

유비쿼터스 환경에서 동적 바인딩을 지원하는 태스크 기반 프레임워크

황윤영 이규철
충남대학교 컴퓨터공학과
{yyhwang, klee}@cnu.ac.kr

The framework for dynamic binding in ubiquitous environment

Yun-Young Hwang Kyu-Chul Lee
Dept. Computer Engineering in Chungnam National University
{yyhwang, klee}@cnu.ac.kr

요 약

유비쿼터스 환경은 디바이스의 상태 및 위치가 자주 변하는 특성을 가진다. 이러한 특성은 사용자가 사용하고 있는 서비스의 오퍼레이션 상태에도 영향을 주게 된다. 즉, 사용하고 있던 서비스 오퍼레이션이 갑자기 물리적 위치가 변경되어 재연결을 요구한다거나, 사용할 수가 없어 다른 오퍼레이션으로 대체해야 하는 경우가 발생할 수 있다. 이 때, 사용자에게 사용하던 오퍼레이션과 같은 기능을 하는 서비스를 발견해 줄 수 있어야 하며, 끊임 없는 서비스를 제공하기 위해 동적 바인딩이 제공되어야 한다.

1. 서 론

유비쿼터스 환경이란 사용자에게 언제, 어디서든, 어떤 서비스든지 끊임 없이 제공해주는 것을 의미한다. 그러나 유비쿼터스 환경에 산재하는 많은 디바이스들은 빈번하게 위치 이동이 발생한다. 이는 사용자가 사용하는 서비스 오퍼레이션의 상태 변화에 영향을 준다. 즉, 사용하고 있던 서비스 오퍼레이션이 갑자기 물리적 위치가 변경되어 재연결을 요구한다거나, 사용할 수가 없어 다른 오퍼레이션으로 대체해야 하는 경우가 발생할 수 있다. 이해를 돕기 위해 이 논문에서는 다음과 같은 시나리오를 설정하고, 이를 해결하는 방향으로 논문을 전개해 나간다.

8 월의 어느 평일 아침, A 의 비서인 K 는 오후에 이사진 긴급 회의를 소집하라는 지시를 받게 된다. K 는 회의를 위해, 오후 시간에 비어 있는 회의 장소를 찾아 예약하고, 이사진들에게 회의 소집을 알린다. 또한, K 는 회의 참석 인원수에 맞추어 회의 자료를 준비하고, 회의 후에 있을 저녁식사 장소를 예약해야 한다.

회의 자료를 출력하던 중 사용하고 있던 프린터가 오류를 발생시켜 더 이상 자료 출력이 불가능하게 되었다. 따라서, K 는 다른 프린터를 찾아 회의 자료 출력을 해야 한다.

위 시나리오에서 K는 사용하고 있던 프린터를 대체해야 하는 상황에 놓여있다. 이렇게 사용하고 있던 디바이스나 서비스 오퍼레이션을 대체해야 하는 상황에서 사용자가 자신이 필요로 하는 서비스 오퍼레이션을 처음부터 다시 발견하고 선별해야 하는

한다면 사용자는 많은 불편함을 갖을 것이다. 따라서 사용자가 사용하던 오퍼레이션을 대체할 수 있는 오퍼레이션을 시스템에서 발견하여 사용자에게 제안함으로써 사용자의 편의성을 도모하는 방법이 필요하다. 또한, 사용하던 오퍼레이션과 동일한 기능을 수행하는 오퍼레이션을 동적으로 바인딩 시켜주는 기능 역시 필요하다. 본인은 [1]에서 사용자의 서비스 조합 요구사항을 태스크로 표현하고, 태스크를 수행할 수 있는 오퍼레이션을 검색하여 사용자에게 제공하는 시스템을 제안하였다. 그러나 [1]은 오퍼레이션의 플로우를 지원하지 못한다는 문제점을 가졌다. 따라서, 이 논문은 [1]의 문제점을 해결하면서 동적 바인딩까지 지원하는 시스템을 제안한다.

이 논문의 구성은 다음과 같다. 2장은 동적 바인딩과 관계된 WS-BPEL에 대해 살펴보고, 3장에서는 이 논문에서 제안하는 유비쿼터스 환경에서의 동적 바인딩을 소개한다. 4장은 기존 연구와의 비교를 통한 결론과 향후연구에 대해 기술한다.

