韓國IT서비스學會誌
第5卷 第2號
2006年 8月

비즈니스 프로세스 관리 표준 도입 전략 및 지침 개발 : 비즈니스 프로세스 실행 언어를 중심으로*

김 동 수**

Development of Adoption Strategy and Guideline of Business Process Management Standards : Focusing on Business Process Execution Language*

Dongsoo Kim**

■ Abstract ■

The objectives of this study is to develop a strategy for the adoption of BPM(Business Process Management) standards and an implementation guideline of the BPM standard for BPM solution developers focusing on BPEL(Business Process Execution Language) which is regarded as the most important BPM standard. In the heterogeneous and distributed IT environments, every type of enterprise software requires standards to enhance interoperability. BPMS(Business Process Management System), which is a type of enterprise software requires BPM standards such as BPEL(Business Process Execution Language), BPMN(Business Process Modeling and Notation), BPQL(Business Process Query Language) and so on to achieve multi-system interoperability and component interoperability with their BPM solutions. It is quite helpful to provide the adoption strategy concerning BPM standards for each type of BPM solution vendors who need the BPM standards. Since the BPEL is conceived as the most important BPM standard and widely adopted by many BPM vendors, we have proposed a reference architecture for BPEL implementation and also developed the detail implementation guideline of core components of the BPM system supporting the BPEL standard. Using the strategy and implementation guideline proposed in this work, BPM solution vendors can establish their own standard adoption strategy and they can also develop their BPM solutions supporting the BPM standards more efficiently.

Keyword : BPM(Business Process Management), Standard Adoption, BPM Solution, BPEL

* 본 연구는 숭실대학교 교내연구비 지원으로 이루어졌음.
** 숭실대학교 산업・정보시스템공학과 조교수

1. 서 론

기업 활동의 근간을 이루는 비즈니스 프로세스들을 투명하게 관리하고 통제하기 위해 비즈니스 프로세스 관리(BPM : Business Process Management)가 주목받고 있다[1]. 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 지칭하고 있다[3]. 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. 종합적으로 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[2].

BPM 솔루션 업체의 특성이 다양하고 그 수도 많아짐에 따라 각 솔루션 업체 전용의 모델링과 관리 기술 구조로 인해 제품 간 상호운용성 확보가 어려워지고, 프로세스 공유도 불가능해 지는 문제점 등을 해결하기 위해 프로세스 정의 및 구현 기술에 관한 표준의 필요성이 제기되었으며[4], 프로세스 모델링, 모니터링, 기업 간 협업, 트랜잭션, 프로세스 실행, 프로세스 질의 등에 관한 국제 표준들이 발표되고 있다[12, 14].

BPM을 도입하고자 하는 조직은 비즈니스 프로세스 수명 주기 전체에 걸쳐 비즈니스 프로세스를 관리하고자 하므로, BPM 솔루션 제공 업체는 프로세스 정의로부터 개선을 위한 분석 기능에 이르기까지의 다양한 기능을 제공해 주어야 한다. 프로세스 모델링 → 프로세스 검증 및 최적화 → 프로세스 실행과 운영 → 모니터링과 통제 → 측정과 분석 → 프로세스 개선에 이르는 일련의 프로세스 수명 주기가 반복되면서 비즈니스 프로세스를 관리해야 하므로 BPM 솔루션은 이러한 과정을 지원하기 위한 표준의 도입이 필요하다. 현재 이들 기능 중 주로 프로세스 모델링, 프로세스 실행과 운영, 모니터링과 통제에 초점을 맞추고 국제 표준화가 진행되고 있다. 각 활동별 관련 국제 표준들을 살펴보면 다음과 같다.

- 프로세스 모델링 : BPMN(Business Process Modeling and Notation), IDEF0, IDEF3, PSL(Process Specification Language), UML 2.0 등
- 프로세스 실행과 운영 : BPEL(Business Process Execution Language), BPML(Business Process Modeling Language), BPSS(Business Process Specification Schema), XPDL(XML Process Definition Language) 등
- 모니터링과 통제 : BPQL(Business Process Query Language), BAML(Business Activity Monitoring Language) 등

모든 표준의 목적이 상호운용성의 확보에 있는 것처럼 BPM 표준이 등장한 이유도 시스템 간 상호운용성 제공에 있다[6]. BPM 시스템을 구성하는 구성요소들은 모델링, 실행 및 운영, 질의 및 모니터링 등의 여러 기능으로 나누어져 있다. BPM 표준 도입을 통한 상호운용성은 여러 BPM 시스템 간 상호운용성 측면과 각 구성요소들 사이의 상호운용성 측면으로 나누어 생각할 수 있다. 특히, BPM 프로세스 모델의 표준화를 통해 비즈니스 프로세스의 재사용성이 높일 수 있고 기업의 프로세스 지식을 축적하고 공유하는데 도움을 줄 수 있다.

BPM 솔루션 제공업체 관점에서는 표준 채택이 자사 솔루션의 마케팅에 많은 영향을 미칠 수 있다. 따라서 표준의 발전 동향을 파악하고, 다른 경쟁 솔루션들이 표준을 채택하고 있는 동향을 파악하는 것이 중요한 문제이다. 또한, 표준은 그 특성상 방대한 양을 다루면서도, 기업 솔루션으로서 갖추어야 할 모든 것을 규정하지 못하는 특징이 있

다. 그러므로 주요 BPM 국제 표준들의 개념과 솔루션에의 구체적인 적용 범위와 한계 등을 인지하는 것은 BPM 업계에서 표준을 채택하는데 있어서 매우 중요한 사항이다. 더욱이 국제 표준화를 주도하고 있지 못한 국내 BPM 업체들에게 있어서 이러한 점은 매우 중요한 문제가 아닐 수 없다.

