

# Java 기반의 Web Service를 이용한 개인 맞춤형 서비스

남호영\*, 이현림\*, 양정진\*

\*가톨릭대학교 컴퓨터정보공학부

## Customize Service using Java Web Service

Nam, Ho Young, Lee, Hyun Lim, Yang, Jung Jin

Catholic University

E-mail : furystyle@naver.com, sor\_love\_nm@hanmail.net, jung.jin@catholic.ac.kr

### 요약

웹이 급속히 발전하면서 방대해진 정보와 자원을 플랫폼과 벤더, 프로토콜에 상관없이 통합하려는 목적으로 웹 서비스가 등장하였다. 웹 서비스가 발전함에 따라 웹 서비스들 간의 통합은 필수적인 요소로 자리 잡았고 이를 지원하는 언어로써 BPEL을 비롯한 많은 언어들이 등장하였다.

현재 개인 맞춤형 서비스는 주로 위젯 형태나 포털 사이트에서 서비스 되고 있다. 하지만 이러한 형태는 서비스 제공자 입장에서 제한된 정보를 가지고 복잡도가 높은 처리과정을 통해 정보를 선별하여 사용자에게 제공하게 된다. 또한 서비스의 추가나 변경, 삭제에 있어서도 유연성과 확장성이 떨어지게 된다.

본 논문에서는 이러한 한계점을 극복하기 위해 개인 맞춤형 서비스를 웹 서비스를 사용하여 구현하였다. 새로운 서비스의 추가나 서비스의 변경에 있어서 웹 서비스는 유연한 특징을 가지고 있고 BPEL을 활용하여 서비스들 간의 조합과 Wrapping이 가능하게 하여 좀 더 구조화 된 서비스를 제공해 줄 것이다.

### 1. 서론

인터넷은 웹의 등장과 함께 통신기술의 발전으로 인해 폭발적인 발전을 이루었다. 하지만 그 발전은 기술적인 측면보다는 양적인 측면이 강했고 웹 사용자들의 증가하는 요구를 충족시키지 못했다. 이러한 요구를 만족하기 위해서는 기존의 웹보다 더욱 효율적인 웹이 요구되었고 웹 2.0이라 불리는 차세대 웹이 등장했다.

이와 함께 기업에서는 분산 컴퓨팅 환경 측면에서 SOA(Service Oriented Architecture)라는 패러다임이 등장하고 발전하였지만 벤더간의 통합이나 사용 환경의 통합이 되지 않아서 실제적으로 구현 되는 것이 힘들었다. 그러나 웹의 발전에 힘입어서

웹이라는 통합된 환경 위에 구축되는 웹 서비스를 이용하여 SOA가 구현 가능해졌다. 웹 서비스는 XML 표준에 기반을 두고 모든 프로그래밍 언어, 프로토콜 또는 플랫폼을 사용하는 느슨하게 결합된(loosely coupled) 애플리케이션 구성요소로서 개발될 수 있는 특징을 가졌다. 이러한 특징 때문에 현재 많은 발전을 이루었으며 널리 사용되어지고 있다.[1]

웹 서비스와 관련된 기술 중 웹 서비스들 간의 통합을 위한 표준으로 WS-BPEL(Business Process Execution Language for Web Services)이 채택되었다. 이는 SOA 환경을 잘 수용하고 있으며 비즈니스 프로세스를 기술하고 구현한다.[2] WS-BPEL를 이용한 웹 서비스 통합이 이루어지

면서 비즈니스 프로세스 단위의 구성과 흐름을 보다 쉽고 실제 사용가능하게 정의할 수 있게 되었다. 또한 BPEL의 등장으로 비즈니스 프로세스의 정의, 최적화, 리엔지니어링이 가능하게 되었으며 고차원적인 웹 서비스가 가능하게 됐다.

