

안드로이드 타블릿 기반 통합 스케줄러 개발

박동연*, 최진아*, 유희수*, 이정준*

*한국산업기술대학교 컴퓨터공학과

e-mail:{kungwoan, cjal114, yooheesoo, jjlee}@kpu.ac.kr

Development of Integrated Scheduler based on Android Tablet

Dong-Yeon Park*, Jin-Ah Choi*, Hee-Soo Yoo*, Jeong-Joon Lee*

*Dept of Computer Engineering, Korea Polytechnic University

요 약

스마트폰 시장의 성장 그리고 스마트폰 시장 내의 안드로이드 시장이 급성장함에 따라 안드로이드 기반의 다양한 앱에 대한 수요가 많아지고 있다. 그중에서도 안드로이드 앱에 대한 수요는 기하급수적으로 증가하고 있다. 또한 스마트폰뿐만 아니라 타블릿PC 시장도 성장하면서 타블릿PC 앱에 대한 수요도 증가하고 있다. 하지만 타블릿PC 전용 앱의 개발은 아직 미흡하다. 본 논문에서는 안드로이드 타블릿PC 기반으로 스케줄링과 가계부를 지원하는 시스템의 설계 및 구현에 대한 내용을 기술한다. 또한 Android의 강점인 Google 계정과의 Synchronization을 통해 PC와 타블릿PC에서의 스케줄 관리를 할 수 있으며 자체적으로 만든 서버를 통하여 공공일정을 자신이 원하는 항목을 선택하여 다운로드하여 관리 할 수 있다. 가계부는 수입과 지출을 관리하고 그래프를 통하여 자신이 어느 정도로 사용을 하였는가를 확인할 수 있다.

1. 서론

안드로이드 시장이 급성장하고 있다. 스마트 시장에서의 안드로이드 점유율은 (그림 1)과 같이 2009년에 1.8%, 2010년 2분기 17.2%, 2011년에는 심비안을 제치고 36%를 차지하면서 스마트폰 시장에서 1위를 차지하는 기업을 토해냈다.

Worldwide Smartphone Sales to End Users by Operating System in 1Q11 (Thousands of Units)

Company	1Q11 Units	1Q11 Market Share (%)	1Q10 Units	1Q10 Market Share (%)
Android	36,267.8	36.0	5,226.6	9.6
Symbian	27,598.5	27.4	24,067.7	44.2
iOS	16,883.2	16.8	8,359.7	15.3
Research In Motion	13,004.0	12.9	10,752.5	19.7
Microsoft	3,658.7	3.6	3,696.2	6.8
Other OS	3,357.2	3.3	2,402.9	4.4
Total	100,769.3	100.054	505.5	100.0

Source: Gartner (May 2011)

(그림 1) 세계 스마트폰 점유율

타블릿PC 시장 또한 상당한 변화가 있었다. 2010년 4분기 970만대가 출하되어 3분기보다 출하량이 2배 이상 증가하였다. (그림 2)는 2011년 4분기 스마트폰 점유 OS의 비율을 나타낸다.

Global Tablet OS Shipments (Millions of Units)	Q3 '10	Q4 '10	2010
Apple iOS	4.2	7.3	14.8
Android	0.1	2.1	2.3
Others	0.1	0.3	0.5
Total	4.4	9.7	17.6

Global Tablet OS Marketshare %	Q3 '10	Q4 '10	2010
Apple iOS	95.5%	75.3%	84.1%
Android	2.3%	21.6%	13.1%
Others	2.3%	3.1%	2.8%
Total	100.0%	100.0%	100.0%

(그림 2) 스마트폰 점유 OS

이 가운데 안드로이드 기기들의 수많은 출시로 인해 3분기 2.3%였던 안드로이드 점유율이 21.6%로 상승하여 애플의 독점을 막았다. 또한 이 점유율은 앞으로 계속 증가할 것으로 보인다. 하지만 타블릿PC의 수요에도 불구하고 전용 앱의 수는 거의 없다. 그리하여 우리는 타블릿PC 전용 앱을 만들어 보았다.

