

BPEL4WS 기반의 비즈니스 프로세스 관리를 위한

통합 모니터링 시스템

이종혁^o 국윤규 김운용 최영근
광운대학교 컴퓨터학과
shineljh@kw.ac.kr

Integrated Monitoring System

for Business Process Management based on BPEL4WS

Jong-Hyuk Lee^o Yoon-Gyu Kook, Woon-Yong Kim, Young-Keun Choi
Dept. of Computer Science, Kwangwoon University

요 약

BPEL4WS는 이기종의 시스템을 사용하는 사내 조직뿐만 아니라 협력 업체 및 고객 전반에 걸쳐 비즈니스 프로세스를 통합해준다. 통합된 비즈니스 프로세스 관리는 이질적인 수행환경 극복, 기업간 프로세스 자동화, 프로세스에 대한 모니터링 및 관리, 비즈니스 트랜잭션 관리, 파트너 상태에 대한 관리가 필요하다. 하지만 비즈니스 프로세스, 비즈니스 트랜잭션, 파트너 상태에 대한 관리에 대한 연구가 미흡하다. 따라서 본 논문에서는 통합 프로세스 모니터링 시스템을 제안함으로써, 프로세스 관리자에게 비즈니스 진행 상태에 대한 가시성을 제공하고, 신뢰성 있는 웹 서비스를 위한 트랜잭션을 관리해 주며, 예외 처리를 위한 파트너들의 모니터링 기능을 제공한다.

1. 서 론

최근 기업 내 또는 기업간 비즈니스 통합을 위한 기술로 웹 서비스가 부각되고 있다. 웹 서비스는 웹 표준과 표준을 확장한 언어 및 프로토콜(SOAP, WSDL, UDDI)을 사용하여, 이질적인 시스템 간의 상호운용을 제공한다. 또한 BPEL4WS는 이기종 시스템을 이용하는 사내 조직 뿐만 아니라 협력 업체 및 고객 전반에 걸쳐 비즈니스 프로세스 및 트랜잭션 서비스를 통합해 준다 [1][2][4].

비즈니스 프로세스는 업무 흐름, 업무 수행 순서, 공유 데이터 형식, 협력업체와의 상호 관계 등을 기술하며, BPEL4WS를 통해 표준화할 수 있다. 이러한 비즈니스 프로세스를 관리하게 위해서는 이질적인 수행환경을 극복하고, B2B 환경에서의 기업간 프로세스를 자동화해야 하며, 프로세스에 대한 모니터링 및 관리 기능을 제공해야 한다. 또한 신뢰성 있는 웹 서비스를 위한 비즈니스 트랜잭션 관리와 프로세스에 참여 중인 파트너들을 관리할 수 있어야 한다[3][8].

SOAP과 WSDL은 이질적인 수행환경을 극복할 수 있도록 해주고, BPEL4WS를 통해 기업간 프로세스를 자동화 해 주며, WS-Coordination[5]과 WS-Transaction [6]을 통해 신뢰성 있는 웹 트랜잭션을 수행할 수 있다. 하지만 비즈니스 프로세스 인스턴스, 장기실행 트랜잭션, 참여 파트너에 대한 관리에 대한 연구가 부족하다.

따라서 본 논문에서는 BPEL4WS 기반에서 비즈니스 프로세스 관리를 위한 프로세스 인스턴스와 장기 실행 트랜잭션, 비즈니스 파트너를 통합하여 모니터링해 주는 시스템을 제안한다. 또한 유연하고 확장성 있는 시스템

제공을 위해 모니터링 결과를 웹 서비스로 제공하고, 결과를 보인다.

본 논문의 구성은 2장에서 관련연구에 대해 살펴보고 3장에서는 통합 모니터링 시스템을 보인다. 4장에서는 모니터링 결과를 웹 서비스로 제공하는 것을 기술하고, 5장에서 결론 및 향후 연구로 마무리한다.

2. 관련 연구

2.1 WS-Coordination과 WS-Transaction

WS-Coordination은 웹 서비스 코디네이션의 일반적인 토대를 제공하고, WS-Transaction은 분산 조직간의 웹 서비스 환경에서 업무 처리의 일관성과 신뢰성을 달성하고자 할 때 필요한 융통성 있는 프로토콜을 제공하고, 비즈니스 프로세스가 진행되는 동안 자원이 폐쇄되지 않도록 단기간 또는 장기간 트랜잭션 기능을 제공한다 [5][6][7].

