

http://dx.doi.org/10.7236/JIWIT.2012.12.3.141

JIWIT 2012-3-18

안드로이드 기반 모바일 스마트 알람

Android Based Mobile Smart Alarm

오진우*, 안병구**

Jinwoo Oh, Beongku An

요 약 안드로이드는 구글(Google)과 오픈 핸드셋 얼라이언스(Open Handset Alliance)가 만들어낸 모바일폰용 오픈 소스 소프트웨어 툴킷(toolkit)이다. 스마트폰 사용자의 증가로 각종 애플리케이션 콘텐츠의 양이 점점 증가하고, 특히 사용자의 편의를 위한, 사용자 중심의 애플리케이션이 많이 개발되고 있다. 본 논문에서는 구글의 안드로이드 플랫폼을 기반으로 한 새로운 애플리케이션 콘텐츠 스마트 알람을 개발한다. 개발된 스마트 알람의 특징은 다음과 같다. 첫째, 사용자의 하루일정과 알람을 간편하게 입력하고 그에 따라 일정을 확인 할 수 있다. 둘째, 기존의 일정관리 애플리케이션과 다르게 사용자는 자신의 일정을 알람 방식뿐 아니라 타이머 형식으로도 입력하여 여러 개의 순차적인 업무처리를 잊지 않고 해결 할 수 있다. 성능평가는 AVD (Android Virture Device)를 사용하여 시뮬레이션 하였다.

Abstract Android that was made by Google and Open Handset Alliance is the open source software toolkit for mobile phone. Nowadays, Smart phone users have been rapidly increasing, by that reason a lot of application contents have been released and especially contents which focus on user convenience have been developed. In this paper, we develop a new application content smart alarm based on Google Android platform. The features of the developed content are as follows. First, users set their day's schedules. So they can notify easily what they have to do at the moment. Second, when users set their alarms not to forget several sequential tasks, they can set their schedules by not only alarm method but also timer method. In the performance evaluation, we show the results of simulation using AVD(Android Virture Device).

Key Words : Android, Mobile system, Smart alarm

1. 서 론

스마트폰은 기존의 휴대전화 기능에 더불어 인터넷 브라우징, 이메일, 문서작성, 일정관리, 음악, 동영상감상, 게임 등 PC에서 하던 대부분의 작업을 수행할 수 있는데, 이러한 기능은 모두 스마트폰에 탑재된 GPOS(General Purpose Operating System)에 기인한다. 따라서 스마트폰은 단순히 부가기능이 확장된 휴대전화로 볼 것이 아니라 음성 통화 기능이 포함된 휴대용 컴퓨터로 이해하

는 것이 더 옳은 해석이다. 안드로이드가 기존의 GPOS인 심비안이나 윈도우즈 모바일과 차별되는 가장 큰 특징은 소스 코드를 공개하여 누구나 안드로이드를 이용하여 소프트웨어와 임베디드 기기를 만들 수 있는 개방형 플랫폼이라는 점이다.

안드로이드는 구글이 주도하는 OHA(Open Handset Alliance)에서 개발한 개방형 운영체제로서 리눅스 2.6 커널을 기반으로 하여 다양한 라이브러리와 멀티미디어 사용자 인터페이스 및 휴대전화 애플리케이션을 제공 한

*준회원, 홍익대학교 컴퓨터정보통신공학과

**중신회원, 홍익대학교 컴퓨터정보통신공학과

접수일자 2012년 5월 17일, 수정완료 2012년 6월 7일,

게재확정일자 2012년 6월 8일.

