

<http://dx.doi.org/10.7236/JIWIT.2012.12.2.111>

JIWIT 2012-2-14

# 안드로이드 기반 스마트 폰 응용을 위한 미들웨어 설계

## Design of a Middleware for Android-based Smart Phone Applications

오선진\*

Sun-Jin Oh

**요약** 스마트 폰 기술의 급속한 발전과 더불어 유비쿼터스 컴퓨팅 환경 하에서 기존의 웹사이트나 포털시스템에서 제공하는 서비스를 이용하고자 하는 모바일 사용자들의 요구가 급증하고 있다. 하지만 이러한 서비스들은 데스크 탑 PC나 노트북 등의 환경에 맞춰져 있어 스마트 폰과 같은 오늘날의 모바일 기기에서는 그대로 사용하기 어려운 상황으로 수정으로 인한 많은 비용이나 시스템 보안에 대한 어려움이 큰 실정이다. 본 논문에서는 기존의 웹서버로부터 제공되는 포털 서비스를 모바일 기기에서 원활하게 이용할 수 있도록 중간에 미들웨어 시스템을 두어 최소한의 수정과 기능 설정만으로 포털시스템과 연동할 수 있으면서 인터넷 보안과 스마트 폰 어플의 장점을 살릴 수 있는 안드로이드 기반 스마트 폰 응용을 위한 미들웨어를 설계하고 구현하였다.

**Abstract** With the rapid growth of smart phone technologies, mobile user's demands for using services provided by existing web or portal sites are highly increased. These services, however, are designed for the environment of desktop PCs or laptops in general, so it requires lots of costs for corrections and has difficulties for system protection to use in current mobile devices. In this paper, we design and implement a middleware for android-based smart phone apps to cooperate with a portal system with minimum corrections and function configuration. Our middleware system is working as an interface between an existing portal system and mobile devices to provide harmonious usage and safe system protection.

**Key Words :** Middleware, Android, Smart phone, Mobile App

### 1. 서 론

최근 스마트 폰 기술의 급속한 발전과 더불어 빠른 속도로 스마트 폰이 일반 대중들에게 보급되어 사용되고 있으며 이러한 스마트 폰 사용자들은 단순한 전화 통화 용도로 스마트 폰을 사용할 뿐만 아니라 인터넷 접속과 검색, 음원 파일이나 동영상 등 멀티미디어 데이터의 처리와 재생 및 모바일 게임과 같은 엔터테인먼트 등 다양

한 응용 용도로 스마트 폰을 사용하게 되었다. 따라서 오늘날 모바일 사용자들에게는 다양하고 많은 스마트 응용들이 절실히 요구되는 실정이다. 모바일 사용자들은 3G 망 등 기존의 전화망을 이용한 인터넷 접속을 하기 보다는 보다 편리하고 빠른 전송속도와 저렴한 비용의 모바일 컴퓨팅 환경에서 시간과 장소에 구애 없이 자유롭게 이동하면서 접속할 수 있는 Wi-Fi망을 통한 인터넷 접속과 사용을 선호하는 편이다.<sup>[1]</sup> 이러한 요구에 부응하여

\*중신회원, 세명대학교 정보통신학부  
접수일자 2012년 3월 10일, 수정완료 2012년 4월 12일  
게재확정일자 2012년 4월 13일

