

관점지향 기법의 상황인지를 지원하는 BPEL 워크플로우 시스템

김민석*, 광동규*, 최종선*, 최재영*¹⁾

*송실대학교 컴퓨터공학과

e-mail:minseok84@gmail.com

A Context-aware BPEL Workflow System using Aspect-Oriented Programming

Min-suk Kim*, Donggyu Kuak*, Jong-Sun Choi*, Jae-Young Choi*

*Computer Science & Engineering, Soong-Sil University

요 약

표준 워크플로우 언어로 가장 인지도가 높은 BPEL은 분기를 통한 플로우선택에 있어 상황정보를 기술하기 어려워 유비쿼터스 컴퓨팅 환경에 적용하기 어렵다. 이를 위해 본 논문에서는 모듈간 낮은 결합도를 보장하는 관점지향 프로그래밍 (AOP: Aspect-Oriented Programming) 기법을 사용하여 기존 BPEL 워크플로우 시스템에 상황인지 기능을 추가한 시스템을 제안한다. 제안하는 상황인지 워크플로우 시스템은 AOP 기법을 사용하여 BPEL 워크플로우에 상황에 따른 서비스를 제공하기 위한 Context 정보를 삽입하는 방식으로 개발함으로써, 기존 BPEL 문서를 수정할 필요 없이 상황인지 기능을 적용한 워크플로우의 생성이 가능하다. 본 논문에서는 시나리오 기반의 실험을 통하여 제안한 시스템을 입증한다.

1. 서론

BPEL (Business Process Execution Language)은 웹 서비스 환경에서 비즈니스 프로세스를 정의하고 실행하는 표준 워크플로우 언어이다 [1]. BPEL은 서로 연계된 작업 흐름을 갖는 실세계 문제들을 해결하기 위한 핵심 솔루션으로 사용되고 있으며, 특히 사무 자동화를 필두로 한 BPMS (Business Process Management System), 의료, 교육, 헬스케어 등 다양한 분야에 적용되어 사용되는 전통적인 IT 시스템이다.

임베디드 장비의 소형화 및 무선통신의 발달로 인하여 보이지 않는 무선네트워크 인프라의 구축이 가능해졌고, 그로 인하여 유비쿼터스 컴퓨팅이 사회 전반에 걸쳐 핵심 인프라로 떠오르게 되었다. 따라서 기존에 사용되고 있던 전통적인 IT 시스템들을 유비쿼터스 컴퓨팅 환경에 적합하게 재구현하거나 새로운 솔루션을 개발하는 작업이 여러 곳에서 이루어지고 있다. 유비쿼터스 컴퓨팅 환경의 어플리케이션은 사용자 주변 상황을 인지하여 적합한 시기에 적합한 서비스를 제공하는 것을 목적으로 한다. 따라서 유비쿼터스 환경을 위한 컴퓨팅 시스템을 개발할 때 가장 고려해야 할 사항은 상황인지 기능이다.

유비쿼터스 컴퓨팅 환경을 위한 상황인지 워크플로우

시스템을 제공하기 위해선 두 가지 개발 방법을 취할 수 있다. 하나는 상황인지 기능을 제공하기 위한 워크플로우 언어를 새롭게 설계하고 그에 해당하는 워크플로우 엔진을 새로 개발하는 방법이고, 또 하나는 기존의 워크플로우 언어를 재사용하여 유비쿼터스 환경의 상황을 기술할 수 있는 문법을 추가하는 방법이다.

첫 번째 방법은 새로운 언어에 대한 학습이 필요하다. 또한 기존에 개발된 응용이나 시스템을 재사용할 수 없다는 치명적인 단점이 존재한다. 반면에 두 번째 방법은 추가로 기술하는 문법 이외에 새로운 언어에 대한 학습이 필요없고, 기존 시스템 및 언어에 대한 재사용이 가능하다는 장점을 지닌다. 그러므로 유비쿼터스 환경을 위한 상황인지 워크플로우 시스템을 개발하기 위해서는 기존에 개발된 워크플로우 어플리케이션, 편집기, 개발환경 등을 재사용할 수 있도록 하는 것이 보다 효율적이다.

기존 워크플로우 언어를 유지시킨 채로 새로운 기능을 추가하려면 새로운 방식의 프로그래밍 방법론이 필요하다. 관점지향 프로그래밍 (AOP) [2]은 관심사를 핵심 관심사 (Core concern)와 횡단 관심사 (Cross-cutting concern)로 분류하고 모듈화 되어있는 핵심 관심사에 횡단 관심사를 직조하는 방식으로 기존 모듈을 유지한 채로 새로운 기능을 추가할 수 있다.

