

# 관점지향 프로그래밍 기법을 이용한 BPEL 기능 추가를 위한 XAS4B 엔진 설계

## A Design of XAS4B for Adding Function in BPEL using Aspect-Oriented Programming

곽 동 규\*, 최 재 영\*\*

Donggyu Kwak\*, Jaeyoung Choi\*\*

### Abstract

The BPEL is standard of workflow. A domain experts not familiar with programming languages and he can write workflow. But he need new function without BPEL. We must develop or modify a BPEL engine for new functions. This method is expensive. In this paper, We propose the XAS4B documents that extend the XML schema. And We describe method that can process new function in BPEL using Aspect-Oriented Programming. AOP can cross-cut concern adding to core concern using weaving without modification. We use an existing B2J (BPEL to JAVA) engine that is transfer a BPEL's document to a JAVA program and B2J execute a JAVA program. This system adding new functions to BPEL's flow using AOP. In this system, A JAVA program is core concern and a program of new function is cross-cut concern. And this system weave a JAVA program made in B2J and a program of new function. This method provide abstract grammar for new functions. And domain experts can easily write new function is a BPEL Document and reuse new function using abstract grammar.

### 요 약

워크플로우의 표준인 BPEL은 프로그래밍 언어에 대한 이해도가 낮은 도메인 전문가도 손쉽게 비즈니스 흐름을 작성할 수 있다. 하지만 특정 도메인에서는 BPEL 문법에 없는 새로운 기능이 요구된다. BPEL 엔진이 새로운 기능을 처리하기 위해서는 새로운 BPEL 엔진을 개발하거나 기존의 BPEL 엔진에 기능을 추가해야 하는데 이는 비용이 많이 필요하다. 본 논문에서는 XML 스키마를 확장하여 XAS4B 문서를 제안하고 관점지향 프로그래밍 기법을 이용하여 BPEL 엔진에 새로운 기능을 추가하는 방법을 보인다. 관점지향 프로그래밍 기법은 직조를 이용하여 핵심관심사의 수정없이 횡단관심사를 추가할 수 있다. 본 연구에서는 기존의 BPEL 엔진으로 B2J (BPEL to JAVA)를 사용하는데 이 엔진은 BPEL 문서를 JAVA 프로그램으로 변환하고 이를 실행한다. 본 시스템은 B2J 엔진이 생성한 JAVA 프로그램을 핵심관심사로 새로운 기능을 처리하는 프로그램을 횡단관심사로 두 프로그램을 직조하여 BPEL 흐름에 새로운 기능의 흐름을 추가한다. 제안하는 방법은 BPEL의 새로운 기능을 추상적으로 제공하여 한 번 개발된 새로운 기능을 도메인 전문가가 재사용하기 손쉽다.

*Key words* : BPEL, Workflow, adding new function, AOP, compiler

\* School of Computer Science and Engineering, Soongsil University

★ Corresponding author  
choi@ssu.ac.kr, +82-2-820-0684

※ This research was partially supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology (2013R1A1A2012118), and partially supported by the MSIP (Ministry of Science, ICT & Future Planning), Korea, under the ITRC (Information Technology Research Center) support program (NIPA-2013-H0301-13-2006) supervised by the NIPA (National IT Industry Promotion Agency)  
Manuscript received Nov. 19. 2013; revised Dec. 09. 2013; accepted Dec. 10. 2013

## 1. 서론

워크플로우는 작업 절차를 통한 정보 또는 업무의 흐름을 의미하며, 워크플로우 시스템은 이를 적용한 시스템이다[1]. 워크플로우 시스템에서는 워크플로우 언어를 통해 작업의 흐름을 기술하는데 워크플로우 언어 중 BPEL (Business Process Execution Language)[1]은 OASIS (Organization for the Advancement of Structured Information Standards)[2]의 표준으로서 웹 기반의 비즈니스 모델

과 공장 자동화 시스템에 도입되어 사용되고 있다. BPEL은 XML을 기반으로 하고 있어 학습이 용이하고 다양한 그래픽 편집기[3]가 있어 컴퓨터 언어에 대한 이해도가 낮은 도메인 전문가도 쉽게 워크플로우를 작성할 수 있다.