Received: 10 March 2012 / Revised: 12 April 2012 /  
Accepted: 13 April 2012

\*\*Corresponding Author: sjoh@semyung.ac.kr

Dept. of Computer & Information Science, Semyung University,  
Korea

다양한 형태의 스마트 폰 응용 어플들이 속속 개발되어 사용되고 있으며 스마트 폰을 통한 인터넷 웹 사이트의 접속 역시 급증하고 있는 실정이다. 하지만 이미 사용 중인 대부분의 인터넷 웹사이트들은 모든 기준이 데스크 탑 PC나 노트북등 큰 화면을 기반으로 연산능력을 갖춘 컴퓨팅 환경에 맞춰 제작이 되어 있으므로 오늘날 스마트 폰 환경에서는 그대로 사용하기 어려운 상황이다. 따라서 각각의 스마트 폰 환경에 맞는 웹 서비스 환경으로 수정이 불가피 하며 또한 스마트 폰에서도 이러한 웹 서비스를 받기 위해서는 스마트 폰을 위한 모바일 앱의 개발과 사용이 필수적이기 때문에 많은 비용과 어려움을 가지고 있다.<sup>[2]</sup> 본 논문에서는 이러한 기존의 웹서버로부터 제공되는 서비스를 스마트 폰에서 원활하게 사용할 수 있도록 기존의 포털 서비스를 제공하는 서버에 스마트 폰의 인터페이스 역할을 할 수 있는 미들웨어를 중간에 두어 스마트 폰의 플랫폼 환경이나 사양이 다양하더라도 최소한의 수정과 기능 설정만으로 쉽게 서버에 접속해서 서비스를 사용할 수 있는 안드로이드 기반 스마트 폰을 위한 미들웨어를 설계하고 이를 서비스할 수 있는 스마트 폰 상의 모바일 앱을 구현하였다. 이렇게 함으로써 다양한 안드로이드 플랫폼 기반의 스마트 폰이나 테블릿 PC 단말에서 미들웨어를 통한 일관된 서비스를 받는 환경을 만들고, 이로 인해 최소한의 미들웨어 구현으로 기존의 포털시스템의 수정 없이 최적의 스마트 폰 웹 서비스 환경을 제공할 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 현재 웹 서비스를 위한 스마트 폰 웹앱 과 모바일 앱 및 미들웨어와 관련된 최근 연구들을 살펴보고, 3장에서는 기존의 포털 시스템에서의 일관된 웹 서비스를 제공하기 위한 미들웨어 시스템 모델을 정의하였고, 4장에서는 본 논문에서 제안한 안드로이드 기반 스마트 폰을 위한 미들웨어 설계와 그것과 연동하는 안드로이드 기반 스마트 폰 상의 모바일 앱의 주요 알고리즘을 알아보고, 5장에서는 그 주요 알고리즘을 구현하여 실행한 결과를 소개하였다. 마지막으로, 6장에서는 향후 연구 과제와 함께 결론을 맺는다.

## II. 관련 연구

스마트 폰의 급속한 발전과 대중화로 인해 네트워크와 디바이스의 한계를 초월한 모바일 앱 개발에 대한 관

심과 연구가 최근 수년 동안 선풍적인 인기를 끌면서, 우리의 인터넷 환경에서 가장 밀집한 웹 개발자들의 고민이 급증하고 있는 실정이다. 그 주요 이유는 자유로운 모바일 접속 환경 때문에 웹앱과 모바일 앱의 경계가 거의 사라졌기 때문이다.<sup>[3]</sup> 또한, 이와 같은 환경에서 새롭게 모바일 앱 개발에 참여하는 웹 개발자들도 급증하였기 때문에 웹과 모바일 단말 장치 모두에서 호환되는 앱을 개발하고자 하는 연구도 활발하게 지속 되고 있다.<sup>[4]</sup> 웹 앱을 구성하기 위한 기본 플랫폼은 HTML, CSS, 그리고 JavaScript나 jQuery로 구성되며 이러한 웹 기반 기술을 토대로 개발한 웹앱을 안드로이드 기반 모바일 단말 장치인 스마트 폰을 통해서 실행 가능한 앱으로 만들기 위한 폰갭<sup>[5]</sup>과 같은 노력들이 주류를 이루고 있다. 다양하고 많은 그룹의 사람들을 만족시키는 앱을 개발하는 일은 불가능에 가까울 정도로 매우 어려운 과제이다. 어떻게 모바일 비전을 현실성 있으며 효과적으로 실행할 수 있고 수많은 사람들이 사용할 수 있게 할 수 있는지에 대한 정답은 바로 웹 기술이 쥐고 있다.<sup>[6]</sup> 즉, HTML5와 모바일 장치들의 결합은 아무리 초보 웹 개발자라 하더라도 사람들의 삶에 도움을 줄 수 있는 모바일 앱을 개발할 수 있는 환경을 제공할 수 있다.<sup>[3]</sup>

구글에서의 안드로이드 운영체제는 오늘날 모바일 컴퓨팅 분야에서 매우 강력한 존재이다. 안드로이드 운영체제는 오픈소스로 누구나에게 무료로 사용할 수 있도록 배포되지만 매우 강력한 기능과 상호 운용성을 가지고 있으며 이를 지원하는 개발 툴들도 강력하고 다양한 기능들을 제공하고 많은 서로 다른 플랫폼 상에서 실행될 수 있게 된다. 오픈 소스인 표준 웹 기술은 유연한 접근성과 편이성을 제공하면서 비용 절감 효과까지 가져오게 되고 이로 인해 수많은 사용자들이 안드로이드 마켓과 같은 앱 서비스 사이트를 통해 쉽게 다운로드 받아 사용하게 되었다. 그러나 이렇게 다양한 안드로이드 기반 플랫폼의 등장으로 웹 개발자들에게는 기존의 웹앱의 형태로 서비스를 고수해야 하는지 아니면 모바일 장치에 맞는 모바일 (네이티브) 앱 형태로 전면 수정을 해야 하는지 많은 혼란을 초래하게 되었다.<sup>[3, 7]</sup>

