

모바일 웹 애플리케이션 개발을 위한 SW 아키텍처 및 인터페이스 관련 연구

이성재*

*포스코 ict 정보제어기술연구소 SW 융합기술팀
e-mail : phantom@poscoict.com

An Architecture of Mobile Web Application

Sungjae Lee*

*SW Convergence Technology Team, posco ict

요 약

본 논문에서는 비즈니스 요구사항 변화에 기민하고 유연하게 대응할 수 있는 모바일 웹 애플리케이션 아키텍처를 제안한다. 모바일 웹 애플리케이션의 뷰를 제공하는 프리젠테이션 레이어, 비즈니스 로직을 처리하는 레가시 레이어와 프리젠테이션 레이어와 레가시의 인터페이스를 담당하는 중재레이어로 구성되어 시스템을 구축한다.

1. 서론

현재 스마트 폰 등의 휴대용 모바일 기기의 급속한 보급 확대에 따라 엔터프라이즈 환경에서도 다양한 비즈니스 업무 들이 모바일 기기에서 제동되어 시간과 공간의 제약이 없이 비즈니스 서비스를 제공할 수 있는 형태로 변화하고 있다. 다양한 모바일 기기 또는 모바일 플랫폼에서의 호환성 제공을 위해 웹 브라우저를 통해 서비스를 제공하여 이 기종 디바이스에 따른 중복 개발을 지양하는 추세이다. 본 논문에서는 비즈니스 요구사항 변화에 기민하고 유연하게 대응할 수 있는 모바일 웹 애플리케이션 아키텍처를 설계한다.

2. 본론

2.1 모바일 아키텍처의 목적

스마트 폰 등에서 사용하는 모바일 웹 애플리케이션 뷰를 제공하는 프리젠테이션 레이어와 비즈니스 로직을 수행하는 기존 레가시 시스템 레이어 그리고 두 레이어 사이에서 프로토콜 연계와 메시지 변환 등의 중재를 담당하는 시스템 아키텍처와 그 방법에 관한 것이다.

2.2 모바일 아키텍처 기대효과

모바일 웹 애플리케이션 개발을 위해서는 사용하고 있는 단말기기에 최적화된 웹 화면을 신규로 개발하

여야 한다. 그러나 화면을 통해 제공되는 정보는 기존의 레가시 시스템에서 웹 애플리케이션 형태로 제공되는 기존 비즈니스 정보와 별반 다를 것이 없다. 이러한 기존 애플리케이션에서 제공되는 비즈니스 로직을 재 사용하기 위해서는 레가시 시스템과의 인터페이스를 통해 기존 애플리케이션의 비즈니스 로직을 재 활용할 수 있도록 시스템을 구성하는 것이 필요하다. 기존의 비즈니스 로직을 레가시 레이어에서 웹 서비스로 서비스를 제공함으로써 재 사용성을 향상시켰으며 웹서비스의 장점을 활용하여 여러 비즈니스 업무에서 공통으로 사용할 수 있도록 하였다.

기존 레가시 시스템은 http, jms, Web Services 등의 서로 다른 전송 프로토콜과 xml, 단순 스트링 등의 다양한 메시지 데이터 포맷을 가지고 있는 경우가 대부분 이다. 이러한 다양한 이 기종 시스템과의 유기적인 통합을 위해 웹 뷰를 제공하는 웹 애플리케이션과 비즈니스 로직을 제공하는 레가시 시스템 레이어 사이에 메디에이션 레이어가 필수적이다. 전송 프로토콜 연계와 메시지 변환 등의 중재자 역할을 수행하는 메디에이션 레이어에 엔터프라이즈 서비스 버스(Enterprise Service Bus)를 배치하여 비즈니스 요구사항 변화에 유연하고 기민한 시스템 아키텍처를 구성 할 수 있다.

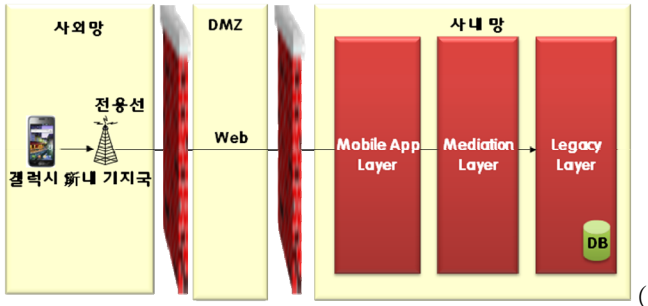


그림 1) 모바일 아키텍처 개념도

웹 애플리케이션 레이어와 레가시 시스템 레이어에 웹서비스 소비자 및 제공자 역할을 한다. 이 두 레이어에 웹서비스 호출을 위한 클라이언트와 호출을 처리하는 받아 처리하는 재 사용 컴포넌트를 제공하여 전송 프로토콜 및 메시지 형태 대한 추상화를 웹 뷰와 비즈니스 로직에 제공하였다.