BPEL은 작업의 흐름을 기술하기 위해서 웹 서비스를 호출하고 조건에 따른 작업의 흐름을 기술할 수 있는데 특정 도메인에 적용할 경우에 새로운 기능을 요구할 수 있다. 예를 들어 복잡한 조건을 추상화시킨 비즈니스 규칙 엔진 (Business Rule Engine)[4]이 필요한 경우나 엔진이 실행되고 있는 환경에서의 응용 프로그램을 호출하는 경우에는 표준 그대로의 BPEL을 사용하기 어렵다. BPEL에 기능을 추가하기 위한 연구로는 JWX (JAVA Weaving XML) 문서를 이용한 기능 추가 방법이 연구되었다[5][6]. JWX는 BPEL에 추가적으로 요구되는 기능을 JWX 문서에 JAVA 프로그래밍 언어로 기술하도록 제안하였다. JWX 문서는 기능이 추가될 BPEL의 위치를 XPath[7]로 기술하고 그 위치에서 실행될 프로그램을 작성할 수 있는 방법을 제공하고 있다. 그리고 BPEL과 함께 JWX에 기술된 추가적인 프로그램을 함께 실행하기 위해 시스템을 보였다. JWX 시스템은 BPEL 엔진으로 B2J (BPEL to JAVA)[8]를 사용하는데 이 엔진은 BPEL 문서를 JAVA 프로그램으로 변환하고 이 JAVA 프로그램을 실행하는 방법을 사용한다. JWX 시스템은 B2J가 생성한 JAVA 프로그램과 JWX 문서에 작성된 추가된 프로그램을 함께 실행하기 위해 관점 지향 프로그래밍 기법[9]의 구현체인 AspectJ[10]를 이용하여 직조 (Weaving)하여 실행시킨다. JWX는 BPEL이 제공하지 않는 기능을 추가할 수 있는 방법을 제공하고 있으나, BPEL과 함께 작성해야 하는 JWX 문서에 새로운 기능을 JAVA 프로그램으로 기술해야 하는 단점을 가지고 있다. 이는 컴퓨터 프로그래밍에 대한 이해도가 낮은 도메인 전문가가 사용하기는 어렵게 한다.

본 논문은 BPEL에 새로운 기능을 추가하기 위해서 XML 스키마[11]를 확장하여 BPEL에서 요구되는 새로운 기능을 JAVA 프로그램으로 작성할 수 있는 XAS4B (XML Attribute Schema for BPEL)를 제안한다. XAS4B는 두 부분으로 구성되어 있다. 한 부분은 추가 기능에 필요한 클래스를 импорт하는 부분이고, 다른 한 부분은 의미 정보를 기술하는 부분이다. 의미 정보를 기술하는 부분은 새로운 기능을 기술하는 XML의 변수를 정의하고 JAVA 프로그램을 기술한다. XAS4B 문서는 프로그래밍 언어에 대한 이해도가 높은 프로그래밍 개발자가 작성하면 BPEL에 사용할 수 있는 새로운 기능에 해당하는 엘리먼트를 생성

할 수 있다. 이와 같은 문서 구조는 프로그래밍 언어에 대한 이해도가 낮은 도메인 전문가가 새로운 기능을 사용할 수 있도록 한다. 표 1은 제안하는 시스템의 사용자에게 따른 요구사항과 개발범위 그리고 사용 도구를 보여준다.

Table 1. Requirement, Development range and Tools in accordance with user

표 1. 사용자에게 따른 요구사항과 개발범위, 사용도구

	요구사항	개발범위	사용도구
프로그래머	추가 기능	프로그램 모듈	XAS4B, JAVA
도메인 전문가	비즈니스 흐름	워크플로우	BPEL 언어 (추가 문법)
최종 사용자	사용자 요구사항	-	BPEL 엔진

표 1과 같이 본 시스템의 사용자는 세 계층으로 구분된다. 프로그래머는 새롭게 추가될 기능을 개발하는 계층으로 요구되는 기능을 기술하기 위한 엘리먼트를 XAS4B 문서로 정의하고 이를 처리하는 JAVA 프로그램을 개발한다. 그리고 도메인 전문가는 해당 도메인에서 요구되는 비즈니스 흐름을 BPEL 문서로 기술한다. 이 때, 프로그래머가 개발한 새로운 기능을 추가된 엘리먼트를 이용하여 기술한다. 최종 사용자는 도메인 전문가가 기술한 비즈니스 흐름에 따라 서비스를 제공받는다. 제안하는 방법은 워크플로우 시스템을 새로운 시스템에 능동적으로 대처하여 사용할 수 있다.

