

# AOP 기법을 적용한 BPEL 엔진 시스템

손인선, 손은미, 곽동규, 최재영  
숭실대학교 컴퓨터학부

e-mail : { [isson](mailto:isson@ssu.ac.kr), [emson](mailto:emson@ssu.ac.kr), [coolman](mailto:coolman@ssu.ac.kr) }@ss.ssu.ac.kr, [choi@ssu.ac.kr](mailto:choi@ssu.ac.kr)

## A BPEL Engine System using Aspect-Oriented Programming(AOP) Approach

In-Seon Son, Eun-mi Son, Dong-Gyu Kwak, Jae-Young Choi  
School of Computing, Soong-Sil University

### 요 약

관점 지향 프로그래밍은 구조적 프로그래밍, 객체 지향 프로그래밍 등 기존 프로그래밍 방법론에서 모듈화하지 못한 여러 요구사항에 걸쳐 있는 부가적인 요구사항을 모듈화 할 수 있는 방법을 제공한다. 모듈화 하기 어려운 요구 사항을 횡단 관심으로 분류하고, 이를 애스펙트 단위로 모듈화하여 재사용성을 극대화시키는 관점 지향 프로그래밍은 새로운 프로그래밍 패러다임으로 활발하게 연구가 진행 중이다.

본 논문에서는 현재 비즈니스 및 분산 컴퓨팅 환경에서 많이 이용되는 워크플로우 표준 언어인 BPEL 에 워크플로우 서비스의 전이조건의 재사용성을 극대화하기 위해 관점 지향 프로그래밍 기술을 적용한 워크플로우 서비스 시스템을 제안한다. 본 시스템은 기존의 BPEL 엔진을 거의 수정하지 않고 AOP 기법의 적용이 가능하다. 또한, 웹 서비스 분기 및 선택을 위한 서비스 전이 조건으로 요구되는 정보를 실행 도중에 변경하여 웹 서비스 흐름을 재구성할 수 있고, AOP 기법을 이용하여 서비스 전이 조건을 변경하는 모듈을 재사용할 수 있다.

## 1. 서 론

관점 지향 프로그래밍(Aspect-Oriented Programming, AOP)[1] 기법은 1997년 Gregor Kiczales 등이 참여하여 제안한 프로그래밍 개발 방법론이다. 어떠한 시스템에서 다루고자 하는 도메인들은 주요 핵심 관심(core concern)으로 분류될 수 있는데, 이러한 핵심관심을 다루는데 가장 많이 사용되는 방법론이 객체 지향 프로그래밍 방법론(OOP)이었다. 그러나 AOP에서는 구조적 프로그래밍, OOP 와 같은 기존의 프로그래밍 방법론으로 모듈화하기 어려운 요구 사항을 횡단 관심(cross-cutting concern)으로 분류하고 이를 애스펙트(Aspect) 단위로 모듈화하여 모듈의 변경이나 재사용성을 높일 수 있으며 [2], 현재 AOP 방법론은 유럽 등지에서 새로운 프로그래밍 패러다임으로 활발하게 연구가 진행 중이다 [3, 4, 5].

본 논문에서는 현재 비즈니스 및 분산 컴퓨팅 환경에서 많이 이용되는 워크플로우 표준 언어인 BPEL[6]에 워크플로우 서비스의 전이조건의 재사용성을 극대화하기 위한 관점 지향 프로그래밍 기술을 적용한 워크플로우 서비스 시스템을 제안한다. 본 시스템은 기존의 BPEL 엔진을 거의 수정하지 않고 AOP 기법의 적용이 가능하다. 또한, 웹 서비스 분기 및 선택을 위한 서비스 전이 조건으로 요구되는 정보를 실행 도중에 변경하여 웹 서비스 흐름을 재구

성할 수 있고, AOP 기법을 이용하여 서비스 전이 조건을 변경하는 모듈을 재사용할 수 있다.

BPEL 을 처리하기 위한 다양한 엔진 중에서 BPEL 을 자바 코드로 변환하여 컴파일하고 실행하는 B2J(BPEL to Java)[7] 엔진과 객체 지향 언어인 자바(Java)에 AOP 개념을 도입한 최초의 AOP 적용 언어인 AspectJ[8]를 이용해 제안한 시스템을 구현했다.

