

BPEL 규칙 모델링을 위한 규칙 기반 BPEL 시스템 설계와 구현

Design and Implementation of Rule-based BPEL System for BPEL Rule Modeling

곽 동 규*, 최 재 영**

Donggyu Kwak*, Jaeyoung Choi**

Abstract

BPEL is a standard executable language for specifying actions within business processes with Web services. BPEL workflow requires rule engine to describe application process in BPEL, as the requirements increase. It is needed to develop new BPEL's engine or modify BPEL's engine in order to add the functionality of rules to BPEL engines, but this method is not easy to implement and it requires very high cost. In this paper, we present an R4BPEL document, which uses the original BPEL grammar and includes a rule document. With this method, it is possible to build rule-based BPEL environment easily by adding web services with rules and R4BPEL document analyzer to the general-purpose Web services. In this paper, we compared a BPEL's document in proposed system with a BPEL's document in the existing system. And we demonstrated the simplicity of the rule-based system.

요 약

BPEL은 작업의 순서를 기술하는 워크플로우의 표준으로서, 웹 서비스를 기반으로 하는 분산 컴퓨팅 환경의 많은 응용 분야에 적용되고 있다. BPEL의 요구사항이 복잡해지면서 규칙기반의 엔진이 요구되고 있다. 본 논문은 BPEL 문법을 그대로 이용하면서 규칙을 추가하기 위해 규칙을 기술할 수 있는 R4BPEL 문서를 제안하고 규칙 웹 서비스를 이용하여 BPEL에 규칙을 사용할 수 있도록 제안한다. BPEL 엔진에 규칙 기능을 추가하기 위해서는 BPEL 엔진을 새롭게 개발하거나 수정해야 하는데, 이 방법은 많은 비용이 많이 필요하다. 제안하는 방법은 기존의 범용 BPEL 시스템에 규칙 웹 서비스와 R4BPEL 문서 분석기를 추가하여 손쉽게 규칙 기반의 BPEL 환경을 구성할 수 있다. 본 논문에서는 제안하는 시스템에서의 BPEL 문서와 기존 방식의 BPEL 문서를 비교하여 제안하는 시스템의 규칙 기반 BPEL 문서가 단순함을 보인다.

Key words : BPEL, Rule Engine, Workflow, Modelling, R4BPEL

* School of Computer Science and Engineering,
Soongsil University

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

※ This research was supported by Basic Science
Research Program through the National Research
Foundation of Korea(NRF) funded by the Ministry of
Education, Science and Technology(2010-0025831)

Manuscript received Aug. 20, 2013; revised Sep. 5, 2013;
accepted Sep. 5, 2013.

1. 서론

워크플로우는 작업 절차를 통한 정보 또는 업무의 흐름이며, 워크플로우 시스템은 이를 적용한 시스템이다[1]. 워크플로우 시스템에서는 워크플로우 언어를 사용하여 작업의 흐름을 기술한다. 워크플로우 언어 중 BPEL (Business Process Execution Language)[2]은 OASIS (Organization for the Advancement of Structured Information Standards)[3]의 표준으로서

웹 서비스를 이용하여 프로세스의 흐름을 작성할 수 있는 XML 기반의 언어이다. BPEL은 XML 변수와 웹 서비스 호출, 조건 분기, 루프, 그리고 비동기식 병렬 실행을 지원한다. 사용자는 이와 같은 BPEL의 기능을 이용하여 비즈니스 프로세스의 흐름을 기술한다.

하지만 BPEL이 제공하는 워크플로우만으로는 복잡한 비즈니스 규칙을 표현하기 위해서는 복잡한 조건문을 작성해야 하는데, 조건이 복잡할수록 기술하기 어렵고 사용자의 의도와 다르게 동작할 가능성이 높아지게 된다. 복잡한 조건을 처리하기 위해서는 규칙을 사용할 수 있다[4]. 규칙은 조건이 충족할 경우의 실행을 식별자로 추상화시킨다. 하지만 BPEL은 규칙을 제공하지 않는다. 규칙은 복잡하고 복잡한 조건을 단순화하고 조건이 충족할 때에 실행할 수 있는 동작을 기술하여 비즈니스 규칙을 손쉽게 모델링할 수 있도록 한다.

