" type="string" />
    <xsd:attribute name="operation" type="string" />
    <xsd:attribute name="ndistribution" type="int" />
  </xsd:complexType>
  <xsd:complexType name="thresholdType">
    <xsd:element name="threshold" type="thresholdType" minOccurs="0" maxOccurs="unbounded" />
    <xsd:element name="alternative" type="alternativeType" minOccurs="0" maxOccurs="unbounded" />
  </xsd:complexType>
  <xsd:complexType name="thresholdType">
    <xsd:attribute name="category" type="string" />
    <xsd:attribute name="threshold" type="int" />
    <xsd:attribute name="unit" type="string" />
  </xsd:complexType>
  <xsd:complexType name="alternative">
    <xsd:attribute name="wsdl" type="string" />
    <xsd:attribute name="operation" type="string" />
  </xsd:complexType>
  <xsd:complexType name="analyzeTimeType">
    <xsd:attribute name="firstHour" type="int" />
    <xsd:attribute name="firstMinute" type="int" />
    <xsd:attribute name="repeatAmount" type="int" />
    <xsd:attribute name="repeatunit" type="string" />
  </xsd:complexType>
</xsd:schema>
```

(그림 1) PoQ 문서 스키마

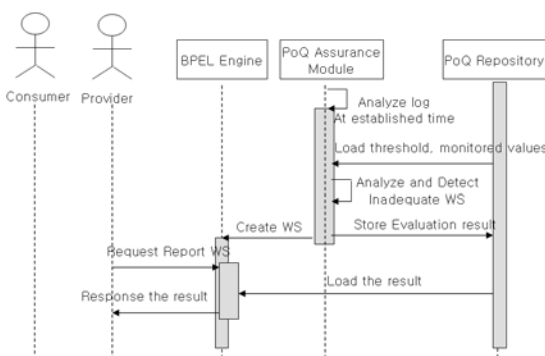
서비스 제공자가 해당 서비스의 품질 관리 및 보증을 위해서 PoQ 문서를 작성을 한 뒤, BPEL Engine에 서비스를 등록 시킬 시에 본 문서를 같이 등록 시켜주면 해당 서비스에 대한 품질 보증 프로세스가 실행 되게 된다. 이와 같은 웹 서비스 품질 보증을 위한 프로세스가 그림 2와 그림 3에 각각 나와 있

다.

그림 2는 PoQ 시스템이 처음 구동되는 절차를 보여주고 있다. 먼저 사용자가 위에서 설명한 PoQ 문서를 작성 하여서 BPEL Engine에 등록시키면, PoQ Process Module이 해당 문서를 분석하고, 그 결과를 PoQ Repository에 저장하게 된다. 그 후에 PoQ Process Module은 PoQ Assurance Module을 호출하게 된다. Assurance Module에서는 저장해 놓은 PoQ 관련 정보를 읽어 와서, 각 웹 서비스를 모니터링 할 준비를 한다. 그 후에 Assurance Module이 Monitoring을 생성하여서, 모니터링을 시작하게 되면, 사용자가 해당 서비스를 사용 할 때마다 그에 대한 로그 정보가 PoQ Repository에 남게 된다. 본 로그 정보는 후에 웹 서비스를 분석할 때 쓰이게 된다. 이로써 웹 서비스 품질 관리에 쓰일 정보를 모을 수 있는 환경이 모두 설정이 되었다.



(그림 2) 품질 보증 절차



(그림 3) 품질 분석 절차

그림 3은 모든 데이터가 모여 있을 때, 시스템에서 웹 서비스 품질 정보를 분석하는 단계를 보여준다. Monitoring Module이 웹 서비스 품질에 대한 정보

를 PoQ Repository에 쌓아 놓으면, PoQ 문서에 정의된 시간마다 Assurance Module이 데이터를 분석을 하게 된다. 분석이 끝나면 그 결과는 PoQ Repository에 저장하게 되고, 최초 분석의 경우는 분석 결과를 보여주는 웹 서비스를 생성하여서, 사용자가 언제든지 통계 분석치를 열람 할 수 있게 해주고, 이미 웹 서비스가 생성 되어있는 경우에는 데이터 갱신만으로도 최신의 정보를 제공할 수가 있다.

### 3.2 웹서비스 품질 분석 방법

웹 서비스의 모니터링을 통해서 쌓은 정보를 바탕으로 현재 웹 서비스의 품질에 대한 기준을 정해서, 그 기준 이하인 경우는 교체를 통해서 전체 품질을 관리 하여야한다. 이를 실현하기 위해서 통계적 방식인 정규분포를 이용하여서, 품질 관리를 하였다.

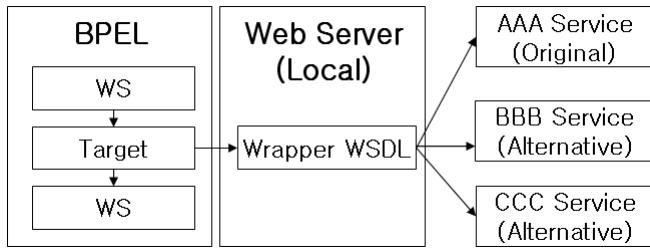
먼저 PoQ 문서에서 사용자가 특정 값(Threshold)을 정해 놓지 않으면 시스템에서는 일정 기간의 데이터를 바탕으로 해당 웹 서비스의 품질에 대한 기준을 세운다. 기준은 정규 분포 공식을 기준으로 하여서 만들어지게 된다. 품질에 대한 기준은 분석 시마다 계속 갱신이 된다. 분석 시에 해당 웹 서비스가 품질 기준에 정해 놓은 정규 분포의 누적 분포 값밖에 있으면 해당 웹 서비스의 품질이 떨어졌다고 판정을 하고 교체를 시도하게 된다. 교체 판정 시에 쓰이는 정규 분포 기준 값은 기본으로 95%이고, PoQ 문서에서 사용자가 임의의 값으로 정할 수 있다.