Received: 17 May, 2012, Revised: 7 June, 2012,

Accepted: 8 June, 2012

Corresponding Author: beongku@hongik.ac.kr

Dept. of Computer and Information Communications Engineering, Hongik University, Korea

다^[1-10]. 코드는 전적으로 공개 돼 있으며 터치는 물론 소리, 음성, 움직임, 기울임 등 다양하고도 미세한 입력 방식으로 휴대폰과 사용자가 상호작용할 수 있는, 인터랙티브한 애플리케이션을 누구든지 개발할 수 있게 됐다. GPS 지도와 데이터베이스 등 자신의 애플리케이션에 임베디드를 해서 활용할 수 있는 수많은 내장 서비스가 제공된다. 고성능 그래픽과 사운드를 지원하며 동영상 재생을 위한 여러 코덱이 장치되어 더욱 흥미로운 게임 및 애플리케이션을 만들 수 있다. 컴포넌트 중심의 아키텍처를 가진 안드로이드는 휴대폰을 음성 통화 장치를 뛰어넘어 웹과 긴밀히 연결된, 모바일 컴퓨팅 장치로 새로 발돋움시켜 주고 있다.

본 논문에서는 스마트폰의 알람과 타이머 방식으로 조합한 일정관리 애플리케이션의 개발을 목표로 하고 있다. 본 논문은 다음처럼 구성되어 있다. II장에서는 안드로이드 구조를 살펴보고, 기존 안드로이드 애플리케이션을 살펴 이들의 장점 및 단점을 비교한다. III장에서는 제안된 안드로이드를 이용한 모바일 스마트 알람의 기본 개념과 아이디어를 설명하고, 이후 구현과 개발된 콘텐츠 스마트 알람의 코드분석에 대해서 설명한다. 그리고 IV장에서는 안드로이드 에뮬레이터를 통한 애플리케이션 콘텐츠 스마트 알람의 성능평가를 한다. 마지막 V장 결론에서는 본 논문이 제시한 애플리케이션의 장점과 향후 과제에 대해 살펴보고 결론을 맺는다.

II. 관련연구

1. 안드로이드



그림 1. 안드로이드 구조
Fig. 1. Android Architecture

그림 1은 안드로이드^[1-10]의 구조를 보여주고 있다. 안드로이드는 다섯 개의 영역들로 나누어진다. 첫 번째 영역은 Application 과 Application Framework 영역으로 응용프로그램과 이 응용프로그램을 데몬 서버 형태로 실행하여 관리하는 역할을 한다. Application Framework는 JNI(Java Native Interface)를 이용해 라이브러리 계층, HAL(Hardware Abstraction Layer)과 통신할 수 있는 일종의 통로 역할을 하게 된다.

Libraries 계층과 HAL영역의 경우 응용프로그램이 개발되는 최상단의 부분과 OS의 영역인 리눅스 커널을 연결해주는 계층이 되며, 개발자가 쉽게 개발할 수 있도록 라이브러리 형태로 제공한다.

Libraries 계층에서는 안드로이드에서 제공하는 기능들을 API 라이브러리 형태로 제공하고 있지만, 여기서 제공되는 라이브러리는 C/C++로 개발되어 있기 때문에 Application Layer에서 라이브러리를 사용하기 위해서는 JNI를 통해서 사용해야 한다. Android Runtime의 경우 핵심라이브러리와 Dalvik 가상머신으로 이루어져 있다. HAL 영역은 안드로이드와 하드웨어를 연결해주는 레이어로 표준화된 API를 제공하고 있고, 사용자 정의 컴포넌트를 추가할 수 있다. 마지막으로 Linux Kernel 계층의 경우 메모리 관리, 프로세서 관리, 드라이버 관리를 담당하게 된다.

2. 안드로이드 애플리케이션

스마트폰 사용자가 점차 늘어나면서 수많은 애플리케이션들이 쏟아져 나오고 있다. 안드로이드 마켓^[10]에서는 전 세계 10만 개 이상의 다양한 애플리케이션들이 등록되어 있다. 건강, 교육, 금융, 날씨, 소셜 콘텐츠, 미디어, 음악, 의료, 커뮤니케이션등 수많은 카테고리에 무궁무진한 애플리케이션들이 개발, 판매되고 있다.

증강현실은 가상환경과 현실 환경을 융합하는 가상현실 시스템이다. 기존에 이들은 의료, 방송, 건축 등에서 주로 활용되어 왔다. 하지만 안드로이드의 이미지처리 모듈과 그래픽 처리모듈을 사용하여 증강현실 시스템을 구축할 수 있게 되었고 이를 이용한 위치기반 서비스나, 모바일 게임등 관련 활용범위가 확장되고 있다. 그 외에 인구 고령화에 따른 의료부분의 U-Health care 시스템을 구축하는 생체정보 모니터링 기술, 안드로이드 기반 모바일 원격 교육 시스템 등에 안드로이드 기술이 광범위하게 응용 적용되고 있다.