## II. 관련연구

본 논문은 워크플로우의 한 표준인 BPEL에 새로운 기능을 추가하기 위해 XML 스키마를 확장하여 XML의 의미 정보를 한정적으로 표현하는 방법을 제안하고 이를 적용하기 위한 시스템을 제안한다. 본 장에서는 BPEL에 기능을 추가하는 시스템과 XML의 의미 정보를 기술하기 위해 제안된 XML 속성 문법을 소개한다.

### 1. JWX4BPEL (JAVA Weaving XML for BPEL)

BPEL은 웹 서비스와 상호작용을 하는 워크플로우 언어의 표준으로서 다양한 응용에서 사용되고 있다. 하지만 특정 응용에서는 BPEL에 없는 추가적인 기능을 요구하므로 BPEL을 바로 적용하기 어렵다. JWX4BPEL (JAVA Weaving XML for BPEL)은 관점지향 프로그래밍 기법을 이용하여 BPEL 엔진에 새

로운 기능을 추가할 수 있는 시스템이다[5][6]. BPEL에 새로운 기능을 추가하기 위해 새로운 기능을 기술할 수 있는 JWX (JAVA Weaving XML) 문서를 정의하였고, 관점지향 프로그래밍 기법을 이용하여 BPEL에 적용한 시스템을 보였다. JWX 문서는 BPEL 문서에 새로운 기능을 추가하기 위하여 JAVA 프로그램을 기술할 수 있는 XML 기반의 문서이다. 관점지향 프로그래밍 기법은 핵심적인 요구사항과 부가적인 요구사항 사이의 낮은 결합도를 보장하고 있으므로 기존의 프로그램을 수정하지 않고 새로운 기능을 추가할 수 있다. 또한 JWX4BPEL은 B2J라는 BPEL 엔진을 관점지향 프로그래밍 기법을 통해 확장하여, JAVA 프로그램과 JWX 문서의 새로운 기능 프로그램을 직조하고 실행시킨다. 그림 1은 JWX-4BPEL 시스템의 구조를 보인다.

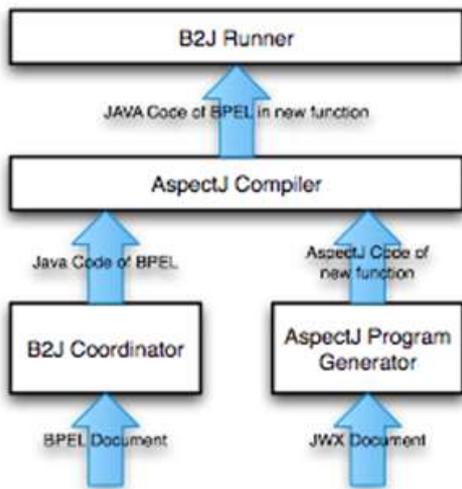


Fig. 1. System Architecture of JWX4BPEL  
그림 1. JWX4BPEL의 시스템 구조

이 방법을 이용하면, 적은 노력과 비용으로 B2J가 제공하는 BPEL의 기능은 그대로 사용하면서 새로운 기능이 추가된 BPEL 문서를 사용할 수 있다. 하지만 JWX 문서를 JAVA로 프로그래밍해야 하므로, JAVA 프로그래밍에 익숙하지 않은 도메인 전문가가 직접 작성하기가 어렵다. 따라서 BPEL 문서를 작성하려면 프로그래밍에 익숙한 개발자의 도움을 받아야 한다.

2. XML 속성 문법

XML 속성 문법은 XML 문서를 위한 컴파일러를 개발하는데 많은 노력이 필요함을 지적하고 XML 컴파일러를 자동으로 생성하기 위해 XML 스키마와 의미 정보를 기술할 수 있는 SML (Semantic Markup Language)를 제안하였다[12]. 그리고 XML 스키마 문

서와 SML 문서를 처리하는 XsCC (XML Schema Compiler-Compiler)를 보였다. 그림 2는 XsCC와 생성된 XML 컴파일러의 구조를 보인다.