웹앱은 스마트 폰에 특화된 웹사이트이다. 주로 이 사이트에서 제공하는 콘텐츠들은 평범한 것들이 주류를 이루고 있으며, 웹앱의 특징은 콘텐츠보다는 웹 표준 기술을 사용한 사용자 인터페이스를 적용하는 것이다. 일반적으로 웹앱은 URL을 통해 접근할 수 있으며, 모바일 단

말 기기의 특성을 반영한 최적화가 가능하고 스마트폰에 직접 설치하지 않아도 사용할 수 있다. 반면 모바일 앱은 자바로 작성하며 안드로이드 기반 스마트폰에 설치해서 모바일 단말의 각종 주변장치들과 연동하여 사용할 수 있다. 다음의 표 1은 앞에 서술한 웹앱과 모바일 앱의 특징과 장단점들을 정리한 것이다.<sup>[3]</sup>

**표 1. 웹앱과 모바일 앱의 비교**  
**Table 1. Comparison between Web App & Mobile App.**

분류	장점	단점
웹 앱	<ul style="list-style-type: none"> <li>- 웹 개발 기존 툴을 사용하여 개발</li> <li>- 기존 웹디자인 기술 이용 가능</li> <li>- 웹브라우저로 다양한 모바일 기기에서 접근 사용 가능</li> <li>- 빠른 개발주기</li> <li>- 실시간 버그 수정</li> </ul>	<ul style="list-style-type: none"> <li>- 모바일 기기 하드웨어의 주변장치 사용 불가</li> <li>- 결제기능 제공 불가, 모바일 뱅킹 등 다른 결제 수단 사용 필요</li> <li>- 세밀한 UI 구성에 한계</li> </ul>
모바일 앱	<ul style="list-style-type: none"> <li>- 신용카드 등 편리한 결제기능 제공 가능</li> <li>- 다양한 모바일 기기 하드웨어 주변장치 제어 가능</li> </ul>	<ul style="list-style-type: none"> <li>- 안드로이드 SDK 등 개발 툴과 개발 방법 숙지 필요</li> <li>- 안드로이드 기반 지정된 모바일 단말에서만 작동 가능</li> <li>- 자바 언어 사용</li> <li>- 비교적 긴 개발 주기</li> </ul>

표 1에서 보인바와 같이 모바일 앱 형태의 웹 개발은 특정 모바일 기기 단말을 위한 개발 플랫폼을 완전히 이해하고 그 환경에 맞는 개발 도구와 언어를 이용해서 새로 앱을 작성해야 하고 기존의 이미 사용하고 있는 웹사이트나 포털 사이트를 그대로 이용할 수가 없으므로 그만큼 개발 기간이나 비용이 많이 소요되는 장점보다는 단점이 더 많은 방법이다. 하지만 모바일 기기가 장착하고 있는 카메라, 스피커, GPS 수신기 등 주변장치에 직접 접근하여 사용할 수 있는 장점이 있다. 반면에, 웹앱은 웹 개발자들에게는 기존의 웹사이트를 개발하면서 익숙한 웹 개발 언어나 툴들을 그대로 사용할 수 있어서 용이하게 콘텐츠를 만들 수 있을 뿐만 아니라 이렇게 작성된 웹 응용들은 웹브라우저를 통해 모바일 기기의 기종에 상관없이 접근하여 사용할 수 있으므로 더욱 이용도를 높일 수 있고 기존의 웹사이트나 포털 사이트의 이용이 가능해 지므로 실용적인 측면에서 더욱 효율적이라 할 수 있다. 하지만 모바일 기기가 가지고 있는 하드웨어 장치의 직접 제어나 사용이 제한되는 단점이 있고 이는 모바일

뱅킹과 같은 지금 사용 중인 다른 방법으로 해결이 가능하다.<sup>[4, 5]</sup> 즉, 두 방법 모두 장점과 단점을 가지고 있으나 웹앱이 모바일 웹에 비해 웹 개발자 입장에서는 구현과 응용이 용이하다.