2. WS-BPEL

WS-BPEL(Web Services Business Process Execution Language)은 BPEL 2.0 (Business Process Execution Language)로써, BPEL 1.1 의 다음 버전을 뜻한다. WS-BPEL 은 웹 서비스를 기반으로 비즈니스 프로세스의 정의와 실행을 위한 언어로 SOAP, WSDL 그리고 UDDI 를 포함하는 웹 서비스 표준을 지원하는 XML 기반의 언어이다[2]. WS-BPEL 을 이용하여 프로세스를 정의하고 기존의 웹 서비스와 연결하여 프로세스를 실행 할 수 있도록 한다. 또한, 각

본 연구는 중소기업청의 산학 공동기술개발지원사업의 연구결과로 수행되었음

프로세스의 변수도 선언 할 수 있도록 하고 있다. 이를 위해 WS-BPEL은 다음과 같은 기능을 제공한다.

● 프로세스(process) 정의

<process> 엘리먼트는 WS-BPEL의 루트 엘리먼트이며, 다양한 하위 엘리먼트를 포함한다. 주요 엘리먼트를 보면, WS-BPEL로 정의되는 비즈니스 프로세스에서 사용하는 웹 서비스를 기술하는 <partnerLinks>, 비즈니스 프로세스에서 사용하는 변수를 정의하는 <variables>, 그리고 비즈니스 프로세스에서 실행되기 위한 액티비티를 정의하는 <activity> 부분으로 구성된다.

● <partnerLinks> 부분

<partnerLinks> 엘리먼트는 WS-BPEL 프로세스를 기존의 웹 서비스와 연결하는 부분을 정의하며, 프로세스에서 사용할 웹 서비스 오퍼레이션(operation)의 집합을 기술한다. 기술된 오퍼레이션들은 WSDL의 <portType> 엘리먼트로 정의되어 있는 웹 서비스 오퍼레이션에 기인한다.

● <variables> 부분

WS-BPEL 프로세스에서 변수는 메시지를 저장하거나 변경하기 위해 사용된다. 교환되는 모든 메시지에 대한 변수와 타입은 정의되어야 한다.

● <activity> 부분

WS-BPEL 프로세스 정의는 다수의 액티비티로 정의되며, 액티비티는 기본 액티비티(basic activity)와 구조화 액티비티(structured activity)의 두 가지 종류가 있다. 이렇게 프로세스를 기반으로 웹 서비스를 플로우에 따라 수행할 수 있도록 가능하게 하는 것이 WS-BPEL이다. WS-BPEL은 기본적으로 프로세스 설계 과정에서 파트너 링크를 정적(static)으로 정의하는 것이 일반적이다. 각 파트너 링크는 개발자가 설계 시점에 지정한 단일 외부 프로세스를 참조하도록 한다. 하지만 대규모 시스템에서는 비즈니스 프로세스가 보다 복잡한 구성을 갖게 되는데, 다수의 외부 서비스와 다수의 파트너 링크가 서로 연동하기도 하고, 파트너 링크들 중 일부는 디자인 타임에 정의되지 않을 수도 있다. 이와 같은 경우 파트너 링크의 선택에 관련한

로직은 비즈니스 프로세스 내부에 구현되어야 하며, 이로 인해 프로세스가 불필요하게 복잡해질 수 있다. 또한 파트너 링크가 계속적으로 추가되면서 프로세스의 규모가 감당할 수 없는 수준으로 커질 수 있으며, 파트너 링크에 대해 변경 작업을 수행할 때마다 전체 비즈니스 프로세스를 수정해야 하는 문제가 발생한다. 이러한 문제점을 해결하는데 필요한 것이 동적 바인딩(Dynamic Binding)이다. 동적 바인딩은 개발자가 설정 파일을 통해, 또는 런타임 입력을 통해 새로운 서비스를 추가하거나 대체할 수 있도록 하는 것을 뜻한다. [3]에서는 WS-BPEL에서 동적 바인딩을 지원할 수 있는 방법을 제안하고 있다. WS-Addressing 표준의 EndpointReference를 이용하여 해결하는 방안이다. EndpointReference는 WSDL 내에서 사용 가능한 서비스 중 하나를 선택하거나, 런타임에 새로운 서비스를 정의하는 것을 가능하게 한다. 비즈니스 프로세스가 portType에 정의된 인터페이스 정보에 정적으로 의존하는 반면, EndpointReference는(서비스에 대한 바인딩의 매핑을 통해) 서비스 위치를 동적으로 재정의할 수 있게 한다. 쉽게 말해, EndpointReference는 WSDL의 정적인 서비스 요소를 대체하는 동적 참조로 활용된다.

