웹서비스를 위한 비즈니스 프로세스 실행 언어 표준 도입 가이드라인 개발

김동수*

*숭실대학교 산업·정보시스템공학과 서울특별시 동작구 상도동 511, 우:156-743.

Abstract

본 논문에서는 웹서비스를 위한 비즈니스 프로세스 실행 언어인 BPEL4WS 표준을 구현하기 위한 표준 도입 가이드라인을 개발하여 제시하였다. BPEL4WS 표준을 지원하는 BPM 솔루션이 갖추어야 하는 참조 아키텍처(Reference Architecture)를 제시하고, 참조 아키텍처를 구성하는 핵심 컴포넌트인 BPEL 디자이너, 프로세스 공개 모듈 및 시작 모듈, BPEL 엔진, 프로세스 모니터링 및 분석 모듈별로 구현 가이드라인을 제시하였다. 본 연구에서 제시한 가이드라인은 BPM 시스템 개발 기업이 BPEL 표준을 적용하여 솔루션을 효율적으로 구축하는데 활용될 수 있다.

1. 서론

기업 활동의 근간을 이루는 비즈니스 프로세스들을 투명하게 관리하고 통제하기 위해 비즈니스 프로세스 관리(BPM : Business Process Management)가 주목받고 있다[2]. Gartner에서는 BPM을 “인적자원과 애플리케이션 수준의 상호작용을 포함한, 프로세스 분석, 정의, 실행, 모니터링 등의 프로세스 관리를 명확하게 할 수 있는 도구와 서비스”로 정의함으로써 프로세스 관리 도구 체계를 지칭한다. Ovum에서는 “조직 내외의 사람 및 시스템이 상호 작용하는 비즈니스 프로세스를 지속적으로 인지하고 관리할 수 있도록 지원하는 일종의 변화 관리 및 시스템 실행 방법론”으로 정의함으로써 개선을 위한 관리임을 강조한다. 결국, BPM은 “기업 이윤 창출을 위한 프로세스의 지속적인 개선을 목표로 프로세스 라이프 사이클(분석, 정의, 실행, 개선)에 걸쳐 기업 내부의 자원을 통합 관리할 수 있는 시스템”, 즉 BPMS(Business Process Management Systems)로 실현된다고 볼 수 있다[1].

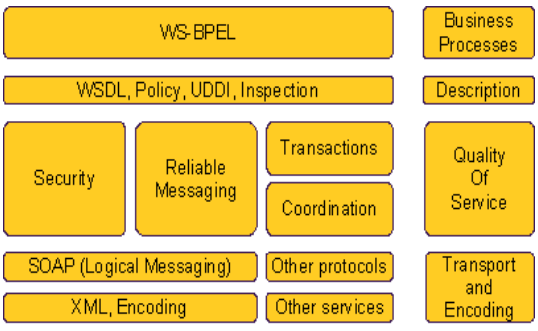
프로세스 모델링 측면, 프로세스 실행과 운영, 프로세스 질의 등에 관한 BPM 표준들이 발표되어 여러 기업에서 활용하고 있다. 여러 BPM 표준 가운데 가장 중요한 표준으로 BPEL(Business Process Execution Language)이 꼽히고 있다[8].

본 연구의 목적은 국내 BPM 업체들의 표준 적용 작업을 용이하게 하기 위하여 가장 주목받고 있는 BPM 국제 표준인 BPEL 표준 적용 가이드라인을 제시하는 것이다. 프로세스 설계, 실행과 운영 측면에서 BPEL 표준을 활용하여 BPM 솔루션을 어떻게 구성하고 각 구성요소의 핵심 기능은 어떤 것이 제공되어야 하는지를 주요 내용으로 하는 표준 도입 가이드라인을 제시하고 있다. 이를 통해 BPM 공급업체들은 자사 솔루션의 특징에 따른 표준 적용 전략을 수립할 수 있고, BPM 표준 적용 및 개발 시의 시행착오를 줄임으로써 보다 효율적인 개발과 적용이 가능하다.

2. BPEL 표준 개요 및 표준화 동향

2.1 BPEL 표준화 동향

웹서비스 환경에서 비즈니스 프로세스를 정의하고 실행하기 위한 표준 언어인 BPEL은 2003년 5월 1.1 버전이 발표된 이후 국제 표준화 기구인 OASIS의 WSBPEL 기술위원회에서 2.0 버전의 스펙을 만들기 위해 표준화 작업을 계속하고 있다. 동 기술위원회에서는 2004년 9월 투표를 통해 2.0 버전의 스펙 부터는 BPEL4WS 라는 명칭 대신 WS-BPEL이라는 명칭을 사용하기로 결정하였다.