이처럼 안드로이드를 이용한 스마트폰 애플리케이션은 사용자의 생활에 큰 편의성을 가져다 줄 수 있다. 컴퓨터와 인터넷을 이용해야 했던 기존의 공간성의 제약을 핸드폰으로 이동시키면서 언제 어디서나 이러한 업무를 처리하고 사용할 수 있게 되었다. 하지만 애플리케이션 개발에 앞서 가장 먼저 고려해야 할 사항은 하드웨어적 제한성, 즉 화면의 한정성과 저장공간, 데이터 처리속도 등의 한계를 어떻게 극복하여 얼마나 사용자가 편리하게, 유용하게 애플리케이션을 사용할 수 있는가가 그 첫 번째 고려 사항이 되겠다. 둘째, 사용자가 손쉽게 이용하고 애플리케이션의 편의성을 느낄 수 있는 개발이 필요하다. 스마트폰을 단지 인터넷과 영상통화의 수단으로 이용하는 사람들이 많은데 그들의 대부분이 애플리케이션의 사용에 어려움과 편의성에 대해 부정적 인식을 가지고 있기 때문이다. 따라서 이러한 사용자들의 어려움과 인식을 해결 할 수 있는 사용자 중심의 애플리케이션 개발이 필요하다.

III. 안드로이드를 이용한 모바일 스마트 알람

1. 기본 개념 및 아이디어

본 논문에서는 안드로이드 플랫폼을 기반으로 한 스마트폰용 애플리케이션 스마트 알람을 제안한다. 제안된 스마트 알람 콘텐츠의 기본 개념 및 아이디어는 다음과 같다. 첫째, 제안된 스마트 알람은 구글의 안드로이드 플랫폼을 기반으로 한다. 둘째, 일간 일정에 중점을 맞추어 간단한 일정, 수시로 변경되는 일정이나 업무 역시 손쉽게 애플리케이션을 이용하여 관리 할 수 있도록 한다. 현재의 일정관리 애플리케이션의 동작 방식은 다음과 같다. 사용자는 자신의 일정과 그 일정의 발생시간, 마감시간을 일일이 입력하고 알람의 여부와 중요도를 체크하게 된다. 이미 짜여진 일정이나 몇일, 몇주등의 앞으로 있을 업무를 위해서는 아주 편리하고 확인하기 용의한 구조이다. 하지만 당일 바로 처리해야 하는 업무를 스마트폰이나 컴퓨터를 이용하여 일정의 내용과 중요도 알람여부 및 시간설정을 모두 하는 것은 오히려 기존의 수기식방식보다 불편하고 느린 방식이다. 몇 분, 몇 시간 후에 처리해야 할 업무를 스마트폰에 입력하기 보단 종이에 써 놓거나 메모장을 이용하는 경우가 더 빈번하다. 그러나 바쁜 업무나 일정으로 인해 종종 이를 놓치고 지나치는

경우가 발생하게 된다. 새롭게 제시되는 모바일 스마트 알람은 이를 방지하고 대체하기 위한 애플리케이션으로서 기존의 알람 방식에 추가로 타이머를 도입, 사용자는 알람의 발생시간을 입력하지 않고 일정에 대해 걸리는 소요시간을 입력함으로써, 자유롭고 짜임새있는 일정관리가 가능하다.