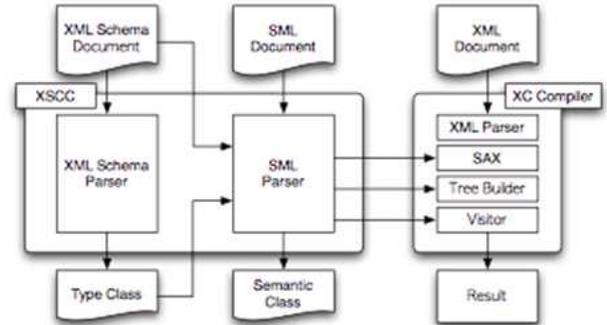


Fig. 2. System Architecture of XsCC and XML Compiler  
그림 2. XsCC와 XML 컴파일러 구조

III. BPEL 문법의 기능 추가

BPEL은 XML을 기반으로 하는 워크플로우 언어로 작업을 흐름을 기술할 수 있어 프로그램을 잘 알지 못하는 도메인 전문가도 사용할 수 있다. BPEL에서 제공하지 않는 추가적인 기능이 필요할 경우에는 새로운 기능을 위한 웹 서비스를 개발하거나 JWX4BPEL 시스템을 이용하여 새로운 기능을 개발해야 한다. 하지만 웹 서비스나 JWX4BPEL을 사용하여 새로운 기능을 개발할 경우 두 가지 문제점을 가진다. 첫 번째는 새로운 기능을 JAVA 프로그램으로 개발해야 하는 것이다. 일반적으로 도메인 전문가는 프로그래밍을 잘 이해하지 못하여 직접 개발하기 어려워 프로그램 개발자에게 도움을 받아야 한다. 두 번째 문제점은 추상적인 문법을 제공하지 않아 재사용하기 어렵다. 웹 서비스나 JWX4BPEL을 사용하여 새로운 기능을 개발할 경우 JAVA 프로그램으로 직접 개발되어 호출되기 때문에 추상적인 문법을 제공하지 않는다. 그러므로 한 번 개발된 기능을 도메인 전문가가 프로그램 개발자의 도움없이 재사용하기 어렵다.

본 연구는 워크플로우의 표준인 BPEL에 새로운 기능을 위한 XAS4B (XML Attribute Schema for BPEL) 문법을 정의하고 이를 적용할 수 있는 방법을 제안한다. XAS4B는 BPEL에 문법을 추가하고 해당 문법의 기능을 JAVA 프로그램으로 작성할 수 있다. 이렇게 개발된 새로운 기능의 문법은 기능을 추상적으로 표현하고 있어 프로그램을 잘 알지 못하는 도메인 전문가가 손쉽게 사용할 수 있다. 또한 한 번 개발된 문법은 필요에 따라 인자를 변경하면서 재사용할 수 있다. 그림 3은 프로그램 개발자와 도메인 전

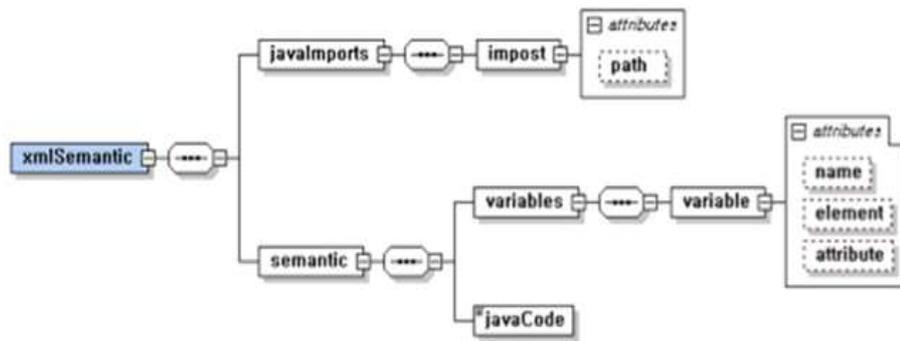


Fig. 5. Schema of XAS4B  
그림 5. XAS4B의 스키마

문가에 따라 사용하는 시스템을 보인다.