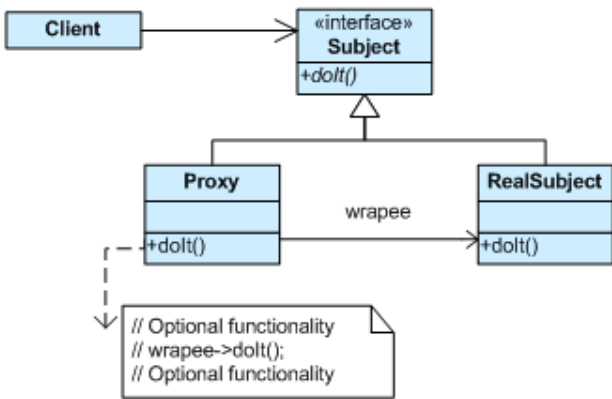
이러한 구조는 비즈니스 로직의 변경 없이 전송 프로토콜을 변경 할 수 있는 유연한 구조로 최소한 영향도로 시스템 요구사항에 맞게 전송 프로토콜을 변경 할 수 있다. 단 이러한 비즈니스 로직과 인터페이스 로직의 분리를 위해 메시지를 키, 값 쌍으로 포맷을 정의하여 페이로드 방식의 기 정의된 메소드 시그니처를 사용해야 하는 제약 조건이 있다. 즉 페이로드 인터페이스 방식에 프락시 패턴을 적용하였다.

기중 시스템의 연동을 위한 인터페이스의 복잡도 증가와 복잡도 증가에 따른 시스템 유지 보수의 어려움을 피 할 수는 없다.

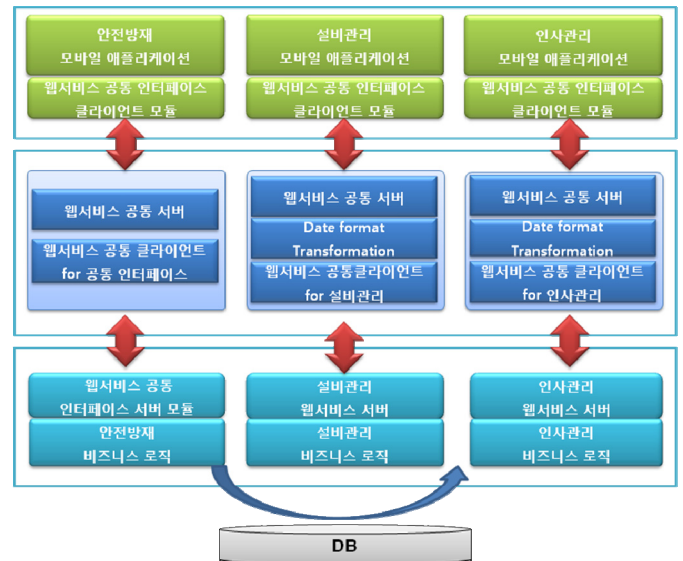
첫번째 뷰와 모델이라는 비즈니스 로직이 분리되어 있지 않는 경우는 기존 웹 애플리케이션의 비즈니스 로직을 재 사용하기가 어려우며 재 사용하기 위해서는 모바일 애플리케이션 인터페이스를 위한 기존 비즈니스 로직 인터페이스의 변경이라는 소스레벨의 수정이 발생한다. 두번째 뷰와 비즈니스 로직이 분리되어 있지만 전송 프로토콜과 메시지의 표준화가 안 되어있는 경우에는 다양한 프로토콜과 메시지 포맷을 고려하여 모바일 애플리케이션을 개발 및 운영해야 하는 단점이 있다. 마지막 방법은 전송 프로토콜과 메시지 표준화를 통해 인티그레이션 로직을 단순화시켰지만 연동되어야 하는 레가시 시스템의 수가 늘어남에 따라 시스템의 복잡도가 증가하고 트랜잭션 데이터의 모니터링 및 장애 감지가 어려운 문제가 존재한다.

2.4 모바일 아키텍처 구성도

전체 구성은 모바일 애플리케이션 레이어, 메디이션 레이어 레가시 시스템 레이어로 구성되어 있으며 각 레이어 간의 인터페이스는 기본적으로 웹 서비스를 사용한다.



(그림 2) 모바일 아키텍처 인터페이스



(그림 3) 모바일 아키텍처 구성도

2.3 기존 아키텍처와 비교

기존 모바일 웹 애플리케이션 개발 환경에서는 웹 화면을 제공하는 뷰와 비즈니스 로직이 하나의 시스템에 배치되어 있는 경우가 대부분이고 웹 화면을 제공하는 뷰와 비즈니스 로직이 시스템 상에서 분리되는 경우와 뷰와 비즈니스 로직이 시스템 상에 분리되어 있으며 개방형 표준 인터페이스를 도입하여 구성하는 경우가 있을 수 있다.