일반적으로 BPEL 엔진에 규칙 기능을 추가하기 위해서는 BPEL 문법을 확장하여 규칙 기능을 작성할 수 있도록 하고, 확장 문법을 처리하는 BPEL 엔진을 새롭게 개발하거나 기존의 BPEL 엔진을 수정해야 한다. 하지만 이 방법은 많은 비용이 많이 필요하다. BPEL 엔진을 수정하지 않고 규칙을 적용하기 위하여 BPEL의 흐름을 인터셉트하여 규칙을 적용한다. BPEL의 흐름을 인터셉트하는 방법은 두 가지가 있다. 첫 번째는 웹 서비스의 호출을 인터셉트하는 방법이고 두 번째는 BPEL의 흐름을 인터셉트하는 방법이다. 웹 서비스의 호출을 인터셉트하는 방법은 Florian Rosenberg의 연구가 있다[5]. BPEL은 “invoke”를 이용하여 웹 서비스로 프로세스를 호출한다. Rosenberg는 invoke의 이전과 이후에 적용할 규칙을 기술하고 이를 처리하는 규칙 엔진을 제안하였다. 하지만 규칙으로 생성된 값을 BPEL의 분기나 다른 서비스의 입력 값으로 사용하기 어려워 규칙 기능을 한정적으로 사용하도록 제한하였다. 그리고 BPEL의 흐름을 인터셉트하는 방법으로는 AspectJ[6]를 이용하는 R4BPEL[7]이 있다. 이 연구에서는 손쉽게 BPEL 흐름에 규칙을 적용하기 위해 R4BPEL 문서를 제안하고 이를 처리하는 엔진을 구현하였다. 이 엔진은 BPEL 엔진 중 하나인 B2J (BPEL to JAVA) 엔진[8]을 사용하는데, B2J 엔진은 BPEL 문서를 JAVA 프로그램으로 변환하고 실행한다. 또한 AspectJ는 JAVA에 관점지향 프로그래밍 기법[9]을 적용한 언어이다. AspectJ는 직조(weaving)를 이용하여 함수를 호출하지 않고 개발되어 있는 프로그램에 기능을 추가할 수 있다. 하지만 이 방법은 B2J 엔진을 제외한 다른 BPEL 엔진에 적용하기가 어렵다.

본 논문은 규칙을 BPEL 흐름에 손쉽게 적용하고 다양한 범용 BPEL 엔진을 제사용하기 위해 R4BPEL 문서를 처리하는 규칙 웹 서비스 기반의 R4BPEL 엔진을 제안한다. 제안하는 시스템은 규칙을 추상적으로 사용하기 위해 BPEL 문법을 확장한 R4BPEL 문법을 사용한다. 그리고 규칙을 처리하기 위해 JAVA 기반의 규칙 엔진을 사용하는데, 범용 BPEL은 JAVA 프로그램을 직접 사용할 수 없다. 그러므로 본 연구에서는 BPEL 엔진에서 규칙을 사용할 수 있도록 규칙 웹 서비스를 개발하고, R4BPEL 문서로 작성된 규칙을 인자로 규칙 웹 서비스에 전송될 수 있도록 범용 BPEL 엔진을 확장할 수 있는 방법을 보인다. 규칙 웹 서비스를 이용한 방법은 기존의 워크플로우 시스템을 그대로 사용하면서 규칙 웹 서비스와 문법 변환기만을 추가하여 규칙을 워크플로우에 적용할 수 있다. 또한, 규칙 엔진으로 Drools[10]를 이용한다. Drools는 JBoss에서 개발한 엔진으로 이클립스 플러그인 형태로 되어 있고 전용 편집기와 XML 기반의 규칙문서를 제공하고 있어 많은 분야에서 사용되고 있다. 그리고 실험에서 규칙을 이용한 BPEL 문서가 기존의 BPEL 문서에 비해 단순함을 보인다.

본 논문은 2장에서 관련 연구를 보이고, 3장에서 규칙 문서의 구조에 관해 설명한다. 4장에서 본 논문에서 제안하는 시스템의 구조를, 5장에서 간단한 실험과 각 시스템을 비교한 후, 6장에서 결론을 맺는다.

