

플랫폼 독립적인 스마트폰 응용 개발을 위한 SOA 기반구조¹⁾최민^{0*}, 정영식⁺, 이용주⁺, 정성태⁺, 임승호⁺⁺원광대학교 컴퓨터공학과⁺한국외국어대학교 디지털정보공학과⁺⁺{mchoi^{0*}, ysjeong⁺, yjlee⁺, stjung⁺}@wku.ac.kr, slim@hufs.ac.kr⁺⁺

SOA Infrastructure for Platform Independent Smart-Phone Application Development

Min Choi^{0*}, Youngsik Jeong⁺, Yongju Lee⁺, Sungtae Jung⁺, Seungho Lim⁺⁺Dept. of Computer Engineering, Wonkwang University⁺Dept. of Digital Information Engineering, Hankuk University of Foreign Studies⁺⁺

요 약

현재의 스마트폰 응용은 플랫폼별로 별도의 SDK를 통해 응용을 개발하고 전용 앱스토어를 통해 애플리케이션을 다운로드하는 방식을 따른다. 본 연구에서는 플랫폼 독립적 스마트폰 응용을 목표로 한다. 한번 구축하면 스마트폰 플랫폼에 구애받지 않고 공통적으로 활용할 수 있다. 이를 위해, 웹 서비스 기반의 SOA 기반구조를 구축하고, 그 위에 공통서비스/요소서비스를 제공한다. 스마트폰 앱 실행환경이 서로 호환되지 않아 유사한 기능의 앱을 처음부터 세가지 다른 플랫폼으로 구현해야 하는 문제를 해결한다.

1. 서 론

우리나라는 그동안 IT분야에서 많은 발전을 이루었고, 단말과 네트워크 중심의 휴대폰 시장에서 성과를 거두었다. 하지만, 최근 몇 년간 애플이나 구글이 주도한 차세대 스마트폰 플랫폼과 애플리케이션 시장에서 다소 뒤쳐진 것이 사실이다. 통신 단말에 칼라 LCD 및 다양한 음원과 화음을 채택하고, 자체적인 무선인터넷 플랫폼을 개발하는 등 단말/네트워크 중심의 경쟁에서는 국내 업체들이 앞섰지만, 애플을 이끌고 있는 스마트폰 시장이 급속도로 발달하면서 전세가 바뀌었다. 즉, 스마트폰 시장에서는 단말자체의 하드웨어 아키텍처 보다는 스마트폰 자원을 관리하는 운영체제와 플랫폼과 스마트폰 애플리케이션의 역할이 더 중요해진 것이다.

현재 널리 활용되는 스마트폰 운영체제로는 심비안, 윈도우 모바일, 애플 아이폰, 안드로이드, 모조, 리모, 바다 등을 꼽을 수 있다. 시장점유율로 볼 때에는 심비안 47.2%, 블랙베리 20%, 애플 아이폰 14.5%, 윈도우 모바일 8.7%, 구글 안드로이드 4.7%, Palm/Linux 3.9%로 심비안과 블랙베리가 상당한 비율을 차지하고 있다[9]. 하지만, 최근 폭발적인 관심을 불러오는 애플 아이폰과 구글의 안드로이드가 머지않아 차세대 스마트폰 시장의 선두주자로 자리매김 할 것이라는 예견이 없다. 실제로, 심비안과 윈도우 모바일 등의 선두 스마트폰 운영체제가 꾸준히 시장 점유율 감소를 보이고, 애플 아이폰과 구글 안드로이드 플랫폼이 지속적으로 활용 증가 추세에 있다. 특히,

애플 앱스토어(appstore)는 최근 다운로드 횟수를 볼 때, 아이폰(iphone)/아이팟(ipod) 합산 기준으로 30억회를 넘어섰다(2010년 1월 기준). 이것은 앱스토어 운영을 시작한지 불과 18개월만에 이루어낸 실적이다 [10]. 애플 앱스토어가 다운로드 10억회를 돌파한 시점이 2009년 4월 23일이고, 20억회에 다다른 시점이 2009년 9월 28일임을 감안하면, 그 성장세는 폭발적이다. 현재 애플 앱스토어는 10만개 이상의 애플리케이션을 보유하고 있다. 구글(안드로이드) 앱스토어가 16,000개 정도의 애플리케이션을 갖고 있다 [5].