2.2 BPEL4WS

BPEL4WS는 비즈니스 관점에서 웹 서비스의 결합을 기술하는 고수준의 언어로 기업간의 비즈니스 프로토콜을 표현할 수 있다. 비즈니스 프로세스는 하나의 트랜잭션을 완료하기 위해 긴 시간을 필요로 하는 경우가 많으며 이를 위해 BPEL4WS는 비즈니스 프로세스에 대응하는 결합 웹서비스의 상태 정보를 기술할 수 있다. 이 때, WS-Coordination과 WS-Transaction이 BPEL4WS를 도와 각기 다른 구현환경에서 다중의 웹 서비스 및 비즈니스 프로세스를 일관되고 신뢰할 수 있게 조정하여 원하는 결과가 달성되도록 한다[4].

3. 통합 모니터링 시스템

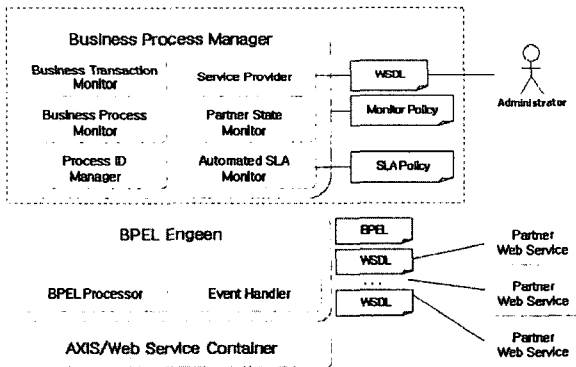
본 시스템은 BPEL4WS 기반의 비즈니스 프로세스 실행 환경에서 동작을 하게 되며, 다음과 같은 기능을 제공한다.

첫째, 비즈니스 프로세스 인스턴스의 진행 상태를 파악할 수 있는 가시성을 제공해 준다. 파트너와의 메시지 교환과 내부적인 메시지 발생에 대해 모니터링해 주고 결과를 보여줌으로서 비즈니스 프로세스의 진행 상황을 파악할 수 있도록 해준다.

둘째, 수행중인 비즈니스 트랜잭션의 상태를 파악할 수 있도록 해준다. 스코프 단위로 비즈니스 트랜잭션을 모니터링하고 결과를 보여주어 트랜잭션 관리를 용이하게 해 준다.

셋째, 파트너 서비스들의 상태를 파악할 수 있도록 해준다. 한 비즈니스 프로세스는 여러 파트너 웹 서비스들의 조합으로 구성이 되며, 참여하고 있는 파트너들의 웹 서비스의 중요한 정보를 파악할 수 있어야 한다.

전체 시스템은 (그림 1)과 같이 구성되며, 본 시스템은 BPEL 엔진 상위에서 동작한다.



(그림 1) 시스템 구성

시스템은 Process ID Manager, SLA Monitor, Partner State Monitor, Business Process Monitor, Business Transaction Monitor, Service Provider로 이루어진다.

Process ID Manager는 프로세스 인스턴스를 관리할 수 있도록 프로세스 식별자를 관리하며, 세부적인 내용은 3.1절에서 언급하도록 한다. SLA Monitor는 서비스의 QoS를 위해 SLA를 측정하는 모듈로 본 논문에서는 언급하지 않는다. Business Process Monitor는 실행중인 비즈니스 프로세스 인스턴스에 대한 모니터링을 하게 되며, 세부적인 내용은 3.2절에서 다룬다. Business Transaction Monitor는 BPEL로 실행되는 장기 실행 비즈니스 트랜잭션을 모니터링 하며, 세부적인 내용은 3.3절에서 다룬다. Partner State Monitor는 참여 파트너의 상태에 대한 모니터링을 하며, 세부적인 내용은 3.4절에서 다룬다. Service Provider는 모니터링하고 있는 결과를 웹 서비스로 보여주는 역할을 하며, 4장에서 세부사항을 다루게 된다.

3.1. 비즈니스 프로세스 식별자

BPEL4WS의 생성은 명시적이지 않고, <receive>나 <pick>을 통해 메시지가 들어왔을 때, correlationSet을 검사해서 새로운 correlationSet인 경우 인스턴스를 생성하는 생명주기를 가지게 된다. 하지만, 관리자 입장에서 식별자를 correlationSet으로 관리하기 용이하지 않으며, 본 논문에서는 이를 해결하기 위해 Process ID Manager를 제안한다.

Process ID Manager는 관리자에게 용이하게 프로세스 인스턴스를 식별을 위해 프로세스 생성자와 생성 시간으로 인스턴스를 식별할 수 있도록 해 준다. 이를 위해 correlatonSet과 새로 생성된 인스턴스에 대한 서비스 요청자와 요청 시간에 대한 정보를 함께 관리하고, 요청자와 요청 시간으로 correlationSet과 매칭시켜 적절한 인스턴스를 반환시키게 된다.