2장에서 기존 타블릿PC 앱의 현황과 특징을 설명하고 3장에서는 요구사항들을 나열하고 4장에서는 요구사항들을 만족시키는 설계와 구현결과를 기술하고 6장에서는 결론을 맺는다.

2. 관련 연구

2.1 타블릿PC의 특징

타블릿PC와 스마트폰은 같은 안드로이드 OS를 탑재하지만 다른 특징을 가진다. 바로 화면(Display)의 크기인데 Display가 크다는 것은 많은 이점이 있다. 먼저 많은 정보를 Display에 보여줄 수 있다. 이는 앱의 화면 UI에서부터의 차이점을 보이게 된다. 스마트폰보다 화면 배치에

따른 Activity의 설계가 바뀌게 되고 보다 많은 정보를 보여줌으로서 사용자(User)에게 편의성을 제공한다. 또한 올해 Google이 안드로이드 기반으로 태블릿PC전용 OS인 허니콤을 출시하면서 아이패드와의 차별성을 두기로 하였다. 허니콤이 탑재가 되면서 인터페이스가 개선되었다.

2.2 태블릿PC전용 앱

기존 태블릿PC 앱은 스마트폰용을 그대로 옮겨와서 사용하고 있다. 그러면서 문제점들이 발생을 하게 되는데 먼저 화면구성이 스마트폰에 맞춰졌기 때문에 화면 배치나 그래픽이 태블릿PC 사용자에게 이질감을 줄 수 있다. 개발 당시 버튼위치나 크기 화면 구조를 비율로 하지 않았을 경우 태블릿PC에서 실행 시 전혀 안 맞는 경우가 많다. 화면이 작아지거나 깨지거나 실행이 되지 않는 문제점이 있다. 아이패드 같은 경우 패드용과 폰용을 따로 판매 및 개발함으로써 이 문제를 해결했다.

2.3 기존의 달력 및 스케줄러

대부분의 스케줄러는 Google계정과의 동기화를 필수적으로 하고 있다. Google은 스케줄러에 중요한 한국의 행사 일정을 함께 보여준다. 화면구성은 일별, 주별, 달별로 나뉘어서 사용자에게 보여주는 형식을 띠고 있다.

3. 연구동기

스마트폰 스마트폰용 달력이나 스케줄러들을 살펴보면 다음과 같은 문제점이 있다. 먼저 화면이 작기 때문에 한번에 보여주는 정보의 양이 적다. 더군다나 작은 화면에 화면을 배치하니 매회 화면전환과 작은 버튼들은 사용하기 불편함이 있었다. 그리고 양력만 표시가 되고 음력은 표시가 되지 않았다. 한국 기념일도 중요한 행사만 표시되고 자잘한 행사들은 표시가 안 되었다. 이러한 문제점을 통해 본 프로젝트의 필요성을 느끼게 되었다.

스케줄러의 문제점을 해결하기 위해 다음과 같은 해결책을 제시하게 되었다. 동기화를 통해 기기에 저장되어있는 사용자의 Google 계정에 태블릿과 동시에 스케줄관리가 되어야 한다. 또한 동시에 사설 스케줄 서버를 운영하여 이용자들이 좀 더 많은 정보를 다운받아서 활용할 수 있게 했다. 다운 시 원하는 콘텐츠만 선택 하여 해당내용만 추려내게 하였다. 태블릿PC가 가지고 있는 화면크기를 이용하여 사용자에게 효율적인 인터페이스를 제공하고 화면에 많은 정보를 제공할 수 있도록 했다. 화면은 주 타깃이 될 기기의 화면크기인 7인치에 최적화되게 구성하였다. 또한 부가적인 요소로 가계부 기능을 추가하여 여성들이 사용할 수 있게 만들며 그래프를 통해 분석가능하게 한다.