II. 관련연구

1. Model-based Verification of BPEL

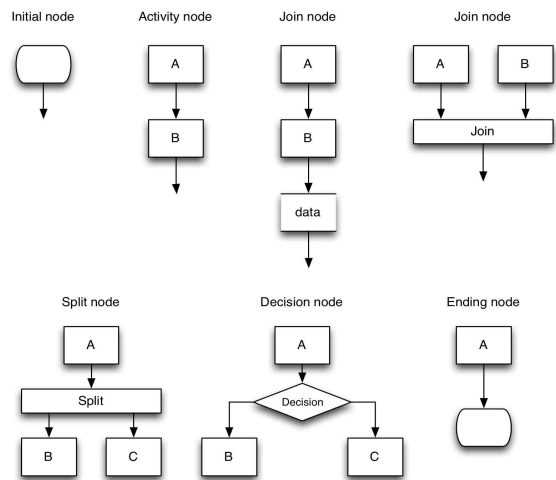


Fig. 1. Type of BPEL business's Rule
 그림 1. BPEL 비즈니스 규칙의 종류

BPEL은 비즈니스 작업의 흐름을 기술하고 조건에 따라 웹 서비스를 호출하여 사용자에게 서비스를 제공한다. 하지만 요구사항이 복잡해지면서 요구사항을 표현하는 BPEL 문서도 복잡해지고 있다. 이에 따라, Cao는 BPEL로 작성된 비즈니스 로직을 검증할 필요성을 지적하고 이를 위해 BPEL 비즈니스 규칙의 종류를 7 가지로 구분하였다[11]. 그림 1은 Cao 연구자가 제안한 BPEL 비즈니스 규칙의 종류를 보인다.

워크플로우의 요구사항이 복잡해지면서 BPEL 문서의 복잡도도 증가하고 있다. 본 연구에서는 BPEL에 규칙을 적용하여 BPEL 문서의 복잡도를 줄인다. 본 논문에서는 Cao 연구자가 제안한 모델링 방법으로 실험 시나리오를 모델링하고 이를 제안하는 시스템으로 변환하여 노드를 갯수를 비교하여 복잡도를 측정한다.

2. BRIB (Business Rule Integration in BPEL)

Rosenberg는 BPEL에 규칙을 추가한 BRIB를 제안하였다. Rosenberg는 BPEL 문법을 그대로 이용하면서 규칙을 적용하기 위해 BPEL 엔진이 웹 서비스를 호출하는 “invoke”를 인터셉트하여 규칙 엔진을 호출하는 방법을 사용하였다. 그림 2는 BRIB에서의 웹 서비스 호출 구조를 보인다.

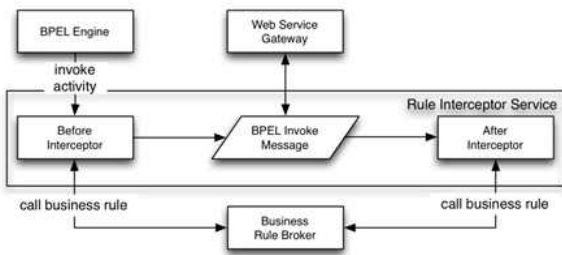


Fig. 2. Rule Intercept Service of BRIB
그림 2. BRIB의 규칙 인터셉트 서비스

BRIB는 BPEL 문법을 수정하지 않고 규칙 맵을 통해 BPEL 엔진에 규칙을 적용할 수 있다. 하지만 규칙 엔진의 호출이 “invoke”에만 국한되어 규칙에 비교 대상을 invoke의 인자 외에는 사용할 수 없어 규칙 엔진을 한정적으로 제한한다.

3. AOP를 이용한 규칙 기반 BPEL

BPEL은 워크플로우의 표준으로 다양한 응용에서 사용되고 있다. BPEL의 요구사항이 복잡해지면서 규칙 기반의 엔진이 요구되고 있다. R4BPEL은 BPEL

문법을 그대로 유지하면서 규칙을 추가하기 위해 제안된 XML 기반 언어이다. 이 연구에서는 횡단관심사를 모듈화한 관점지향 프로그래밍을 이용하여 새롭게 추가된 규칙 기능을 추가한다. 그림 3은 관점지향 프로그래밍 기법 기반의 R4BPEL 시스템의 구조를 보이고 있다.