3.2 비즈니스 프로세스 모니터링

Business Process Monitor는 비즈니스 프로세스의 메시지 교환을 모니터링하게 된다. (표 1)에서는 메시지 교환과 관련된 요소들을 보이며, 본 논문에서는 이 요소들에 대해 모니터링을 한다.

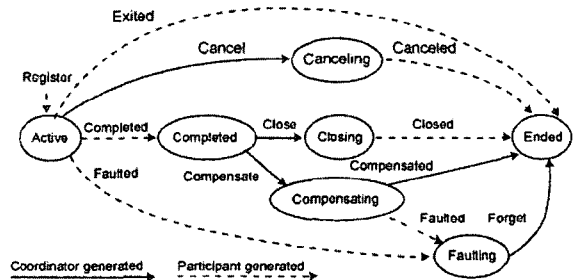
Element	Description
<receive>	프로세스가 제공하는 WSDL 오퍼레이션의 입력
<pick>	이벤트 핸들러가 포함되어 있으며, 알람과 메시지 도착에 대한 핸들링을 함
<invoke>	프로세스에 의해 파트너에 의해 제공되는 웹 서비스함 제공
<reply>	프로세스 요청한 파트너가 응답 액티비티가 필요할 경우, 응답

(표 1) BPEL4WS에서 메시지 교환과 관련된 요소

Business Process Monitor에서는 (표 1)에 있는 요소들의 수행 상태를 모니터링하며, Service Provider의 요청이 있을 때 결과를 반환한다.

3.3 트랜잭션 모니터링

BPEL4WS에서는 신뢰성 있는 장기 실행 트랜잭션을 위한 보상 핸들러를 제공하며, 보상 핸들러는 스코프 단위로 동작한다. 이 때, 스코프간의 에러 및 보상 핸들링을 하기 위해 WS-Transaction을 사용하게 된다. (그림 2)는 BPEL4WS에서 사용하는 WS-Transaction의 BusinessAgreement Protocol을 보여준다[2][6].



(그림 2) Business Agreement Protocol State

각 상태는 아래의 (표 2)와 같이 이루어져 있다.

State	Description
Active	수행되지 않은 상태
Completed	프로세스를 마친 상태(종료 또는 보상을 기다리게 됨)
Canceling	상위 스코프에서 취소시킨 상태
Compensating	보상 진행 상태
Closing	종료 진행 상태
Faulted	오류 발생한 경우(일반 프로세스 또는 보상 프로세스)
Ended	종료된 상태(보상할 수 없게 됨)

(표 2) 비즈니스 트랜잭션의 상태

Business Transaction Monitor는 각 스코프의 트랜잭션 상태를 모니터링하게 되며, Service Provider의 요청이 있을 때 결과를 반환한다.

3.4. Partner State

실행중인 프로세스의 관리 외에 참여하고 있는 파트너들의 상태를 파악함으로써 각종 예외에 대해 빠르게 대응할 수 있게 된다. MonitorPolicy에 모니터링할 오퍼레이션과 모니터링 주기 등의 정책을 설정하게 되고, Partner State Monitor는 정책에 따라 참여 파트너들을 모니터링하고, Service Provider의 요청이 있을 때 결과를 반환한다.

4. 모니터링 웹 서비스

본 논문에서는 유연성과 확장성을 높이기 위해 모니터링 결과를 웹 서비스로 제공한다.

Service Provider는 각 모니터는 관리자에게 요청자와 요청 시간을 입력 받아 Process ID Manager에게 요청을 하여 식별자를 받아오고, 식별자를 통해 각 모니터에게 정보를 인스턴스 정보를 받아오게 된다.

(그림 3)은 모니터링한 결과를 제공하는 웹서비스의 WSDL 문서이며, 비즈니스 프로세스의 실행상태, 실행 중인 트랜잭션의 상태, 파트너의 상태를 보여주는 오퍼레이션들과 식별자를 얻어오는 오퍼레이션을 가진다.