본 연구에서는 BPEL, BPMN, BPQL 등 주요 BPM 표준의 개념과 현황을 소개하고, 자사 솔루션 개발에 적용하고자 하는 BPM 벤더 유형별로 참조할 수 있는 BPM 표준 도입 전략을 제시하였다. 프로세스 모델링 측면, 프로세스 실행과 운영, 프로세스 질의 등에 관한 BPM 표준들 가운데 가장 중요한 표준으로 BPEL(Business Process Execution Language)이 꼽히고 있다. 다시 말해 BPM 표준 가운데 프로세스 실행과 운영을 위한 BPEL 표준이 시장에서 가장 수용도가 높고 그 중요성이 크다. 이러한 배경에서 본 논문에서는 비즈니스 프로세스 실행 언어(BPEL)를 구현하기 위한 구현 가이드라인(또는 지침)을 개발하였다. 본 논문에서 제시한 프로세스 실행 언어 구현 지침에서는 프로세스 설계, 실행과 운영 측면에서 BPEL 표준을 활용하여 BPM 솔루션을 어떻게 구성해야 하는지에 관한 참조 아키텍처를 제시하고, 각 구성요소별로 어떠한 핵심 기능이 제공되어야 하는지를 상술하고 있다.

BPM의 세 가지 측면인 실행, 모델링, 질의처리 관점에서의 중요 표준에 대한 상세한 가이드라인은 한국전산원의 연구보고서[2]에 제시되어 있으므로, BPMN, BPQL 등 기타 BPM 표준별 상세 가이드라인은 이를 참조하면 된다. 각 표준별 구현 지침은 BPM의 세 가지 측면을 다루고 있으므로 상호 보완적인 가이드라인이라고 할 수 있다. 본 연구에서 제안안 내용은 BPMN, BPQL 구현지침과 함께 한국정보통신기술협회(TTA)의 단체표준으로 제안된 상태이다. TTA의 전자거래프로젝트그룹에서는 '비즈니스 프로세스 관리기술의 BPMS 적용 지침', '비즈니스 프로세스 모델링 표기법' 등 BPM 관련 표준화 작업을 수행하고 있다[17].

본 연구의 결과물을 활용하여 BPM 솔루션 공급업체들은 자사 솔루션의 특징에 따른 표준 적용 전략을 수립할 수 있다. 또한, BPM 표준 적용 및 개발 시의 시행착오를 줄임으로써 보다 효율적인 개발과 적용이 가능하다. 특히, 본 연구에서 제시한 비즈니스 프로세스 실행 언어 구축을 위한 가이드라인은 BPM 시스템 개발 기업이 BPEL 표준을 적용하여 솔루션을 효율적으로 구축하는데 활용될 수 있다.

본 논문은 다음과 같이 구성된다. 2장에서는 BPM 표준화 동향을 기술하였고, 3장에서는 BPM 공급업체의 유형별 장단점에 근거한 BPM 표준 도입 전략을 제시하였다. 4장에서는 BPEL 구현 참조 아키텍처를 제시하고, 참조 아키텍처의 핵심 컴포넌트의 기능 요구사항 및 구현지침을 5장에 기술하였다. 끝으로 6장에는 본 연구의 결론을 기술하였다.

2. BPM 표준화 동향

본 장에서는 BPEL, BPMN, BPQL 등 주요 BPM 표준화 현황을 간단히 소개하고, 상세 구현 지침 개발의 대상인 BPEL 표준의 현황과 주요 내용을 설명하였다.

2.1 BPM 표준화 개요

포레스터 리서치의 자료에 의하면 가장 중요한 BPM 표준으로 BPEL과 BPMN이 선정되고 있다[12]. 동 자료에서는 BPM 시장에서의 점유율 확보를 위해 경합하고 있는 표준들로 다음의 표준을 소개하고 있는데 이들 가운데 BPEL이 가장 중요한 표준으로 꼽히고 있다.

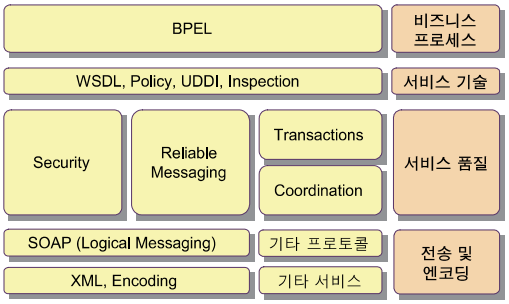
- BPEL(비즈니스 프로세스 실행 언어) : 웹 서비스를 사용해 비즈니스 프로세스를 조정하는 실행언어로 IBM, Microsoft 및 BEA에 의해 최초로 제안되었으며, 현재는 대다수의 BPM 솔루션

업체들에 의해 최고의 BPM 표준으로 인정받고 있음

- BPML(비즈니스 프로세스 모델링 언어) : Intalio가 도입하고, BPMI.org가 후원한 비즈니스 프로세스 모델링용 메타언어로 BPEL과 정면으로 경쟁하고 있으나 BPEL로 인해 빛을 보지 못하는 것으로 평가됨[7]
- BPMN(비즈니스 프로세스 관리 표기법) : 비즈니스 프로세스 모델링을 위한 표준 표기법으로 BPEL 등 다른 BPM 표준들과 보완적인 관계임. BPMN은 그래픽 기반의 프로세스 모델링 방법론으로서 모든 비즈니스 프로세스 관련자들이 쉽게 프로세스를 이해할 수 있는 모델링 요소들을 제공함[8, 9, 13]
- BPQL(비즈니스 프로세스 질의 언어) : 범용의 BPMS에 대한 표준적인 질의 및 조작 언어의 개념에서 출발한 것이 BPQL임. BPMI.org의 BPQL 표준화 작업이 시작된지 오랜 시간이 지났으나 여전히 공식적인 표준안 발표가 미루어지고 있으며, 드래프트 버전 또한 제시되고 있지 않은 상태임
- Wf-XML : WfMC에서 제정한 표준으로 OASIS ASAP(Asynchronous Service Access Protocol) 기반으로 BPM 엔진 간 상호운용성을 제공
- XPDL : 전체 프로세스를 설명하는 비즈니스 프로세스 정의 언어로써 모델링 도구들과 BPM 시스템들 간의 상호교환 표준으로 개발됨. 일부 BPM 벤더들이 채택하고 있지만, 향후 BPEL로 대체될 것으로 전망

2.2 BPEL 표준화 동향 및 주요 내용

BPEL은 2003년 5월 1.1 버전이 발표된 이후 국제 표준화 기구인 OASIS의 WSBPEL 기술위원회에서 2.0 버전의 스펙을 만들기 위해 표준화 작업을 계속하고 있다. 본 논문에서는 명시적으로 1.1 버전의 BPEL 표준을 의미하기 위해 BPEL4WS로 표기하였으며, 2.0 버전을 의미하는 경우에는 WS-BPEL이라는 용어를 사용하였다. 그 외 일반적인 비즈니스 프로세스 실행언어를 지칭할 때에는 BPEL을 사용하였다.