스마트폰 플랫폼과 애플리케이션 분야의 급속히 대중화 됨에 따라 국내외에서 많은 개발자들이 다양한 스마트폰 애플리케이션(응용)을 쏟아내고 있다. 대부분의 스마트폰 응용은 여러 기반기술 서비스가 합쳐져 새로운 기능을 제공하는 매쉬업(mash-up) 방식이 많다. 스마트폰 응용은 일반적인 데스크톱 응용과 달리 가입자의 현재 위치 등의 상황정보를 활용하기 때문이다. 예를 들어, 위치기반서비스(location based system)와 소셜 네트워킹 서비스(social networking service)를 결합하여 주변에서 개인취향에 가장 적합한 음식점을 자동으로 추천하는 등의 서비스가 가능하다. 따라서, 스마트폰 응용 개발분야에 있어서는 이와 같은 서비스 기반 매쉬업에 의한 애플리케이션 구축 기법이 활성화 되어가는 추세라 할 수 있다.

오늘날 개인 개발자들은 서로 다른 플랫폼에서 중복적으로 필요한 기능을 구현하는데 추가적인 노력을 기울인다. 애플 아이폰, 안드로이드, 윈도우 모바일의 각 플랫폼이 제공하는 개발자 SDK와 프로그래밍 언어가 서로 다

1) 이 논문은 2010학년도 원광대학교의 교비지원에 의해서 수행되었음.

르기 때문이다. 애플에서 제공하는 10만개의 응용을 안드로이드나 윈도우 모바일에서 활용할 수 있다면 어떨까? 물론, 애플사 입장에서는 자사가 보유한 다양한 응용 프로그램이 그 자체로 또 하나의 경쟁력이다. 따라서, 이것을 다른 플랫폼과 공유할 생각이 없겠지만, 스마트폰 응용 프로그램 개발을 전문으로 하는 회사나 개발자 입장에서는 다르다. 한번 개발한 스마트폰 응용 프로그램을 서로 다른 3개의 플랫폼에 맞추어 중복 개발해야 한다. 사용자가 근거리에서 위치한 저렴한 주유소를 찾아주는 스마트폰 응용 서비스를 원하는 경우, 이것이 어떤 플랫폼을 통해서 제공되는지에 대해서 사용자는 관심 없다. 단지 서비스만 활용하면 된다. 앱스토어에서 다운로드하고 설치하는 과정도 번거롭다. 사용자는 단지 필요할 때 웹브라우저로 접속해서 서비스를 받으면 된다. 이것은 최근 몇 년간 IT 서비스 패러다임이 애플리케이션 기반에서 웹 기반 정보시스템으로 변화하게 된 근본적인 이유이다. 실제로, IBM은 이미 LotusLive [1]등과 같은 자사의 IT서비스를 스마트폰 플랫폼에서 제공할 때 아이폰용 응용과 함께 웹브라우저에서 동작하는 응용서비스를 함께 2가지 버전으로 제공하고 있다. 사용자는 본인의 취향에 따라 웹 버전이나 네이티브 앱 버전을 선택하여 활용할 수 있다. 하지만, 이러한 접근방법은 동일 기능을 두가지 버전으로 개발해야 하므로 개발사 부담이 크다는 단점이 있다.