따라서 일반적으로 간단하고 단순한 모바일 응용과 모바일 기기상의 하드웨어를 이용한 응용을 위한 경우에는 모바일 앱 형태의 앱 개발이 훨씬 효율적이지만 기존의 웹사이트나 포털시스템을 연동하여 그것을 이용한 응용이나 웹 접속 서비스에는 웹앱 형태의 앱 개발이 사용자 측면에서나 웹 개발자 측면에서 모두 효율적임을 알 수 있다. 아울러, 시스템의 보안적인 측면에서는 웹앱 형태의 응용의 경우 기존의 웹서버나 포털 서버의 수정이나 직접적인 접속 없이 중간에 미들웨어를 두어 이를 통한 데이터 교환이나 서비스가 보다 안전하고 효과적인 보안을 구현할 수 있어 바람직하다고 본다.<sup>[7]</sup> 본 논문에서는 이러한 장단점을 감안하여 기존의 포털시스템과 같은 웹 서버를 그대로 이용하기 위해 모바일 기기에서 간접적으로 접근할 수 있도록 미들웨어를 중간에 두고 그때 획득된 정보 등을 용이하게 안드로이드 기반 모바일 장치에서 사용할 수 있도록 하는 모바일 앱을 구현하여 연동할 수 있는 안드로이드 기반 미들웨어를 설계하고 구현하였다.

### III. 미들웨어 시스템 모델

본 논문에서 제안한 미들웨어는 기존의 포털시스템의 사용 환경은 전혀 변경하지 않고 단지 데이터의 교환과 공유를 위해 웹상으로 포털시스템에 접속하여 연동하며 이렇게 획득된 중요한 정보는 모바일 기기의 특성에 맞게 그 사이즈나 형태 등을 수정하여 모바일 앱을 통해 전송해 줄 수 있는 중간 인터페이스 역할을 하는 시스템으로 기존의 포털시스템의 보안을 유지하면서 모바일 사용자들의 요구에 맞게 모바일 단말에서 웹서비스를 제공할 수 있는 기능을 수행한다. 본 논문에서 제안하는 시스템 환경은 다음의 그림 1과 같다.

그림에서 보인바와 같이 기존의 포털시스템은 내부 인트라 망을 통해 멤버 사용자들에게 예전처럼 서비스가 제공되고 있으며, 외부 사용자들의 접근을 인증과정 등을 통해 차단하고 있어 보안이 유지되고 있다. 추가로 학생 등 외부 사용자들에게 제한적인 포털사이트의 정보를

제공하기 위해 중간에 미들웨어를 두고 외부 사용자들은 모두 이 미들웨어를 거쳐 제한적인 포털 정보만을 검색할 수 있도록 하였고, 이들 외부 사용자들은 Wifi망이나 3G 망을 통해 인터넷으로 접속하는 안드로이드 기반 모바일 기기를 사용하고 있으며 이러한 환경의 서비스를 원활하게 제공할 수 있도록 미들웨어에서 보안과 적절한 형태의 정보 가공 과정을 거쳐 서비스를 하도록 구성되었다.

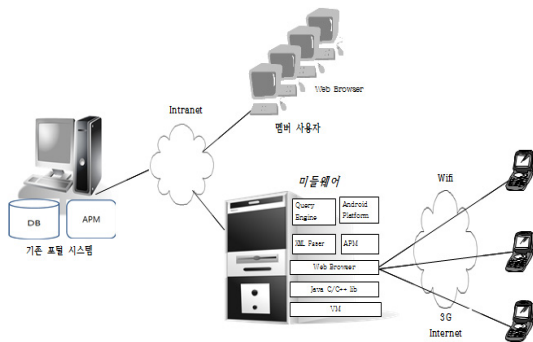


그림 1. 미들웨어 시스템 모델  
Fig. 1. Middleware System Model

스마트 폰 등의 모바일 기기로부터 발생한 외부 사용자의 정보 요청은 미들웨어를 통해 접수되며 이때 필요한 데이터는 일정한 인증 작업을 거쳐 데이터베이스 쿼리 엔진을 가동하여 포털시스템에 요청되고 이어 포털시스템의 데이터베이스가 검색되어 요청되었던 데이터는 미들웨어로 반환되게 된다. 그리고 나서 미들웨어에서는 이 데이터에 대한 모바일 기기에 맞는 데이터 형식으로 전환되고 XML 파서를 통해 파싱되어 안드로이드 플랫폼에 맞게 전환되어 다시 Wifi망이나 3G 망을 통해 보내짐으로써 모바일 사용자의 요청에 대한 서비스가 완료된다. 이때 모바일 기기 상에 보여 지는 요청의 결과는 모바일 앱에 의해 작성된 레이아웃에 근거하여 모바일 기기창을 통해 디스플레이 되게 된다.