3. 유비쿼터스 환경에서의 동적 바인딩 지원

3.1 태스크 기반의 서비스 조합 방법

[1] 태스크 기반의 서비스 조합 방법을 제공함으로써, 사용자의 요구사항을 보다 정확하게 표현하기 위한 것이다. 1장에서 소개된 시나리오를 도시화 하면 [그림 1]과 같이 표현 될 수 있다.

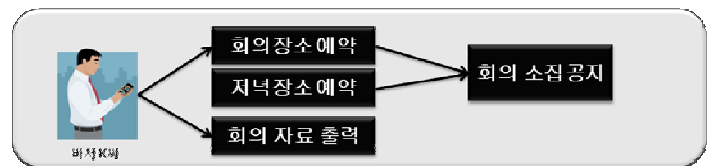


그림 1 시나리오

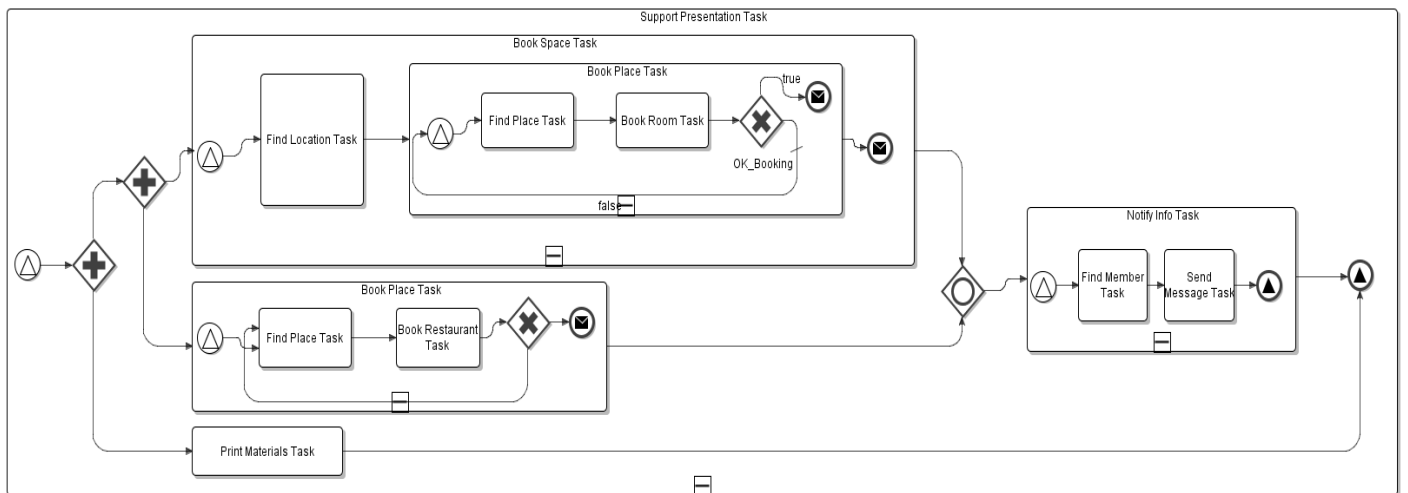


그림 2 BPMN을 이용한 표현

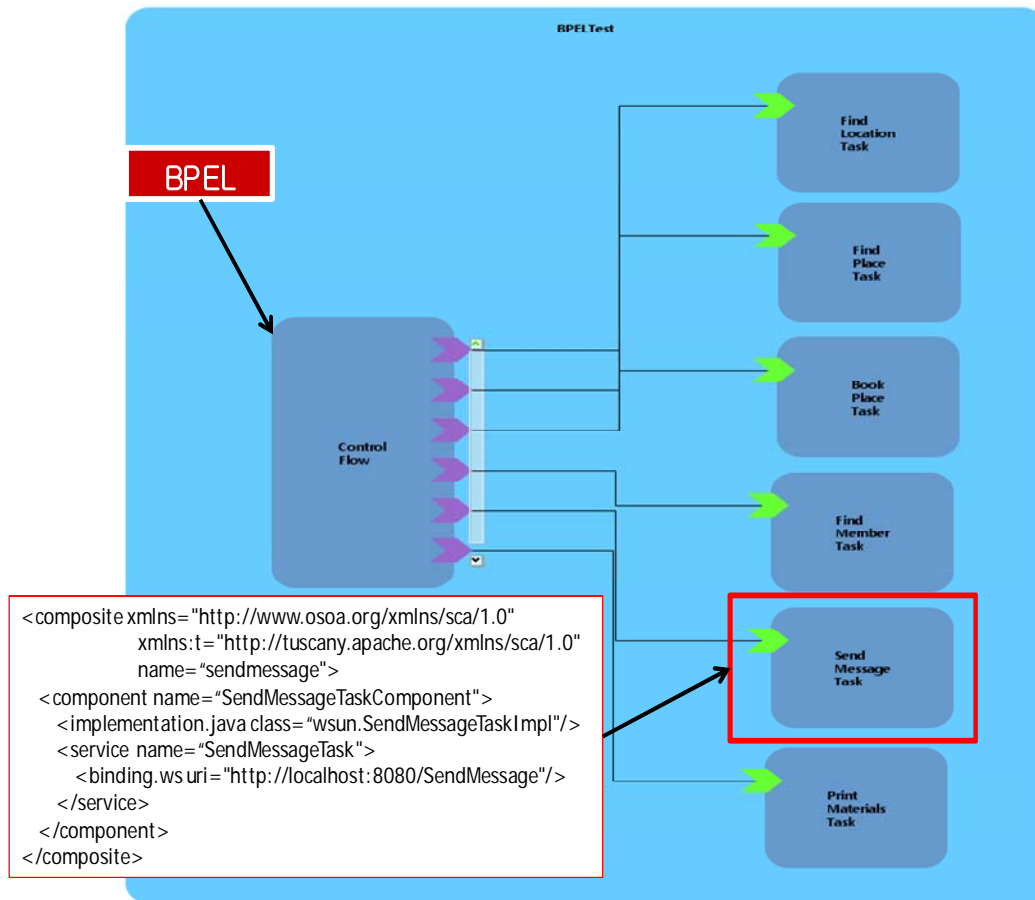


그림 3 SCA로 구현한 태스크 및 프로세스 플로우

이러한 시나리오는 BPMN(Business Process Management Notation)을 이용하여 [1]에서는 [그림 2]와 같이 표현하였다. 이와 같은 플로우에서 순차적 흐름만 지원하고, 병렬구조나 if-then-else, 반복구조 같은 것들은 추후 연구로 남겨 놓았었다. 따라서, 이 논문에서는 [1]에서 해결할 수 없었던 태스크 플로우를 해결하고, SCA로 구현하면서 복잡성을 갖게 되었던 동적 태스크 대체, 즉 동적 바인딩일 지원하는 방법을 소개한다.

3.2 동적 바인딩 지원 방법

동적 바인딩 및 태스크 플로우를 지원하기 위해, BPMN 으로 표현된 [그림 2]를 WS-BPEL 과 SCA(Service Component Architecture)를 이용하여 [그림 3]과 같이 표현하였다. 즉, WS-BPEL 을 SCA 컴포넌트로 작성하고, 그것과 연결된 다른 SCA 컴포넌트들은 태스크를 의미한다. 즉, [그림 3]의 Control Flow 컴포넌트는 WS-BPEL 로 구현되었으며, 그 컴포넌트와 연결된 Find Location Task, ..., Print Materials Task 컴포넌트들은 태스크를 의미한다. [그림 4]는 WS-BPEL 로 표현된 태스크 간의 플로우 중 병렬구조를 나타내고 있다.

```

<bpel:flow bpmn:label="Parallel_Gateway"
  name="Parallel_Gateway" bpmn:id="_Z7VnJg8MEd-TpvQP3PuuAg">
  <bpel:sequence>
    <bpel:empty bpmn:label="Print Materials Task"
      name="Print_Materials_Task" bpmn:id="_Z7VnJQ8MEd-TpvQP3PuuAg"/>
  </bpel:sequence>
  <bpel:flow bpmn:label="Parallel_Gateway"
    name="Parallel_Gateway-1" bpmn:id="_Z7VnJg8MEd-TpvQP3PuuAg">
    <bpel:sequence>
      <bpel:scope bpmn:label="Book Space Task"
        name="Book_Space_Task" bpmn:id="_Z7VnJg8MEd-TpvQP3PuuAg"></bpel:scope>
    </bpel:sequence>
    <bpel:sequence>
      <bpel:scope bpmn:label="Book Place Task"
        name="Book_Place_Task-1" bpmn:id="_Z7VnJw8MEd-TpvQP3PuuAg"></bpel:scope>
    </bpel:sequence>
  </bpel:flow>
</bpel:flow>
    
```