한번에 웹기반 서비스로 이동하기 힘들다면 중간단계도 가능하다. 핵심요소기술과 공통기술에 대해서만 플랫폼 독립적인 서비스로 구현하고 이를 별도의 운영시스템에 의해 보안/과금/관리 하는 방식이다. 웹기반응용 개발 회사/개발자는 자신의 핵심 요소기술을 플랫폼 독립적인 서비스로 만들어 등록하고, 각 플랫폼에 적합한 언어와 SDK로 외부 인터페이스 및 단말 제어를 구현한다. 디자인 패턴의 MVC(model, view, control) 모델 [8]에 따른 애플리케이션 설계가 가능하다. 컨트롤과 데이터 모델 영역은 수정을 가하지 않고 뷰 인터페이스만 바꾸어 서로 다른 플랫폼에서 스마트폰 앱을 개발할 수 있다. 따라서, 현재와 같이 스마트폰 앱 실행환경이 서로 호환되지 않아 유사한 기능의 앱을 처음부터 세가지 다른 플랫폼으로 구현해야 하는 문제를 해결하는 방안을 제시하고자 한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련연구를 살펴보고, 3장에서 플랫폼 독립적 스마트폰 앱을 위한 SOA기반구조를 살펴본다. 4장에서는 결론 및 향후 연구 방향에 대해 정리한다.

2. 관련연구

현재의 스마트폰 웹 애플리케이션이 갖는 가장 큰 단점 중 하나는 네이티브 애플리케이션(아이폰이나 안드로이드 플랫폼 API를 활용하는)과 달리 단말의 하드웨어에 대한 접근이 어렵다는 것이다. 예를들어, 단말의 각종 센서, 카메라, 주소, 연락처 정보, 그리고 단말에 저장된 스케줄 등의 내용을 연계한 다양한 응용 프로그램을 개발하기 어렵다.

이러한 스마트폰 웹 응용의 문제점을 해결하기 위해 W3C [2]를 중심으로 모바일 웹 플랫폼을 위한 Device API 표준화가 진행되고 있다. 2008년 12월에 Device API와 관련되는 다양한 표준화 이슈를 제기한 바 있고, 2009년 6월에 DAP Work Group을 발족하였다. W3C는 2010년말까지 1차 표준화를 마치고 Device API 표준안을 공개할 예정이다. 세부적으로 W3C 표준안에 적용하기 위해서 노력하는 단체들은 여러곳이 있으나 JIL(joint innovation lab)과 OMTP BOND이 대표적이다 [6].

JIL [3]은 OS나 플랫폼에 관계없이 다양한 애플리케이션을 사용할 수 있도록 하기 위한 통합 미들웨어 프로젝트로 2008년 4월 최초 결성되었다. JIL 프로젝트에는 차이나모바일, 소프트뱅크, 버라이즌 와이어리스 등 세계 굴지의 기업들이 참여하고 있으며, JIL 회원사 가입자만 10억명 이상이다. 표준화 플랫폼인 JIL이 확산되면 미국 개발자가 버라이즌을 위해 만든 애플리케이션을 중국의 차이나 모바일 사용자들이 이용할 수 있다. 일본이나 유럽 사용자들도 마찬가지다. 애플 앱스토어가 통신 서비스 회사들과 경쟁구도 및 갈등을 형성하는 반면, JIL은 철저하게 단말 제조사 및 통신 서비스 회사와 공조를 통한 모델이므로 시장에서 우호적인 편이다. 최근 JIL 프로젝트를 통해 삼성전자, LG전자, RIM, 샤프의 4개사가 2010년부터 JIL 미들웨어를 탑재한 단말을 공급할 예정이다.

OMTP는 사용자 지향의 모바일 서비스와 데이터 비즈니스 성장을 목표로 통신사업자들에 의해 만들어진 포럼이다. BOND이 OMTP에서 만든 웹 기반 응용 혹은 위젯이 모바일 폰 기능을 보완적인 방법으로 접근하게 하는 모바일 웹 런타임 플랫폼이다. HTML5 [7], 자바 스크립트, CSS, AJAX 등의 웹 표준 기술로 작성된 응용이 모바일 폰의 내부 장치 및 정보(카메라, 센서, 메시징, 전화, 연락처 등)에 접근할 수 있음을 의미한다.