본 논문은 AOP를 사용하여 BPEL에 상황인지 기능을 추가한 시스템을 제안한다. 그리고 상황인지를 기술하기 위한 CAML이라는 새로운 문서 모델을 제안한다. 제안하

이 논문은 2011년도 정부(교육과학기술부)의 재원으로 한국연구재단의 기초연구사업 지원을 받아 수행된 것임(2010-0025831)

1) 교신저자

는 시스템은 기존 BPEL 언어의 재사용이 가능하면서 BPEL 워크플로우에 상황인지 처리를 가능하게 한다는 특징을 가진다.

본 논문은 2장에서 관련연구를 보이고 3장에서 시스템의 구조를 보인다. 그리고 4장에서 CAML 문서 모델을 소개한 후, 5장에서 결론에 대해 논한다.

2. 관련연구

본 논문의 목적은 관점지향 프로그래밍 기법을 이용하여 BPEL 언어 수준에서 기술하기 어려운 상황인지 기능을 추가하는 방법을 보이고 이를 이용하여 유비쿼터스 환경에서 사용 가능한 워크플로우 시스템을 설계하는데 있다. BPEL에 상황인지 기능을 추가하려는 대표적인 연구인 Context4BPEL은 BPEL 문법에 상황인지 문법을 추가하고, 그에 맞추어 BPEL 엔진을 새롭게 구성하였다 [3]. 이를 통하여 BPEL 문서 수준에서 상황인지를 기술 가능하게 되었지만, 기존의 BPEL 문서에 상황인지를 기술하기 위해선 BPEL문서 자체를 수정해야 하고 Context-4BPEL로 생성한 BPEL 문서를 다른 범용 BPEL 엔진에서 사용할 수 없다는 한계를 지니고 있었다. BPEL 문서를 유지하면서 상황인지 기능을 추가하기 위해선 다른 방식의 접근이 필요하다.

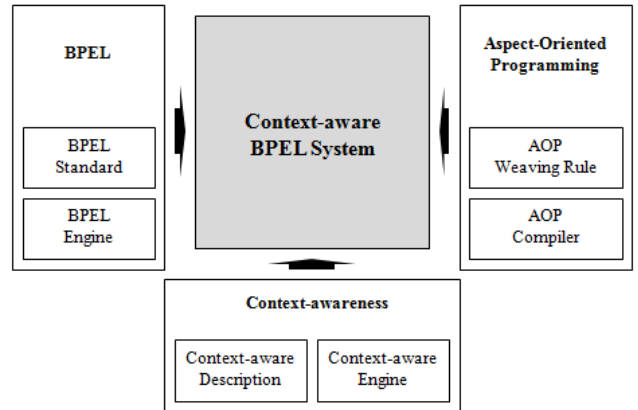
AO4BPEL이라는 연구는 Aspect-Oriented Programming 이라는 방법을 통하여 BPEL 문서를 유지한 채로 새로운 기능을 추가하는 합리적인 방법을 제안하였다 [4]. AO4BPEL은 BPEL에서 제기된 모듈화의 어려움을 해결하고 서비스를 동적으로 지원하기 위하여 AOP 기법을 사용하였다. BPEL의 기존 실행 단위 (Activity)들은 모두 수용하고 AOP기법을 적용한 실행 단위들을 추가하였다. 즉 BPEL 문서 내에 AOP기법에 해당하는 관점 (Aspect), 교차점 (Pointcut), 충고 (Advice) 등을 일컫는 실행 단위들이 포함된다. 또한 이를 실행하기 위해 BPEL 엔진 시스템에 관점을 등록하고 활성화하는 도구와 관점의 실행을 제어하는 관점 매니저를 추가하여 설계하였다. AO4BPEL에서는 주로 부가적인 관심사를 모듈화하고 컴포지션 변경을 모듈화할 수 있음을 보여준 연구로 의의가 크다. 그러나 AO4BPEL은 AOP 기법으로 워크플로우의 컴포넌트들에 대한 제어 이외의 부가 기능들을 추가하였을 뿐 워크플로우의 컴포넌트들을 제어하지는 못하였다.

이러한 한계를 극복하기 위하여 본 논문에서는 유비쿼터스 환경의 워크플로우를 위하여 상황인지 기능을 지원하기 위한 언어 수준에서 복잡한 규칙과 상황정보를 추가할 수 있는 시스템을 AOP 기법을 적용하여 제안한다.