<그림 1> WS-* 스택 상에서의 BPEL 위치

<그림 1>은 웹서비스 표준 스택 상에서

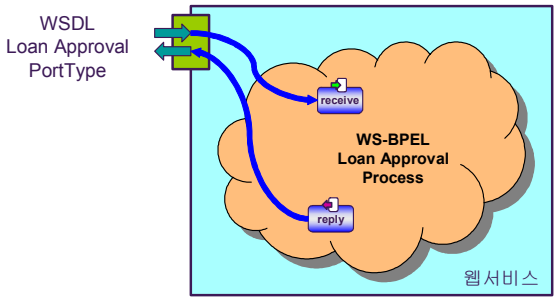
BPEL의 위치를 보여 주고 있다[4]. BPEL은 웹서비스 구성(composition) 표준의 기초가 되고 있다. 함께 발표된 웹 서비스 통신 규약인 WS-Coordination, WS-Transaction과 함께 기업간 프로세스, 즉 협력업체 및 고객 전반에 걸쳐 다중 비즈니스 프로세스 및 트랜잭션 서비스를 신뢰성 있게 통합할 수 있도록 한다.

2.2 BPEL4WS 1.1 표준의 주요 내용

BPEL4WS는 실행용(executable) 비즈니스 프로세스와 추상(abstract) 비즈니스 프로세스를 지원한다. 실행용 비즈니스 프로세스는 기존 서비스를 조합하여 새로운 서비스를 정의하는 것이다. 추상 비즈니스 프로세스는 비즈니스 프로토콜이라고도 불리는데 내부적인 작동을 공개하지 않고 프로토콜 참여자의 가시적인 메시지 교환 동작을 기술하는 것을 말한다

BPEL4WS는 WSDL(Web Services Description Language) 1.1, XML Schema 1.0, XPath 1.0, WS-Addressing에 의존적인 스펙이다. 이 중 WSDL이 BPEL4WS에 가장 큰 영향을 미친 표준이다. BPEL 프로세스 모델은 WSDL로 정의된 서비스 모델의 맨 위에 위치한다. BPEL 프로세스 모델의 핵심은 WSDL로 표현된 서비스들 사이의 상호작용(peer-to-peer interaction)이라는 개념이다.

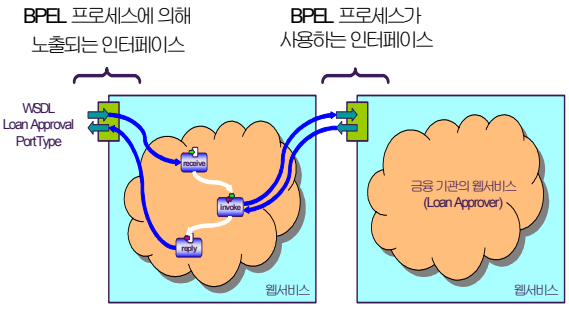
BPEL4WS에서는 프로세스와 그 파트너 모두가 WSDL 서비스로 모델링된다. 비즈니스 프로세스는 프로세스 인스턴스와 그 파트너들 간의 상호작용을 어떻게 조정할(coordinate) 것인가를 정의한다. 이런 의미에서 BPEL 프로세스 정의는 다수의 WSDL 서비스를 제공하거나 사용한다. 또, BPEL 프로세스 정의는 프로세스에 참여하는 파트너와 자원과 관련한 프로세스 인스턴스의 행동과 상호작용에 대한 표현을 웹서비스 인터페이스를 통해 제공한다. 다시 말해 BPEL4WS는 상호작용 상의 정해진 역할을 하는 비즈니스 프로세스가 지켜야 하는 메시지 교환 프로토콜을 정의한다.



<그림 2> BPEL 프로세스의 WSDL 서비스화

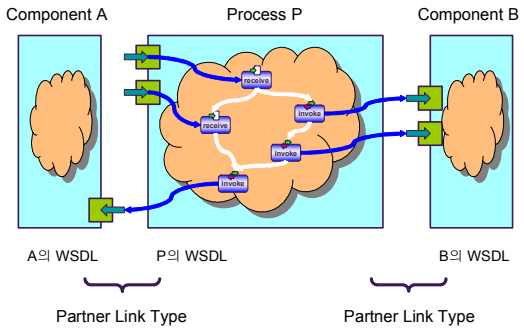
<그림 2>는 BPEL4WS와 WDSL의 관계를 보여 주고 있다. 그림에서 보는 것처럼 BPEL 프로세스 모델은 WDSL 포트 타입을 통해 웹서비스로 제공된다. 이 때 메시지 교환은 WDSL 오퍼레이션에 대응된다.

BPEL 프로세스는 비즈니스 파트너가 제공하는 WSDL 서비스를 통해서 다른 프로세스들과 상호작용한다. <그림 3>에서 보는 것처럼 WSDL 포트 타입을 통해 다른 서비스를 호출하기도 하고 호출되기로 한다.