웹이 발전을 거듭하면서 개인 사용자는 방대한 자료에서 자신에게 필요한 정보를 얻기 위해 많은 노력을 해야만 했다. 이에 따라 개인의 특성과 요구 사항에 맞는 정보를 선별하여 제공하는 서비스들이 등장하기 시작했다.

현재 개인 맞춤형 서비스는 주로 워젯 형태나 포털 사이트에서 서비스 되고 있다. 하지만 이러한 형태는 서비스 제공자 입장에서 제한된 정보를 가지고 복잡도가 높은 처리과정을 통해 정보를 선별하여 사용자에게 제공하게 된다. 또한 서비스의 추가나 변경, 삭제에 있어서도 유연성과 확장성이 떨어지게 된다.

본 논문에서는 이러한 한계점을 극복하기 위해 개인 맞춤형 서비스를 웹 서비스를 사용하여 구현하였다. 새로운 서비스의 추가나 서비스의 변경에 있어서 웹 서비스는 유연한 특징을 가지고 있으며 BPEL을 활용하여 서비스들 간의 조합과 Wrapping이 가능하게 되면 좀 더 구조화 된 서비스를 제공할 수 있다.

본 논문에서는 먼저, 오픈소스 개발 진영인 apache의 프로젝트와 Java를 이용하여 웹 서비스 모델을 제시 한다. 이 모델은 기술적으로 표준사항과 최신 스펙을 지원하고 있으며 비용적인 측면에서도 유리하다. 우리의 이러한 모델은 개발자 측면에서의 편의성을 제공하므로 개발자들에게 효율적인 개발을 도와주는 역할을 할 것이다. 현재 많은 모델들이 제시되고 있는 상황에서 프로젝트의 성격에 맞춰 모델의 특성을 파악하여 이를 이용하는 것이 중요하다.

본 논문의 2장에서는 본 연구에 필요한 관련 연구로써 웹 서비스와 BPEL과 기존의 개인 맞춤형 서비스의 문제점에 대해 논하고 있으며 본격적인 자바 기반 웹 서비스의 구조와 개인 맞춤형 서비스의 모델에 대해 이야기하고 있다. 3장에서는 결론 및 향후 연구를 소개하였다.

## 2. 본론

### 2.1. 관련연구

웹 서비스 모델의 기반이 되고 있는 웹 서비스의 개념과 웹 서비스 통합을 기술하고 있는 BPEL을 설명하고 있으며 기존 개인 맞춤형 서비스의 문제점을 간단히 살펴본다.

#### 2.1.1 웹 서비스

웹 서비스(Web Service)는 특정 소프트웨어 패키지에 종속적이지 않은 개방형 구조를 가지고 있다. 또한 인터넷 공간을 기본 채널로 이용하며 표준 기술을 바탕으로 한 인터페이스를 갖는 모듈화된 일련의 소프트웨어 구성요소들이라 할 수 있다. 웹 서비스는 내부 또는 외부의 사용자들에게 공개되며 검색이 가능하고 다른 웹 서비스들을 이용하여 새로운 서비스를 구성하는 것이 가능하다.