3. 시스템 구성

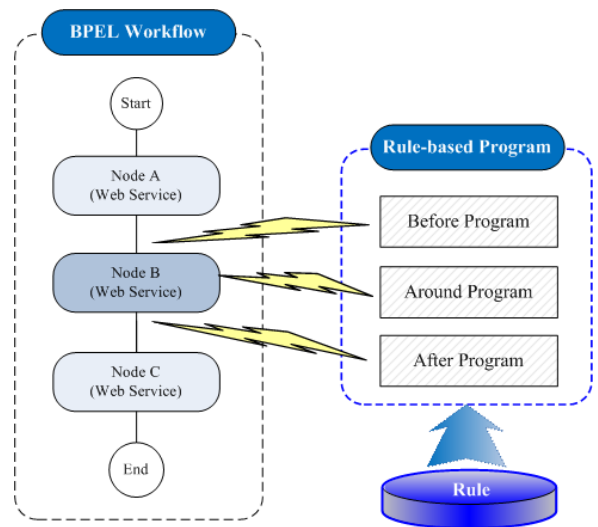
본 논문은 상황인지 BPEL 워크플로우를 설계함에 있어 그림 1과 같이 BPEL 시스템의 언어 표준 및 BPEL 엔진, AOP의 직조규칙 및 AOP 컴파일러, 그리고 상황인

지 시스템을 위한 상황 표현 및 상황인지 처리 엔진을 이용하여 시스템을 구성하였다.



(그림 1) 제안하는 워크플로우 시스템의 개념도

제안하는 시스템을 통해 생성되는 워크플로우의 형태는 그림 2와 같다.

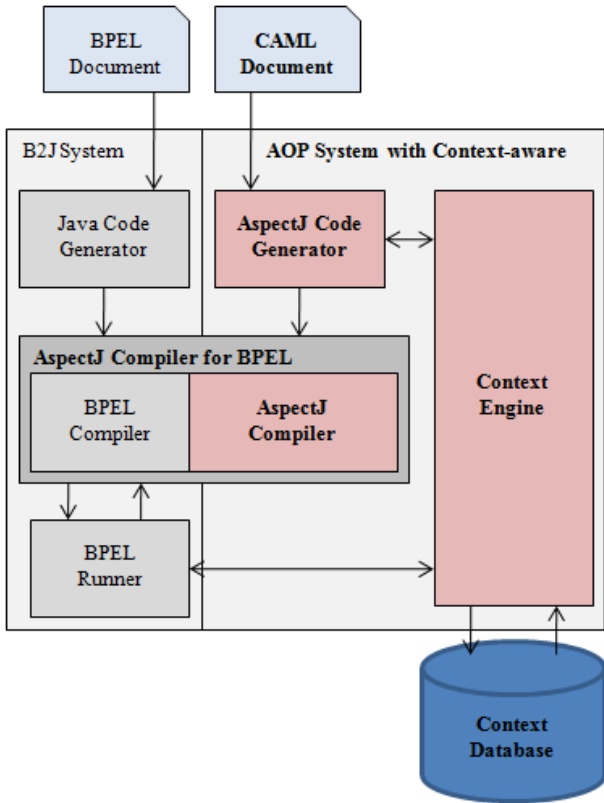


(그림 2) 상황인지 기능이 추가된 워크플로우

그림 2와 같이 제안한 시스템에서 생성된 워크플로우 프로그램은 기존의 BPEL 워크플로우를 유지한 채 추가되는 상황인지 기능을 적용시키는 형태를 보인다. BPEL 문서를 수정하지 않고 새롭게 추가된 상황인지 기능들을 처리하기 위하여, AOP 기법을 적용한 상황인지 정보를 기술하는 문서를 추가적으로 정의한다.

3-1. Context-Aware B2J System Architecture

시스템을 구축하기 위하여 BPEL 엔진은 Eclipse에서 제공하는 B2J [5]를, AOP 프로그램은 Eclipse에서 제공하는 자바 기반의 AspectJ 프로그램 [6]을 선택하였다. 새로운 문서의 처리를 포함하는 BPEL 시스템의 동작 구조는 그림 3과 같다.