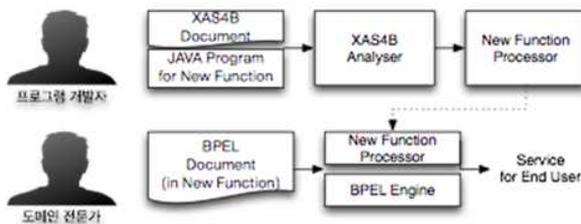


Fig. 3. System using User Level  
그림 3. 사용자별 시스템

그림 3에서 프로그램 개발자는 요구되는 새로운 기능을 분석하여 추가될 문법을 XAS4B 문서로 작성하고 새로운 기능을 실행하는 JAVA 프로그램을 작성한다. 이 문서와 JAVA 프로그램은 XAS4B 분석기를 통해 분석되어 새로운 기능을 처리하는 모듈 (New Function Processor)을 생성한다. 그리고 도메인 전문가는 새로운 기능을 위한 문법을 포함하여 BPEL 문서를 작성하고 이 문서는 새로운 기능을 처리하는 모듈과 BPEL 엔진을 통해 최종 사용자에게 서비스된다.

도메인 전문가가 작성한 서비스의 흐름이 사용자에게 제공되기 위해서는 BPEL 엔진과 새로운 기능을 처리하는 엔진이 하나의 흐름으로 실행되어야 한다. 하지만 기존에 개발된 BPEL 엔진에 새로운 기능을 동적으로 추가하기 어렵다. 본 논문에서는 관점지향 기법을 이용하여 BPEL 문법을 처리하는 모듈을 핵심 관심사로 새로운 기능을 처리하는 부분을 횡단관심사로 개발하여 직조하는 방법을 사용한다. 관점지향 프로그램 기법은 핵심관심사의 수정없이 횡단관심사를 추가할 수 있는 방법을 제공한다. 그림 4는 BPEL의 흐름과 새로운 기능의 흐름간의 관계를 보인다.

본 연구에서는 BPEL 엔진으로 B2J (BPEL to JAVA)를 사용한다. B2J는 BPEL 문서를 JAVA 프

그램을 변환하여 실행한다. 그러므로 B2J 엔진이 생성하는 JAVA 프로그램을 핵심관심사로 새로운 기능 JAVA 프로그램을 횡단관심사로 하여 두 프로그램을 직조한다. 이 방법은 관점지향 프로그래밍 기법을 이용하여 기존 엔진의 BPEL 흐름을 수정하지 않고 새로운 기능을 추가한다.

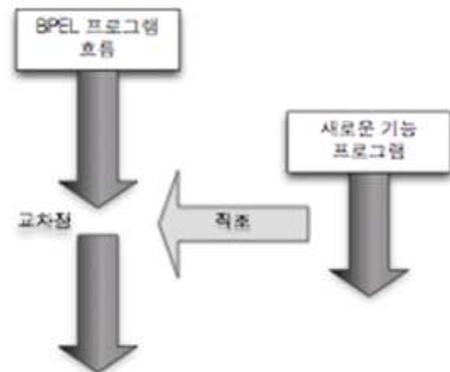


Fig. 4. Relation of BPEL and New Function  
그림 4. BPEL과 새로운 기능간의 관계

#### IV. XAS4B 문서 (XML Attribute Schema for BPTEL)

본 논문은 BPTEL에 추가적인 기능이 요구될 때 추가 기능을 XML 스키마로 정의하여 BPTEL 문법을 확장하고 확장된 문법의 프로그램을 작성하는 문서를 제안하고 이를 위한 처리 엔진을 보인다. 본 논문에서는 추가 기능을 정의하는 방법으로 XML 스키마와 함께 추가 기능의 프로그램을 기술할 수 있는 XAS4B를 제안한다. 그림 5는 XAS4B의 스키마를 보인다. XAS4B는 두 부분으로 구성되어 있다. 한 부분은 “javaImports”로 추가 기능에서 요구되는 JAVA 클래스를 импорт (import)할 수 있도록 제공한다. 그리고 다른 한 부분은 “semantic”으로 변수 기능을 하

```

01 <?xml version="1.0" encoding="UTF-8"?>
02 <xsd:schema
03   targetNamespace="http://www.coolman.org/XMLSchema/BPEL/wait"
04   elementFormDefault="qualified"
05   xmlns="http://www.w3.org/2001/XMLSchema"
06   xmlns:wait="http://www.coolman.org/XMLSchema/BPEL/wait"
07   xmlns:xsd="http://www.w3.org/2001/XMLSchema"
08   xmlns:xas4b="http://www.coolman.org/XMLSchema/xas4b">
09   <xas4b:javaImports>
10     <xas4b:import path="coolman.sync.Wait" />
11   </xas4b:javaImports>
12   <xsd:element name="wait">
13     <xsd:complexType>
14       <xsd:attribute name="time" type="xsd:long" />
15     </xsd:complexType>
16     <xas4b:semantic>
17       <xas4b:variables>
18         <xas4b:variable name="time" element="wait" attribute="time" />
19       </xas4b:variables>
20       <xas4b:javaCode>
21         <![CDATA[
22           Wait.wait($time);
23         ]]>
24       </xas4b:javaCode>
25     </xas4b:semantic>
26   </xsd:element>
27 </xsd:schema>

