

BPEL4WS를 이용한 웹서비스 통합 관점의 SLA보증기법

정승준^o 국윤규, 김운용, 최영근
광운대학교 컴퓨터학과
excepter@kw.ac.kr

SLA Assurance on the integrated web services using BPEL4WS

Rudy Jung^o Yoon-Gyu Kook, Woon-Yong Kim, Young-Keun Choi
Dept. of Computer Science, Kwangwoon 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 통합 환경 상의 품질 관리가 필요하나 확실한 대안책이 없었다. 이에 본 논문에서는 사용자와 공급자 사이의 협정을 통해 품질 보증하는 SLA를 이용하여, BPEL4WS로 통합한 환경에 적합한 품질 보증기법을 제시한다. 또한 제시한 기법의 효율적인 활용을 위한 웹 서비스 통합 프레임워크를 제시 한다.

1. 서 론

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 시스템 구성의 필요성이 생겨났다. 이를 구현하기 위한 표준으로써 웹 서비스가 제시되었다. 웹 서비스는 이기종 간의 데이터 통합을 위한 XML과 전송 통합을 위한 HTTP 기반 위에서, 새로이 만들어진 SOAP, WSDL 그리고 UDDI 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)가 제시되었다. BPEL4WS는 이미 구축되어있는 여러 웹 서비스들을 표준적으로 어떻게 통합을 하여서, 하나의 새로운 서비스를 만들 것인가를 기술한 모델 언어로써, 웹 서비스 통합 표준 언어이다.

이러한 통합된 서비스의 가장 큰 문제점 중 하나가 신뢰성의 확보이다. 통합된 서비스 중에서 하나의 서비스라도 오작동을 하는 경우에는 전체 프로세스의 신뢰성에 치명적일 수가 있다. 그러므로 포함된 웹 서비스들과 통합된 전체 서비스에 모두에 대한 신뢰성 확보가 필요하다.

이러한 신뢰성 확보를 위해서는 서비스의 품질을 보증 해주어야한다. 그 방법 중 하나는 SLA(Service Level Agreement) 보증이다. SLA는 서비스 제공자와 서비스 사용자 사이에 서비스 수준에 대한 구체적인 기준을 정하는 것이다. 이러한 기준의 보증을 통해서 서로 간에 신뢰성 있는 서비스를 제공하고, 사용 할 수 있다.

지금까지 제시된 기존의 웹 서비스 상의 품질 보증 시스템들은 통합 환경에서 필요한 기능들을 제공하지 못하였다 [3][6][7]. 또한 제공되는 시스템은 기존 시스템 외부에

새로운 시스템을 구축해서 구현이 가능했었다[6]. 이에 본 논문에서는 통합된 웹 서비스 환경에서 품질 보증이 가능하고, 외부 시스템을 구축할 필요가 없는 통합 웹서비스 SLA 보증 시스템을 제안한다. 그리고 통합 환경에서 SLA 보증을 위한 XML 스키마인 SLA4BPEL(SLA for BPEL4WS)과 본 시스템을 구현한 웹 서비스 통합 프레임워크를 제시한다.

본 논문은 2장에서 BPEL4WS와 SLA 품질 보증에 관한 연구를 보이고, 3장에서 품질을 보증하는 기법을 제시한다. 그리고 4장에서는 이를 구현한 프레임워크를 제시하고, 마지막 5장에서는 결론과 향후 과제에 대해 이야기 한다.

2. 관련연구

2.1 BPEL4WS

BPEL4WS는 여러 웹 서비스들을 하나의 통합된 프로세스로 만드는 표준으로 채택된 언어이다. 웹 서비스의 통합 시에 순차적으로 나열하는 방식이 아닌, 표준 기술방법[4]를 통해서 분기, 순환, 연산 등이 필요한 복잡한 프로세스도 표현 할 수 있다.