[그림 1] WS-* 스택 상에서의 BPEL 위치

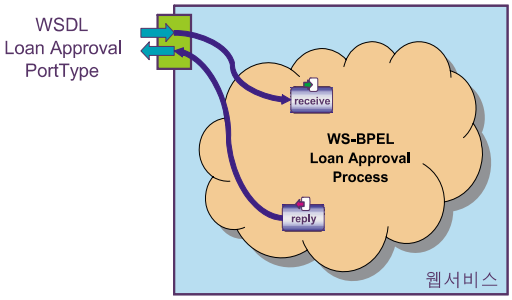
[그림 1]은 웹 서비스 표준 스택 상에서 BPEL의 위치를 보여 주고 있다[10]. BPEL은 웹 서비스 구성(composition) 표준의 기초가 되고 있다. 함께 발표된 웹 서비스 통신 규약인 WS-Coordination, WS-Transaction과 함께 기업 간 프로세스, 즉 협력업체 및 고객 전반에 걸쳐 실행되는 다중 비즈니스 프로세스 및 트랜잭션 서비스를 신뢰성 있게 통합할 수 있도록 한다.

BPEL은 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.

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

BPEL에서는 프로세스와 그 파트너 모두가 WSDL

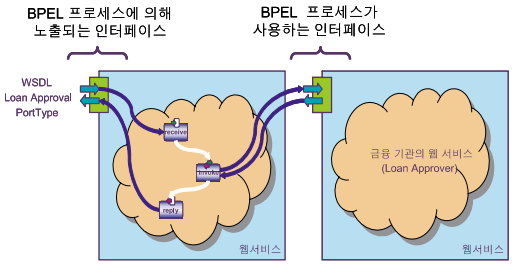
서비스로 모델링된다. 비즈니스 프로세스는 프로세스 인스턴스와 그 파트너들 사이의 상호작용을 어떻게 조정할(coordinate) 것인가를 정의한다. 이런 의미에서 BPEL 프로세스 정의는 다수의 WSDL 서비스를 제공하거나 사용한다. 또, BPEL 프로세스 정의는 프로세스에 참여하는 파트너와 자원과 관련한 프로세스 인스턴스의 행동과 상호작용에 대한 표현을 웹 서비스 인터페이스를 통해 제공한다.



[그림 2] BPEL 프로세스의 WSDL 서비스화

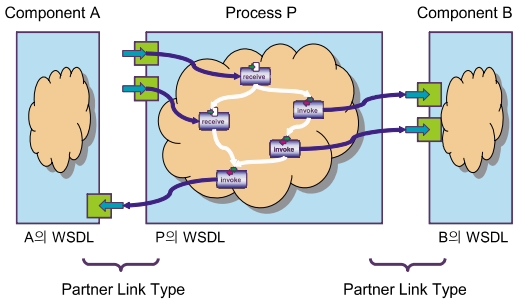
[그림 2]는 BPEL 프로세스 모델과 WDSL의 관계를 보여 주고 있다. 이 그림에서 보는 것처럼 BPEL 프로세스 모델은 WDSL 포트 타입을 통해 웹 서비스로 제공된다. 이 때 메시지 교환은 WDSL 오퍼레이션에 대응된다.

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]는 BPEL을 이용한 웹 서비스 구성을 보여 주고 있다. 프로세스 P와 웹 서비스 컴포넌트 A, B가 WSDL 인터페이스를 활용하여 웹 서비스를 구성하고 있다. BPEL 프로세스는 메시지를 받기도 하고 각 컴포넌트의 서비스를 호출하기도 한다.



[그림 4] BPEL을 이용한 웹 서비스 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. BPM 표준 도입 전략

본 장에서는 BPM 솔루션 공급업체 유형을 몇 가지로 구분하여 각 유형별로 비즈니스 프로세스의 표준을 BPMS에 적용하기 위한 전략적인 방안을 모색해 보았다.

3.1 BPM 솔루션 기업의 유형 분류 및 특장점

BPM 솔루션 기업의 유형화를 위해 해당 기업들의 기술적 배경과 기존 제품 특성을 유형화를 위한

기준으로 사용하였다. 가트너의 BPM 공급업체 유형 분류 등 관련 분류를 참조하여 다음의 6개 유형으로 BPM 솔루션 기업을 구분하였다.

1) EAI 제품으로부터 상향식(bottom-up) 형태로 성장한 BPM 솔루션 기업
2) 워크플로우 제품으로부터 하향식(top-down) 형태로 성장한 BPM 솔루션 기업
3) 기존의 미들웨어 제품으로부터 확장한 BPM 솔루션 기업
4) 특화된 전문 유틸리티 제공 업체의 BPM
5) ERP 및 기업용 애플리케이션 공급 업체의 BPM
6) 웹 서비스, XML 및 B2B 표준 기술을 활용한 새로운 BPM 솔루션 업체

위의 구분에서 1)과 2)번 유형은 BPM 시장에서 일반적으로 제품군을 크게 양분할 때 사용하는 기준이며, 3)번에서 6)번까지는 최근 몇 년 사이에 BPM 시장에 신규로 진입하고 있는 제품들의 기술적인 특성을 반영하여 추가된 유형이다.

본 연구에서 제시한 BPM 솔루션 기업 유형 구분은 각 유형이 모든 BPM 제품들을 포함하지 않을 수도 있으며, 특정 BPM 제품이 정확히 하나의 유형에 소속됨을 보장하는 것은 아니다. 그러나 일반적으로 BPM 솔루션을 구별할 때 자주 인용되는 구분과 상당히 유사하므로 국내 관련 업계에서 이에 대한 이해와 수용이 상대적으로 용이할 것으로 예상된다.