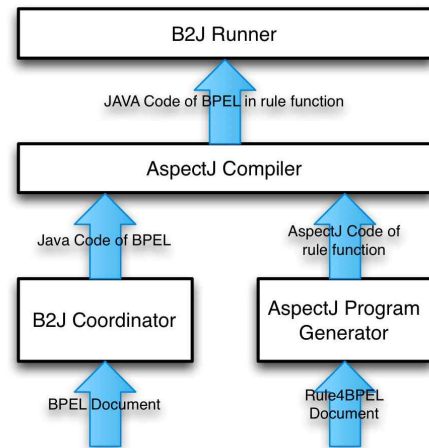


Fig. 3. System Structure of R4BPEL using B2J based AOP
그림 3. R4BPEL using B2J based AOP의 시스템 구조

관점지향 프로그래밍 기법을 이용한 R4BPEL 시스템은 직조(weaving)를 이용하여 BPEL 엔진을 수정하지 않고 규칙을 적용할 수 있다. 이 방법은 관점지향 프로그래밍 기법을 사용하기 때문에 B2J (BPEL to JAVA) 엔진을 사용한다. B2J 엔진은 BPEL 엔진 중 하나로 BPEL 문서를 JAVA 프로그램으로 변환하여 이 프로그램을 실행하여 BPEL 문서의 요구사항을 실행한다. 이 시스템은 B2J 엔진이 변환한 JAVA 프로그램에 직조될 규칙을 관점지향 프로그램으로 생성하여 B2J 엔진을 수정하지 않고 규칙을 적용한다. 하지만 이 연구에서 사용한 방법은 B2J 엔진에만 적용할 수 있어 기존에 B2J를 사용하지 않은 시스템에서 사용하기 위해서는 BPEL 엔진을 교체해야 한다.

III. 규칙문서 (Rule4BPE : Rule for BPEL)

본 논문에서 제안하는 시스템은 규칙을 BPEL 문서의 다양한 위치에 사용할 수 있도록 정의하여, 규칙이 생성한 결과를 분기에 사용하거나 서비스의 입력으로 사용할 수 있는 장점을 가진다. 그림 4는 제안하는 규칙이 적용된 워크플로우를 보인다.

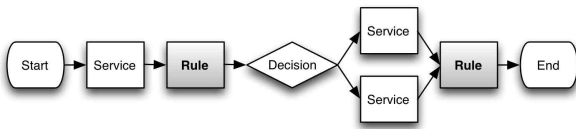


Fig. 4. Workflow using Rules
 그림 4. 규칙이 사용된 워크플로우

규칙 엔진은 상태에 따라 동작을 정의하는 규칙을 시행하는 프로그램이다. 규칙 엔진 중에서 Drools는 JBoss에서 개발한 엔진으로 이클립스 플러그인 형태로 되어 있고 전용 편집기도 제공하여 많은 분야에서 사용되고 있다. Drools는 전용 비즈니스 규칙 언어를 갖는다. 그림 5는 Drools가 제공하는 XML 기반의 규칙 언어를 보이는 예제이다.

```

01<?xml version="1.0" encoding="euc-kr" ?>
02 <rule-set name="Hello" ... >
03 <import java.lang.Object</import>
04 <import java.lang.String</import>
05 <java:functions>
06 public static void helloWorld(String hell) {
07     System.out.println(hello + " World"); }
08 public static void goodbyeWorld(String goodbye) {
09     System.out.println(goodbye + " Cruel World"); }
10 </java:functions>
11 <rule name="Hello World">
12 <parameter identifier = "hello">
13 <class>String</class>
14 </parameter>
15 <java:condition>hello.equals("Hello")</java:condition>
16 <java:consequence>helloWorld(hello)</java:consequence>
17 </rule>
18 <rule name="Goodbye Cruel World">
19 <parameter identifier = "goodbye">
20 <class>String</class>
21 </parameter>
22 <java:condition>goodbye.equals("Goodbye")</java:condition>
23 <java:consequence>goodbyeWorld(goodbye);</java:consequence>
24 </rule>
25 </rule-set>
    
```

Fig. 5. Drools's Rule Document-based XML
 그림 5. Drools의 XML 기반 규칙 문서

그림 5는 간단한 규칙 문서의 예로서 규칙의 특징을 보여준다. 규칙은 복잡한 조건과 조건에 합당할 경우의 실행을 결합하여 식별자 (Identifier)로 추상화한다. 규칙을 시스템에 적용할 경우 사용자에게 제공할 서비스를 독립적인 문서로 모델링할 수 있는 방법을 제공하고 복잡한 조건을 단순화할 수 있다. 본 연구에서는 서비스의 흐름을 기술하는 BPEL과 함께 규칙을 사용할 수 있는 방법을 보인다. 본 연구에서 보이는 규칙 문서는 Drools의 XML 기반 규칙 문서를 확장하여 정의한다. 그림 6은 규칙 문서인 R4BPEL의 스키마를 보인다.