논문의 구성은 다음과 같다. 2 장에서는 AOP 기법을 적용한 다른 연구와 비교한다. 3 장에서는 본 논문에서 BPEL 의 실행을 위한 B2J 엔진, AOP 기법을 적용한 BPEL 엔진 시스템, 그리고 시나리오 기반의 워크플로우 서비스를 실행한 시스템 과정에 대해서 설명한다. 마지막으로 결론으로 본 논문을 끝맺음 한다.

## 2. 관련 연구

BPEL 에 AOP 를 적용한 AO4BPEL[9, 10]은 BPEL 에 있는 액티비티를 모두 수용하고, AOP 기법을 위한 태그들을 BPEL 에 추가하였다. BPEL 문서 내에 AspectJ 와 유사한 코드를 작성하는데, 애스펙트, 교차점, 충고 등을 일컫는 태그들이 포함된다. 또한 이를 처리하기 위해서 엔진을 설계하고 구현하

였다.

그러나 BPEL 문서가 아닌 AOP 가 적용된 AO4BPEL 문서를 입력으로 처리하기 때문에 워크플로우 표준 언어 본래의 의미가 손실된다.

본 논문에서 제안한 AOP 기법을 적용한 BPEL 엔진 시스템은 BPEL 은 수정하지 않고 자바로 설계된 엔진에서 AspectJ 코드가 처리될 수 있다. 이는 횡단 관심사 모델링을 통해 직조된 BPEL 워크플로우는 기존의 워크플로우와 완전히 호환되므로 어떤 BPEL 엔진에서도 처리될 수 있다. 또한 횡단 관심사 모델링을 통해 재사용성을 극대화하고 차후에 애스펙트로 컨텍스트 처리기를 추가하여 유비쿼터스 컴퓨팅 환경에서 발생하는 컨텍스트 정보를 효율적으로 처리할 수 있다.

### 3. AOP 를 적용한 워크플로우 서비스 시스템

현재 비즈니스 및 분산 컴퓨팅 환경에서 많이 이용되는 워크플로우 표준 언어인 BPEL 에 워크플로우 서비스 전이조건인 재사용성을 극대화하기 위해 관점 지향 프로그래밍 기법을 적용한 BPEL 엔진 시스템을 제안한다.

본 시스템은 기존의 BPEL 엔진을 거의 수정하지 않고 AOP 기법의 적용이 가능하다. 또한, 웹 서비스 분기 및 선택을 위한 서비스 전이 조건으로 요구되는 정보를 실행 도중에 변경하여 웹 서비스 흐름을 재구성할 수 있고, AOP 기법을 이용하여 서비스 전이 조건을 변경하는 모듈을 재사용할 수 있다.

본 시스템을 설명하기에 앞서 BPEL 을 처리하기 위한 다양한 엔진 중에서 BPEL 을 자바 코드로 변환하여 컴파일하고 실행하는 B2J(BPEL to Java) 엔진에 대해 설명한다.

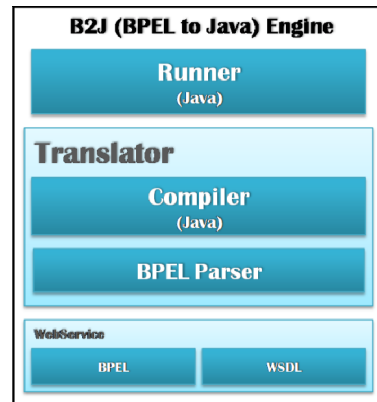
#### 3.1. B2J(BPEL to Java) 엔진

B2J 는 BPEL 문서를 입력 받아 BPEL 프로세스들을 처리하고 결과를 보여주는 엔진이다. B2J 엔진은 <그림 2>에서 보는 바와 같이 BPEL 과 WSDL 로 구성된 웹 서비스, 그리고 웹 서비스를 분석하고 파싱하여 자바 소스 파일로 변환하고 이를 컴파일하여 자바 클래스들로 변환하는 변환기, 변환된 자바 클래스들을 실행하는 실행기로 이루어진다. 다음은 B2J 의 각 부분에 대한 설명이다.