미들웨어 시스템 모델에서의 모바일 사용자를 위한 모바일 앱 개발환경은 Windows 7 운영체제의 Desktop PC에 JAVA JDK 1.6 update 23과 Eclipse 3.6.1 툴, ADT 플러그인, Android SDK 2.3 그리고 AVD 생성 및 옵션 등을 설치하여 구성하였다. 한편, 미들웨어 서버 시스템의 개발환경과 주요 시스템의 사양은 표 2와 같다.

표 2. 미들웨어 시스템 개발 환경

Table 2. Development Environment of a Middleware System

품명	사양
CPU	AMD athlon II X2 250 Processor
메인보드	NVIDIA GeForce 6150/6100
그래픽 카드	NVIDIA GeForce 7025 + nForce 630a
사운드 카드	Realtek HD Audio(ALC662)
랜 카드	NVIDIA Ethernet
OS	Windows7 (IIS, FTP설치)
Database	Oracle 10g, mySql 5.1
개발언어	Jsp, XML
서버 툴	Apache Tomcat 7.0.26 JDK 1.7.0_03 Eclipse Helios EE 32bit JDBC(Connector/J 5.1.18)
개발 툴	Editplus3

#### IV. 안드로이드기반 스마트 폰 앱 설계

이 장에서는 본 논문에서 제안한 미들웨어를 둔 모바일 환경에서의 안드로이드 기반 스마트 폰 앱을 설계하고 그 주요 알고리즘을 제안한다. 본 논문에서 구현한 스마트 폰 응용은 이미 포털시스템을 통해 서비스 중인 세명대학교 정보통신학부의 주요 학사 일정, 교수진 소개, 공지사항, 게시판, 도서관 도서목록 검색, 도서관 자리 예약 조회, 캠퍼스 안내, 그리고 학과 주요 정보들을 기존의 포털시스템으로부터 정보를 획득하여 미들웨어를 통해 안드로이드 기반 모바일 기기인 삼성 갤럭시 폰을 통하여 접속해서 포털시스템 정보를 사용할 수 있는 모바일 앱을 구현하였다. 이때 기존의 포털시스템은 단지 데이터베이스에 저장되어 있는 정보를 미들웨어의 요청에 따라 제공해 줄 뿐 인증되지 않은 사용자의 어떠한 수정이나 변경을 허용하지 않으며 또한 인증되지 않은 직접적인 모바일 기기로부터의 접근 또한 허용하지 않는다. 이렇게 함으로써 인트라넷을 구성하고 있는 포털시스템의 외부로부터의 공격이나 보안 문제를 해결하였다.

미들웨어는 각 모바일 사용자들의 모바일 기기를 통한 적당한 인증절차에 의한 접근을 허용하는 접속 접점을 가지고 있으며, 모바일 사용자의 질의에 의해 요청된 서비스를 수행하여 필요로 하는 정보를 포털시스템으로부터 획득하여 모바일 기기 환경에 맞게 가공하여 모바일

일 사용자의 단말 창을 통해 결과를 서비스하게 된다. 다음은 포털시스템에 있는 학부 공지사항과 게시판의 정보를 요청하는 데이터베이스 질의를 처리하는 알고리즘을 보여준다.

```
private void write() { // DB에 쿼리를 넣는 부분
    try {
        HttpClient client = new DefaultHttpClient();
        String postURL = "http://211.114.147.131/boardfree/board_write_insert.jsp";
        HttpPost post = new HttpPost(postURL); // 웹페이지상의 DB post주소
        List<BasicNameValuePair> params = new
            ArrayList<BasicNameValuePair>();
        params.add(new BasicNameValuePair("dbname",
            edt1.getText().toString()));
        params.add(new BasicNameValuePair("dbsubject",
            edt2.getText().toString()));
        params.add(new BasicNameValuePair("dbmemo",
            edt3.getText().toString()));
        // 보낸 데이터 리스트로 묶어 editText의 DB이름으로 맞춤
        UriEncodedFormEntity ent = new
            UriEncodedFormEntity(params, "euc-kr");
        post.setEntity(ent);
        HttpResponse responsePost = client.execute(post); // 클라이언트에서 질의
        HttpEntity reEntity = responsePost.getEntity();

        if (reEntity != null) {
            Log.w("RESPONSE", EntityUtils.toString(reEntity));
        }
    } catch (Exception e) {
    }
}
```