(그림 3) 상황인지 BPEL 시스템 동작구조

우선 BPEL 문서 및 CAML 문서는 각각의 생성기 (Generator)를 통하여 워크플로우 자바코드 및 상황인지 코드를 포함한 AspectJ 코드로 변환된다. 그리고 생성된 자바 소스 및 AspectJ 문서는 AspectJ 컴파일러를 통하여 상황인지 정보가 직조된 워크플로우 프로그램을 생성하게 된다. 자바 컴파일러 모듈은 B2J에서 사용하던 컴파일러와 AspectJ 컴파일러를 결합하여 CAML 문서의 여부에 따라 상황인지 워크플로우 뿐만 아니라 기존의 워크플로우도 생성이 가능하다.

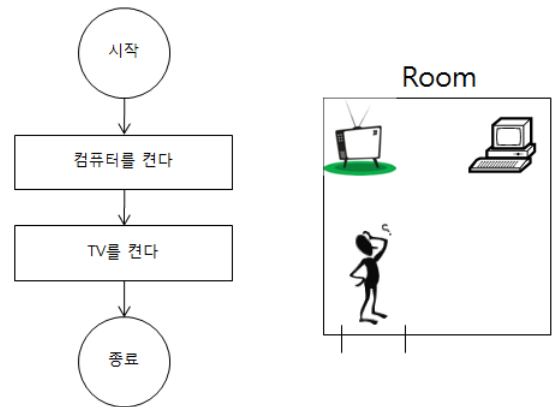
4. CAML (Context-aware Aspect-oriented Markup Language)

본 논문에서는 상황인지의 기술을 위하여 CAML라는 문서양식을 사용한다. CAML는 BPEL에 추가로 기술될 상황인지 정보를 기술하기 위해 새롭게 정의한 XML 형태의 문서 모델이다. CAML문서는 AsectJ Code Generator를 통하여 상황인지 코드가 매핑된 AspectJ 문서를 생성한다. 이후, B2J 엔진에서 자바 파일로 변환된 BPEL 문서와 함께 컴파일되어 상황인지 기능이 추가된 워크플로우를 생성한다. 그림 4는 상황인지 워크플로우를 위한 간단한 시나리오를 나타낸 것이다.

1. Room에 사람이 들어온다.
2. 들어온 사람이 John이면 컴퓨터를 켜다.
3. 들어온 사람이 Ann이면 TV를 켜다.

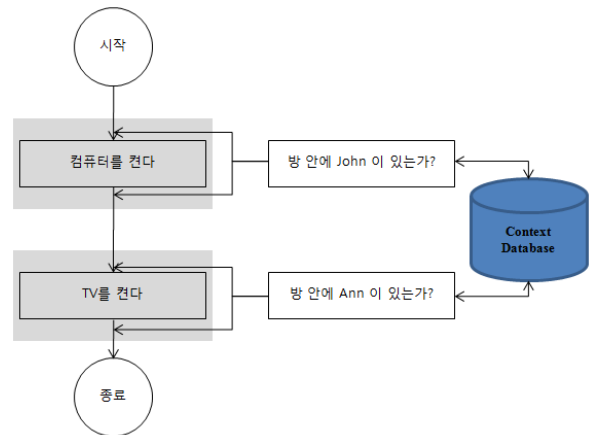
(그림 4) 워크플로우 시나리오

위 시나리오는 방에 들어온 사람을 인지해서 특정 사람에 맞는 동작을 수행하는 시나리오이다. 기존의 BPEL 워크플로우의 경우, 사람이 누군지 확인하여 서비스를 분기하는 상황인지 처리 기능을 수행할 수 없다. 기존 BPEL 시스템에서 상황인지를 제외한 워크플로우를 구성한다면 그림 5와 같이 나타낼 수 있을 것이다.



(그림 5) BPEL 워크플로우

사람을 구별할 수 없으므로 컴퓨터를 켜는 서비스와 TV를 켜는 서비스를 구분하여 실행하지 못하는 워크플로우 프로그램이 된다. 본 논문에서 제안한 시스템에서는 그림 6과 같이 기존의 BPEL 워크플로우를 유지한 채 목적하는 상황인지 워크플로우를 생성할 수 있다.