그림 6과 같이 규칙 문서는 규칙을 기술하는 부분 (rules)와 규칙의 결합을 기술하는 부분 (apply-rule)으로 구성되어 있다. 규칙을 기술하는 부분은 Drools에서 제공하는 XML 기반 규칙 문서를 사용하고, 규칙 문서는 R4BPEL 문서 외부의 문서를 참조할 수 있다. 규칙 문서의 외부 참조는 서비스 모델링과 워크플로우간의 독립성을 보장한다. 그리고 규칙의 결합을 기술하는 부분은 규칙이 BPEL 문서에 결합되어야 할 위치와 규칙 적용에 필요한 정보를 기술한다. 이렇게 작성된 문서는 Drools 규칙 문서로 변환되어 규칙 웹 서비스를 통해 실행된다.

IV. 시스템 구조

본 장에서는 3장에서 보인 R4BPEL 문서에 작성된 규칙이 BPEL 문서에 적용하기 위한 시스템 구조를 보인다. 제안하는 시스템은 워크플로우가 작성되어 있는 BPEL 문서와 규칙을 이용하여 서비스를 모델링한 규칙 문서를 입력으로 하는 워크플로우와 서비스 모델링을 제공한다. 그림 7은 본 논문에서 제안하는 시스템의 실행 흐름을 보여준다.

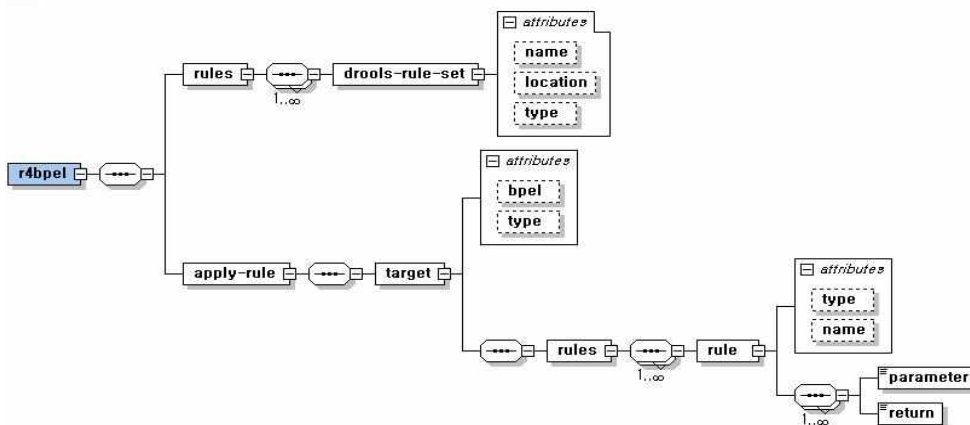


Fig. 6. Schema of R4BPEL
 그림 6. R4BPEL 스키마

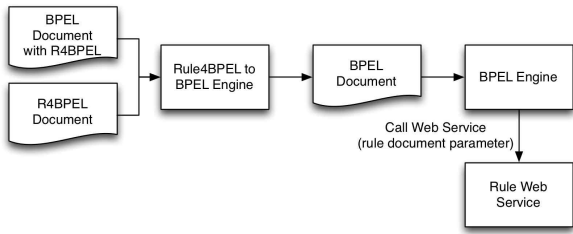


Fig. 7. Execution Flow of Propose System
 그림 7. 제안하는 시스템의 실행 흐름

그림 7과 같이 제안하는 시스템은 Rule4BPEL 변환기 (Rule4BPEL to BPEL Engine)와 BPEL 엔진 (BPEL Engine), 규칙 웹 서비스 (Rule Web Service)로 구성되어 있다. 시스템의 입력은 BPEL 문서와 R4BPEL 문서인데 BPEL 문서에는 작업의 흐름이 기술되어 있고 R4BPEL에는 작업의 흐름간에 호출될 규칙이 기술되어 있다. BPEL 문서와 R4BPEL 문서는 Rule4BPEL 변환기를 통해 BPEL 문서로 변환되고 규칙은 변환된 BPEL 문서의 변수에 할당되어 규칙 웹 서비스가 호출될때 인자로 전달되게 된다. 규칙 웹 서비스는 전달받은 인자로 사용자에게 비즈니스 규칙을 제공한다. 그림 8은 Rule4BPEL 변환기의 구조를 보인다.