본 연구에서는 위와 같은 요구사항을 충족하는 시스템을 설계 개발하고자 한다.

4. 설계 및 구현

4.1 개발 환경

개발 환경은 크게 클라이언트-서버로 분류된다. 개발 툴들과 개발 H/W를 분류하면 <표 1>과 같다.

<표 1> 개발환경

	H/W	개발 툴
서버	AMD Athlon(tm)64 x 2 Dual PC서버	ASP Microsoft Server 2000
클라이언트	Identity Tab	eclipse IDE for JavaEE Developers Android 2.2

4.2 시스템 구조

(그림 3)은 안드로이드 태블릿 기반 다기능 스케줄러에 서버와 태블릿PC, 개인PC간의 연동장면을 보여준다.



(그림 3) 시스템 구조

4.3 시스템 설계의 특징

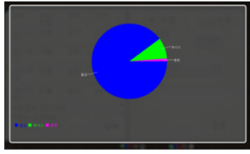
서버는 기존의 웹 환경에서 제공되는 서비스를 기반으로 ASP.NET 2.0기반으로 작성되어 있으며 태블릿PC에 데이터를 전송해 주기 위해서 XML을 전송을 하고 클라이언트는 pullparser를 이용하여 데이터를 파싱을 하게 된다. 서버 내에 있는 XML에서는 주로 클라이언트 DB에 맞춘 데이터로 구성을 하여 클라이언트가 손쉽게 파싱할 수 있도록 설계를 하였다. 서버에서 데이터를 받아오면 화면에서는 테이블에 저장되고 리스트에 뿌려지게 된다. (그림 4)는 서버 내 XML 데이터 파싱 과정을 보여준다.

```
<?xml version="1.0" encoding="utf-8" ?>
- <event>
  <name>FINAL TEST</name>
  <startdate>2011.06.27</startdate>
  <enddate>2011.06.27</enddate>
  <starttime>pm 16:30</starttime>
  <endtime>pm 18:30</endtime>
  <location>E-517</location>
  <description>final test ssession 4</description>
  <name>FINAL TEST2</name>
  <startdate>2011.06.27</startdate>
  <enddate>2011.06.27</enddate>
  <starttime>pm 17:30</starttime>
  <endtime>pm 19:30</endtime>
  <location>E-518</location>
  <description>final test ssession 3</description>
</event>
```



(그림 4) 서버 내 XML 데이터 파싱 과정 가계부에서의 그래프 표시를 위해 achartengine를 사용하였다. achartengine은 오픈소스로써 안드로이드 내에서 사

용할 수 있는 라이브러리를 제공한다. 여기서 원형 그래프를 사용해서 품목별 분석을 할 수 있게 만들었다. (그림 5)는 구현된 그래프이다.



(그림 5) 그래프 표현

4.4 동기화 설계방식 및 구현

내부 캘린더와의 동기화를 위해 안드로이드에서 기본으로 제공하는 Content Values 클래스와 Content Resolver 자료형을 사용하였다. 내부 캘린더를 Content Resolver를 사용하여 클라이언트와 간접적으로 연결할 수 있게 하였다. Key와 Value를 세트로 하여 여러 개의 데이터 세트를 입력시킬 수 있는 Content Values를 이용하면 손쉽게 캘린더에 필요한 데이터를 입력할 수 있다. 이를 위해 안드로이드에서 제공하는 라이브러리를 추가하였다. 내부 캘린더의 주소를 URI로 나타내고 URI를 파싱해 content Resolver로 content Values에 입력한 데이터와 함께 Insert시키는 방식으로 내부 캘린더에 본 프로그램과 동시에 입력될 수 있게 하였다. 아래의 (그림 6)은 내부 캘린더와 본 프로그램과의 동기화 구현내용이다.

```
Uri.Builder eventUri = Uri.parse("content://com.a:
ContentResolver contentResolver = context.getCont

ContentValues cEvent = new ContentValues();
cEvent.put("calendar_id", 1);
cEvent.put("title", title);
cEvent.put("dtstart", startTime);
cEvent.put("dtend", endTime);
cEvent.put("hasAlarm", 1);
cEvent.put("eventLocation", location);
cEvent.put("description", description);
contentResolver.insert(eventUri.build(), cEvent);
```

```
Uri eventUri = Uri.parse("content://com.android.calendar/events");
int id = Integer.parseInt(event_id);
Uri uri = ContentUris.withAppendedId(eventUri, id);

contentResolver.update(uri, cEvent, null, null);

Uri eventUri = Uri.parse("content://com.android.calendar/events");
int id = Integer.parseInt(event_id);
Uri uri = ContentUris.withAppendedId(eventUri, id);

contentResolver.delete(uri, null, null);
```