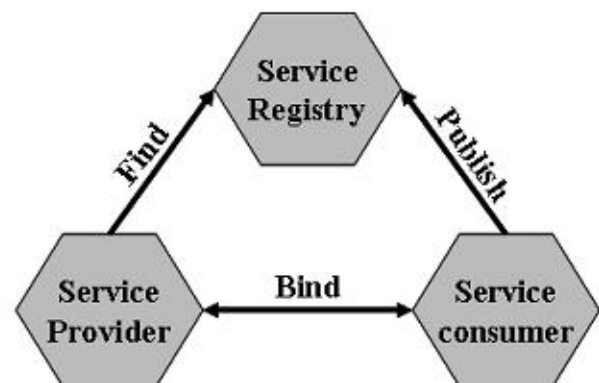


그림 1. 웹 서비스 컴포넌트

위의 그림 1과 같이 웹 서비스 아키텍처는 Service Registry(UDDI)에 사용 가능한 웹 서비스를 등록하고 원하는 웹 서비스를 찾기 위해 Service Registry를 검색한다. 웹 서비스 사용자는 원하는 웹 서비스에 접근 및 호출을 위해 웹 서비스 제공자와의 협의 및 연결을 위한 기본적인 오퍼레이션을 지원한다. 세부적으로 각각의 웹 서비스 컴포넌트에 대해 설명하자면 웹 서비스 제공자(Web Service Provider)는 웹 서비스를 운영하며 Service Registry에 사용 가능한 웹 서비스를 등록

한다. 웹 서비스 중개자(Web Service Broker)는 제공자와 소비자 사이에서 제공자가 서비스를 출판하고 이를 소비자가 검색할 수 있도록 편의를 제공한다. 또한 웹 서비스 사용자(Web Service Consumer)는 Service Registry를 통해 원하는 웹 서비스의 위치를 발견하게 되고 발견한 위치를 사용하여 서비스를 사용하게 된다.

웹 서비스에서 사용하는 대표적인 기술들로는 XML(Extensible Markup Language), SOAP(Simple Object Access Protocol), WSDL(Web Service Description Language), UDDI(Universal Description, Discovery and Integration)가 있다.[3]

### 2.1.2 BPEL

Business Process Execution Language for Web Services(BPEL 또는 BPEL4WS)는 웹 서비스 환경에서 비즈니스 프로세스를 정의하고 실행하기 위한 표준 언어이다. BPEL은 이전의 두 가지 워크플로우 언어, 즉 Web Services Flow Language(WSFL)와 XLANG의 성과를 물려받은 차세대 언어이다. BPEL은 웹 서비스의 composition, orchestration, coordination을 통해 탐-다른 방식으로 SOA(Service Oriented Architecture)를 구현한다. BPEL을 이용하면 웹 서비스에서 "비즈니스 프로세스"를 구현하기 위한 컴포지트 서비스(composite service)를 쉽게, 그리고 직관적으로 구현할 수 있다.[4]

BPEL은 SOAP, WSDL, UDDI, WS-Reliable Messaging, WS-Addressing, WS-Coordination, WS-Transaction 등을 포함하는 웹 서비스 테크놀로지 스택을 지원하기 위해 XML을 기반 언어로 사용한다.

BPEL은 두 가지 접근방식을 조합하였다. BPEL은 두 가지 프로세스 모델링을 지원하는데 하나는 추상적 프로세스 모델링으로 실제 행위 보다는 집단 간의 메시지 교환 행동에 대한 비즈니스 프로토콜을 모델링하는 것이며 두 번째는 수행 가능한 프로세스 모델링으로 프로세스를 구성하는 액티비티 사이의 수행 흐름을 모델링하는 것이다.[2]

BPEL은 비즈니스 파트너 간의 통합을 효과적으

로 지원 할 수 있다. BPEL은 비즈니스 프로세스의 정의를 통해 비즈니스 프로세스의 최적화, 리엔지니어링을 가능하게 하고, 가장 적합한 프로세스를 선택함으로써 기업 환경을 최적화할 수 있도록 한다. BPEL에 의해 기술된 비즈니스 프로세스 정의는 기존 시스템에 영향을 미치지 않으며, 따라서 업그레이드가 용이하다. 웹 서비스의 확산과 더불어 BPEL의 중요성 역시 계속적으로 증가하는 추세를 보일 것이다.[4]

아래 그림은 BPEL을 이용한 웹 서비스 통합 구조를 간단히 나타낸 것이다. 이때 파트너와의 협력은 WSDL에 의존하여 이루어진다.