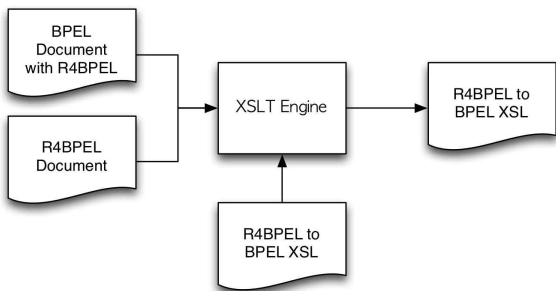


Fig. 8. Structure of Rule4BPEL's Translator
 그림 8. Rule4BPEL 변환기 구조

본 시스템에서 Rule4BPEL 변환기는 XML 문서 변환의 W3C 표준인 XSLT(Extensible Stylesheet Language Transformations)[]를 사용한다. XSLT를 사용하면 XSL 문서만으로 기존의 XSLT 엔진을 사용하여 문서의 변환을 할 수 있다. 제안하는 방법은 기존의 BPEL 실행 환경에 Rule4BPEL 변환기를 추가하여 BPEL에 규칙을 추가하여 사용할 수 있어 기존의 시스템을 재사용할 수 있다.

V. 실험 및 시스템 비교

본 논문에서는 제안하는 시스템의 사용 예제로서, 인사정보를 이용한 급여 시스템을 보인다. 실험에서 예로 사용하는 급여 시스템은 인사정보 데이터베이스로부터 인사 정보를 받아 급여에 관한 규칙을 적용하는 급여 정보 데이터베이스에 저장한다. 그림 9는 제안하는 실험 시나리오를 보여주고 있다.

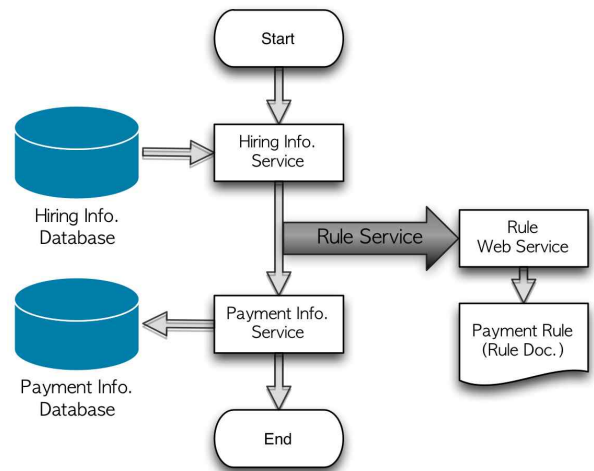


Fig. 9. Structure of Payment System
 그림 9. 급여 시스템 구조

그림 9에서 인사정보 데이터베이스는 기업의 인사정보를 저장하고 있으며, 인사정보 서비스는 데이터베이스로부터 인사정보를 출력한다. 그리고 인사정보에 따라 급여지급 규칙이 적용되는데, 이는 규칙문서인 R4BPEL에 작성되어 있고 규칙 서비스를 통해 적용된다. 제안하는 시스템은 BPEL 문서를 자바 프로그램으로 변환하여 실행하는 BPEL을 이용하여 규칙을 적용할 수 있다.

BPEL은 조건과 웹 서비스 실행을 제공하고 있어 다른 추가적인 기능이 없더라도 조건에 따른 실행을 기술할 수 있다. 하지만 복잡한 조건에 따른 실행에서는 사용자의 요구사항에 맞는 문서를 작성하기 어렵다. 복잡한 조건에 따른 실행을 이해하기 쉽게 작성하기 위해서는 규칙을 사용할 수 있는데 하지만 BPEL 문서에는 규칙을 제공하지 않아 BPEL에서 사용할 수 있는 웹 서비스를 이용하여 규칙을 사용해야 한다. 그러나 이 경우에는 사용자가 규칙을 위한 변수를 직접 다루어야 하기 때문에 사용하기 쉽지 않다. 그러므로 본 연구에서는 BPEL 문법을 확장하여 R4BPEL을 제안한다. 본 장에서는 5장 실험에서 보인 규칙 기반의 급여 시스템의 요구사항을 두 가지 형태

로 구현한 경우를 보인다. 표 1은 본 장에서 보이는 두 가지 시스템의 특징을 보인다.