그림 4 WS-BPEL 표현

[1]에서는 태스크 삽입 삭제를 위해서 태스크 모두를 조합 컴포넌트로 작성해서 대체하는 방법을 이용하였다. 그러나, WS-BPEL을 이용하여 태스크 흐름을 조정하는 방법은 태스크를 단일 컴포넌트로 작성하고 태스크 컴포넌트의 수정 없이 단순히 WS-BPEL 컴포넌트만 수정

하는 방법을 사용할 수 있다. [1]에서 언급된 바와 같이, SCA에서는 런타임 시에 컴

포넌트 수정을 지원하지 않음으로 WS-BPEL 컴포넌트를 직접 수정하지 않고 다른 컴포넌트로 대체하는 방법을 활용한다.

동적 바인딩을 지원을 위해 각 태스크 컴포넌트는 특정 오퍼레이션을 지정하고 있는 것이 아니라, 오퍼레이션 그룹을 가르키도록 한다. [그림 3]에서 자바로 구현된 “SendMessageTaskImpl”에서는 메시지 전송 오퍼레이션 그룹에 해당하는 모든 오퍼레이션을 검색하게 된다. 즉, 사용하고 있던 오퍼레이션이 다른 오퍼레이션으로 대체될 필요성이 발생하면, 같은 오퍼레이션 그룹의 오퍼레이션으로만 대체되어 사용될 수 있도록 지원한다.

4. 결론 및 향후연구

이 논문은 WS-BPEL 및 SCA를 이용하여 사용자의 오퍼레이션 조합 요구사항을 해결하고, 런타임시에 태스크를 수정하고 다른 오퍼레이션으로의 대체를 지원하는 방법을 제안했다.

아래 표는 [1]의 연구 결과를 향상시키기 위해 WS-BPEL을 사용함으로써 얻을 수 있는 장점을 정리한 것이다. BPMN과 WS-BPEL만을 사용하면, 사용자 태스크

를 수정하는 것이 어렵다. 그러나 SCA를 접목하면 태스크 수정이 가능하며, 태스크에서는 중계 웹 서비스를 이용한 오퍼레이션 검색으로 인해 동적 바인딩 지원도 가능하다.

현재, 이를 지원하기 위한 시스템은 구축 완료 단계에 있으며, 클라이언트 개발을 완료하는 시점에서 함께 선보일 예정에 있다.

5. 참고문헌

- [1] 황윤영, 이규철, “동적 서비스 조합을 위한 서비스 컴포넌트 아키텍처의 확장”, 한국전자거래학회, 지식정보산업연합학회 창립기념 학술대회, 369-377, 2008.12.10
- [2] A. Alves, et al. “Web Service Business Process Execution Language 2.0”, <http://www.oasis-open.org>.
- [3] S. Carey. “다이내믹 BPEL 프로세스의 구현”, http://www.oracle.com/technology/global/kr/pub/articles/bpel_cookbook/carey.html

	BPMN+BPEL+SCA	BPMN+BPEL
태스크 표현	Client: BPMN Server: SCA	Client: BPMN Server: BPEL
태스크 Control Flow	Client: BPMN Server: BPEL	Client: BPMN Server: BPEL
검색 단위	Operation	Web Services
실행 단위	Operation	Operation
BPEL 바인딩	SCA Component	Web Services
BPEL 실행	태스크	Operation
태스크 바인딩	중계 WS(사용자가 원하는 오퍼레이션 실행을 위함)	Web Services