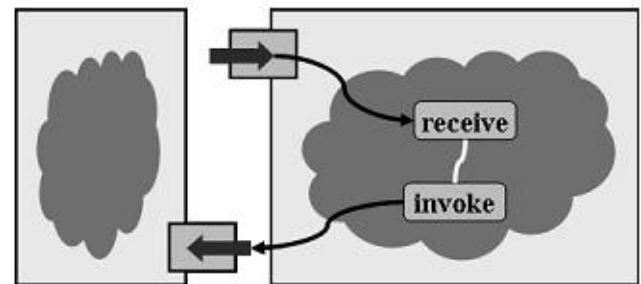


그림 2 간단한 BPEL 프로세스

### 2.1.3 기존 개인 맞춤형 서비스의 문제점

개인 맞춤형 서비스는 여러 형태로 제공되고 있다. 데스크 탑에서 사용되는 위젯 형태나 웹상의 포털사이트에서 보여주는 서비스를 그 예라고 할 수 있다.

제공해주는 서비스는 사용자가 미리 선택 입력하거나 개인의 특성이나 패턴을 학습하여 서비스를 제공해주는 방법이 있을 수 있다. 개인 맞춤형 서비스의 등장 배경이 적은 노력과 반복 작업 없이 개인에 특화 된 서비스를 구현하기 위한 것인 만큼 자동화 되고 지능화 된 서비스 제공방법이 필요하다.

기존 개인 맞춤형 서비스에서는 특정 환경에 제한을 받는 문제점이 있다. 위젯 서비스를 예를 들면 필요한 클라이언트 프로그램을 설치한 후 자신이 원하는 서비스를 이용하게 되는데 이때 모든 관련 정보와 설정을 하게 된다. 하지만 이런 정보는 하나의 데스크 탑에 종속적이게 되고 환경이 바뀌면 동일한 서비스를 이용하지 못하게 된다.

사용자에 관한 정보를 서버에 저장하는 방법을 사용하더라도 개인마다 처리과정이 다른 서비스를 요구하는 경우나 이 기종간의 정보 교환이 필요한 경우, 서비스의 변경, 추가가 빈번할 경우와 같이 복합적이고 단계적으로 일어나는 프로세스를 처리가 요구 될 때에는 기본적인 개인 맞춤형 서비스 모델은 적용하기가 힘들다.

유비쿼터스 환경을 가정하였을 때 사용자 측면에서는 사용자가 자신의 데스크탑에서 입력한 일정이 자신의 휴대전화와 직장의 데스크탑까지 영향을 주어 변경되고 적합하게 처리된 결과를 서비스 받아 보는 것이 편리하다.

본 논문에서는 이러한 문제점을 해결하기 위해서 미들웨어의 성격을 가지는 웹서비스를 플랫폼으로 하여 개인 맞춤형 서비스를 구현하는 모델을 제시한다.

## 2.2 Java 기반 웹 서비스

먼저 본 논문에서 제시하고 있는 모델의 구현한 환경은 Web Server와 Servlet Container의 역할로 Apache Tomcat 6.0[5]을 사용하였다. 또한 Web Service Container는 Apache AXIS 2.0[6]로 구현하였다. AXIS는 Apache SOAP project가 발전한 것으로 SOAP 스펙과 JWS DP의 API를 지원하고 있으며 이를 이용하여 쉽게 웹 서비스를 개발하고 배치할 수 있게 해 준다. 또한 클라이언트 측에서도 AXIS의 API를 이용하면 쉽게 SOAP 메시지 처리를 하는 클라이언트 프로그램을 만들 수 있다. BPEL Engine은 Active Endpoint의 ActiveBPEL 4.0[7]로 구현하였다. Active Endpoint는 BPEL Engine 뿐 아니라 비주얼 적으로 도식화하여 BPEL를 생성시킬 수 있는 틀을 제공한다.