프로세스는 메시지를 교환하는 주체가 되는 파트너 정보가 있고, 작업의 흐름은 활동으로 나타내진다. 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.[4]

2.2 SLA

SLA는 서비스의 제공자와 사용자가 서비스의 수준에 대해서 맺는 협정이다. SLA를 맺는 대상으로는 응답성, 가용성, 퍼포먼스, 사용빈도, 보안, 비용 등이 있다. 협정 내용은 협정의 적용 기간과 품질의 기준이 되는 구체적인 값을 포함한다[2].

3. BPEL4WS에서 SLA 적용

3.1 통합된 웹서비스에서 품질 관리

분산되어 있는 웹 서비스를 하나의 프로세스로 통합 할 경우, 각각의 웹 서비스가 신뢰성 있는 동작을 해야지만 전체 프로세스가 신뢰성이 있게 된다. 또한 통합된 프로세스의 품질이 떨어질 경우 그 원인을 찾고 해결하기 위해서도 전체 프로세스뿐만 아니라 개별적인 웹서비스에 대한 품질 보증법 및 관리법이 필요하다.

통합 환경에서 품질 보증을 하는 주요 요소는 응답성, 가용성, 사용빈도이다. 특히 응답성과 가용성은 프로세스 전체와 개별 웹 서비스들 각각을 측정 하여서 품질 측정뿐만 아니라 품질 개선의 자료로도 이용 된다. 사용빈도는 개별 서비스 중에서 빈번하게 사용해서 품질관리가 중요한 웹 서비스 관리 목적 등으로 사용 된다.

이러한 통합된 프로세스의 품질 보증을 BPEL4WS의 엔진에서 해주는 것이 사용자가 쓰기 가장 쉬운 방법이다. 그리고 품질 보증의 한 가지 방법으로써, SLA를 이용하여서 서비스 제공자와 소비자가 정해 놓은 기준에 맞는 품질을 보증하는 것 또한 품질 보증을 위한 최선의 방법이다.

```
<?xml version="1.0" encoding="UTF-8" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" >
  <xs:element name="SLA" type="slatype" />
  <xs:complexType name="slatype">
    <xs:sequence>
      <xs:element name="slo" type="sloType" minOccurs="1" maxOccurs="unbounded" />
      <xs:sequence>
        <xs:attribute name="startDate" type="xsd:date" />
        <xs:attribute name="endDate" type="xsd:date" />
      </xs:sequence>
    </xs:complexType>
    <xs:sequence>
      <xs:element name="evalFunc" type="evalFuncType" minOccurs="1" maxOccurs="unbounded" />
      <xs:element name="reportTarget" type="reportTargetType" minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
    <xs:attribute name="sloId" type="xsd:int" />
    <xs:attribute name="dayTimeConstraint" type="xsd:string" />
    <xs:attribute name="operator" type="xsd:string" />
    <xs:attribute name="method" type="xsd:string" />
    <xs:attribute name="evalWhen" type="xsd:string" />
  </xs:complexType>
  <xs:complexType name="evalFuncType">
    <xs:attribute name="name" type="xsd:QName" />
    <xs:attribute name="operator" type="xsd:string" />
    <xs:attribute name="Threshold" type="xsd:int" />
    <xs:attribute name="unit" type="xsd:string" />
  </xs:complexType>
  <xs:complexType name="reportTargetType">
    <xs:attribute name="name" type="xsd:string" />
    <xs:attribute name="email" type="xsd:string" />
  </xs:complexType>
</xs:schema>
```

(그림 1) SLA4BPEL 스키마

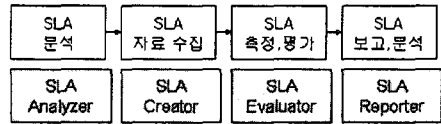
통합 환경의 이와 같은 특징을 적용하여서 그림 1의 SLA4BPEL 스키마로 표현했다. SLA4BPEL은 BPEL4WS로 통합된 웹 서비스 상의 SLA를 보증을 위한 XML 스키마이다.

SLA4BPEL의 루트인 SLA는 여러 개의 SLO(Service Level Objective)로 구성되어있고, SLO에는 측정 기준, 보고대상, 날짜정보, 측정 대상 정보 등이 있다. SLO에서는 측정 대상을 전체 프로세스 혹은 하나의 특정 웹 서비스를 선택할 수 있다. 측정 기준에는 측정 기준에 대한 정보가 있고, 보고 대상에는 측정 결과를 알려줄 대상에 대한 정보가 있다.