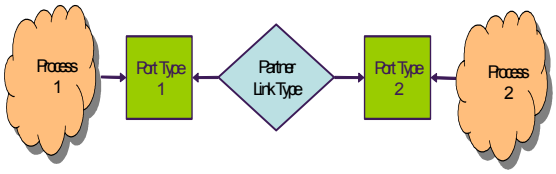
<그림 3> WSDL 포트 타입을 통한 재귀적 프로세스 구성

<그림 4>에서 프로세스 P와 웹서비스 컴포넌트 A, B가 WSDL 인터페이스를 활용하여 웹서비스를 구성하고 있다. BPEL 프로세스는 메시지를 받기도 하고 각 컴포넌트의 서비스를 호출하기도 한다.



<그림 4> BPEL을 이용한 웹서비스 구성

<그림 4>에서 WSDL 간 연결을 위해 파트너 링크 타입이 존재함을 알 수 있다. 파트너 링크 타입은 파트너와 일대일 대화(peer-to-peer conversation)를 하는데 필요한 채널을 의미한다. <그림 5>에서 보는 것처럼 파트너 링크 타입은 요구되거나 제공되는 포트타입을 지정한다.

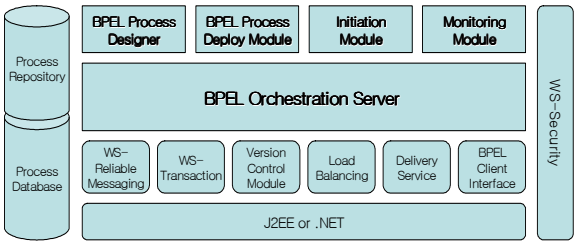


<그림 5> 파트너 링크 타입

3. BPEL4WS 구현 참조 아키텍처

BPEL 표준 스펙을 구현하여 시스템을 구축하기 위해서는 BPEL 프로세스를 설계하기 위한 프로세스 모델러(혹은 디자이너)와 정의된 프로세스 모델을 실행하기 위한 BPEL 엔진이 기본적으로 필요하다. 그 외에도 프로세스 실행을 관리 감독하기 위한 모니터링 프로그램을 많은 벤더들이 제공하고 있다.

본 장에서는 BPEL을 지원하는 BPM 시스템을 구현하기 위해 참조할 수 있는 공통 아키텍처를 <그림 6>과 같이 제시하였다. 제안된 참조 아키텍처는 BPEL 표준을 지원하는 각종 솔루션들을 분석하여 BPEL 구현에 반드시 필요한 공통적 핵심 컴포넌트를 뽑아내어 작성한 것이다. 제시한 BPEL 구현 참조 아키텍처는 BPEL 엔진 리스트 가운데 자료가 공개되어 있는 일부 솔루션과 국내 BPEL 지원 BPM 솔루션들을 참고하여 만든 것이다.



<그림 6> BPEL 구현 참조 아키텍처

- BPEL Engine or Orchestration Server: BPEL 프로세스를 실행시키고 비동기적인 상호작용을 상호연관시키고 조정하여 협업 비즈니스 흐름을 관장함
- BPEL Process Designer: 프로세스 정의 기능을 담당
- BPEL Process Deploy Module: 정의된 프로세스를 서버에 공개하는 기능 수행
- Initiation Module: 서버에 공개된 프로세스를 시작시킴
- Monitoring Module: 프로세스 진행 상태를 모니터링하는 기능을 제공
- WS-Transaction: 느슨하게 결합된 서비스들에 걸친 비즈니스 트랜잭션의 보상 작업을 처리함
- Version Control Module: BPEL 프로세스의 버전 관리 기능을 담당함
- Delivery Service: SOAP, JMS, e-mail 등의 메시지가 원격 사이트에 전달될 수 있도록 해 줌. 인증, 암호화, 부인방지 등의 기능을 포함
- WS-ReliableMessaging: 메시징에 신뢰도를 높여주는 것으로, 목적지까지 메시지가 확실하게 도착하는 것을 보장
- Load Balancing: 업무 부하의 균형을 위한 기능을 수행
- BPEL Client Interface: 프로세스 디자이너, 모니터링 모듈 등 각종 BPEL 클라이언트 프로그램과의 인터페이스를 담당
- Process Repository: BPEL 프로세스의 저장소
- Process Database: BPEL 프로세스의 실행시 발생하는 각종 데이터를 관리하기 위한 데이터베이스
- WS-Security: 웹서비스 안전성을 위한 보안 기능을 담당

열거된 여러 요소들 가운데 가장 핵심적인 컴포넌트라 할 수 있는 BPEL 엔진, 프로세스 디자이너, 프로세스 공개 모듈 및 시작 모듈(Initiator), 프로세스 모니터링 모듈을 중심으로 각 컴포넌트들이 갖추어야 할 기능을 자세히 설명하였다.