앞서 제시한 6개의 BPM 업체 유형별로 전반적인 특징과 장단점을 살펴보기 위해 레거시 시스템 통합, 대용량 트랜잭션 지원, 비즈니스 프로세스(BP)의 중점 관리 및 지원 등 12개 항목의 기준을 설정하고, 6개의 업체 유형별 강약점을 <표 1>에 정리하였다. 표에 나타난 내용이 해당 범주에 속하는 모든 업체에 그대로 적용되는 규칙을 나타내는 것은 아니기 때문에, 업체별로 일정한 범위 내에서의 변동사항이 있을 수 있다. <표 1>에서 제시한 강약점은 절대적인 평가라 할 수 없으며, 일반적으로 인식되거나 지적되고 있는 사항들을 기초로 하고 있다.

〈표 1〉 BPM 업체 유형별 특징 및 장단점

강 : ◎ 중 : ● 약 : ○

항목 \ 유형	EAI의 bottom-up형	Workflow의 top-down형	미들웨어의 middle-out형	특화된 유틸리티형	ERP 및 기업 App형	WS, XML 및 B2B 신규형
레거시/시스템 통합	◎	○	●	○	◎	●
대용량 트랜잭션 지원	◎	●	◎	◎	◎	●
BP의 중점 관리/지원	○	◎	●	○	◎	◎
고객층/레퍼런스	◎	◎	◎	○	◎	●
최신 기술 요소 지원	○	●	○	●	○	◎
표준 지원	●	◎	◎	●	○	◎
안정성/통합성	◎	●	◎	○	●	●
비주얼 환경	○	◎	○	●	●	◎
다양한 전문 기능	○	○	●	◎	●	◎
높은 비용 투자	◎	●	●	○	◎	○
자료/시스템 접근성	●	◎	◎	◎	○	◎
사용자 Interaction	○	◎	○	◎	●	◎

주) 표의 내용은 절대적인 것이 아니며, 업체별로 범위 내에서 변동사항을 가질 수 있음.

3.2 BPM 업체 유형별 표준 도입 전략

본 절에서는 여섯 가지로 구분된 BPM 업체 유형별로 어떻게 BPEL, BPMN 및 BPQL 등 비즈니스 프로세스 관리 시스템 관련 표준을 도입해 나가야 하는지에 대한 전략을 제시하였다. 즉, 앞서 설명한 업체 유형별 장단점에 입각해서 BPM 솔루션 업체가 상대적으로 취약한 부분을 극복하고 자사의 장점을 강화하기 위한 BPM 표준 도입 전략을 제시하였다. 여기서 제시된 전략은 모든 업체들이 반드시 따라야 할 성질의 것이기 보다는 자사의 표준 도입 전략을 수립할 때 참고할 수 있는 유용한 지침으로 활용할 수 있다. 본 연구에서 제시한 도입 전략을 활용함에 있어 자사의 BPMS를 어떤 유형으로 규정할 것인가에 대한 명확한 개념 정립이 필요하다.

3.2.1 EAI 제품으로부터 성장한 BPM 업체

이미 자사의 독자적인 방식으로 전체 시스템이 구현되어 작동 중인 상황에서 새로운 표준으로 전체 구조를 재구성하는 방식은 상당한 비용과 개발 기간을 요하는 작업이 된다. 따라서 이 유형의 BPM 솔루션 개발업체는 새로운 표준 인터페이스를 기존의 기반 구조 위에 추가하는 방식을 고려해 볼 수 있다. 이렇게 제공된 인터페이스를 통해 표준 지원과 자료 및 시스템 접근성을 확보할 수 있을 것이다. 이러한 방식은 특히, 기존에 자사의 시스템을 사용 중에 있는 많은 고객층들의 기존 투자를 살릴 수 있다는 점에서도 의미 있는 방법이다.

기존의 EAI 제품들은 대용량 데이터의 중계 및 변환에 초점을 두고 있으므로, 사용자 및 시스템과의 상호작용 요소가 많은 비즈니스 프로세스의 관리와 지원에는 취약한 구조이다. 데이터의 중계 및 변환은 단시간에 많은 이벤트와 처리가 발생하는 반면에 BP의 경우 장시간에 걸쳐서 서서히 진행되는 특징을 가진다. 따라서 BPEL과 같은 비즈니스 프로세스 표준을 지원하기 위해서는 기존에 갖추고 있는 하위 수준의 시간 단위에 특화된 실행 엔진 외에도 보다 상위 수준의 시간 단위에 적합한 실행 엔진의 개발이 필요하다.

EAI 제품군들의 기존 특징이 기업 정보 시스템의 후단에서 백그라운드로 실행되는 성격이 강하기 때문에, 실제 프로세스의 관리 및 모니터링을 위한 사용자 GUI 환경의 지원에 취약한 면을 보이게 된다. 특히 비즈니스 프로세스의 모델링 및 모니터링과 관련하여서는 사용자 상호작용도 함께 강화한 보다 높은 수준의 그래픽 환경이 필요하다. 이를 위해서는 BPMN과 같은 그래픽 표기를 완벽히 지원하는 것이 중요하다.

3.2.2 워크플로우 제품으로부터 성장한 BPM 업체

이 유형의 업체에서는 기존에 사용하던 비즈니스 프로세스 정의와 새로운 표준과의 호환성을 확보하는 데에 많은 어려움이 있을 것으로 예상된다. 매우 복잡한 형태의 프로세스를 독자적인 방식이나 XPDL과 같은 기존의 표준을 사용하여 구현한 상태이므로, 기존 프로세스 모델을 BPEL과 같은 새로운 표준으로 전환하는 과정이 기술적으로 난해하다고 볼 수 있다.