먼저, <그림 1>의 하단에 있는 웹 서비스는 입력 받은 BPEL 문서와 BPEL 문서에 기술된 WSDL 을 나타낸다. B2J 엔진의 두 번째 부분은 BPEL 파일을 WSDL 을 이용하여 분석하고 파싱하여 자바 소스 파일로 변환하고 이를 컴파일하여 자바 클래스들로 변환하는 변환기이다. B2J 엔진은 BPEL 파일이 입력되면 이를 실행하고 처리한 결과를 보여준다. BPEL 은 C, 자바 언어들과 같이 목적 코드를 가지고 컴파일하여 실행하는 언어와는 다르다. 그래서 BPEL 은

파싱한 자료 구조를 표현하기 위한 처리기가 있어야 한다. 처리기는 먼저 BPEL 파일 내에 기술된 WSDL 을 토대로 BPEL 파일을 분석한다. 분석한 BPEL 파일은 파싱 과정을 거쳐 자바 소스 파일로 변환된다. 이제 BPEL 파일은 자바 소스 파일로 변환이 되었다. 그러나 자바 소스 파일은 실행할 수 없다. 따라서 변환된 자바 소스 파일은 자바 컴파일러를 거쳐 자바 클래스들로 컴파일된다.

마지막으로 위와 같은 변환 과정을 거쳐서 나온 자바 클래스들을 실행하는 실행기가 있다. 실행기는 BPEL 파일이 실행 환경, 변환기를 거쳐서 나온 자바 클래스들을 B2J 엔진 내에 있는 API 들과 통신하여 실행하고 결과를 보여준다. 자바 클래스들은 BPEL 이 가지고 있는 내용을 모두 포함하나, 이를 처리하기 위해선 엔진 내에 있는 API 들을 이용해야 한다. 본 논문에서는 위의 설명을 위해 B2J 엔진을 분석한 후 로컬 엔진 부분을 응용하여 시스템에 적용하였다.



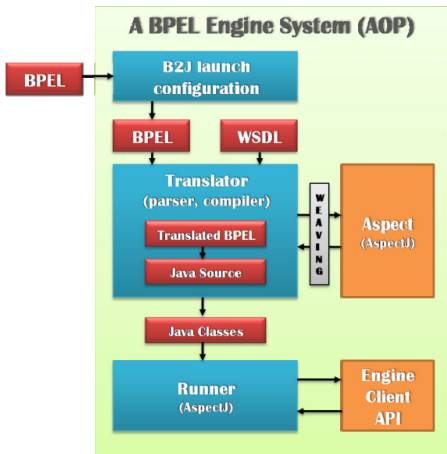
<그림 1> B2J 엔진 시스템

#### 3.2. AOP 기법을 적용한 BPEL 엔진 시스템

이 절에서는 3.1 에서 설명한 B2J 엔진을 기반으로 관점 지향 프로그래밍 기법을 적용한 BPEL 엔진 시스템을 설명한다.

이 시스템은 AOP 기법을 적용하여 웹 서비스의 흐름을 변경하고, 애스펙트를 작성하기 위해 AspectJ 언어를 선택했다. AspectJ 언어로 작성된 애스펙트를 직조하기 위해서는 기존 B2J 엔진을 수정해야 한다. B2J 엔진을 수정하기 위해 먼저 B2J 엔진의 실행 과정을 분석하였다. 수정한 부분은 <그림 1>에서 B2J 엔진의 변환기 내부와 실행기이다. <그림 2>는 본 시스템을 나타낸다. 변환기 내부의 자바 컴파일러를 AspectJ 컴파일러로 변경하고 실행기 또한 AspectJ 로 변경하였다. 이는 컴파일할 때 BPEL 파일을 변환한 자바 소스 파일뿐만 아니라, 애스펙트 또한 컴파일하고 직조하기 위함이다. 이로써 BPEL 파일을 변환한 자바 소스 파일 뿐만 아니라, 웹 서비스