4. BPEL4WS 핵심 컴포넌트 기능 요구사항

본 장에서는 BPEL 지원 BPM 시스템을 개발하기 위해 참조할 수 있도록 프로세스 정의, 서버로의 공개, 프로세스 실행, 모니터링으로 구성되는 일련의 BPEL 표준 활용 시나리오에 따라 핵심 컴포넌트 구현 지침을 제시하였다. BPEL 표준 스펙이 프로세스 모델링 규약을 다루고 있기 때문에 많은 내용이 프로세스 디자이너를 구현하기 위한 지침에 할애되고 있으나, 엔진 또한 BPEL 프로세스 모델의 해석과 실행을 관장하므로 디자이너 구현 지침에서 설명한 내용은 엔진의 실행 모듈을 구현하는데 있어서도 참조할 수 있다.

4.1 BPEL 프로세스 디자이너

BPEL 프로세스 디자이너는 BPEL 스펙에서 정의한 BPEL 프로세스 모델을 설계하는 기능을 제공하는 프로그램으로 BPEL 준수 프로세스 모델을 정의할 수 있는 기능을 제공해

야 한다. 여기서는 BPEL 표준 프로세스 모델을 구성하는 주요한 요소를 중심으로 BPEL 프로세스 디자이너를 개발할 때 고려해야 할 점을 설명하였다.

BPEL 프로세스를 설계하기 위한 일반적인 과정은 다음과 같다.

- 먼저 관련된 웹서비스의 구조를 이해한다.
- BPEL 프로세스를 위한 WSDL을 정의한다.
- 파트너 링크 타입을 정의한다.
- BPEL 프로세스를 정의한다. (파트너 링크를 정의하고, 변수를 선언하고, 프로세스 로직을 정의함)

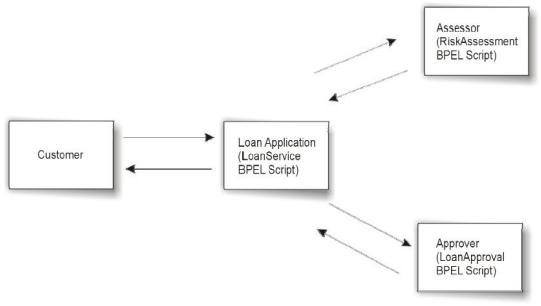
위에서 설명한 프로세스 생성하기 위한 전 과정을 프로세스 디자이너 프로그램이 지원해야 한다. 그리고 BPEL 프로세스 디자이너는 사용자 편의성을 제고하기 위해 GUI 형태로 프로세스 정의 기능을 제공하는 것이 바람직하다.

관련된 웹서비스 구조에 대한 이해를 바탕으로 프로세스를 설계하는 과정에 따라 프로세스 디자이너의 주요 기능을 설명하였다.

4.1.1 웹서비스 구조에 따른 WSDL의 정의

BPEL 프로세스 정의 작업을 시작하기 전에 구현할 비즈니스 프로세스에서 호출하는 웹서비스에 대해 먼저 이해할 필요가 있다. 이를 흔히 파트너 웹서비스(Partner Web Service)라고 부른다. 클라이언트가 자신을 호출할 수 있도록 WSDL 포트 타입을 정의해야 하고, 자신이 호출하는 파트너 웹서비스의 포트 타입도 정의해야 한다. 따라서, BPEL 프로세스 디자이너는 정의하고자 하는 프로세스를 위한 WSDL 정의 기능을 제공해야 한다.

여기서는 <그림 7>에서 개념적으로 표현한 대출 승인(Loan Approval) 서비스 예제를 가지고 WSDL 정의 기능을 설명하고자 한다.



<그림 7> 대출 승인 BPEL 프로세스

본 예제는 BPEL4WS 1.1 표준에서 제시하고 있는 예제 가운데 하나로 고객이 대출 신청을 할 수 있는 포트를 제공하는 간단한 대출 승인 웹서비스이다. 해당 서비스를 이용하는 고객은 개인정보와 요청 금액을 입력하여 대출 신청을 한다. 입력된 정보를 이용하여 대출 서비스(Loan Service)는 '대출 승인' 또는 '대출 불가' 메시지를 생성하는 간단한 프로세스를 수행한다.

요청된 액수와 요청한 사람의 신용 위험도에 따라 두 가지 다른 방식으로 대출 승인 결정이 내려진다. 대출 요청 금액이 적은 경우와 위험이 낮은 경우의 개인 대출 신청의 경우는 승인이 자동적이다. 금액이 크고, 위험도가 중간 이상인 경우 좀 더 자세하게 검토할 필요가 있다. 따라서 대출 신청을 처리하기 위하여 대출 서비스는 두 가지 다른 서비스에 의해 제공되는 기능을 사용한다. 대출 신청 금액이 적은 경우 신속한 대출 신청자 위험도 평가를 위해서 '위험 평가(Risk Assessment)' 서비스가 사용된다. 모든 사항을 포함하는 '대출 승인(Loan Approval)' 서비스(대출 전문가의 직접적인 관여가 대개 필요함)는 보다 자세한 신용도 평가를 하기 위해 사용된다.