그림 2. 데이터베이스 질의 처리 알고리즘  
Fig. 2. DB Query Processing Algorithm

그림 2에 보인바와 같이 포털시스템의 공지사항이나 게시판 정보는 데이터베이스에 저장되어 있고 이를 미들웨어를 통해 질의하는 알고리즘으로 이렇게 획득된 정보는 jsp로 페이지를 생성하여 그 내용을 모바일 기기에 알맞게 XML 페이지로 변환시킨 후 파싱과정을 거쳐 안드로이드 기반 모바일 기기에 보내지게 된다. 또한 게시판의 새 글쓰기의 경우 모바일 기기 상에서 editText를 만들고 그곳에 쓰인 Text를 List로 묶어 euc-kr 포맷으로 변환한 후 http posting으로 미들웨어 상의 jsp의 form tag의 action으로 보내어서 최종 데이터베이스 상에서 처리하도록 한다. 이 알고리즘에서 데이터베이스에 질의 처리를 위해 우선 http 통신을 위한 클라이언트 객체를 생성한 후 데이터를 웹 페이지로 보내기 위한 URL을 String 변수 하나로 생성하여 넣어준다. 이때 URL은 웹 페이지에서 form 태그의 action에서 넘어가는 부분의 URL을 넣어주는데 여기서 웹 페이지가 데이터베이스 삽

입 처리부분을 담당하는 board1\_write\_insert.jsp라는 파일로 데이터를 보내게 하기 위해 URL을 넣어주게 된다. 그리고 데이터를 editText로 가져와서 리스트로 데이터를 묶은 다음에 웹 페이지 형식에 맞게 euc-kr 포맷으로 인코딩 시킨 후 클라이언트 객체를 이용하여 데이터베이스 질의를 전송하게 된다.

```
try { //XML변환페이지, 파서 팩토리 생성, 파서생성
    URL url = new URL("http://211.114.147.131/boardfree/board_xml.jsp");
    XmlPullParserFactory parserfac = XmlPullParserFactory.newInstance();
    XmlPullParser parser = parserfac.newPullParser();
    parser.setInput(url.openStream(), "euc-kr"); //euc-kr로 인코딩
    int parserEvent = parser.getEventType();
    String b_num = ""
    String b_title = "" -중략-
    while (parserEvent != XmlPullParser.END_DOCUMENT) {
        switch (parserEvent) {
            case XmlPullParser.START_TAG:
                tag = parser.getName();
                if (tag.equals("sun")) { // 시작 태그일 경우
                    b_num = parser.getAttributeValue(null, "aid");
                    b_title = parser.getAttributeValue(null, "subject");
                    -중략- // XML 상 데이터 변수에 저장
                } break;
            case XmlPullParser.END_TAG:
                if (tag.equals("sun")) { // 종료 태그
                    BoardFree nb = new BoardFree(b_num, b_title, b_date,
                        b_text, b_name);
                    boardfree.add(nb); // 리스트로 묶음
                } tag = "nothing" break;
            } parserEvent = parser.next();
        }
    } catch (Exception e) { // TODO Auto-generated catch block
        e.printStackTrace();
    }
    setListAdapter(new CustomRow(this));
}
```