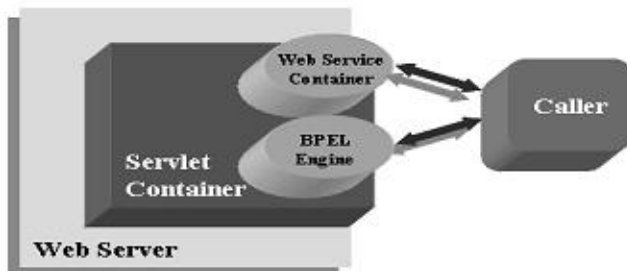


그림 3 자바 기반 웹 서비스 모델

위 그림 3에서 Web Server와 Servlet Container는 자바 기반 웹 서비스를 개발할 때 필요한 웹 어플리케이션 서버역할을 한다. Web Service Container는 자바 기반 웹 서비스 개발을 위한 라이브러리를 제공하고 배포된 웹 서비스들을 관리한다. 웹 서비스 Container는 Servlet Container에 배포된 하나의 Servlet 형태로 동작한다. 또한 개발을 위한 각종 도구와 웹 서비스 배포 및 모니터링, 테스트를 위한 도구를 지원한다.

따라서 웹 서비스를 JWSDP(java web service developer pack)을 사용하여 하위 수준에서 구현하는 것 보다 비교적 쉽게 구현할 수 있다. 이는 개발자에게 편리성을 제공해준다. 웹 서비스 Container는 가장 작은 단위의 웹 서비스, 즉 아토믹 웹 서비스를 제공하며 이 서비스들은 BPEL Engine에서 호출되어 하나의 큰 서비스 구성 요소가 될 수 있다.

BPEL Engine은 BPEL로 기술된 비즈니스 프로세스를 배포하고 이를 실행한다. BPEL Engine 역시 Servlet Container에 배포된 하나의 Servlet 형태로 동작한다. 비즈니스 프로세스는 작업의 흐름에 따라 각각의 서비스들을 호출하게 된다. 이러한 전체적인 프로세스의 흐름을 BPEL Engine을 통해 개발자가 한 눈에 확인할 수 있으며 수정 가능하다.

본 논문에서 제시한 Web Service 모델에서 Caller의 서비스 호출 경로는 두 가지로 볼 수 있다. 첫째는 Web Service Container에 직접 접근하여 호출하는 경로이며 두 번째는 BPEL Engine를 통한 호출로 비즈니스 프로세스를 실행하는 것이다.

첫 번째 경로는 Web Service Container에 배치되어 있는 atomic 웹 서비스를 공개하였다는 가정 하에 Caller가 직접 웹 서비스를 호출할 수 있다. 이러한 경우에는 가장 단순한 Web Service의 모습을 보이며 제공자 측에서 작성한 서비스의 프로세스만이 실행이 된다. 하지만 SOA 패러다임 안에서 서비스의 재사용성이 강조되는 점을 볼 때 복잡한 프로세스를 하나의 서비스에 작성하는 것은 큰 의미가 없다.

두 번째 경로는 BPEL로 기술된 웹 서비스를 호출하였을 경우 BPEL Engine은 이를 해석하여 구



성된 서비스들을 실행해준다. 프로세스 진행 중에 호출되는 서비스는 Local에 있는 Web Service Container에서 서비스되고 있는 서비스일 수도 있고 원격에 있는 서비스들 일 수도 있다. 각 서비스들은 BPEL 안에 기술되어 있는 호출 액티비티에 따라 호출 되게 된다.

## 2.3 개인 맞춤형 웹 서비스

개인 맞춤형 서비스는 웹 서비스 플랫폼을 통해 유연하고 확장 가능한 특성을 가질 수 있다. 이러한 모델은 그림 4와 같이 표현할 수 있으며 Server는 앞에서 살펴본 Java 기반 웹 서비스 모델을 사용한다.

사용자는 실시간이나 특정 시간, 아니면 자신이 원하는 때에 서버 측에 사용자 정보를 전송하게 되고 서버는 이를 저장소에 저장한다. 이렇게 저장된 사용자 정보를 토대로 사용자에게 특화된 서비스를 제공하게 된다.