(그림 6) 상황인지 BPEL 워크플로우

목적하는 상황인지 워크플로우를 위해 그림 7과 같이 상황인지를 위한 CAML문서를 작성하였다.

```

1 <CAML>
2   <bpel
3     target="/process/invoke[@operation=W"onComW"]"
4     position="around">
5     <context target="/default/">
6       <subject>John</subject>
7       <verb>locatein</verb>
8       <object>Room</object>
9     </context>
10    <operation>enable</operation>
11  </bpel>
12 ...
13 </CAML>
    
```

(그림 7) CAML 문서 모델의 예

그림 7은 그림 6의 워크플로우에 대한 CAML 문서의 일부분을 나타낸 것이다. 2~4줄은 대상 BPEL 컴포넌트를 의미한다. 여기서 target은 BPEL 문서에서의 적용 위치를 XPath [7] 형태로 기술한 것이다. position은 직조하기 위한 Aspect 위치정보를 나타낸 항목이다. 5~9줄은 상황인지를 위한 부분으로, 상황인지 처리를 위하여 주어, 동사, 목적어 기술 방법으로 상황에 대한 표현을 나타내고 있다. 5줄에서의 target은 상황인지 엔진의 위치를 기술하는 것으로, “/default/”는 로컬에 존재하는 엔진을 지시하는 표현이다. 6~8줄에 표현되는 주어, 동사, 목적어 기술 방법은 사용하려는 대상 Context database에 포함된 항목들을 기준으로 작성하여야 한다. 10줄의 operation은 해당 상황인지에 대한 BPEL 컴포넌트의 처리방식을 기술하는 것으로 enable은 컴포넌트를 해당 상황에서만 사용하도록 하는 명령이다. 그림 7의 문서를 AspectJ Code Generator가 받아 그림 8과 같은 AspectJ 코드를 생성한다.

```

1 import bpel.context.EngineManager;
2 import org.eclipse.stp.b2j.core....Message;
3 import org.eclipse.stp.b2j.core....RunnerInterface;
4 aspect onComComponent {
5   pointcut point() : execution(* onCom(*));
6
7   Object around(): point() {
8     EngineManager contextEngine =
9     new EngineManager("/default/");
10
11     if(contextEngine.checkContext(
12       "locatein", "John", "Room")) {
13       return proceed();
14     }
15     return null;
16   }
17 }
18 }
...
    
```

(그림 8) 상황인지를 위한 AspectJ 코드

자바 코드를 대상으로 프로토타입 시스템을 작성하였기 때문에 자바 코드를 위한 AspectJ 문서를 생성한다. 1

줄은 상황인지 엔진의 접근 및 상황인지를 위한 매니저 클래스를 나타낸다. 6줄에서 onCom() 컴포넌트가 수행되는 시점을 pointcut으로 정의한 뒤, 해당 메소드의 수행부를 6~16줄의 around()에서 Intercept하고 대체 수행하게 된다. onCom() 메소드는 12줄의 상황인지 결과가 참인 경우에 14줄의 proceed() 메소드로써 수행된다.

자바 파일로 변환된 BPEL 문서와 그림 8과 같이 생성된 AspectJ문서를 AspectJ 컴파일러를 통해 컴파일하는 방식으로 그림 6과 같은 상황인지 워크플로우 프로그램이 생성되게 된다.

5. 결론

BPEL을 사용하려는 응용의 범위가 다양해짐에 따라서 BPEL 워크플로우에서의 요구사항이 증가하고 있다. 그러한 요구사항을 충족시키기 위하여 BPEL 문법에 새로운 기능을 추가하는 Context4BPEL 같은 연구와, BPEL 문서는 유지한 채 원하는 기능을 추가하는 AO4BPEL 등과 같은 연구가 진행되었다. 하지만 Context4BPEL은 기존의 BPEL 엔진 및 문서를 재사용할 수 없다는 한계를 지니고, AO4BPEL은 상황인지 기능을 추가하기 어렵다는 한계를 지니고 있었다.

이러한 한계를 극복하고 각 시스템의 장점을 취합하기 위하여 본 논문에서는 AOP 기술을 적용한 BPEL 기반의 상황인지 워크플로우 시스템을 제안하였다. 본 시스템은 기존의 BPEL 문서의 재사용이 가능하고, BPEL 워크플로우에서 상황인지 기능을 사용할 수 있다는 특징을 지니고 있다.

참고문헌

- [1] BPEL, <http://www-128.ibm.com/developerworks/library/specification/ws-bpel/>.
- [2] Gregor Kiczales, “Aspect-oriented programming”, ECOOP’97.
- [3] Matthias Wieland, Oliver Kopp, Daniela Nicklas, Frank Leymann, “Towards Context-aware Work-flows”, Citeseer.
- [4] Anis Charfi and Mira Mezini, “Aspect-oriented web service composition with AO4BPEL“, Lecture Notes in Computer Science, 2004, Volume 3250/2004, 168-182
- [5] eclipse B2J, <http://www.eclipse.org/stp/b2j/>.
- [6] eclipse AspectJ, <http://www.eclipse.org/aspectj/>.
- [7] XPath, <http://www.w3.org/TR/xpath/>.