(그림 6) 동기화에 사용된 함수내용

5. 구현결과

5.1 동기화 및 서버 연동

아래 (그림 7)은 스케줄 입력 화면이다.



(그림 7) 스케줄 입력

인터넷이 연결되어 있다면 입력, 수정, 삭제를 함과 동시에 Google 계정에 동기화를 시켜 같은 내용을 저장하게 된다. 태블릿PC에서 하는 작업은 동기화된 계정에도 동시에 수행된다.

사실 서버와의 연동은 Download버튼을 통해 사용할 수 있다. 버튼을 누르게 되면 콘텐츠 선택 창이 뜨고 그 중 원하는 것을 선택할 수 있다. 그 후 사실 서버를 통해 서버에 저장된 내용을 파싱하여 list에 저장하고 원하는 내용만 화면에 뿌려준다.

5.2 효율적 인터페이스

아래 (그림 8)은 최초 화면 및 스케줄러 화면이다. 메인 화면에서 사용자는 가계부와 스케줄러를 선택한다.



(그림 8)메인화면 및 스케줄러 화면

스케줄러 화면을 보면 달력과 당일의 스케줄 등이 보여 태블릿PC의 넓은 화면을 활용해 많은 정보를 한눈에 볼 수 있게 한 것을 알 수 있다.

5.3 가계부

가계부를 선택하면 수입과 지출항목을 볼 수 있고 그래프 버튼을 클릭 시 사용빈도를 그래프로 표현해준다. 아래 (그림 9)는 가계부 화면이다.



(그림 9)가계부와 수입, 지출그래프

6. 결론

본 연구를 통하여 태블릿PC용 앱의 필요성을 제시할 수 있다. 또한 Google의 동기화 기능을 통하여 스케줄뿐만 아니라 나아가 더욱더 다양한 기능을 사용할 수 있으리라고 본다. 그리고 achartengine처럼 안드로이드에 없는 API같은 경우 다른 개발자가 만들어서 사용할 수도 있기 때문에 유용한 함수를 꼭 안드로이드 API에서 찾을 필요가 없다. 그리고 우리가 만든 사실 서버를 통하여 광고의

효과를 낼 수도 있다. 행사나 이벤트 같은 항목도 넣어서 사용을 하면 다양하게 활용을 할 수가 있을 것이다. 이제 곧 있으면 허니콤을 탑재한 타블릿이 나오는데 현재 Google에서 15개 정도의 허니콤용 앱이 개발됐다고 하는데 스마트폰과 타블릿PC를 같이 보는 것이 아닌 독립적인 것으로 보고 접근해야 하지 않을까 싶다.

참고문헌

- [1] 김상형 "안드로이드 프로그래밍 정복" 한빛미디어
- [2] 케이시 시에라 "Head First Java" 한빛미디어
- [3] <http://developer.android.com> "안드로이드 API 정보"
- [4] 리토 마이어 "프로페셔널 안드로이드 애플리케이션 개발" Jpub
- [5] Gartner "Worldwide Smartphone Sales to End User by Operating System"