4.1.2 파트너 링크 타입의 정의

파트너 링크 타입은 BPEL 프로세스와 관련 개체들 간의 상호작용을 정의하는데 사용된다. 여기에는 BPEL 프로세스가 호출하는 웹서비스와 BPEL 프로세스를 호출하는 클라이언트에 대한 정의가 포함된다. BPEL 프로세스 디자이너는 BPEL 프로세스와 관련 웹서비스와의 상호작용을 정의하기 위한 파트너 링크 타입 정의 기능을 제공해야 한다.

동기식으로 호출되는 서비스에 대한 포트 타입은 작업이 일방향으로 호출되므로 각 파트너 링크에는 하나의 역할(role)이 사용된다. 반대로 비동기식 서비스에 대한 포트 타입은 2개의 역할을 가진다. 일반적으로 바로 결과가 반환될 수 있는 작업에는 동기식 서비스를 사용하는 것이 좋고, 비교적 오랜 시간이 걸리는 작업에는 비동기적 서비스를 사용하는 것이 좋다. 비동기식 웹서비스를 호출하는 BPEL 프로세스 역시 비동기식으로 구현하는 것이 일반적이다.

4.1.3 BPEL 프로세스 정의

파트너 링크 타입 정의가 완료되면 BPEL 프로세스를 본격적으로 정의할 수 있다. 일반적으로 BPEL 프로세스는 클라이언트로부터 메시지를 받아 비즈니스 프로세스를 실행하는

형태로 동작한다.

프로세스 디자이너는 아래와 같은 타겟 네임스페이스와 각 서비스의 WSDL을 액세스하기 위한 네임스페이스 정의 기능을 제공해야 한다.

```
<process name="loanApprovalProcess"
   targetNamespace="http://acme.com/loanprocessing"
   xmlns="http://schemas.xmlsoap.org/ws/2003/03/business-process/"
   xmlns:lns="http://loans.org/wsdl/loan-approval"
   suppressJoinFailure="yes">
```

BPEL 프로세스 디자이너 프로그램은 BPEL 프로세스와 통신하는 다른 개체들을 정의하기 위해 파트너 링크를 모델링하는 기능을 지원해야 한다. 각 파트너 링크는 특정 파트너 링크 타입과 연결되어 있으며, 최대 두 가지 속성을 가진다. 'myRole'은 비즈니스 프로세스의 역할을 의미하고, 'partnerRole'은 파트너의 역할을 의미한다. 동기식 요청/응답 작업에서는 하나의 역할을 가지고, 비동기식 작업에서는 두 개의 역할이 사용된다.

```
<partnerLinks>
  <partnerLink name="customer"
     partnerLinkType="lns:loanPartnerLinkType"
     myRole="loanService"/>
  <partnerLink name="approver"
     partnerLinkType="lns:loanApprovalLinkType"
     partnerRole="approver"/>
  <partnerLink name="assessor"
     partnerLinkType="lns:riskAssessmentLinkType"
     partnerRole="assessor"/>
</partnerLinks>
```

BPEL 프로세스는 파트너 간 메시지 교환을 위해 변수를 사용한다. 변수는 비즈니스 프로세스의 상태를 구성하는 메시지들을 보관하는 수단이 된다. BPEL 프로세스 디자이너는 변수 정의 기능을 제공해야 한다. 각 변수는 범위(scope)를 가지며 전체 프로세스 범위에서 선언된 변수는 전역변수라 하고 그렇지 않은 변수를 지역변수라고 한다. 각 변수마다 타입을 지정해 주어야 하는데, 변수의 타입에는 WSDL 메시지 타입, XML 스키마 단순 타입(simple type), XML 스키마 엘리먼트 등이 있다.

프로세스 디자이너 프로그램은 BPEL 스펙에 정의되어 있는 기본 액티비티와 구조화 액티비티를 지원해야 한다.

BPEL4WS 1.1 버전에서의 액티비티 분류는 <표 1>과 같다. BPEL에서 사용되는 기본 액티비티 중에 외부 세계와 사용할 때 사용되는 기본 액티비티는 <receive>, <reply>, <invoke>이다.

구조화 액티비티는 프로세스 실행시 액티비티가 수행되는 순서를 규정한다. 액티비티 사이에서 순서 제어는 sequence, switch와 while을 사용하여 정의할 수 있다. 액티비티들 사이의 동시성과 동기성은 flow에 의하여 정의된다. 외부 이벤트에 기초한 비결정론적인(non-deterministic) 선택은 pick에 의하여 제공된다.

<표 1> BPEL4WS의 액티비티 분류

액티비티 분류	종류
기본 액티비티	<receive>, <reply>, <invoke>, <assign>, <throw>, <wait>, <empty>
구조화 액티비티	<sequence>, <switch>, <while>, <pick>, <flow>
기타 특수 액티티비	<scope>, <compensate>, <terminate>

BPEL 프로세스 디자이너 프로그램에서는 구조화 액티비티의 재귀적인 설계가 지원되어야 한다. 즉, 구조화 액티비티는 중첩되어 사용되거나 다양하게 조합하여 사용될 수 있다.