위의 스키마에서 측정할 대상을 웹서비스, 프로세스로 정할 수 있고, 측정 항목으로 응답성, 가용성, 퍼포먼스, 사용빈도 등을 결정 할 수 있다.

3.2 SLA 측정 방법

SLA 보증 단계는 그림 2에 나와 있다. 먼저 SLA 문서를 분석을 한 후에 분석 결과를 바탕으로 자료를 수집한다. 그리고 수집한 자료를 측정하고 평가한 후에 이를 보고하고 분석하는 것이다



(그림 2) SLA 측정 단계

실제 SLA를 맺고 시스템에 해당 SLA를 등록하여서, 사용자가 해당 서비스를 사용 시에 SLA 항목에 대한 기록을 하는 과정이 그림 3에 나타나있다.

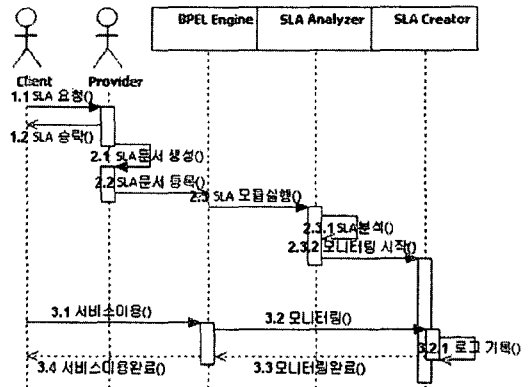
SLA를 측정하기 위해서는 먼저, 공급자와 사용자 사이에 SLA 협정을 맺어야한다. 보통 사용자가 SLA 문서를 작성 하면 공급자가 이를 승인함으로써 SLA에 대한 협정이 이루어진다.

이 SLA 문서를 바탕으로 공급자가 SLA4BPEL 문서를 생성을 해서, BPEL4WS Engine에 문서를 등록하게 된다. Engine에서는 문서가 등록이 되면 SLA Module을 실행하게 된다.

SLA Module에서는 먼저 Analyzer가 SLA4BPEL 문서를 파싱한 후에 분석을 한다. 분석한 결과를 Creator가 모니터링 하기 위한 내부 웹 서비스를 생성하게 되어서, 모니터링이 시작이 되게 된다.

실제 공급자가 BPEL 문서를 BPEL4WS를 엔진에 등록하면서 SLA4BPEL 문서를 같이 등록하고, 모니터링도 등록된 BPEL4WS가 시작이 되면서 같이 시작하게 된다.

사용자가 SLA를 맺은 서비스를 이용할 경우에는 먼저 엔진에서 사용자의 요구를 처리하고, 엔진이 실행 중인 모니터에게 SLA로 지정한 항목을 SLA Repository에 기록하게 한다.

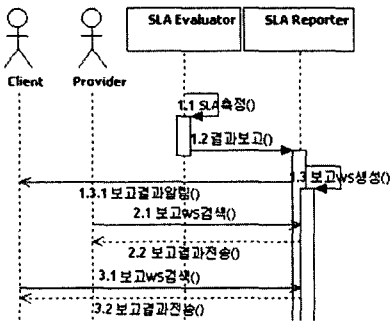


(그림 3) SLA 측정 시나리오

3.3 SLA 결과 보고 방법

SLA에 대한 로그가 쌓인 후에는 쌓인 로그를 분석을 하고 지정한 기준에 맞는지 평가를 하여야 한다. 이를 위한 시나리오가 그림 4에 나와 있다.