서버는 실시간이나 특정 시간, 아니면 사용자가 요청을 했을 때 특정 프로세스가 수행된 결과를 전송한다. 이 때 서버 내부에서는 BPEL로 기술된 비즈니스 프로세스가 수행되며 이는 내부적인 처리과정과 함께 외부 자원을 사용하는 프로세스를 통해 결과를 도출해낸다.

이러한 시스템에서는 사용자가 이용하는 여러 가지 서비스가 모두 하나의 저장소에 저장되므로 중복 입력이 필요가 없게 되고 자유로운 참조가 가

능해지면서 개인의 특성을 좀 더 고려할 수 있다. 앞에서 예를 들었던 일정관리를 다시 생각해보면 우리가 제시한 모델에서는 사용자가 집에 있는 데스크탑에서 입력한 일정이 자동적으로 휴대폰에서 참조되어 동기화가 되며 알람 서비스도 일어날 수 있다. 또한 회사에 있는 데스크탑에서도 동일한 일정관리 내용을 확인 할 수 있게 되며 모든 작용은 역으로도 가능하게 된다.

## 2.4 개인 맞춤형 웹 서비스 시나리오 예제

개인 맞춤형 웹 서비스에서 사용자와 서버의 상호작용은 크게 세 가지로 나누어 볼 수 있다. 첫 번째는 사용자의 정보를 서버에게 전송하는 패턴이고 두 번째는 서버가 사용자에게 메시지를 전송하는 패턴이다. 세 번째는 사용자가 특정한 서비스를 요구하면 서버는 그에 상응하는 결과를 응답해주는 패턴이다.

이러한 패턴은 기본적인 메시지 교환 패턴(MEP)에서의 단방향 메시지 패턴, 알림 메시지 패턴, 요청-응답 메시지 패턴과 대응된다고 볼 수 있다.[8]