BOND이 [4] 는 자바스크립트 확장(extension)과 이에 대한 접근 제어(access control)로 구성된다. BOND이는 AT&T, T-모바일, 보다폰(vodafone) 등의 여러 이동통신 사업자가 결합하여 구성한 것으로, 웹 개발자들이 웹 중심의 실행환경에서 쉽고 편리하게 모바일 웹 서비스를 개발할 수 있는 서비스 인터페이스 및 보안 문제에 대한 해결 방안을 제시한다. OMTP는 오픈소스 참조구현(open source reference implementation)과 명세 기반 참조구현(specification driven by reference implementation)을 통해 빠른 규격화 및 W3C 표준화를 주도하고 있다. 실제로, OMTP BOND이는 2009년 5월 API Spec 1.0 버전을 공표한 이래 W3C 표준화를 위해 지속적으로 노력중이다.

지금까지 언급한 관련연구들은 공통적으로 웹 기반 응용이 단순 네트워크형 응용의 한계에서 벗어나 폰 내부 자원을 표준에 근거한 방법으로 접근할 수 있도록 하여, OS나 플랫폼에 독립적인 모바일 응용을 개발할 수 있도록 한다. 따라서, 본 연구가 추구하는 플랫폼 독립적 서비스 기반구조가 이와 함께 연동될 때, 보다 완벽한 서비스 기반 웹 응용 플랫폼을 실현할 것으로 기대한다. 즉, W3C의 표준안이 공표되는 2010년 하반기에 본 연구결과와 함께 더욱 큰 시너지를 창출할 것으로 예상된다.

3. 플랫폼 독립적 스마트폰 앱을 위한 SOA 기반 구조

현재 스마트폰 앱개발 환경은 아이폰, 안드로이드, 윈도우 모바일 등으로 구분된다. 플랫폼별로 별도의 SDK를 통해 응용을 개발하고 전용 앱스토어를 통해 애플리케이션을 다운로드하는 방식을 따른다.

본 연구에서 추구하는 플랫폼 독립적 스마트폰 응용은 한번 구축하면 스마트폰 플랫폼에 구애받지 않고 공통적으로 활용할 수 있는 응용이다. 데스크탑의 경우에 비유하자면 기존의 IT서비스 응용을 웹 기반 시스템으로 개발하는 것과 유사하다. 운영체제에 관계없이 웹 브라우저를 통해 어디서든 접속할 수 있다. 게다가, 모바일 웹 응용의 형태로 구현된 정보 시스템 응용은 아이폰, 안드로이드폰과 같은 스마트폰 뿐 아니라, PMP, 미니노트북(넷북) 등 모바일기기에서 변환이나 설치 없이 구동할 수 있다.

하지만, 완전한 형태의 플랫폼 독립적인 앱 개발/운영 환경을 실현하기에는 아직 해결되어야 할 부분이 많다. 가장 시급한 부분은, 모바일 웹 플랫폼을 위한 Device API 표준화이다. 이 작업은 현재 W3C를 중심으로 진행 중이며, 2010년 하반기에 표준안이 발표될 예정이다. 이 표준이 마련되기 전까지는 ①, ②와 같이 플랫폼별 네이티브 앱 인터페이스에 어느정도 의존할 수 밖에 없다. 현재 대부분의 앱개발이 ①과 같이 플랫폼별 네이티브 API를 활용한다. W3C 표준안이 발표되기 전까지는 ②의 절충안이 가장 현실적이면서도 향후 ③으로 확장 가능한 모델이다. 따라서, 본 연구는 최종적으로 ③의 모델을 목표로 하지만, 현실적으로 ②의 구조를 1차적인 목표로 진행하고 있다.