상기 세가지 모두 전송 프로토콜 연계와 다양한 메시지 포맷의 연동을 위한 메시지 변환의 역할을 수행하는 중재 레이어의 부재로 시스템의 확장 시 이

2.4.1 모바일 애플리케이션 레이어

모바일 웹 애플리케이션의 화면 뷰를 제공하는 레이어로 웹 화면을 제공하기 위해 웹 애플리케이션 서버가 배치되어 있으며 메디이션 레이어를 통해 기존 레가시 시스템과 인터페이스 한다. 레가시 시스템에서 제공하는 웹 서비스를 사용하는 클라이언트 역할을 수행하며 웹 서비스 공통 인터페이스 클라이언트 모듈이 프레임워크의 재 사용 컴포넌트로서 사용

된다.

각 모바일 애플리케이션에서 웹 서비스 호출을 위한 파라미터를 키/값 쌍으로 맵핑하여 기 정의된 웹 서비스 공통 인터페이스 클라이언트 모듈에서 제공하는 메소드를 호출하여 처리하고자 하는 비즈니스 로직을 수행한다.

■ 웹 서비스 공통 인터페이스 클라이언트 모듈

웹 서비스 공통 인터페이스 클라이언트 모듈은 웹 서비스 서버의 변경에 따른 영향도를 최소화 하기 위해 다양한 파라미터를 수용하기 위해 키/값 형식의 스트링 파라미터로 넘겨 웹 서비스를 호출 할 수 있는 공통 인터페이스를 구현하여 라이브러리 형태로 제공한다. 호출하고자 하는 웹 서비스의 엔드포인트와 오퍼레이션 네임을 공통 인터페이스 호출 시 파라미터로 같이 넘긴다.

2.4.2 모바일 중재 레이어

모바일 애플리케이션에서 필요로 하는 정보를 기존 레가시 시스템과 인터페이스 하기 위해 필요한 다양한 프로토콜과 데이터 포맷의 중재를 위한 역할 수행하여 기존 레가시 시스템과 모바일 애플리케이션의 변경에 대한 영향도를 최소화 한다.

웹 서비스 공통 인터페이스 모듈로 제공되는 웹 서비스 인터페이스 이외에도 기존 웹 서비스에서 정의되어 있는 다양한 웹 서비스 오퍼레이션의 연계를 위해 웹 서비스 메시지 포맷을 각 레가시 시스템에서 요구하는 형태로 변환 할 수 있는 메시지 변환 기능을 제공한다.

레가시 시스템에 웹 서비스를 요청 할 때는 키/값 쌍으로 정의되어 있는 스트링을 파싱하여 웹 서비스 요청을 위한 요청 SOAP 구조체를 생성하며, 레가시 시스템의 웹 서비스 응답으로 받은 응답 SOAP 구조체를 키/값 스트링 형태로 생성하여 모바일 애플리케이션에 리턴한다.

■ 웹 서비스 공통 서버 모듈

웹 서비스 공통 서버는 기존 레가시 시스템에 웹 서비스를 요청하기 위한 모바일 애플리케이션과 인터페이스를 위하여 웹 서비스를 제공하는 역할로 모바일 애플리케이션은 레가시 시스템과 맵핑되어 있는 웹 서비스 공통 서버에 서비스를 요청함으로써 접속하고자 하는 레가시 비즈니스 로직에 수행을 요청 할 수 있다.

■ 메시지 포맷 변환

웹 서비스 공통 인터페이스 서버 모듈로 개발된 레가시 시스템 레이어의 비즈니스 로직은 키/값으로 정의되어 있는 스트링 파라미터를 파싱하여 처리하여 비즈니스 로직을 수행하나 웹 서비스 공통 인터페이스 서버 모듈로 개발되지 않는 비즈니스 로직은 웹 서비스 오퍼레이션에서 개별적으로 정의된 메소드 시그니처에 맞춰 파라미터를 파싱하여 호출해 줘야 한다. 또한 레가시 시스템 웹 서비스로 받은 응답도 키

/값 형태의 스트링으로 결과 값을 생성하여 모바일 애플리케이션에 보내줘야 한다.

웹 서비스 요청 시는 키/값으로 쌍으로 맵핑되어 있는 스트링을 기 정의되어 있는 구조체 맵핑 정보를 참조하여 레가시 시스템에서 정의한 웹 서비스 요청 구조체로 변경한다.

웹 서비스 응답 처리 시는 웹 서비스 응답 구조체를 기 정의된 구조체 맵핑 정보를 참조하여 키/값 쌍으로 맵핑하여 스트링을 생성한다.