Table 1. Comparison of System Feature

표 1. 시스템 특징의 비교

	BPEL	Rule based BPEL
Rule4BPEL	미사용	사용
Rule Service	미사용	사용
BPEL	사용	사용

표 1에서와 같이 본 장에서 비교하는 시스템은 두 가지이다. 한 가지는 BPEL 문서만을 이용한 경우 (BPEL 시스템)이고 다른 하나는 제안하는 규칙 기반 문서를 사용하는 경우 (Rule based BPEL 시스템)를 보인다. 첫 번째 BPEL 시스템은 사용자가 직접 BPEL에서 제공하는 조건과 웹 서비스를 이용하여 모든 조건에 따른 실행을 작성해야 한다. 그림 10은 BPEL 문서만을 이용한 시스템을 2장에서 보인 Cao 연구자의 모델링 방법으로 보인다.

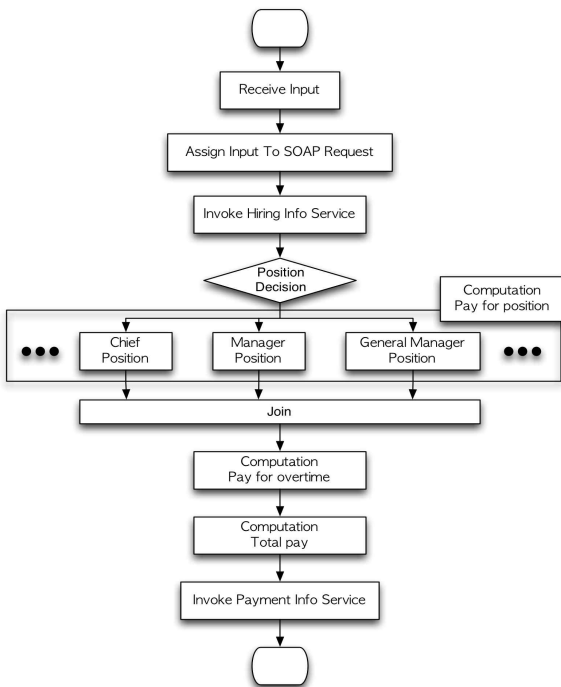


Fig. 10. Payment System Model using BPEL Document
 그림 10. BPEL 문서를 이용한 급여 시스템 모델

그림 10과 같이 BPEL 문서만을 이용하여 급여 시스템을 작성하기 위해서는 워크플로우에서 인사 정보 웹 서비스를 호출한 후 사용자에게 직급에 따라 분기하여 직급에 따른 급여를 결정한다. 그리고 시간 외 수당을 연산한 후 급여 웹 서비스를 호출하여 데이터를 저장한다. 이 방법은 총 10개의 노드와 함께 직급의

개수 만큼의 노드가 필요하다. 그림 11은 본 연구에서 제안하는 규칙 문서를 이용한 시스템 모델을 보인다.

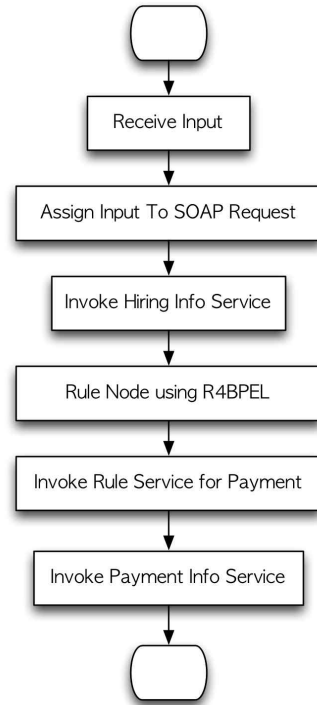


Fig. 11. Payment System Model using R4BPEL Document
 그림 11. R4BPEL 문서를 이용한 급여 시스템 모델링

그림 11과 같이 인사 웹 서비스를 호출한 후 규칙 문서를 통해 규칙이 적용된다. 규칙이 적용된 급여 정보는 급여 웹 서비스를 통해 실행되어 급여 정보에 저장된다. 이 방법은 8개의 노드가 사용되고 있으며 규칙을 다루기 위한 변수의 할당을 사용자가 작성할 필요가 없다. 표 2는 앞서 소개한 2 가지 경우를 비교하고 있다.