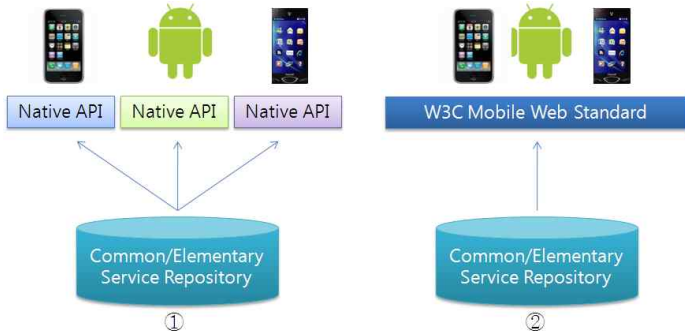


그림 1 SOA 웹 서비스 기반 스마트폰 앱 구조

그림1은 본 연구에서 제시하는 절충안 ②와 최종적으로 목표하는 바인 요소/공통 기술 웹서비스를 활용한 플랫폼 독립적 아키텍처 ③을 소개한다.

- ① 플랫폼별 네이티브 앱
- ② 플랫폼별 네이티브 앱 인터페이스 + 플랫폼 독립적 요소/공통 서비스
- ③ 플랫폼 독립적 웹 서비스 기반 앱

특히, ②는 현재의 스마트폰 플랫폼 제조사와 앱개발사를 위한 상호 절충안이자 모바일 웹 응용을 위한 표준안이 확정되지 않은 현재로서는 가장 현실적인 방법이다.

특정 플랫폼에 적합한 SDK를 이용해 플랫폼별 전용 앱을 개발하지만, 개발사 입장에서는 자사의 공통/요소 서비스를 다른 스마트폰 플랫폼 개발을 위해서도 활용할 수 있는 방법이므로. 가장 현실성 있는 방안이라 할 수 있다. 그리고, 현재 스마트폰 웹 애플리케이션으로는 불가능한 센서 및 단말 하드웨어 접근을 가능하도록 외부 인터페이스에 네이티브 API를 사용하는 방법이다.

③은 W3C의 모바일 웹 애플리케이션 표준안이 발표되는 2010년 이후 실현 가능한 모델로서, 본 연구가 궁극적으로 추구하는 방향이다. 동일기능을 서로 다른 플랫폼에서 중복으로 구현하는 것을 방지하기 위해 웹 서비스 기반 SOA 기반구조를 구축하고, 그 위에 공통서비스/요소서비스를 제공한다. 공통/요소 서비스의 웹 서비스 구현 및 서비스 저장소(repository)구축이 선행되어야 한다. 이러한 기반환경이 마련된다면, 공통/요소 웹 서비스와 단말의 로컬 자원을 함께 활용하는 모바일 스마트폰 웹 응용의 개발이 활성화될 것이다. 실제로, 서로 다른 모바일 단말 플랫폼(아이폰, 안드로이드, 윈도우 모바일)이라 할지라도, 공통적으로 활용되는 스마트폰 공통 서비스(SMS, MMS, 광고, 주소록 DB, 우편번호 안내, 114안내 DB 등)/요소 서비스(음성처리, 영상처리, HCI, 위치기반서비스, 유비쿼터스, 클라우드 컴퓨팅, 이러닝, 과학계산 등)를 플랫폼 독립적인 서비스로 구현할 필요성이 제기되고 있다. 개인 개발자들은 서로 다른 플랫폼에서 중복적으로 필요한 기능을 구현하는데 불필요한 노력을 기울인다. 즉, 스마트폰 앱 실행환경이 서로 호환되지 않아 같은 기능의 앱을 세가지 다른 플랫폼으로 구현해야 한다. 이러한 기능들을 플랫폼 독립적인 서비스로 개발하여 서비스 저장소(service repository)에 넣어두면, 개발자들은 서비스의 조합으로 스마트폰 앱을 개발한다. 이 때, 개발자는 본인이 원하는 기능(서비스)을 앱창작터 서비스 저장소에서 UDDI, WSDL에 의해 검색하여 사용할 수 있다.