프로세스 디자이너의 경우 기존에 제공하던 디자이너 환경 외에 BPMN의 형태로 제공되는 디자이너가 추가될 필요가 있다. 사용자의 요구에 따라 BPEL 형태의 표준으로 기술된 프로세스 정의를 BPMN 디자이너가 지원하여 생성할 수도 있으나, 내부적으로 기존에 사용되어 오던 프로세스 정의 방식으로도 생성할 수 있는 방법이 보다 중요하다. 이는 만일 그렇지 않을 경우 표준을 위해 새로운 프로세스 엔진을 추가로 작성하여 두 개의 실행 엔진을 운영해야 하는 상황이 발생할 수도 있기 때문이다.

이 유형의 기업의 경우 BPQL 표준을 지원하는 것은 상대적으로 중요한 의미를 지닌다. 이미 비즈니스 프로세스의 정의와 실행을 위한 기본 요소들을 충분히 갖추고 있다는 점에서 볼 때, BPQL 표준의 지원이 다른 유형의 업체들과 차별화라는 점

에서 중요하다. 아직까지 BPQL 표준이 제시되고 있지 않은 상황이므로 독자적인 BPQL 기능을 제공하되, 많은 워크플로우 및 BPM 연구들에서 공통적으로 강조되는 기본 요소들을 중심으로 제한할 필요가 있다. 필요 이상의 추가 기능이 오히려 BPQL 표준의 발표로 인해 걸림돌이 될 수도 있기 때문이다.

3.2.3 미들웨어 제품으로부터 성장한 BPM 업체

미들웨어 제품들은 기존에 갖추고 있는 사용자 환경과 비즈니스 프로세스 지원 요소들이 취약하기 때문에 해당 요소들을 전문화된 컴포넌트나 라이브러리를 구입하여 자신들의 미들웨어 제품과 통합하여 제공하는 전략이 바람직하다. 이러한 전략을 통해 상대적으로 빠른 시간에 중요 요소를 갖출 수 있다. 기존에 사용하던 비즈니스 프로세스 정의 양식이나 디자이너 요소들이 없기 때문에 기존 사용자를 위한 유지보수 부담 없이 BPEL이나 BPMN과 같은 표준을 곧바로 도입할 수 있다.

이 유형에서는 기존에 유지하고 있는 미들웨어의 컨테이너 구조와 새롭게 추가될 BPM 요소를 어떻게 조화시킬지가 매우 중요한 요소이다. 이미 미들웨어의 컨테이너는 표준 모델이 있어서 이를 통해 서블릿(Servlet), EJB(Enterprise Java Beans), JSP(Java Servlet Page) 등의 요소를 실행시키고 있는 상황에서 BPM 표준 모델이 컨테이너 운영 및 수명 주기 관리방식과 잘 통합되어야 한다. 비즈니스 프로세스에 대한 표준이 BPEL과 같이 제공되고 있지만 이러한 비즈니스 프로세스의 수명 주기가 어떻게 관리되어야 하는지에 대한 표준은 아직 없다는 점도 고려해야 한다.

3.2.4 특화된 전문 유틸리티로부터 성장한 BPM 업체

이 유형의 BPM 업체에서 가장 먼저 취해야 할 전략은 안정적인 운영 환경의 확보이다. 개별적인 영역에서 특화된 전문도구들을 중심으로 구성된 시스템에서 안정적이고 통합된 비즈니스 프로세스 운용 환경을 갖추는 것이 필요하다.

이미 이들 유형의 제품 속에는 BPEL이나 BPMN과 같은 표준들을 제공하는 도구들이 포함되어 있을 가능성이 높으므로 이를 운용하기 위한 컨테이너나 미들웨어 환경을 표준이나 참조 모델에 근거하여 구축하는 것이 요구된다. 비즈니스 프로세스 관리를 위한 도구로 중앙의 프로세스 저장소, 고성능의 프로세스 실행 엔진 등이 이러한 운용 환경에 구축되어야 하고, 이들 유형의 업체의 최대 강점이라 할 수 있는 다양한 전문 기능을 프로세스 중심적인 측면에서 통합하는 작업이 진행될 필요가 있다.

3.2.5 ERP 및 기업용 애플리케이션으로부터 성장한 BPM 업체

이 유형의 업체는 앞서 살펴보았던 EAI 유형의 BPM 제품군과 유사한 면을 보인다. 이 두 유형의 경우 해당 유형의 약점을 극복하기 위해 최근까지의 기술적 발전과 표준 요소들을 적극적으로 수용하기 위한 노력이 필수적이다. ERP 및 기업용 애플리케이션으로부터 성장한 BPM 업체의 경우 기업의 비즈니스 전반에 걸쳐 거의 모든 프로세스를 실행하고 관리한다는 점에서 상당히 유리한 위치에 있다. 문제는 이러한 프로세스 지원 요소들이 모두 고정된 비즈니스 프로세스 정의를 가지고, 실행 환경만을 집중적으로 관리하고 모니터링 한다는 점이다.

따라서 응용프로그램을 통해 고정된 형태로 제공되고 있는 기존의 프로세스 정의와 관리의 모든 측면이 동적인 환경을 지원하도록 변경되어야 한다. 우선 비즈니스 프로세스를 새롭게 정의할 수 있는 BPMN 디자이너 환경과 BPEL 실행 환경이 추가되어야 한다. 이미 프로그램의 화면 형태로 하드 코딩되어 있는 프로세스 정의는 갖추고 있으므로 이를 BPMN과 BPEL 표준을 이용하여 기술하고 실행하는 것이 많은 자원을 소비하지는 않을 것으로 예상된다.

이 유형의 제품에서도 BPQL을 지원할 필요가

있다. ERP와 같은 패키지의 경우 기반 데이터에 대한 접근성이 매우 낮기 때문에 사용자의 시스템 접근을 원활하게 하기 위한 방안으로 BPQL의 질의 기능이 제공될 필요가 있다. BPQL이 기존의 SQL과 같은 정도의 기능성을 갖도록 정의되어 ERP나 기업 응용프로그램이 내부적으로 관리하고 있는 데이터에 대한 접근에 활용될 수 있도록 하는 것이 우선 요구된다.