### 3.3 웹서비스 교체 방법

해당 웹서비스가 사용자가 PoQ 문서에 지정해 놓은 품질 기준을 만족 시키지 못하였거나, 시스템이 3.2의 방식으로 분석하여서, 웹 서비스의 품질이 저하 되었다고 판단하면 해당 웹 서비스를 다른 서비스로의 대체를 시도하게 된다.

사용자가 PoQ 문서에서 대안 웹 서비스를 설정 해 놓은 경우는 해당 웹 서비스의 동작 여부를 테스트 한 후에 그림 4에서와 같이 교체를 하게 된다. 대안 웹 서비스를 설정 해 놓은 경우는 실제로 처음 웹 서비스를 BPEL 엔진에 등록 시킬 때 해당 웹 서비스를 직접 연결 하는 방식이 아닌 일종의 래퍼(wrapper) 서비스를 서버에 만들어서 연결하게 된다. 래퍼 웹 서비스는 기존의 웹 서비스와 대안 웹 서비스를 BPEL 엔진 입장에서는 같은 인터페이스를 가지고 있는 것처럼 보이게 하여서, 품질 저하에

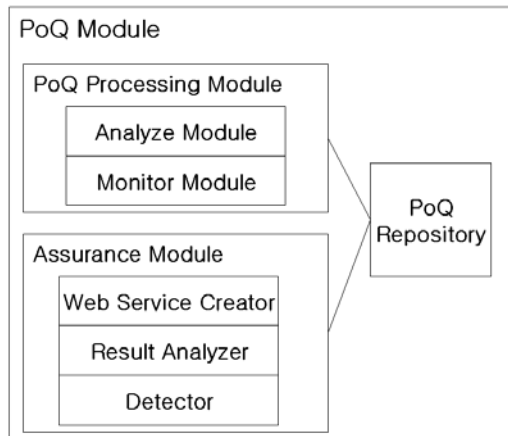
따른 실시간 웹 서비스 교체를 가능하게 해준다.



(그림 4) 웹 서비스 교체 방법

#### 4. PoQ 모듈

본 시스템 전체를 PoQ 모듈이라고 보았을 때 PoQ 모듈은 크게 3부분으로 이루어져있다. 먼저 PoQ Processing Module은 PoQ 문서를 분석하고 저장하는 Analyze Module과 웹 서비스의 품질을 측정하고 저장하는 Monitor 모듈이 있다. 두 번째로 Assurance 모듈은 분석 결과를 보여주는 웹 서비스를 만드는 Web Service Creator와 실제 결과를 분석하는 Result Analyzer, 그리고 결과를 토대로 교체 할 웹 서비스를 발견하는 Detector 모듈이 있다. 마지막으로 본 시스템의 저장소인 PoQ Repository가 있다.



(그림 5) PoQ 모듈

#### 5. 결론 및 향후 과제

본 논문은 BPEL4WS로 웹 서비스 통합 시에 통합된 서비스의 신뢰성을 높이기 위해서, 서비스의 품질을 보장해주는 시스템을 제안하였다. 본 시스템을 통하여서, 웹 서비스의 품질이 떨어지는 시기에 교

체를 함으로써, 하나의 웹 서비스의 불능으로 인해서 전체 프로세스가 사용 불능이 되는 것을 예방할 수 있다. 또한 각각의 웹 서비스의 품질을 측정하고 분석하여서 통계적 결과를 봄으로써 서비스 제공자가 보다 안정되고 높은 품질의 서비스를 제공하는데 중요한 정보를 제공할 수 있다.

본 시스템에서는 사용자가 지정한 웹 서비스만을 교체를 시도하였으나, 시스템 자체에서 유사한 웹 서비스를 스스로 찾아내어서 교체를 시도하는 분야의 연구도 필요하다. 또한 좀 더 유연하고 신뢰성 있는 실시간 웹 서비스 교체를 위하여, 다각도에서의 연구도 필요 할 것이다.

#### 참고문헌

- [1] W3C Web Services Activity, <http://www.w3.org/2002/ws>
- [2] W3C. Simple Object Access Protocol (SOAP) 1.2, W3C Candidate Recommendation, 2002. <http://www.w3.org/TR/soap12-part0/>
- [3] W3C. Web Services Description Language (WSDL) 1.1, W3C Note, 2001. <http://www.w3.org/TR/wsdl> (W3C Working Draft for version 1.2 is available at <http://www.w3.org/TR/wsdl12>.)
- [4] Oasis Committee. Universal Description, Discovery and Integration (UDDI), Version 3 Specifications, 2002. <http://www.uddi.org>.
- [5] "Specification: Business Process Execution Language for Web Service Version 1.1" 2003, <http://www-106.ibm.com/developerworks/library/ws-bpel/>
- [5] ActiveBPEL "The Open Source BPEL Engine", [www.activebpel.org](http://www.activebpel.org), 2004 <http://www.activebpel.org/info/intro.html>
- [6] Daniel A.Menasce, "QoS Issues in Web Services" IEEE INTERNET COMPUTING Vol.6 No.6 pp72-75 2002.12
- [7] 임익친, 김현수, 김승렬, "웹 서비스의 평가, 인증제도에 관한 탐색적 연구," 정보화정책 제10권 제1호 pp.99-119, 2003.
- [8] 정승준, 국윤규, 김운용, 최영근. "BPEL4WS를 이용한 웹서비스 통합 관점의 SLA보증기법" 정보과학회