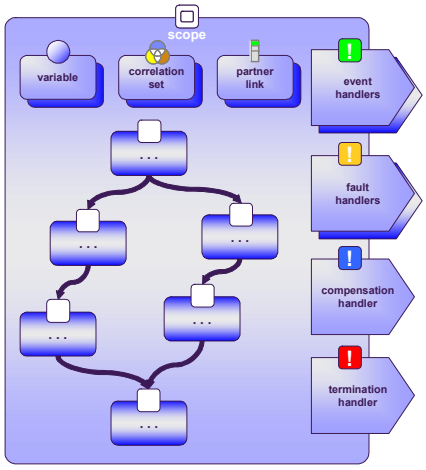
BPEL 프로세스 모델에서 각 액티비티의 행위 문맥(behavior context)은 스코프(범위, scope)에 의해 제공된다. BPEL 프로세스 디자이너 프로그램은 스코프 정의 기능과 이벤트 처리기능을 제공해야 한다.

하나의 스코프는 오류 핸들러, 이벤트 핸들러, 보상 핸들러, 데이터 변수, 상호연관 집합(correlation sets)을 제공할 수 있다.

상호연관 집합은 상호연관된 그룹 내의 모든 메시지에 의해 공유되는 특성들의 집합을 말하는데, BPEL4WS에서는 상호연관 시나리오를 통해 서비스 인스턴스 내에서 상호연관된 오퍼레이션 그룹을 지정하는 선언적 메커니즘을 제공한다. 다시 말해 파트너와의 비동기 메시지를 주고받을 때 여러 번의 비동기 메시지를 하나의 대화로 묶어야 할 필요성이 있을 경우 각 통신에 주고받는 메시지가 동일한 대화에 속해 있음을 나타내는 것을 메시지 상호연관이라고 한다. 상호연관 집합을 통해 프로세스들이 상태 지속형 대화(stateful conversations)에 참여할 수 있다. BPEL 프로세스 디자이너는 이러한 메시지 상호연관 집합 정의 기능을 제공해야 한다.

이벤트 핸들러(event handler)는 메시지 이벤트나 시간 이벤트를 처리한다. 오류 핸들러(fault handler)는 다른 예외적 상황을 다루기 위한 핸들러이다. 보상 핸들러(compensation handler)는 이미 완료된 액티비티의 영속적 실행 결과를 취소하기 위한 핸들러를 말한다. 종

료 핸들러(termination handler)는 강제 스코프 종료를 처리하는 핸들러이다. <그림 8>에 스코프와 핸들러들의 관계를 나타내었다.



<그림 8> 스코프와 핸들러

4.2 프로세스 공개 및 시작 모듈

BPEL 프로세스 디자이너를 이용해서 비즈니스 프로세스를 정의하고 나면 작성된 프로세스를 컴파일하여 공개(Deploy)하고 테스트하여 실행시킨다.

프로세스 공개는 BPEL로 작성된 프로세스 실행을 위해 서버에 정의된 프로세스를 공개하는 과정으로 프로세스의 실행을 관장하는 엔진이 프로세스를 제대로 동작시키기 위해 필요한 기능이다. 이를 위해 프로세스를 서버에 공개하는 기능을 제공하는 모듈이 필요하다. 많은 경우 프로세스 공개 기능은 프로세스 디자이너에 포함되어 제공되기도 한다.

프로세스 공개를 효율적으로 하기 위해서 프로세스 서술자(descriptor)를 정의해서 사용할 수 있다. 프로세스 서술자는 BPEL 표준 스펙에 정의되어 있지 않으며, 따라서 각 솔루션마다 다른 방식으로 구현된다. BPEL 표준에 따라 정의된 프로세스를 각기 다른 BPEL 엔진에 적용해야 하는 경우 프로세스 서술자만 수정하면 된다.

예를 들어, 오라클의 프로세스 서술자는 XML 파일로 BPEL 프로세스에 대한 자세한 정보(BPEL 소스 파일명, BPEL 프로세스 명(id), 파트너 링크 웹서비스의 WSDL 위치정보, 기타 구성 정보 등)를 구현하고 있다.

프로세스 서술자가 준비되면 BPEL 엔진을 사용하여 프로세스를 시작할 준비가 완료된 상태가 된다. 서버에 프로세스를 공개하기 위해서는 복잡한 컴파일 작업과 공개(deploy) 작업을 위한 환경설정이 필요한데, Ant 유틸리티와 같은 빌드(build) 유틸리티를 포함하게 되면 이러한 작업이 편리해 진다.

Ant 유틸리티와 같은 빌드 유틸리티를 이용해서 서버에 성공적으로 BPEL 프로세스를 공개하고 나면 BPEL 프로세스를 실행할 준비가 끝나게 된다.