스의 흐름을 변경할 수 있는 애스펙트를 컴파일할 수 있게 되었다. 또, 실행기에서는 BPEL 파일을 변환하여 컴파일한 자바 클래스들과 AspectJ 로 작성한 애스펙트의 직조가 가능해졌다. 이는 향후에 B2J 엔진이 업데이트 되어도 컴파일러 부분만 수정을 하면 되기 때문에 AOP 기법을 이용하기 위한 본 기술의 적용이 쉽다는 장점을 갖는다. 본 시스템은 앞서 말한 바와 같이 BPEL 문서를 입력으로 받는다. 입력은 B2J 실행 환경에서 받는데, 매개변수로 BPEL 문서를 입력하고 엔진을 실행한다. 입력된 BPEL 문서는 내부에 기술된 WSDL 문서와 함께 변환기를 거치게 된다. 변환기는 먼저 WSDL 문서를 따라 BPEL 문서를 분석하고, 파서와 컴파일러를 거친다. 파서는 분석된 BPEL 문서를 파싱하고, 파싱된 BPEL 문서는 변환기를 통해 자바 소스로 변환된다. 변환된 자바 소스는 변환기내의 AspectJ 컴파일러에 의해 컴파일된다. 이 때 AspectJ 로 작성된 애스펙트도 같이 컴파일된다. 자바 소스 파일은 컴파일되어 자바 클래스들로 변환되고, 애스펙트는 컴파일되어 AspectJ 클래스들로 변환된다. 원래 B2J 엔진에서는 자바 소스만이 컴파일되었으나, AOP 기법을 적용하기 위해 컴파일러를 AspectJ 컴파일러로 수정했기 때문에 가능한 일이다. 처음에 입력한 BPEL 문서와 애스펙트는 실행기에 의해 실행된다. 이 때, 실행기는 애스펙트에 기술한 내용에 따라서 자바 클래스들과 직조되고, 엔진 내부의 클라이언트 API 와 통신하여 결과를 나타낸다. 이 시스템을 쉽게 설명하기 위해 크게 실행환경 그리고 변환기, 실행기를 거치는 과정으로 표현했으나, 매개변수인 BPEL 문서를 입력하면 일련의 과정들이 한 번에 발생한다. 따라서, 사용자가 애스펙트로 서비스 전이 조건을 변경하는 모듈을 작성하고, BPEL 문서를 입력하면 이 시스템은 결과를 보여준다.



<그림 2> AOP 기법을 적용한 BPEL 엔진 시스템

### 3.3. 시나리오 기반 워크플로우

본 시스템은 입력으로 BPEL 문서를 받는다. 그래서 시나리오 기반의 워크플로우를 BPEL 문서로 작성하였다. <그림 3>의 ②는 회의 준비 시나리오를 작성한 BPEL 문서의 예이다.

<그림 3>의 ②는 John 의 일정을 나타낸다. John 이 자신의 사무일정을 PDA 에 기록하여 저장한 후, 오전 9 시 이전에 출근하여 책상 앞에 앉으면 워크플로우 서비스가 활성화된다. Display Schedule 에서는 John 이 오전 9 시 이전에 출근하여 책상 앞에 앉으면, 일정을 확인할 수 있도록 일정정보를 보여준다.

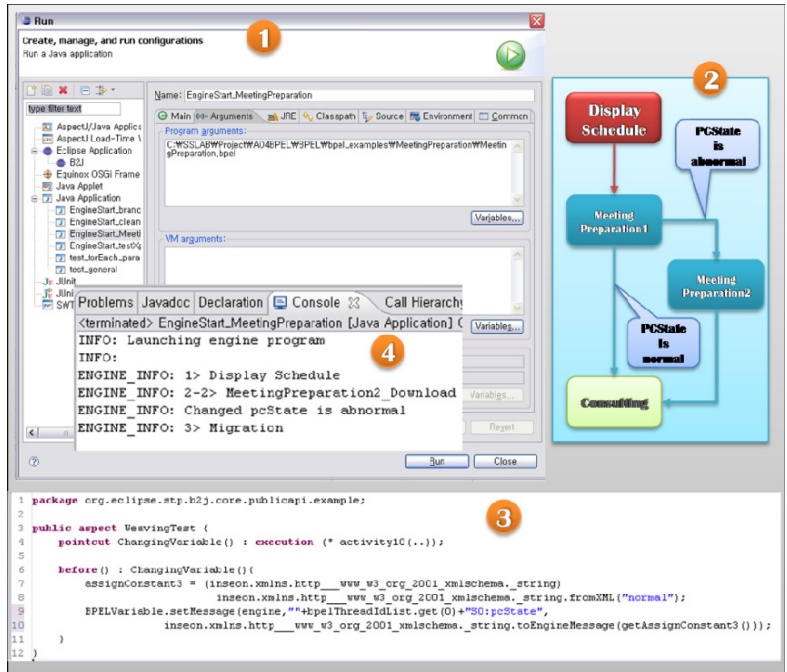
Meeting Preparation1 에서는 오전 10 시 313 호 회의실에서 발표자료를 John 의 컴퓨터로부터 313 호에 있는 컴퓨터로 다운로드한다. Meeting Preparation2 에서는 313 호의 컴퓨터에 이상이 있는 경우, 310 호 회의실에서 10 분 후에 미팅을 진행할 수 있도록 발표자료를 John 의 컴퓨터로부터 310 호에 있는 컴퓨터로 다운로드 한다.

마지막으로 Consulting 에서는 오후 1 시에 고객 상담을 위해 예약된 311 호 상담실로 작업환경을 John 의 컴퓨터로부터 311 호에 있는 컴퓨터로 마이그레이션 해준다.