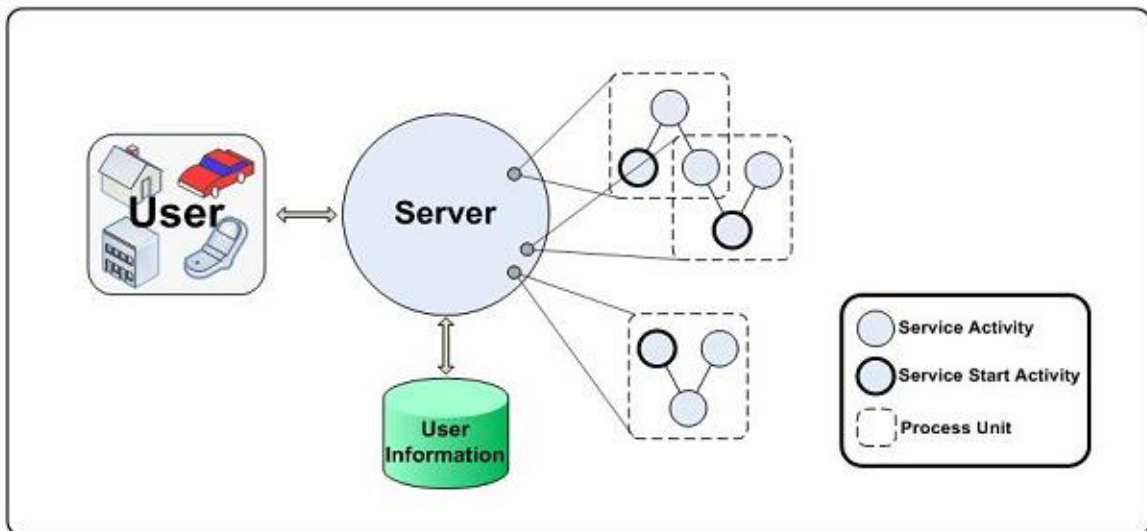


그림 4 개인 맞춤형 웹 서비스 모델

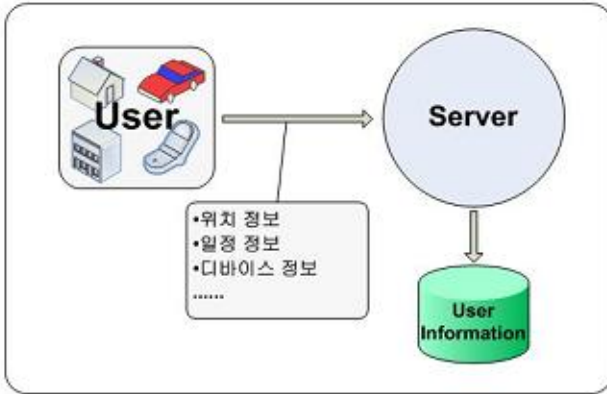


그림 5 사용자 측에서의 정보 송신

단방향 메시지 패턴의 경우, 그림 5와 같이 사용자가 서버에게 자신의 다양한 정보를 전송하게 된다. 여기에는 서비스 제공에 대한 설정도 포함된다.

예를 들면 현재 자신이 사용 중인 디바이스의 정보나 전송하거나 원하는 서비스의 목록이나 알림 방법, 시간들을 정하는 것이다. 만약 자동차와 같이 이동 중인 장치의 디바이스에서 서비스를 이용하게 된다면 실시간으로 위치 정보를 보내주는 경우도 필요하게 되고 이 경우도 단방향 메시지 패턴에 해당한다고 볼 수 있다.

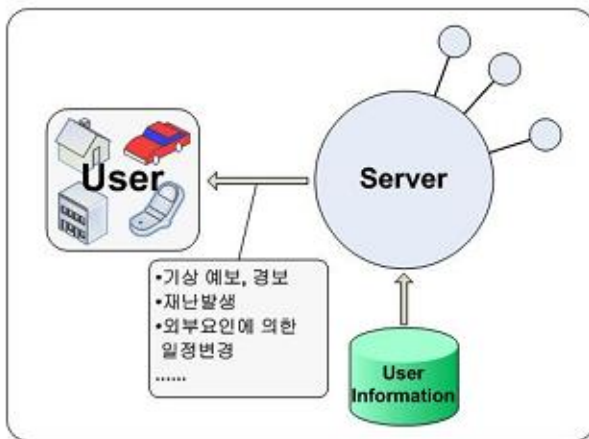


그림 6 알림 서비스

그림 6과 같은 알림 메시지 패턴은 서버가 사용자에게 메시지를 전송하는 패턴으로 서버 측에서 사용자에게 특별한 알림을 전송할 때 사용된다. 예를 들면 기상 예보나 경보 지역이 사용자의 현재 위치와 근접하다면, 혹은 앞으로의 일정과 연관된다면 바로 사용자에게 알림 메시지를 보내게 된다.

이는 사용자가 미리 설정해놓은 경우만 동작하는 것이 아니라 서버가 여러 가지 자원을 사용하여 판단하여 그 결과를 사용자에게 전송하는 것이다.

위와 같은 시나리오가 정상적으로 동작하기 위해서는 서버는 항상 외부 세계의 가능한 많은 정보를 계속해서 주시하고 있어야 하며 많은 서비스들과 연관되어 있어야 한다. 기상, 교통, 금융 분야와 같은 사회 기반 서비스는 필수적인 요소라고 할 수 있다.