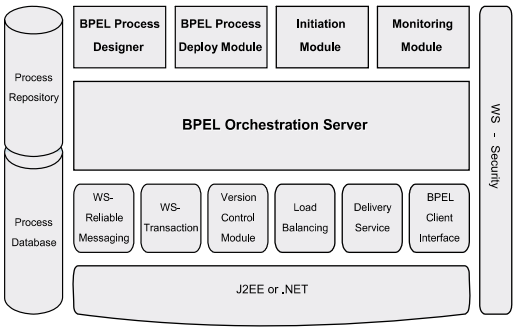
3.2.6 웹 서비스, XML 및 B2B로부터 성장한 BPM 업체

이미 BPMN, BPEL 등의 표준이 이 유형의 제품 내에 포함되어 있을 가능성이 다른 유형에 비해 가장 크다. 그 정도로 BPM 표준에 대한 지원은 매우 우수하지만 상대적으로 낮은 인지도와 레퍼런스로 인해서 시스템의 성능과 안정성 등에 대해서는 시장에서 검증되지 않았다는 인식이 많다.

이 유형에서는 전체 표준 스펙의 완전 구현과 같은 전략을 고려해 볼 만하다. 일반적으로 표준의 구현에서는 선택적인 요소들이 존재하는데 대부분의 제품 유형에서는 필수 요소들을 위주로 구현하고, 선택적인 요소들에 대해서는 많은 투자를 하지는 않는 경우가 많은 실정이다. 이러한 점에 착안하여 전체 표준의 완전 구현과 보다 향상된 표준의 활용 방안을 제시하는 접근법이 이 유형의 기업에서는 효과적이다. 이를 위해서는 BPMN으로 기술된 비즈니스 프로세스를 BPEL을 통해 실현시키기 위한 추가적인 기술과 요소들의 개발을 고려해 볼 수 있다.

4. BPEL 구현 참조 아키텍처

본 장에서는 BPM 표준 가운데 가장 중요한 표준인 BPEL 표준을 구현하기 위한 참조 아키텍처를 제시하였다. 제안한 BPEL 구현 참조 아키텍처는 [그림 6]과 같다. 제안된 참조 아키텍처는 BPEL 표준을 지원하는 각종 솔루션들[5, 15]을 분석하여 BPEL 구현에 반드시 필요한 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 작성하였다.



[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 : 웹 서비스의 안전성을 위한 보안 기능을 담당

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. 앞서 열거된 여러 요소들 가운데 가장 핵심적인 컴포넌트는 프로세스 디자이너, BPEL 엔진, 프로세스 모니터링 모듈, 프로세스 공개 및 시작 모듈이라 할 수 있고, 다음 장에서는 특히 프로세스 디자이너와 BPEL 엔진을 중심으로 핵심 컴포넌트들이 갖추어야 할 기능 요구사항 및 구현 지침을 자세히 설명하였다.

5. BPEL 핵심 컴포넌트 구현 지침

본 장에서는 BPEL을 지원하는 BPM 시스템을 개발하는데 참조할 수 있도록 핵심 컴포넌트 구현 지침을 제시하였다. 프로세스 정의, 서버로의 공개, 프로세스 실행, 모니터링으로 구성되는 일련의 BPEL 표준 활용 시나리오에 따라 핵심 컴포넌트 구현 지침을 기술하였다. 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.

5.1 BPEL 프로세스 디자이너

BPEL 프로세스 디자이너는 BPEL 스펙에서 정의한 BPEL 프로세스 모델을 설계하는 기능을 제공하는 프로그램으로 BPEL 준수 프로세스 모델 정의 기능을 제공해야 한다. 여기서는 BPEL 표준 프로세스 모델을 구성하는 주요한 요소를 중심으로 BPEL 프로세스 디자이너를 개발할 때 고려해야 할 점을 설명하였다. 관련된 웹 서비스 구조에 대한 이해를 바탕으로 프로세스를 설계하는 과정에 따라 프로세스 디자이너의 주요 기능을 설명하였다.

BPEL 프로세스를 설계하기 위한 일반적인 과정은 다음과 같다.

- 먼저 관련된 웹 서비스 구조 이해
- BPEL 프로세스를 위한 WSDL 정의
- 파트너 링크 타입 정의
- BPEL 프로세스 정의(파트너 링크를 정의하고, 변수를 선언하고, 프로세스 로직을 정의함)

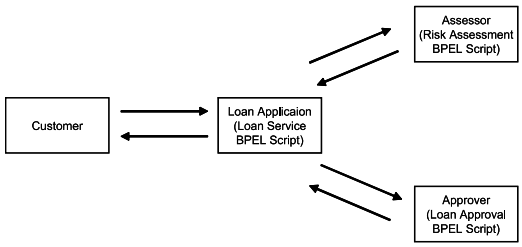
프로세스 디자이너 프로그램은 기본적으로 비즈니스 프로세스를 생성하기 위한 전 과정을 지원해야 한다. 그리고 프로세스 디자이너는 사용자 편의성을 제고하기 위해 GUI 형태로 프로세스 정의 기능을 제공하는 것이 바람직하다.

5.1.1 웹 서비스 구조에 따른 WSDL의 정의

BPEL 프로세스 정의 작업을 시작하기 전에 구현할 비즈니스 프로세스에서 호출하는 웹 서비스-파트너 웹 서비스(Partner Web Service)-에 대해 먼저 이해할 필요가 있다. 클라이언트가 자신을 호출할 수 있도록 WSDL 포트 타입을 정의해야 하고, 자신이 호출하는 파트너 웹 서비스의 포트 타입도 정의해야 한다. 따라서 BPEL 프로세스 디자이너는 정의하고자 하는 프로세스를 위한 WSDL 정의 기능을 제공해야 한다.

여기서는 [그림 7]의 대출 승인(Loan Approval) 서비스 예제를 가지고 WSDL 정의 기능을 설명하고자 한다. 본 예제는 BPEL4WS 1.1 표준에 포함된

예제 프로세스로 고객이 대출 신청을 할 수 있는 포트를 제공하는 간단한 대출 승인 웹 서비스이다.



[그림 7] 대출 승인 BPEL 프로세스

해당 서비스를 이용하는 고객은 자신의 개인정보와 대출 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. 보통 대출 전문가의 직접적인 관여가 필요한 '대출 승인(Loan Approval)' 서비스는 보다 자세한 신용도 평가를 하기 위해 사용된다.