```

Fig. 6. XAS4B Document in XML Schema

그림 6. XML 스키마 내부에 기술된 XAS4B 문서

```

...
01 <bpel:receive name="receiveInput" partnerLink="client" portType="tns:exam"
02   operation="initiate" variable="input" createInstance="yes"/>
03 <bpel:empty name="FIX_ME-Add_Business_Logic_Here"/></bpel:empty>
04 <wait:wait time="10" />
05 <bpel:invoke name="callbackClient" partnerLink="client" portType="tns:examCallback"
06   operation="onResult" inputVariable="output" />
...

```

Fig. 7. BPEL Document adding Wait's Function

그림 7. 대기 기능이 추가된 BPEL 문서

는 엘리먼트에 관한 정의와 JAVA 프로그램으로 구성되어 있다. 그림 6은 XML 스키마 내부에 작성된 XAS4B 문서의 예를 보여준다. 그림 6은 BPEL에 대기 기능을 추가하기 위한 XAS4B 문서이다. BPEL은 흐름을 잠시 대기 하는 기능이 없다. 하지만 특정 도메인에서는 서비스 호출간 사이에 동기화나 서비스 처리를 위해 잠시 대기하는 루틴이 필요할 수 있다. 그림 4에서 9줄 ~ 11줄에 새로운 기능에 요구되는 JAVA 클래스가 импорт (import)되어 있고 12줄 ~ 15 줄에는 새로운 기능에 사용될 엘리먼트를 정의한다. 그리고 16줄 ~ 25줄에 새로운 기능에 대한 변수와 JAVA 프로그램이 기술되어 있다. 그 중 18줄은 새로운 기능인 대기 기능에서 필요한 시간을 변수로 정의한다. 그리고 20줄 ~ 24줄은 JAVA 프로그램을 기술하는데 18줄에서 기술된 변수를 "\$time"으로 접근한다. 그림 7은 그림 6에서 작성한 XAS4B 문서를 적용한 대기 기능을 포함하고 있는 BPEL 문서이다. 그림 7에서 4줄은 새롭게 추가된 "wait" 엘리먼트를 보이

고 있다. 문서에서 BPEL의 엘리먼트와 새로운 기능의 엘리먼트는 네임스페이스 (namespace)로 구분된다. 기능을 추가하게 되는 XAS4B 문서는 컴퓨터 프로그래밍 언어에 대해 이해도가 높은 개발자가 작성하게 되고 BPEL에 새로운 기능에 해당하는 엘리먼트를 추가할 수 있다. 이 후 프로그래밍 언어에 대한 이해도가 낮은 도메인 전문가는 엘리먼트를 사용함으로써 추가 기능을 사용할 수 있다.

## V. 결론

BPEL은 워크플로우의 표준으로 웹 기반의 비즈니스 모델과 공장 자동화에 도입되어 사용되고 있다. 또한 XML을 기반으로 하고 있어 학습이 용이하고 다양한 편집기가 있어 컴퓨터 언어에 대한 이해도가 낮은 도메인 전문가도 쉽게 워크플로우를 작성할 수 있다. 하지만 BPEL은 제공되고 있는 기능만을 사용할 수 있고 새로운 기능이 요구될 경우 새로운 문법을 설계

해야 하고 이 문법을 처리하기 위해 새로운 BPEL 엔진을 개발하거나 기존의 BPEL 엔진을 수정해야 한다. 그러나 새로운 BPEL 엔진을 개발하거나 기존의 엔진을 수정하기 위해서는 많은 비용이 필요하다.