Table 2. Comparison of System Modelling

표 2. 모델링 비교

	BPEL	Rule based BPEL
노드의 갯수	10 + a	8
조건 분기	있음	없음
규칙변수할당	없음	없음

표 2와 같이 세 가지 경우를 노드의 갯수와 조건 분기의 유무, 복잡한 규칙 변수 할당을 비교하면 제안하는 시스템이 문서의 복잡한 정도가 낮다.

VI. 결론

BPEL은 비즈니스 프로세스를 정의하고 실행하기 위한 워크플로우의 표준으로서, XML 기반으로 하여 학습이 용이하여 다양한 응용에서 사용되고 있다. BPEL은 조건을 처리할 수 있는 조건문을 제공하고 있는데 조건의 복잡도가 높아질수록 기술하기 어렵고 이는 사용자의 의도와 다르게 동작할 가능성을 높인다.

본 논문은 BPEL에 규칙을 추가하기 위해 R4BPEL 문서를 제안하고 이를 위한 시스템을 보인다. 일반적으로 BPEL 엔진에 규칙 기능을 추가하기 위해서는 엔진을 새롭게 개발하거나 기존의 엔진을 수정해야 한다. 하지만 새로운 엔진의 개발이나 기존 엔진의 수정은 많은 비용이 필요하다. 이에 따라 관점지향 프로그래밍 기법과 B2J 엔진을 이용한 R4BPEL 시스템은 BPEL 엔진 중 하나인 B2J 엔진을 수정하지 않고 관점지향 프로그래밍 기법의 직조를 통해 규칙을 적용한 시스템이 있었다. 하지만 이 시스템은 B2J 엔진에만 적용할 수 있어 기존의 BPEL 환경에 바로 적용하기 어렵다.

제안하는 시스템은 기존의 범용 BPEL 엔진을 사용하는 환경에 쉽게 적용할 수 있는 규칙 기반 BPEL 엔진이다. 본 연구에서는 규칙 엔진 중 하나인 Drools를 이용하여 규칙 웹 서비스를 개발하고 규칙이 기술되어 있는 R4BPEL 문서를 웹 서비스의 인자로 전달하여 규칙을 수행하도록 한다. 본 시스템은 기존의 BPEL 환경에 규칙 웹 서비스 추가와 R4BPEL 문서 처리기만을 도입하여 규칙 기능을 추가한다.

References

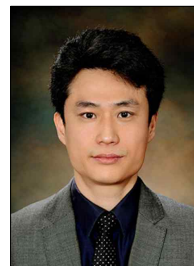
- [1] O. Zimmermann, V. Doubrovski, J. Grundler, K. Hogg, "Service-oriented architecture and business process choreography in an order management scenario: rationale, concepts, lessons learned", *Proceeding OOPSLA '05 Companion to the 20th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications*, pp.301-312, 2005.
- [2] BPEL, <http://www-128.ibm.com/developerworks/library/specification/ws-bpel/>.
- [3] OASIS, <http://www.-oasis-open.org>.
- [4] JBoss Rules, <http://labs.jboss.com/jbossrules/docs>.
- [5] F. Rosenberg and S. Dustdar, "Business Rules Integration in BPEL - A Service-Oriented

Approach", *In Proceedings of the 7th International IEEE Conference on E-Commerce Technology (CEC 2005)*, 2005.

- [6] AspectJ, <http://www.eclipse.org/aspectj/>
- [7] 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.
- [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] Drools, <http://www.jboss.org/drools/>.
- [11] Honghua Cao, Shi Ying, Dehui Du, "Towards Model-based Verification of BPEL with Model Checking", *Computer and Information Technology 2006*, pp.190-194, 2006.

BIOGRAPHY

Donggyu Kwak (Member)



2002: B.S. in Dept. of Applied Mathematics, Seokyeong University, Seoul, Korea
 2004: M.S. in School of Computer Science and Eng. Soongsil University, Seoul, Korea
 2012: Ph.D. in School of Computer Science and Eng. Soongsil University

Jaeyoung Choi (Member)



1984: B.S. in Dept. of Control and Instrumentational Eng., Seoul National University
 1986: M.S. in Dept. of Electrical Eng., University of Southern California
 1991: Ph.D. in School of Electrical Engineering, Cornell University
 1995~present: Professor, School of Computer Science and Engineering, Soongsil University,