■ 웹 서비스 공통 클라이언트 모듈

레가시 시스템 레이어에서 개발된 웹 서비스를 호출하기 위한 웹 서비스 호출 공통 클라이언트로 웹 서비스 호출을 위한 URL 와 실제 비즈니스 로직을 수행하기 위한 오퍼레이션 명을 설정하여 웹 서비스를 호출 할 수 있다. 웹 서비스 호출을 위한 파라미터는 키/값 쌍으로 맵핑하여 웹 서비스 공통 클라이언트에 넘기면 맵핑된 키/값을 파라미터 명과 값으로 웹 서비스 요청 구조체를 생성하여 레가시 시스템 레이어에 제공하는 웹 서비스를 호출하고, 서비스 응답 메시지를 키/값 쌍으로 맵핑하여 결과 값을 리턴한다.

2.4.3 모바일 레가시 시스템 레이어

기존 레가시 시스템에서 사용하고 있는 비즈니스 로직과 인터페이스를 통해 모바일 애플리케이션에서 필요로 하는 정보제공 및 비즈니스 로직을 수행하는 레이어로 다양한 전송 프로토콜 및 데이터 포맷이 공존 하는 레이어 이다. 또한 웹 서비스 공통 인터페이스 모듈의 서버 측 모듈이 개발 프레임워크의 재사용 컴포넌트로 제공된다.

레가시 시스템 레이어에서는 모바일 애플리케이션에서 필요로 하는 비즈니스 로직을 수행한다.

웹 서비스 공통 인터페이스 서버 모듈에 의해 비즈니스 로직 인터페이스가 개발되거나 개별적으로 웹 서비스 인터페이스를 정의하여 개발된다.

■ 웹 서비스 공통 인터페이스 서버 모듈

웹 서비스 공통 인터페이스로 정의되어 있는 스트링 파라미터를 키/값으로 쌍으로 파싱하고 정의되어 비즈니스 서비스 이름으로 수행하고자 하는 비즈니스 로직을 식별하여 호출한다. 키/값으로 맵핑되어 있는 처리 결과를 스크링으로 만들어 웹 서비스 공통 인터페이스에서 정의된 스트링 형태로 변환하여 응답을 보낸다.

3. 결론

현재 스마트 폰 등의 휴대용 모바일 기기의 급속한 보급 확대에 엔터프라이즈 환경에서도 다양한 비즈니스 업무 스마트 오피스 또는 스마트 워크라는 명칭으로 시간과 공간의 제약을 구애 받지 않는 모바일 애플리케이션들이 개발되어, 비즈니스 서비스들이 스마

트 기기들을 기반으로 진화하고 있다.

모바일 플랫폼 독립적인 모바일 웹 애플리케이션을 변화하는 비즈니스 요구사항에 유연하고 기민하게 대응할 수 있는 모바일 애플리케이션의 뷰를 제공하는 모바일 애플리케이션 레이어와 비즈니스 로직을 제공하는 기존 레가시 시스템 레이어와 이 둘 사이의 전송 프로토콜 연계와 메시지 변환 등의 중재 기능을 제공하는 매디에이션 레이어로 아키텍처를 구성하였고 매디에이션 레이어를 통해 통합되어 있는 두 레이어의 변경에 대한 영향도를 최소화 하기 위해 웹 애플리케이션 개발 프레임워크에서 재 사용 가능한 컴포넌트 형태로 상호 인터페이스가 가능 하도록 설계 및 개발 하였다.

향후 Smart Works 라 불리는 모바일 웹 애플리케이션 프로젝트 업무에 모바일 아키텍처 및 모바일 뷰와 비즈니스 로직을 연계하는 통합 인터페이스 모듈(서버/클라이언트)을 적용하여 모바일 시스템 개선효과를 검증 할 예정이다.

참고문헌

- [1] David A. Chappell, "Enterprise Service Bus", O' Reilly, 2004.
- [2] Thomas Earl, Service-Oriented Architecture : Concepts, Technology, and Design, Prentice Hall PTR, 2005.
- [3] Martin Keen, Amit Acharya, Susan Bishop, Alan Hopkins, Sven Milinski, Chris Nott, Rick Robinson, Jonathan Adams, Paul Verschueren
Patterns: Implementing an SOA Using an Enterprise Service Bus, IBM, 2004
- [4] Anurag Goel, Enterprise Integration EAI vs. SOA vs. ESB
- [5] Colombe Herault, Gael Thomas, and Philippe Lalanda, Mediation and Enterprise Service Bus A position paper, 2005
- [6] Erich Gamma, Design Patterns, Addison Wesley, 2003
- [7] SOAP version 1.2, <http://www.w3.org/2000/TR/Group>