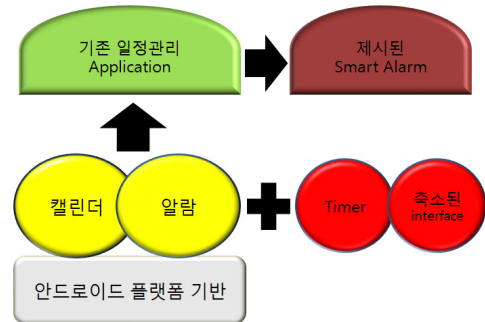


그림 2. 스마트 알람의 기본 개념
Fig. 2. Concept of Smart Alarm

2. 모바일 스마트 알람 구현 및 코드 분석

본 절에서는 기존의 알람과 타이머 방식을 조합한 본 연구에서 개발한 스마트 알람 애플리케이션 콘텐츠의 구현 및 코드를 설명한다. 첫째, 스마트 알람의 기본이 되는 캘린더 구조를 위한 코드와 데이터 입력구조를 보여주고 설명한다. 둘째, 스케줄입력의 구조를 위한 데이터 베이스 코드를 보여주고 설명하며, 마지막으로 알람과 noti 피케이션(notification)을 알려주기 위한 코드를 설명한다.

가. 캘린더 구조

| | |
|------|-----------------------|
| 패키지 | com.android.scheduler |
| 액티비티 | CalendarViewAcitivity |

캘린더의 구조를 구성하고 있는 주요한 기능들로는 달력구성, 날짜 확인, 당일 설정등이 있다.

① 달력 구성

```

rightNow = Calendar.getInstance();
gCal = new GregorianCalendar();
iYear = rightNow.get(Calendar.YEAR);
iMonth = rightNow.get(Calendar.MONTH);
maxDay = gCal.getActualMaximum(
    ((Calendar.DAY_OF_MONTH));
for(int i=1;i<=maxDayi++)
{
    daylist.add(Integer.toString(i));
}
    
```

현재 시간을 구하고 양력 달력을 생성하여 현재 년도와 달을 구한 뒤 그달의 마지막 날짜를 구하여 배열에 들어 있는 날짜를 달력에 입력한다. java.util.Calendar와 java.util.GregorianCalendar의 메소드를 사용하기 때문에 이를 import 시켜줘야 한다^[7].

② 날짜 확인

```
protected void onTouched(final Oneday touchedDay){
    if(isInside(touchedDay, basisDay, during)){
        Calendar cal = Calendar.getInstance();
        cal.set(basisDay.getYear(),
            basisDay.getMonth(), basisDay.getDay());
        cal.add(Calendar.DAY_OF_MONTH,during);
        Toast.makeText(this,
            (cal.get(Calendar.MONTH) + 1)+"월"+
            cal.get(Calendar.DAY_OF_MONTH) + "일 이후 선택
            가등", Toast.LENGTH_SHORT).show();
        return
    }
    AlertDialog.Builder builder = new
    AlertDialog.Builder(CalendarView.this);
    builder.setTitle("다음 날짜로
    설정하시겠습니까?");
    builder.setMessage(year + "." + month + "." +
    date + "(" + touchedDay.getDayOfWeekKorean()+")");
    builder.setPositiveButton("예", new
    DialogInterface.OnClickListener(){
    }
    );
    builder.setNegativeButton("아니오", null);
    builder.show();
}
```

화면에서 날짜 클릭시 현재 날짜 이전으로 선택 할 수 없도록 설정하고 선택시 Toast 메시지로 알려준다. 가능한 날짜를 선택시 AlertDialog를 이용하여 스케줄 입력창으로 또는 취소를 할 수 있도록 하였다.

③ 당일 선택 메뉴

```
public boolean onCreateOptionsMenu(Menu menu) {
    super.onCreateOptionsMenu(menu);
    MenuInflater mi = getMenuInflater();
    mi.inflate(R.menu.calendar_menu, menu);
    return true
}
@Override
public boolean onOptionsItemSelected(MenuItem
item) {
    switch (item.getItemId()) {
        case R.id.menuitem_calendar_0:
            gotoToday();
            return true
    }
}
```

onCreateOptionsMenu 메소드를 사용하여 메뉴창을 만들고 메뉴를 클릭했을 경우 현재일로 돌아가도록 구성한다^[7].

나. 데이터 베이스

| | |
|------|----------------------|
| 패키지 | com.android.database |
| 액티비티 | DataBase |

스케줄 입력 사항을 안드로이드내의 데이터베이스에 입력한다.