프로세스 시작 모듈(Initiator)은 BPEL 서버에 공개된 프로세스 실행을 시작하기 위한 프로그램을 말한다. BPEL 표준 스펙에 따라 구현된 BPM 솔루션은 설계된 프로세스를 서버에서 실행하여 프로세스 인스턴스를 생성시키는 기능을 포함해야 한다. 프로세스를 실행시키기 위해서는 사용자에게 HTML 폼과 같은 GUI를 제공하는 것이 바람직하다. 제공된 사용자 인터페이스를 이용하여 필요한 데이터를 입력하여 프로세스를 시작할 수 있다.

4.3 BPEL 엔진

BPEL 엔진은 정의된 프로세스 모델을 해석하여 런타임 프로세스 실행을 관리하는 서버로써 BPEL 컴포넌트 가운데 가장 핵심적인 요소라 할 수 있다. BPEL 엔진은 프로세스 해석 및 흐름 처리, XML 처리 기능, 프로세스 실행 상태 관리, 서비스 호출 처리 등의 기능을 수행한다.

4.3.1 BPEL 프로세스 해석 및 흐름 제어

BPEL 엔진은 프로세스 디자이너 부분에서 설명한 BPEL 프로세스 정의 관련 내용을 이해하여 정확하게 프로세스 모델을 해석하고 흐름(flow)을 제어할 수 있어야 한다. 따라서, BPEL 엔진을 구현하기 위해서는 프로세스 디자이너 부분에서 설명한 내용에 대한 이해가 필요하다.

BPEL 엔진의 프로세스 해석 및 흐름 제어 기능은 정의된 프로세스의 컴파일 기능과 실행 기능으로 구성된다. 프로세스 디자이너를 통해 작성된 프로세스는 BPEL 컴파일러를 사용하여 컴파일된다. J2EE 플랫폼의 경우 컴파일하면 자바코드를 생성하게 된다. BPEL 엔진은 프로세스 모델의 컴파일 기능과 컴파일된 BPEL 코드를 실행할 수 있는 환경을 제공해야 한다. BPEL 코드 실행 환경을 위해 실시간 트랜잭션 기능이 제공되어야 하는데 이는 EJB 환경을 통해 자동으로 제공될 수 있다.

장기 트랜잭션의 효율적인 수행을 위해 프로세스를 구성하는 액티비티의 특성을 체크하여 자동으로 프로세스를 동면(hibernation) 상태로 만드는 기능이나 프로세스 실행 상태 정보를 제공하기 위한 기능도 제공할 수 있다.

4.3.2 XML 처리 기능

BPEL 엔진은 런타임 XML Schema 및 XPath 처리 기능을 제공해야 한다. BPEL 표준 스펙에서 요구하는 XPath 표현식과 Xpath 함수 처리 기능, XML Schema 처리 기능, WSDL 메시지 처리 기능, XML 문서 인스턴스 생성, XML Schema 구문 적합성 확인 기능 등이 프로세스 실행 관리를 위해서 필요하다.

4.3.3 프로세스 실행 상태 및 메시지 영속성 (Persistence) 관리

BPEL 엔진은 프로세스 인스턴스와 액티비티 인스턴스의 상태 관리 기능을 제공해야 한다. 액티비티들로 구성되는 프로세스가 실행되면 프로세스 인스턴스가 생성된다. 프로세스 인스턴스는 애플리케이션 레벨 메시지를 수신함으로써 생성되고 프로세스 인스턴스의 마지막 액티비티가 종료되면 그 실행이 완료된다. 프로세스 인스턴스는 명시적 식별자(identifier token)보다는 특정한 애플리케이션 메시지 필드값인 상호연관 집합(correlation sets)에 의해 식별된다.

각 액티비티는 다음과 같은 상태(state)를 가질 수 있다.

- INACTIVE: 프로세스가 시작되었으나 아직 비활성화 되어 있는 액티비티 상태
- READY: 실행할 준비가 된 상태
- EXECUTING: 현재 실행중인 상태
- FINISHED: 오류 없이 정상적으로 종료한 상태
- FAILURE: 오류가 발생하여 비정상적으로 종료한 상태
- TERMINATED: 강제 종료된 상태

BPEL 엔진은 장기간 수행되는 비즈니스 프로세스의 효율적인 관리를 위해 프로세스의 상태를 저장하고, 프로세스 실행 시 사용되는 메시지의 내용을 저장하는 기능을 제공해야 한다. 이를 위해 데이터베이스와 파일 시스템을 사용할 수 있다. 예를 들어, 프로세스 상태와 각 변수의 상태를 데이터베이스에 저장하고, 메시지의 내용을 별도의 파일 시스템에 내부적 규칙에 따라 저장할 수 있다.

4.3.4 메시지 라우팅 및 서비스 호출 지원

BPEL 엔진은 수신 메시지 라우팅 기능과 파트너 웹서비스 호출 기능을 제공해야 한다.