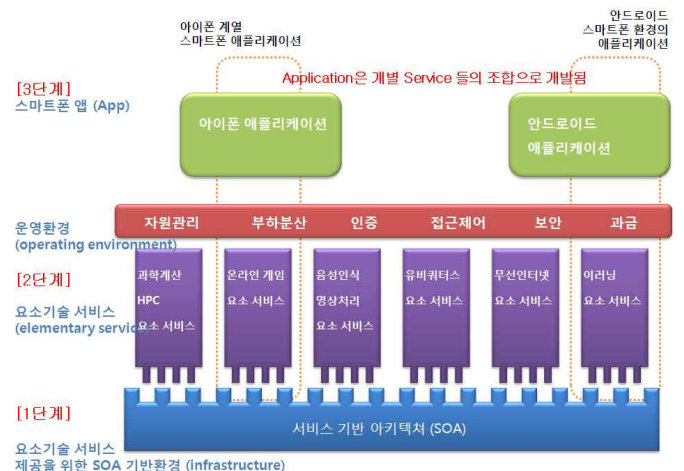


그림 2 요소/공통 서비스를 위한 SOA 기반구조

스마트폰 앱 개발에 필요한 공통 서비스/요소 서비스, 특히 기술을 SOA 기반구조에서 서비스로 제공하면 개발자들은 이를 활용/조합하여 새로운 스마트폰 앱을 만든다. 여기서 서비스는 SOA (Service-Oriented Architecture)

를 실현하는 소프트웨어 부품의 개념이다. SOA 기반구조 운영센터는 스마트폰 앱개발 서비스를 제공함과 동시에, 국내 스마트폰 앱개발 활성화를 위한 차원에서 SOA 지원 프레임워크를 구축한다. 스마트폰 타겟 플랫폼(아이폰, 안드로이드, 윈도우 모바일)에 구분 없이 활용가능한 플랫폼 독립적인 서비스로 공통기술/요소기술(음성처리, 영상처리, HCI, 위치기반서비스, 유비쿼터스, 클라우드 컴퓨팅, 이러닝, 과학계산 등)을 제공한다.

본 연구에서는 이와 같이 플랫폼 독립적 스마트폰 앱개발 지원체제를 확립하기 위해서, 그림3과 같은 3단계 기반구조 구축을 고안하였다.

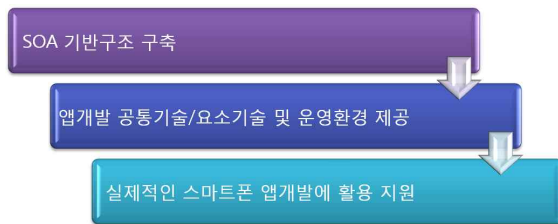


그림 3 SOA 기반구조 구축을 위한 3단계 전략

1단계 : 스마트폰 앱개발 활성화를 위한 SOA 기반구조(infrastructure) 구축

- 스마트폰 앱개발 분야를 활성화하기 위해서는 손쉽게 앱개발에 뛰어들 수 있는 개발환경을 마련해야 한다. 빠르고 손쉬운 애플리케이션 개발(RAD)을 실현하는 개발환경을 제공할 필요가 있다.
- 실제로 최근 널리 알려진 Object C 기반의 아이폰 앱개발은 이미 컴포넌트기반 개발(CBD: Component-Based Development) 기반으로 가능(전문적인 애플리케이션 개발 지식이 없는 고등학생들도 쉽게 스마트폰 응용 프로그램을 개발하는 상황)
- 스마트폰 앱개발을 위해 공통적으로 필요한 기능은 플랫폼 독립적인 웹서비스로 구현 → 서비스 저장소(repository)에 등록하여 센터에서 관리 → 앱개발자들이 개발자 인증 후 활용하도록 제공한다.