기록된 로그에 대한 측정은 “매일 오후6시”와 같은 정해진 시간에 행하게 된다. 정해진 시간이 되면, Evaluator가 SLA Repository에 있는 로그 파일을 정해진 기준대로 측정을 하게 된다. 그리고 측정한 결과를 Reporter에게 보내게 되면, Reporter는 그 결과를 볼 수 있는 웹 서비스를 생성하고, 지정한 사람에게 보고 결과를 전송 해준다. Reporter에 의해서 보고 웹 서비스가 한번 생성이 되면, 제공자나 사용자 등은 웹 서비스를 통해서, 보고 결과를 수시로 볼 수가 있다



(그림 4) SLA 보고 시나리오

4. SLA Engine Framework

본 논문에서 제안한 SLA 보증 시스템은 오픈 소스 형태로 진행 중인 Active BPEL 1.1 Engine[1]에 추가하여 쓸 수 있는 방식으로 구성되었다. Active BPEL 1.1 version에 SLA Engine Module을 추가함으로써, 기존 엔진 기능에 SLA 보장 기능이 추가되게 된다.

그림 4는 Active BPEL4WS 엔진에 SLA Module을 추가한 그림이다. SLA Module과 SLA 보증을 위한 저장 공간인 SLA Repository를 추가함으로써 엔진에서 SLA를 보증해 줄 수 있다. SLA Module은 Analyzer, Creator, Evaluator 그리고 Reporter로 구성되어있다.

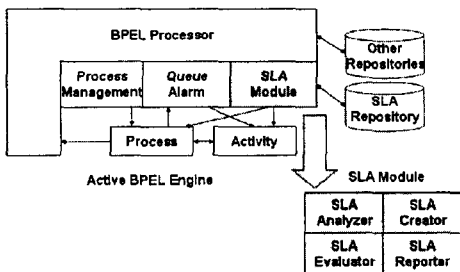


그림 5 Active BPEL Engine framework

5. 결론 및 향후 과제

본 논문에서는 통합 관점의 웹 서비스 품질 보증 시스템을 제안하였다. 제안한 시스템은 BPEL4WS Engine Framework 상에서 SLA 보증을 해주고 있다. 이를 통해서 개개의 웹서비스와 통합된 하나의 서비스를 모두 품질 보증을 할 수가 있다.

이를 통해 분산된 환경에서 웹 서비스 통합을 위한 가장 큰 문제 중에 하나인 서비스의 신뢰성을 확보 할 수가 있다. 또한 개개의 웹서비스들의 품질 관리를 통해서, 품질을 저하 시키는 웹서비스의 빠른 발견 및 교체도 가능하게 한다.

본 시스템의 사용자 입장에서는 BPEL을 등록 시에 SLA4BPEL 파일 하나를 추가함으로써, 엔진 상에서 자동으로 SLA 보증을 해주므로 손쉽게 이용 할 수 있다. 또한 보증 결과를 웹 서비스로 제공을 해주므로 언제든지 손쉽게 결과를 열람 할 수가 있다.

본 시스템은 웹 서비스 통합 시에 신뢰성 구축을 위한 하나의 방법으로 제시가 되었으며, 이 외에도 다른 방향에서의 신뢰성 구축을 위한 연구가 필요하다고 생각한다. 그리고 품질보증 과정 중에서 결과를 분석 및 보고 하는 방법에 대한 표준방법의 연구가 필요하다고 생각된다.

6. 참고문헌

- [1] ActiveBPEL Engine "The Open Source BPEL Engine", <http://www.activebpel.org/info/intro.html> 2004
- [2] Akhil Sahai, Anna Durante, Vijay Machiraju. "Towards Automated SLA Management for Web Services" HP Laboratories
- [3] Akhil Sahai, Vijay Machiraju, Mehmet Sayal, Aad van Moorsel, Fabio Casati. "Automated SLA Monitoring for Web Services" HP Laboratories
- [4] "Specification: Business Process Execution Language for Web Service Version 1.1" 2003, <http://www-106.ibm.com/developerworks/library/ws-bpel/>
- [5] Petia Wohed, Wil M.P., Marlon D, Arthur H.M., "Analysis of Web Services Composition" 2003
- [6] 서승희, 박상근, 최덕재, "Mobile Agent를 이용한 사용자 관점의 웹서비스 진단 시스템 설계 및 구현" 2001
- [7] 이흥준, 박상원, 정성호, 이강찬, 전종홍, "Web Service에서 QoS 지원방향에 관한 연구" 2003