수신 메시지로부터 상호연관 값을 추출하여 타겟 프로세스 인스턴스를 식별하는데 사용하는 메커니즘을 수신 메시지 라우팅이라 한다. 이 때 상호연관값과 프로세스 인스턴스 매칭을 위한 알고리즘이 필요하다. 특정 프로세스 인스턴스와 매칭되는 메시지는 해당 인스턴스의 입력 메시지 큐(input queue)에 저장된다.

외부 파트너 웹서비스의 호출은 IIOP, JMS, 고유 자바 호출(native Java calls) 등의 J2EE 프로토콜을 사용할 수 있다. 외부 서비스의 호출을 지원하는 데에는 웹서비스 호출 프레임워크(Web Services Invocation Framework)를 사용할 수 있다.

4.4 프로세스 모니터링 및 분석 프로그램

BPEL 엔진을 통해 실행하고 있는 프로세스의 실행상태를 브라우징하는 프로그램을 프로세스 모니터링 프로그램이라고 한다. 프로세스 모니터링 프로그램은 사용자 편의성을 높이기 위해 프로세스 인스턴스 실행상태의 비주얼한 흐름을 GUI(Graphical User Interface) 형태로 제공하는 것이 바람직하다. 모니터링 프로그램은 프로세스 인스턴스 실행 상태(진행중, 완료됨, 취소됨, 중지됨 등)와 프로세스 실행 과정(어느 액티비티가 진행되고 있는가)에 대한 정보를 제공해야 한다.

모니터링 기능과 함께 프로세스 분석 기능과 시뮬레이션 기능이 제공될 수 있다. 프로세스 분석 기능은 설계한 프로세스의 성과를 분석하여 보고함으로써 효과적인 의사결정을 수행할 수 있도록 해 준다. 시뮬레이션 기능은 실제로 프로세스를 실행하기 이전에 모의시행해 봄으로써 프로세스의 로직 상 문제점을 사전에 분석할 수 있고, 여러 프로세스 대안들을 비교 평가할 수 있다.

5. 결론

본 연구에서는 BPEL4WS 표준을 지원하는 BPM 솔루션이 갖추어야 하는 참조 아키텍처를 제시하고, 참조 아키텍처를 구성하는 핵심 컴포넌트의 구현 가이드라인을 개발하였다. 본 연구에서 제시된 참조 아키텍처는 BPEL4WS 표준을 지원하는 국내외 BPM 솔루션들을 분석하여 BPEL4WS 구현에 반드시 필요한 공통적 핵심 컴포넌트를 추출한 것이다.

BPEL4WS 지원 BPM 솔루션을 구성하는 여러 요소들 가운데 가장 핵심적인 컴포넌트라 할 수 있는 BPEL 엔진, 프로세스 디자이너, 프로세스 공개 모듈 및 시작 모듈, 프로세스 모니터링 모듈을 중심으로 각 컴포넌트들이 갖추어야 할 기능을 자세히 설명하고 프로세스 정의 단계에서 출발하여 정의된 프로세

스의 서버로의 공개, 프로세스 실행, 모니터링으로 구성되는 일련의 시나리오에 따라 핵심 컴포넌트 구현 지침을 제시하였다.

본 연구의 결과물을 통해, BPM 솔루션 개발업체에서는 BPEL4WS 표준 적용 및 개발 시의 시행착오를 줄임으로써 보다 효율적인 개발과 적용이 가능할 것으로 기대된다. 아울러 본 연구의 결과물은 기업들이 참조할 수 있도록 가이드라인 혹은 지침서 형태의 표준안으로 작성하여 전자상거래표준화통합포럼(ECIF)과 정보통신기술협회(TTA)의 단체 표준으로 제정될 예정으로 향후 많은 관련 기업이 활용할 수 있을 것으로 예상된다.

참고문헌

[1] 한국전산원, 비즈니스 프로세스 관리기술 표준적용을 위한 지침 연구, 2004
[2] 한국전산원, 비즈니스 프로세스 관리 기술 표준의 BPMS 적용에 관한 지침 연구, 2005
[3] BPEL Resource Guide, http://bpelsource.com
[4] Dieter König, Web Services Business Process Execution Language (WS-BPEL), OASIS Open Standards Day XTech 2005 Conference Amsterdam, May 24, 2005
[5] P. Wohed, W.M.P. van der Aalst, M. Dumas, and A. H. M. ter Hofstede, Pattern Based Analysis of BPEL4WS, Technical Report FIT-TR-2002-4, QUT
[6] Wikipedia, http://en.wikipedia.org/wiki/BPEL
[7] Petia Wohed, Wil M.P. van der Aalst, Marlon Dumas, Arthur H.M. ter Hofstede, Pattern Based Analysis of BPEL4WS, Technical Report FIT-TR-2002-04, 2002
[8] Kimberly Hill, Enterprises Face BPM Standards Maze, October 21, 2004 http://www.cio-today.com/story.xhtml?story_id=27761