5.1.2 파트너 링크 타입의 정의

파트너 링크 타입은 BPEL 프로세스와 관련 개체들 간의 상호작용을 정의하는데 사용된다. 여기에는 BPEL 프로세스가 호출하는 웹 서비스와 BPEL 프로세스를 호출하는 클라이언트에 대한 정의가 포함된다. BPEL 프로세스 디자이너는 BPEL 프로세스와 관련 웹 서비스와의 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 웹 서비스를 호출하는 BPEL 프로세스 역시 비동기식으로 구현하는 것이 일반적이다.

5.1.3 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 네임스페이스를 정의해야 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 링크 모델링 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.

BPEL 프로세스는 파트너 간 메시지 교환을 위해 변수를 사용한다. 변수는 비즈니스 프로세스의 상태를 구성하는 메시지들을 보관하는 수단이 된

다. BPEL 프로세스 디자이너는 변수 정의 기능을 제공해야 한다. 각 변수는 스코프(scope, 범위)를 가지며 전체 프로세스 범위에서 선언된 변수는 전역변수(global variable)라 하고 그렇지 않은 변수를 지역변수(local variable)라고 한다. 각 변수마다 타입을 지정해 주어야 하는데, 변수의 타입에는 WSDL 메시지 타입, XML 스키마 심플타입(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 2>와 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>, <switch>와 <while>을 사용하여 정의할 수 있다. 액티비티들 사이의 동시성과 동기성은 <flow>에 의하여 정의된다. 외부 이벤트에 기초한 비결정론적인(non-deterministic) 선택은 <pick>을 사용하여 정의한다.

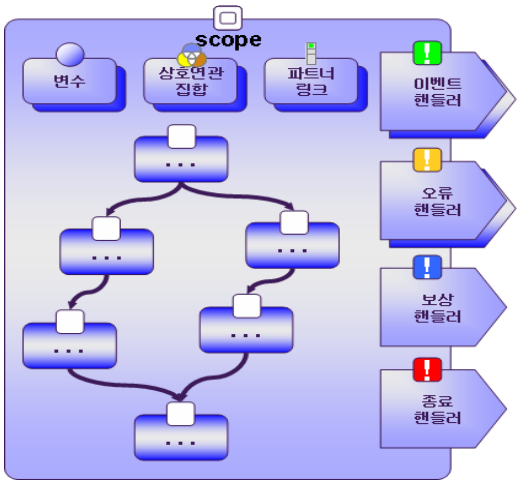
〈표 2〉 BPEL4WS의 액티비티 분류

액티비티 분류	종류
기본 액티비티	<receive>, <reply>, <invoke>, <assign>, <throw>, <wait>, <empty>
구조화 액티비티	<sequence>, <switch>, <while>, <pick>, <flow>
기타 특수 액티비티	<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 스코프에 의해 제공된다. BPEL 프로세스 디자이너 프로그램은 스코프 정의 기능과 이벤트 처리 기능을 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. 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. 종료 핸들러(termination handler)는 강제 스코프 종료를 처리하는 핸들러이다. [그림 8]에 스코프와 핸들러들의 관계를 나타내었다.



[그림 8] 스코프와 핸들러

5.2 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 정의된 프로세스를 서버에 공개하는 과정으로 프로세스의 실행을 관장하는 엔진이 프로세스를 제대로 동작시키기 위해 필요한 과정이다. 이를 위해 프로세스를 서버에 공개하는 기능을 제공하는 모듈이 필요하다. 많은 경우 프로세스 공개 기능은 프로세스 디자이너에 포함되어 제공되고 있다.

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. 보통 프로세스 서술자는 XML 파일로 작성되며, BPEL 프로세스에 대한 자세한 정보인 BPEL 소스 파일명, BPEL 프로세스 명, 파트너 링크 웹 서비스의 WSDL 위치정보, 기타 구성 정보 등을 포함한다.

프로세스 서술자가 준비되면 BPEL 엔진을 사용하여 프로세스를 시작할 준비가 완료된 상태가 된다. 서버에 프로세스를 공개하기 위해서는 복잡한 컴파일 작업과 공개 작업을 위한 환경설정이 필요한데, Ant 유틸리티와 같은 빌드(build) 유틸리티를 사용하면 이러한 작업이 편리하다. 빌드 유틸리티를 이용해서 성공적으로 BPEL 프로세스를 서버에 공개하고 나면 BPEL 프로세스를 실행할 준비가 완료된다.

프로세스 시작 모듈(Initiator)은 BPEL 서버에 공개된 프로세스 실행을 시작시키기 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 사용자는 필요한 데이터를 입력하여 프로세스를 시작할 수 있다.

5.3 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.

5.3.1 BPEL 프로세스 해석 및 흐름 제어

BPEL 엔진은 프로세스 모델을 정확하게 해석하고 흐름을 제어할 수 있어야 한다. 따라서 BPEL 엔진을 구현하기 위해서는 5.1절에서 설명한 BPEL 프로세스 디자이너 부분에서 설명한 프로세스 정의와 관련한 내용의 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 프로세스를 컴파일하면 자바 코드를 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(Enterprise Java Beans) 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.

5.3.2 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.

5.3.3 프로세스 실행 상태 및 메시지 영속성 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 식별자에 의해 식별되는 것이 아니라 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.

5.3.4 메시지 라우팅 및 서비스 호출 지원

BPEL 엔진은 수신 메시지 라우팅 기능과 파트너 웹 서비스 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 웹 서비스의 호출은 IIOP(Internet Inter-ORB Protocol), JMS(Java Message Service) [11], 고유 자바 호출(native Java calls) 등의 J2EE 프로토콜을 사용할 수 있다. 외부 서비스의 호출을 지원하기 위해 웹 서비스 호출 프레임워크(Web Services Invocation Framework)를 사용할 수 있다[16].