JWX는 B2J에 새로운 기능을 추가하기 위해서 새로운 기능을 JAVA 프로그램으로 작성하고 관점지향 프로그래밍 기법을 이용하여 B2J 엔진 새로운 기능을 추가한다. 이 방법은 BPEL 엔진을 수정하지 않고 새로운 기능을 추가할 수 있다. 하지만 이 방법은 새로운 기능을 추가할 때 마다 JAVA 프로그램을 작성할 수 있는 프로그래머의 도움이 필요하다. 또한 필요에 따라 개발된 JAVA 프로그램은 추상적인 문법을 제공하지 않아 프로그래머의 도움없이 재사용하기 어렵다.

본 논문에서는 BPEL에 새로운 기능을 추가하는 방법으로 XML 스키마를 확장하여 BPEL에서 요구되는 새로운 기능을 JAVA 프로그램으로 작성할 수 있는 XAS4B를 제안하고 새로운 문법을 처리하는 시스템을 설계하였다. XAS4B 시스템의 사용자는 세 계층으로 구분되는데, 그 세 계층은 프로그래머와 도메인 전문가, 최종 사용자이다. 프로그래머는 새롭게 추가될 기능을 개발하는 계층으로 요구되는 기능을 기술하기 위한 엘리먼트를 XAS4B 문서로 정의하고 이를 처리하는 JAVA 프로그램을 개발한다. 그리고 도메인 전문가는 해당 도메인에서 요구되는 비즈니스 흐름을 BPEL 문서로 기술한다. 마지막으로 최종 사용자는 도메인 전문가가 기술한 비즈니스 흐름에 따라 서비스를 제공받는다. 제안하는 방법은 프로그래머가 개발한 새로운 기능을 추상적인 문법으로 제공하여 프로그램을 잘 이해하지 못하는 도메인 전문가도 손쉽게 기능을 재사용할 수 있다.

## 참고문헌

- [1] BPEL, <http://www-128.ibm.com/developerworks/library/specification/ws-bpel/>.
- [2] OASIS, <http://www.oasis-open.org>.
- [3] BPEL Designer, <http://www.eclipse.org/bpel/>.
- [4] Drools, <http://www.jboss.org/drools/>.
- [5] Donggyu Kwak, Jaeyoung Choi, "Design and Implementation of a BPEL Engine for Dynamic Function using Aspect-Oriented Programming" *Journal of Korean Institute of Information Scientists and Engineers Vol.37 No.4*, pp.205-214, Aug. 2010.
- [6] Donggyu Kwak, Jaeyoung Choi, Chae-woo Yoo, "Rule based BPEL System using Aspect-Oriented Programming", *Journal of Korean Institute of Information Scientists and Engineers Vol.39 No.2*, pp.153-161, Feb. 2012.
- [7] XML Path Language, "<http://www.w3.org/TR/xpath/>".
- [8] B2J, <http://www.eclipse.org/stp/b2j/>
- [9] Gregor Kiczales, John Lamping, Anurag Mendhekar, Chris Maeda, Cristina Videira Lopes, Jean-Marc Loingtier and John Irwin, "Aspect-Oriented Programming", *ECOOP*, pp. 220 ~ 242, 1997.
- [10] AspectJ, <http://www.eclipse.org/aspectj/>
- [11] XML Schema, <http://www.w3.org/TR/xmlschema-1>.
- [12] Jong-Myung Choi, Ho-Byung Park, "An SML Compiler Generator Using Attribute Grammar and XMLSchema", *Journal of Korean Institute of Information Scientists and Engineers Vol.33 No.9*, pp.810-821, Jun. 2006.

## BIOGRAPHY

### Donggyu Kwak (Member)



He received the B.S. degree in Department of applied mathematics from Seokyeong Univesity, Seoul, Korea, in 2002, the M.S. and Ph.D. degrees in School of Computer Science and Engineering from Soongsil University, Seoul Korea, in 2004 and 2012, respectively.

### Jaeyoung Choi (Member)



He received the B.S. degree in Department of Control and Instrumentational Engineering, from Seoul National University, Seoul, Korea, in 1984, the M.S. degree in Dept. of Electrical Engineering, Univ. of Southern

California in 1986, and the Ph.D. degree in School of Electrical Engineering from Cornell University in 1991. He has previously worked at Oak Ridge National Laboratory (1992-1994) and University of Tennessee, Knoxville (1994-1995) as a postdoctoral research associate and a research assistant professor, respectively, where he had been involved with the ScaLAPACK project.

He is currently a professor of School of Computer Science and Engineering at Soongsil University, Seoul, Korea.