① DataBase 생성

```
private static class DatabaseHelper extends
SQLiteOpenHelper
{
    DatabaseHelper(Context context)
    {
        super(context, DATABASE_NAME, null,
        DATABASE_VERSION);
    }
    @Override
    public void onCreate(SQLiteDatabase db)
    {
        db.execSQL(DATABASE_CREATE);
    }
    @Override
    public void onUpgrade(SQLiteDatabase db, int
oldVersion, int newVersion)
    {
        Log.w(TAG, "Upgrading database from version
" + oldVersion + " to "
        + newVersion + ", which will destroy
all old data");
        db.execSQL("DROP TABLE IF EXISTS " +
        DATABASE_TABLE);
        onCreate(db);
    }
}
```

Data 생성자, SQLiteOpenHelper의 생성자를 불러서 데이터베이스를 생성한다. onCreate 메서드를 호출하여 데이터 베이스를 생성하고 onUpgrade 메서드를 호출하여 오래된 테이블 삭제를 위한 구문 지정을 한다^[7].

② 데이터 입력

```
public long createDB(String today, String state, String
title, String timeText, int year, int monthOfyear, int
dayOfMonth, int hour, int minute)
{
    ContentValues initialValues = new
ContentValues();
    initialValues.put(KEY_TODAY, today);
    initialValues.put(KEY_SCHEDULE_MENU_STATE,
state);
    initialValues.put(KEY_TITLE, title);
    initialValues.put(KEY_TIMETEXT, timeText);
    initialValues.put(KEY_YEAR, year);
    initialValues.put(KEY_MONTHOFYEAR,
monthOfyear);
    initialValues.put(KEY_DAYOFMONTH,
dayOfMonth);
    initialValues.put(KEY_HOUR, hour);
    initialValues.put(KEY_MINUTE, minute);
    return mDb.insert(DATABASE_TABLE, null,
initialValues);
}
```

데이터 테이블에 입력될 날짜, 제목, 시간 등을 입력한다. 사용자가 입력하는 일정에 대한 날짜, 제목, 시간들을 받아와서 입력하게 된다.

③ 데이터 수정

```
public boolean updateDB(long rowId, String today, String state, String title, String timeText, int year, int monthOfYear, int dayOfMonth, int hour, int minute)
{
    ContentValues args = new ContentValues();
    args.put(KEY_TODAY, today);
    args.put(KEY_SCHEDULE_MENU_STATE, state);
    args.put(KEY_TITLE, title);
    args.put(KEY_TIMETEXT, timeText);
    args.put(KEY_YEAR, year);
    args.put(KEY_MONTHOFYEAR, monthOfYear);
    args.put(KEY_DAYOFMONTH, dayOfMonth);
    args.put(KEY_HOUR, hour);
    args.put(KEY_MINUTE, minute);
    return mDb.update(DATABASE_TABLE, args, KEY_ROWID + "=" + rowId, null) > 0;
}
```

입력되는 데이터 값을 각각의 항목에 수정하여 입력한다. rowId 값을 이용하여 값을 수정한다.

④ 데이터 삭제

```
public boolean deleteDB(long rowId)
{
    return mDb.delete(DATABASE_TABLE, KEY_ROWID + "=" + rowId, null) > 0;
}
```

rowId를 이용하여 선택되어진 필드를 삭제한다.

다. 알람 및 noti피케이션

| | |
|------|------------------|
| 패키지 | com.android.alam |
| 액티비티 | AlramActivity |

① Alarm 설정

```
AlarmManager am = (AlarmManager) getSystemService(this.ALARM_SERVICE);
Intent intent = new Intent(this, AlarmReceiver.class);
intent.putExtra("number", cursor.getInt(cursor.getColumnIndex(DataBase.KEY_ROWID)));
intent.putExtra("title", title);
PendingIntent sender = PendingIntent.getBroadcast(TodayScheduleEditActivity.this, cursor.getInt(cursor.getColumnIndex(DataBase.KEY_ROWID)), intent, PendingIntent.FLAG_UPDATE_CURRENT);
am.set(AlarmManager.RTC_WAKEUP, time, sender);
```

알람의 설정은 일정을 입력하면서 시스템상에서 AlarmManager 메소드를 호출한다. 이후 AlarmReceiver를 호출하는 intent를 실행한다. PendingIntent는 바로 아래 코드인 am.set의 메소드를 주기적으로 실행하도록 하기 위한 방법이다^[7].