5.4 프로세스 모니터링 및 분석 프로그램

BPEL 엔진을 통해 실행하고 있는 프로세스의 실행 상태를 브라우징하는 프로그램을 프로세스 모니터링 프로그램이라고 한다. 프로세스 모니터링 프로그램은 사용자 편의성을 높이기 위해 프로세스 인스턴스 실행 상태의 비주얼한 흐름을 GUI (Graphical User Interface) 형태로 제공하는 것이 바람직하다. 모니터링 프로그램은 프로세스 인스턴스의 실행 상태(진행중, 완료됨, 취소됨, 중지됨 등)와 프로세스의 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 제공함으로써 효과적인 의사결정을 수행할 수 있도록 해 준다. 시뮬레이션 기능은 실제로 프로세스를 실행하기 이전에 모의시행 해 볼 수 있는 기능으로 이러한 프로세스 시뮬레이션 기능을 통해 프로세스의 로직 상 문제점을 사전에 분석할 수 있고, 여

러 프로세스 대안들을 비교 평가할 수 있다.

6. 결 론

BPM은 오늘날 실시간기업(RTE : Real Time Enterprise)의 실현 도구, 인터넷을 통한 기업 간 협업 프로세스의 유기적 관리 도구로써 각광받고 있다. 국내외에서 여러 BPM 솔루션 기업들이 다양한 제품을 출시하고 있으며, 제품 간 상호운용성을 목적으로 여러 BPM 표준이 등장하고 있다. BPM 관련 국제 표준을 주로 수용하고 있는 국내 BPM 업체들이 BPM 표준화 동향을 정확히 이해하고 자사 실정에 맞는 표준 도입 전략을 수립하는 것이 매우 중요하다.

본 연구에서는 BPEL, BPMN, BPQL 등 BPM 표준화 현황 및 주요 내용을 소개하고, BPM 솔루션 개발 업체 유형별 표준 도입 전략을 제시하였다. 특히, BPM 표준 가운데 시장에서의 영향력과 중요성이 가장 큰 BPEL 표준의 주요 내용을 설명하고, BPM 솔루션 개발 업체가 참조할 수 있는 BPEL 구현 지침을 개발하였다.

BPM 표준 도입 전략을 제시하기 위하여 BPM 솔루션 업체 유형을 제품 특성과 기술적 배경을 기준으로 6가지 분류하였다. 각각의 유형에 대해서 강점과 약점을 살펴보고, 이를 기준으로 하여 업체 유형별로 표준을 구현하기 위해 집중해야 할 요소들을 제시하였다. 본 논문에서 제시된 전략은 모든 업체들이 반드시 따라야 할 절대적인 성질의 것이라기보다는 자사의 표준 도입 전략을 수립할 때 참고할 수 있는 유용한 지침이라고 볼 수 있다

BPEL 구현 가이드라인에서는 BPEL 표준을 지원하는 BPM 솔루션이 갖추어야 하는 참조 아키텍처를 제시하고, 참조 아키텍처를 구성하는 핵심 컴포넌트의 구현 지침을 기술하였다. BPEL을 지원하는 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 프로세스의 서버로의 공개, 프로세스 실행, 모니터링으로 구성되는 일련의 시나리오에 따라 핵심 컴포넌트 구현 지침을 제시하였다.

본 연구에서 제시한 BPM 표준 도입 전략을 바탕으로 BPM 솔루션 기업들은 자사 솔루션의 특징에 따른 표준 적용 전략을 수립하고, BPM 표준 적용 및 개발 시의 시행착오를 줄임으로써 보다 효율적으로 시스템을 개발할 수 있을 것으로 기대된다. 또한, BPM 솔루션 개발 업체는 BPEL 표준 적용 및 개발 시의 본 연구에서 제시한 참조 아키텍처와 상세 구현 지침을 참조할 수 있다.

참 고 문 헌

[1] 임홍순, “BPM 기술 현황 및 전망”, 한국전자거래학회 월례세미나, 2004, http://iems.net/vod/calsec/monthly_seminar/2004/october/index.html.

[2] 한국전산원, 「비즈니스 프로세스 관리 기술 표준의 BPMS 적용에 관한 지침 연구」, 2005.

[3] 한국전산원, 「비즈니스 프로세스 관리기술 표준적용을 위한 지침 연구」, 2004.

[4] 한국전산원, 「XML 기반의 기업간 워크플로우 통합 및 상호연동성에 대한 연구」, 2003.

[5] BPEL Resource Guide, http://bpelsource.com.

[6] BPM.com, Standards and BPM, http://www.bpm.com/FeaturePrint.asp?FeatureId=145.

[7] BPMI.org, http://www.bpmi.org.

[8] BPMN 1.0 specifications, 2004.

[9] BPMN homepage, http://www.bpmn.org.

[10] Dieter König, “Web Services Business Process Execution Language (WS-BPEL)”, *OASIS Open Standards Day XTech 2005 Conference Amsterdam*, 2005.

[11] Java Message Service, http://java.sun.com/products/jms/.

[12] Kimberly Hill, "Enterprises Face BPM Standards Maze", 2004 http://www.cio-today.com/story.xhtml?story_id=27761.
[13] Martin Owen and Jog Raj, *BPMN and Business Process Management*, 2003.
[14] 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.
[15] Wikipedia, http://en.wikipedia.org/wiki/BPEL.
[16] Welcome to WSIF : Web Services Invocation Framework, http://ws.apache.org/wsif/.
[17] http://www.tta.or.kr/Home2003/committee/CommitToR.jsp?commit_code=PG403.

◈ 저 자 소 개 ◈

김 동 수 (dskim@ssu.ac.kr)

현재 숭실대학교 산업·정보시스템공학과 조교수로 재직 중이며, 서울대학교 산업공학과에서 학사, 석사 및 박사를 취득하였다. IEEE Internet Computing, Lecture Notes in Computer Science 등의 국제학술지 및 전자거래학회지, 한국SI학회지, 정보화정책, 산업공학, 경영과학, 대한의료정보학회지 등의 국내학술지에 논문을 게재한 바 있다. 주요 관심분야는 BPM(Business Process Management), e-business 기술 및 표준, SOA(Service Oriented Architecture) 구현, u-비즈니스 모델 및 시스템, u-Health 등이다.