이제 회의 준비 시나리오 기반으로 작성된 BPEL 문서가 AOP 를 적용한 워크플로우 시스템에서 실행되는 과정을 설명한다. 그림 5 는 실행 과정을 나타낸다.

<그림 5>의 ①은 시스템에서 BPEL 문서를 입력할 수 있는 환경이다. 3.3 에서 설명한 회의 준비 시나리오 를 기반으로 한 BPEL 문서를 입력하였다. 시스템이 실행되면 워크플로우가 처리되는데, 실행 중에 교차점을 만나면 <그림 5>의 ③의 AspectJ 코드가 직조된다. 회의 준비 시나리오 기반의 워크플로우는 먼저 일정 정보를 보여준다. 그리고 미팅 준비를 위해 발표 자료를 John 의 컴퓨터로부터 313 호 회의실에 있는 컴퓨터로 다운로드한다. 그러나 3 번의 AspectJ 코드가 미팅 준비전에 직조되면서 pcState 를 초기 상태인 "normal"에서 "abnormal"로 변경하기 때문에 컴퓨터에 이상이 있다고 판단하고 310 호 회의실에서 10 분 후에 미팅 준비를 할 수 있도록 발표 자료를 John 의 컴퓨터로부터 310 호 회의실에 있는 컴퓨터로 다운로드 한다. 마지막으로 고객 상담을 위해 예약된 311 호 상담실로 John 의 작업환경을 자신의 컴퓨터로부터 311 호에 있는 컴퓨터로 마이그레이션 해준다.

<그림 5>의 ④는 위의 실행 과정을 엔진에서 출력한 결과이다. 실행 중에 변경된 pcState 의 값도 출력하였다.



<그림 3> 시스템 실행 과정

### 5. 결론

본 논문에서는 현재 비즈니스 및 분산 컴퓨팅 환경에서 많이 이용되는 워크플로우 표준 언어인 BPEL 에 워크플로우 서비스의 전이조건의 재사용성을 극대화하기 위한 관점 지향 프로그래밍 기법을 적용한 BPEL 엔진 시스템을 제안하였다. 본 시스템은 기존의 BPEL 엔진을 거의 수정하지 않고 AOP 기법의 적용이 가능하다. 또한, 웹 서비스 분기 및 선택을 위한 서비스 전이 조건으로 요구되는 정보를 실행 도중에 변경하여 웹 서비스 흐름을 재구성할 수 있고, AOP 기법을 이용하여 서비스 전이 조건을 변경하는 모듈을 재사용할 수 있다. 향후 연구에서는 본 시스템을 확장하여 애스펙트로 컨텍스트 처리기를 추가하면 유비쿼터스 컴퓨팅 환경처럼 컴퓨팅 환경뿐만 아니라 센서, 사용자 정보 및 환경으로부터 얻을 수 있는 모든 정보들을 이용하는 서비스들이 실행될 수 있다.

#### 참고문헌

[1] Gregor Kiczales, John Lamping, Anurag Mendhekar, Chris Maeda, Cristina Videira Lopes, Jean-Marc Loingtier, and John Irwin, "Aspect-Oriented Programming," ECOOP, pp220-242, 1997

[2] "Aspect-Oriented Programming으로 모듈화 향상시키기: 자바 언어에 AOP를 가져다주는

AspectJ", <http://www.ibm.com/developerworks/kr/library/j-aspectj/#resources>

[3] Carine Courbis and Anthony Finkelstein, "Towards Aspect Weaving BPEL Engine", ACP4IS, 2004

[4] Carine Courbis and Anthony Finkelstein, "Towards Aspect Weaving Applications", ICSE'05, 2005

[5] Anis Charfi and Mira Mezini, "Aspect-Oriented Workflow Languages", OTM Conferences, 2006

[6] BPEL, "<http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html>"

[7] B2J, "<http://www.eclipse.org/stp/b2j>"

[8] Gregor Kiczales, Erik Hilsdale, Jim Hugunin, Mik Kersten, Jeffrey Palm and William G. Griswold, "An Overview of AspectJ"

[9] Anis Charfi and Mira Mezini, "Aspect-Oriented Web Service Composition with AO4BPEL", ECOWS 2004, LNCS 3250, pp. 168-182, 2004

[10] Anis Charfi and Mira Mezini, "AO4BPEL: An Aspect-Oriented Extension to BPEL", World Wide Web Journal: Recent Advances in Web Services (special issue), 2007