그림 3. XML 파싱 알고리즘  
Fig. 3. XML Parsing Algorithm

그림 3은 데이터베이스로부터 질의를 통해 획득한 데이터를 모바일 기기에 가져와서 보여주기 위해 XML 형식에 맞게 파싱을 하는 알고리즘의 일부를 보여준다. XML 파싱을 하기 위해서는 우선 파서 팩토리를 생성하고 여기서 파서 객체를 얻는다. getEventType으로 현재 XML에 있는 데이터를 검색하고 next를 이용하여 다음 데이터를 검색해 나가면서 최초 시작 태그에서 부터 끝나는 태그까지 각 데이터마다 getName을 이용하여 tag나 text 내용을 검사하여 원하는 정보를 추출한다. 이 알고리즘에서는 XML의 데이터의 내용을 추출하여 각각의 변수에다가 넣어놓고 루프를 돌리면서 더 이상 데이터 내용이 없을 때까지 계속하게 된다.

```

<%@ page contentType="text/html;charset=euc-kr" %>
<%@ page language="java" import="java.sql.*, javax.servlet.http.*,java.util.*" %>
<%
String action = request.getParameter("action"); // 요청 접수
try{
    Class.forName("oracle.jdbc.driver.OracleDriver");
    } catch(ClassNotFoundException e) {
    out.println(e.getMessage()); } // 포털의 오라클 DB
String url="jdbc:oracle:thin:@localhost:1521:ORCL";
String id="system"; // 포털 시스템 아이디
String pass="mmcl23"; // 포털 시스템 비밀번호
Connection connection= DriverManager.getConnection(url,id,pass);
Vector<String> isbn=new Vector<String>(); // DB 연결시도
Vector<String> title=new Vector<String>();
Vector<String> author = new Vector<String>();
Vector<String> publisher = new Vector<String>();

try{
    statement = connection.createStatement();
    String query = null;
    -중략- // DB 질의 구성
query = "select * from book_info where "+ item+" like '%" +String+"%' ";
result=statement.executeQuery(query); // DB 질의 실행
while(result.next())
{
    isbn.add(result.getString("isbn"));
    title.add(result.getString("title")); // DB 질의 항목
    -중략-
}
}
%>
<html> <meta name="viewport" content="width=device-width;
user-scalable=no, target-densitydpi=medium-dpi">
<head> <title>도서 정보 리스트</title> // 질의 결과 보고서
<script>
function bookSearch(){
    var searchString = document.form1.search.value;
    -중략-
    document.form1.submit();
} </script> </head>
<table align="center" border="1" cellpadding="5" width="90%">
<tr> <td bgcolor="white">도서번호</td>
<td bgcolor="white">책제목</td>
<tr>
<td><%=isbn.get(i)%></td>
<td><%=title.get(i)%></td>
-중략-
</html>

```

그림 4. 도서관 도서목록 검색 알고리즘  
Fig. 4. Library Book Search Algorithm

그림 4는 미들웨어 시스템에서 모바일 사용자로부터 요청 받은 도서관 도서목록 검색요청에 따라 인터넷을 통해 포털시스템에 도서목록 검색 정보 요청을 하는 알고리즘을 보여준다. 그림에 보인바와 같이 jsp를 이용해서 인터넷 내의 포털시스템에 접속을 시도하고 데이터베이스에 저장되어 있는 도서관 도서목록 정보를 검색하여 이를 일정 형식의 결과 보고서 양식에 맞게 구성하여 미들웨어로 가지고 온 다음, 사용자의 모바일 기기에 맞게 XML 파싱 과정을 거쳐 보여주게 된다. 이렇게 함으로서 포털시스템의 보안은 안전하게 유지한 채 직접 모바일 사용자들의 포털시스템 접근 없이 미들웨어를 통해 포털시스템이 서비스하고 있는 정보를 이용하게 된다.

이 외에 학사일정, 식단 목록 조회, 도서관 자리 예약 조회 등의 서비스들을 동일한 방법으로 미들웨어를 통해 모바일 사용자들에게 서비스 할 수 있다.

## V. 구현 결과 및 고찰

이 장에서는 본 논문에서 제안한 안드로이드 기반 스마트 폰을 위한 미들웨어를 이용하여 스마트 폰 상에서 구현한 모바일 앱을 사용하여 실행한 결과를 보여준다.



그림 5. 모바일 앱의 실행 결과와 공지사항 화면  
Fig. 5. Execution Results of a Mobile App

그림 5에 보인 바와 같이 본 논문에서 제안한 미들웨어를 이용한 포털 서비스를 사용하는 모바일 앱을 실행시킨 결과를 보여 준다. 왼쪽의 그림은 앱을 처음 실행했을 때의 실행 결과로 여기서 학부 소개와 교수진 소개 메뉴를 선택할 수 있으며 이 명령에 의해 포털의 정보를 미들웨어를 통해 가져와 보여주게 된다. 또한 가운데와 오른쪽 그림은 주 메뉴의 두 번째 탭인 “보”를 선택했을 때의 공지사항과 게시판을 검색하는 화면과 공지 사항의 내용을 미들웨어가 포털시스템으로부터 가져와 모바일 기기를 통해 보여주는 화면이다. 이렇게 미들웨어를 통해 포털의 정보를 가져와 서비스하는 방법과는 달리, 반대로 모바일 기기를 통해 사용자에게 의해 입력된 내용을 미들웨어를 통해 포털시스템의 데이터베이스에 저장하여 포털에서 사용할 수 있도록 하는 서비스 역시 본 논문에서 설계한 미들웨어를 통해 제공된다. 그림 6은 스마트폰 상에서 모바일 앱을 통해 게시판에 글쓰기를 하는 실행화면과 도서관 도서 목록을 조회하고 검색한 결과를 보여주는 실행 화면이다. 여기서 모바일 사용자는 직접 포털시스템의 DB를 접근할 수 없고 다만 미들웨어를 통해 필요한 데이터를 질의를 통해 요청할 수 있다. 기존의 포털시스템은 기업이나 단체 내부의 인터넷 포털을 사용하여 인증된 사용자와 시스템만의 접근이 허용되고,