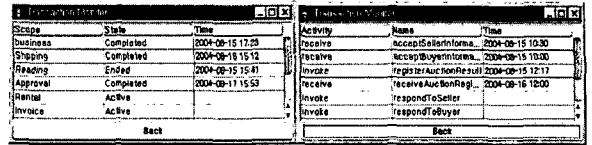
```

13 <?xml version="1.0" ?>
14 <wsdl:types?>
15 <wsdl:message name="getSLAReporRequest"?>
16 </wsdl:message?>
17 <wsdl:message name="getCurrentStateResponse"?>
18 </wsdl:message?>
19 <wsdl:message name="getVariableRequest"?>
20 </wsdl:message?>
21 <wsdl:message name="getTransactionLogRequest"?>
22 </wsdl:message?>
23 <wsdl:message name="getTransactionLogResponse"?>
24 </wsdl:message?>
25 <wsdl:message name="generateTIDResponse"?>
26 </wsdl:message?>
27 <wsdl:message name="getCurrentStateRequest"?>
28 </wsdl:message?>
29 <wsdl:message name="getSLAReporResponse"?>
30 </wsdl:message?>
31 <wsdl:message name="generateTIDRequest"?>
32 </wsdl:message?>
33 <wsdl:message name="getVariableResponse"?>
34 </wsdl:message?>
35 <wsdl:portType name="MonitoringService"?>
36 <wsdl:operation name="getVariable"?>
37 <wsdl:input message="getVariableRequest"?>
38 </wsdl:input?>
39 <wsdl:output message="getVariableResponse"?>
40 </wsdl:output?>
41 </wsdl:operation?>
42 <wsdl:operation name="generateTID"?>
43 <wsdl:input message="generateTIDRequest" parameter="OrderId"?>
44 </wsdl:input?>
45 <wsdl:output message="generateTIDResponse" parameter="OrderId scope"?>
46 </wsdl:output?>
47 </wsdl:operation?>
48 <wsdl:operation name="getSLARepor"?>
49 <wsdl:input message="getSLAReporRequest" parameter="OrderService"?>
50 </wsdl:input?>
51 <wsdl:output message="getSLAReporResponse" parameter="OrderService scope"?>
52 </wsdl:output?>
53 </wsdl:operation?>
54 <wsdl:operation name="getTransactionLog"?>
55 <wsdl:input message="getTransactionLogRequest" parameter="OrderId scope"?>
56 </wsdl:input?>
57 <wsdl:output message="getTransactionLogResponse" parameter="OrderId scope"?>
58 </wsdl:output?>
59 </wsdl:operation?>
60 </wsdl:portType?>
61 <wsdl:binding name="MonitoringServiceSoapBinding" type="tns:MonitoringService"?>
62 <wsdl:operation name="getVariable" soapAction="tns:getVariable"?>
63 </wsdl:operation?>
64 <wsdl:operation name="generateTID" soapAction="tns:generateTID"?>
65 </wsdl:operation?>
66 <wsdl:operation name="getSLARepor" soapAction="tns:getSLARepor"?>
67 </wsdl:operation?>
68 <wsdl:operation name="getTransactionLog" soapAction="tns:getTransactionLog"?>
69 </wsdl:operation?>
70 </wsdl:binding?>
71 </wsdl:service?>
72 </wsdl:definitions?>
    
```

(그림 3) 모니터링 웹 서비스

웹 서비스로 제공되기 때문에, 다른 시스템에 쉽게 통

합 응용될 수 있으며, (그림 4)은 Java로 결과를 보인 것이다.



(그림 4) 비즈니스 관리 시스템

5. 결론

본 논문에서는 프로세스 관리자에게 비즈니스 진행 상태에 대한 가시성을 제공하고, 신뢰성 있는 웹 서비스를 위한 트랜잭션을 관리해 주며, 예외 처리를 위한 파트너들의 모니터링 기능을 제공하는 통합 프로세스 모니터링 시스템을 제안하였다. 또한 유연성과 확장성을 위해 모니터링 결과를 웹 서비스로 제공하였다.

향후 연구 과제로는 SLA 자동화에 대한 연구가 더 필요하다.

참고문헌

- [1] Curbera F., et al., "Unraveling the Web Services Web: An Introduction to SOAP, WSDL and UDDI. IEEE, 2002
- [2] P.Wohed, et al., "Pattern Based Analysis of BPEL4WS," Technical Report FIT-TR-2002-4, QUT, 2002. 4
- [3] Pires, P.F., Benevides, R.F. et al., "Web Transact: A Framework for Specifying and Coordinating Reliable Web Service Compositions", In: Technical Report ES-578/02 PESC/Coppe Federal University of Rio de Janeiro, 2002. 4
- [4] T. Andrews, et al., "Business Process Execution Language for Web Services(BPEL4WS) 1.1," 2003. 5, [http://www-106.ibm.com/developerworks/webservices/library/ws-bpel]
- [5] F. Cabrera et al., "Web Service Coordination (WS-Coordination)", 2003. 9, [http://www-106.ibm.com/developerworks/library/ws-coor/]
- [6] F. Cabrera et al., "Web Services Transaction(WS-Transaction)", 2002. 8, [http://www-106.ibm.com/developerworks/library/ws-ttranspec/]
- [7] Tom Freund, Tony Storey, "Transactions in the world of Web services", 2002. 8, [http://www-106.ibm.com/developerworks/webservices/library/ws-wstx1/]
- [8] 김민수, "통합 비즈니스 프로세스를 통한 기업정보 시스템의 상호운영성", 정보과학회지, 2004. 7