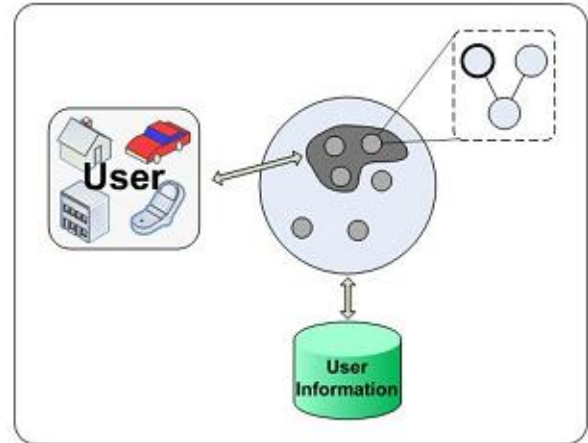


그림 7 요청 및 응답

그림 7과 같은 양방향 패턴의 경우는 가장 일반적으로 사용자가 특정한 목적을 가지고 서비스를 요청하면 서버는 서비스를 실행하고 그 결과를 응답하는 패턴이다. 이 때 사용자는 자신이 서비스를 요청하는지 꼭 의식할 필요는 없다.

한 가지 혹은 여러 가지의 사용자 요청에 따라 서버는 외부의 서비스와 저장소에 있는 사용자의 정보를 참고하여 논리적인 처리 결과에 따라 결과를 응답하게 된다.

### 3. 결론

개인 맞춤 서비스는 개인의 특성이나 사용 패턴을 고려하는 자동화되고 지능적인 처리를 요구한다.

하지만 사용자에 대한 정보는 한 시스템이나 플랫폼에 종속적인 형태가 많고 이에 따라 여러 가지의 사용 환경을 사용하는 사용자에겐 불편을 가져다주었다.

본 논문에서는 이러한 기존 개인 맞춤형 서비스의 문제를 해결하기 위해서 Service Oriented Architecture 패러다임을 도입하여 웹서비스 형태로 개인 맞춤형 서비스를 구현하는 모델을 제시하였다. 이를 이용하면 좀 더 유동적이고 재사용 가능하며 확장성이 있는 서비스 형태를 가질 수 있다. 또한 사용자 정보도 시스템이나 플랫폼에 종속적으로 관리되지 않으며 한 곳에 개인에 관한 정보를 저장함으로써 개인에게 특성화된 서비스를 제공할 수 있게 된다.

하지만 반대로 생각해 보면 사용자 정보를 한 곳에서 집중 관리하게 되면서 야기될 수 있는 위험성 문제와 모든 프로세스를 서버가 처리하면서 일어나는 서버의 과부하 문제는 성능 개선을 위해 반드시 고려해야 할 점이며 앞으로 개선해야 할 사항이라고 할 수 있다.

또한 각각의 사용자가 이용하는 서비스들을 개별적으로 개발 할 때 일어나는 비즈니스 프로세스의 중복 문제도 처리가 필요하다.

이와 더불어 사용자의 요청을 의미기반으로 해석하고 서비스의 검색, 등록, 조합, 조정의 지능화와 자동화를 위한 서비스 명세인 SAWSDL이나 에이전트 기반의 추천 시스템 도입으로 웹 서비스를 지능화 시킬 필요가 있고 이러한 주제들은 향후 연구의 중요한 과제라고 볼 수 있다.

## 후 기

본 연구는 유비쿼터스 자동화 컴퓨팅과 네트워크 프로젝트, 정보통신부(MIC) 21세기 프론티어 R&D 사업의 연구결과로 수행되었음

## [참고문헌]

- [1] Developing Java Web Services, Ramesh Nagappan 외 2명, 사이텍미디어
- [2] BPEL4WS를 이용한 동적이고 재사용가능한 웹 서비스 통합 모델, 김운용, 한국 인터넷 정보학회 2004년
- [3] 웹 서비스, 정지훈, 한빛미디어, 2002
- [4] [http://www.oracle.com/technology/global/kr/pub/articles/matjaz\\_bpel1.html](http://www.oracle.com/technology/global/kr/pub/articles/matjaz_bpel1.html)

[5] <http://tomcat.apache.org/>

[6] <http://ws.apache.org/axis2/>

[7] <http://www.active-endpoints.com/open-source-documentation.htm>

[8] SOA : 서비스 지향 아키텍처(개념에서 설계 구현까지), 토마스 얼 지음, 에이콘