방화벽으로 보안이 철저히 이루어지므로 단지 인증된 미들웨어를 통해 질의에 의해 데이터를 요청하게 되고 이때 미들웨어는 포털시스템의 보안 프로그램에 의해 접근이 제한적으로 허용된 자료들만을 질의를 통해 간접적으로 접근 사용하게 함으로써 포털시스템의 보안을 보장하게 된다. 또한 미들웨어를 도입함으로써 이 기종의 모바일 단말들의 정형화된 접근 방법을 제공하게 된다.



그림 6. 게시판과 도서목록 검색 실행 결과  
Fig. 6. Execution Results of Bulletin Board and Library Book Search

## VI. 결론

최근 스마트 폰 기술의 급속한 발전과 더불어 유비쿼터스 컴퓨팅 환경 하에서 기존의 포털시스템에서 제공하는 서비스를 이용하고자 하는 모바일 사용자들의 요구가 급증하고 있다. 하지만 이런 서비스들은 스마트 폰과 같은 모바일 기기에서는 그대로 사용하기 어려운 상황이다. 따라서 수정으로 인한 많은 노력과 비용이 요구되며 포털시스템의 보안에도 매우 위협적이다.

본 논문에서는 기존의 웹서버로부터 제공되는 포털 서비스를 모바일 기기에서 원활하게 이용할 수 있도록 중간에 미들웨어를 두어 인터넷 상의 보안을 유지한 채 다양한 모바일 기기 상에서 서비스를 할 수 있는 안드로이드 기반 스마트 폰 응용을 위한 미들웨어를 설계하고 구현하였다. 또한 구현한 미들웨어를 이용하여 포털 시스템에 접속하여 서비스 받을 수 있는 안드로이드 기반 모바일 앱도 구현하였으며 이를 이용하여 포털시스템에 접속하여 학사일정, 공지사항과 게시판 및 도서관 도서목록검색과 도서관 자리예약 조회 기능을 실행하였고, 포털시스템에 직접 외부 모바일 사용자들의 접근 없이 보안을 유지한 채 미들웨어를 통해 원활하게 웹 서비스가 이루어짐을 확인할 수 있었다. 향후 연구 과제로는 안드로이드뿐만 아니라 아이폰 기반 어플을 구현하고 미들

웨어의 기능을 컴포넌트 화하는 작업이다.

## 참고 문헌

- [1] Gabhart, K., Gordon, J., "Wireless Web Services with J2ME," WebServices Journal, Volume 2, Issue 2, 2002.
- [2] Carpa L. et al., "Middleware for Mobile Computing," In Proceedings of the 8th Workshop on Hot Topics in Operating Systems, Germany, 2001.
- [3] J. Stark, "Building Android Apps with HTML, CSS, and JavaScript", O'Reilly Media Inc., 2010.
- [4] Efstratiou, C., et al., "Architectural Requirements for the Effective Support of Adaptive Mobile Applications," in Proceedings of IFIP/ACM International Conference on Distributed Systems Platforms and Open Distributed Processing (Middleware-2000, New York, USA, 2000.
- [5] Coulouris, G., Dollimore, J., Kindberg, T., "Distributed Systems Concepts and Design", Addison-Wesley, 2001.
- [6] Gu, T., et al., "A Middleware for Building Context-Aware Mobile Services," In Proceedings of IEEE Vehicular Technology Conference (VTC-Spring 2004), Milan, Italy, 2004.
- [7] Lohse M, Repplinger M, Slusallek P., "Session Sharing as Middleware Service for Distributed Applications, Interactive Multimedia on Next Generation Networks," Proceedings of First International Workshop on Multimedia Interactive Protocols and Systems, Naples, Italy, 2003.

## 저자 소개

### 오 선 진(중신회원)



- 제 6권 제2호 참조
- 현재 세명대학교 정보통신학부 교수  
<주관심분야 : 스마트 응용, 그린IT, MANETs, 모바일컴퓨팅, USN 등>