2단계 : 스마트폰 앱개발 공통서비스/요소서비스 및 운영환경 구축

- 1단계에서 컴퓨팅 SOA를 위한 기반구조가 마련되면, 스마트폰 애플리케이션 구성에 필요한 여러가지 공통 서비스/요소서비스를 플랫폼 독립적으로 구현가능.
- 분야별로 세부적인 요소서비스 구축
- 스마트폰 애플리케이션에 공통적으로 사용되는 서비스나 음성/영상 정보처리를 위한 요소기술 서비스를 플랫폼(아이폰, 안드로이드, 윈도우 모바일) 구분 없이 활용할 수 있도록 웹서비스(web service) 기술을 선택하였다.
- 웹서비스는 SOA를 실현하는 테크닉 중 하나로 플랫폼, 언어, 객체 모델에 중립적인 형태로 다른 프로그램에서 표준화된 웹서비스 프로토콜을 통해 호출할 수 있는 서비스이다.
- 공통 서비스(elementary service)와 함께 클라우드 서

비스 운영환경(OE: Operating Environment)이 필요

- 서비스 운영환경(OE)은 자원관리 및 부하분산, 접근 제어, 인증, 보안, 과금 등의 프레임워크 운영관리를 담당한다. 웹 서비스에 대한 활용은 MD5 지문(fingerprint)이나 공인인증서에 의해 인증된 경우에만 접근 가능하다. 이를 위해, 접근제어, 인증, 보안, 과금 등의 기능이 필요하다.

3단계 : 실제적인 스마트폰 앱개발에 활용 지원

- 국내 스마트폰 앱개발 인력양성 및 서비스 활성화를 위해서는 애플리케이션에 대한 서비스 추상화(abstraction)가 필요하다.
- 애플리케이션 사용자는 서비스 사용시 그 서비스가 어떠한 원리로 동작하며 어떻게 구현되었는지에 대해 알 필요 없음, 뿐만 아니라 SOA에 의한 서비스 기반 앱개발자 입장에서 서비스 내부 구현 원리를 인지할 필요 없다.
- 이러한 서비스 추상화는 강력하고 편리한 기능의 스마트폰 앱을 보다 쉽고 부담없이 개발할 수 있어 서비스 대중화의 바탕이 된다.

4. 결론 및 향후 계획

본 연구는 향후 다양한 SOA 플랫폼을 비교검토한 뒤, 이를 모바일 요소/공통 기술을 서비스하는데 적합하도록 구현·변경하여 SOA 기반구조로 채택할 예정이다. 그리고, 구축된 서비스 기반구조 인프라를 바탕으로, 서비스 기반 앱개발 활성화를 실현하기 위해 각종 요소 서비스와 운영환경 서비스를 제공할 것이다. 실제로 이를 모델링 하기 위해 웹서비스 플랫폼을 구축하고 특정 분야의 기본적인 요소 서비스와 운영환경을 제공할 것이다. 또한, 구축된 SOA 플랫폼과 요소 웹서비스, 운영환경을 활용하여, 서비스 활성화를 실현할 것이다. 웹서비스 개념은 플랫폼 독립적이기 때문에 어떠한 스마트폰 플랫폼에서도 손쉽게 연동할 수 있다. 결국, 일반 데스크탑 애플리케이션, 모바일 장치 애플리케이션과 함께 연동하여 누구나 손쉽게 서비스를 활용할 수 있도록 한다.

5. 참고문헌

- [1] IBM Lotus Live, <http://www.lotuslive.com>
- [2] W3C(World Wide Web Consortium) Mobile Web Initiative, <http://www.w3.org>
- [3] JIL (Joint Innovation Lab), <http://www.jil.org/>
- [4] OMTP (Open Mobile Terminal Platform) BOND, <http://bondi.omtp.org>
- [5] 전종홍, 이승윤, "차세대 모바일 웹 애플리케이션 표준화 동향", 전자통신동향분석 제25권 제1호, 2010년 2월
- [6] John Arne Saeteras, Mobile Web vs. Native Apps. Revisited, Mobiletech AS, April 9, 2010
- [7] W3C, Working Draft, "HTML 5", <http://www.w3.org/TR/html5>
- [8] Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, "Design Patterns : Elements of Reusable Object-Oriented Software", Addison-Wesley, 1994
- [9] 2009 Smart Phone Sales, Gartner, 2010
- [10] Iphone 3G reviews, CNET.com, 2010