② 알람 및 noti피케이션 호출

```
public void onReceive(Context context, Intent intent)
{
    Bundle bundle = intent.getExtras();
    nm_number = bundle.getInt("number");
    title_text = bundle.getString("title");
    showNotification(context, R.drawable.icon);
}
```

알람 호출이후 리시버를 통하여 noti피케이션과 벨소리를 켜게 된다. 받은 일정의 내용을 화면에 보여주고, 이와 동시에 벨소리를 울리게 한다.

IV. 성능평가

본 논문에서 제안된 애플리케이션은 안드로이드 SDK 2.2버전을 기준으로 작성된 코드를 PC에서 스마트폰용 애플리케이션 구동을 테스트해 볼 수 있는 AVD(Android Virture Device)를 이용하여 동작 실험을 한다. 실제 디바이스(device)와 유사한 환경을 지닌 에뮬레이터는 이클립스와 안드로이드 SDK를 이용하여 실행할 수 있으며, 애플리케이션 실행 시 발생하는 버그들에 대해 리포팅 기능 및 디버깅 작업을 할 수가 있다^[6-9].

표 1. 실험 환경

Table 1. Simulation Environments

| | |
|-------------|-------------|
| 안드로이드 SDK | SDK ver2.2 |
| Java JDK[6] | JDK 6 |
| 이클립스 | Classic 3.7 |
| AVD | ver2.2 |
| 가상 메모리카드 용량 | 64MB |

성능평가를 위해 스마트알람의 주기능인 일정을 입력하고 그에 따른 알람 발생여부를 테스트하여 동작 유무를 확인해 보기로 한다. 그림 3은 스마트알람 애플리케이션의 메인메뉴화면을 보여주고 있다.



그림 3. 스마트알람 애플리케이션 메인메뉴화면
Fig. 3. Main Menu of Smart Alarm Application

그림 4는 스마트 알람의 기본화면이다. 메뉴를 클릭하면 보이는 바와 같이 일정추가/수정과 추가일정메뉴를 선택하여 입력 할 수 있다.



그림 4. 스마트 알람 기본 화면
Fig. 4. Base View of Smart Alarm

그림 5는 스마트 알람의 일정추가화면이다. 일정의 내용을 입력하고 시간 입력 탭을 클릭하면 시간입력 창이 나오며 시간을 입력 할 수 있게 된다. 일정을 다 입력하고 나면, 스크롤을 내리면 저장할 수 있는 버튼이 있다. 현재 기본일정은 8개까지 입력이 가능하다.



그림 5. 스마트 알람 일정추가 화면
Fig. 5. View of daily schedule addition of Smart Alarm

그림 6은 스마트 알람의 일정 입력 후의 화면이다. 에뮬레이터에서는 한글입력이 되지 않아서 숫자로 입력을 하였다. 1번 일정은 17시 14분에 알람 설정이 2번은 1번의 1분 후로 3번은 2번의 2분 후에 알람이 울리도록 설정이 되어져 있다. 위의 그림은 1번 알람이 울렸을 당시 화면이다. 토스트로 “알람!”을 알려주고 위에 noti피케이션(notification) 창이 활성화 된 것을 볼 수 있다. 각 일정을 클릭하면 일정을 삭제 시킬 수 있다.



그림 6. 스마트알람 일정 입력 화면
Fig. 6. After input the plan

그림 7은 스마트 알람의 noti피케이션 화면이다. 17시 14분에 첫 알람이 울린 후 설정해놓은 대로 1분뒤 15분에 두 번째 알람, 15분의 2분 뒤인 17분에 세 번째 알람이 울린 것을 확인 할 수 있다. 각각의 noti피케이션을 클릭시 일정의 내용을 볼 수 있다.



그림 7. 스마트알람 noti피케이션 화면
Fig. 7. Notification View of Smart Alarm

V. 결론

본 논문에서는 안드로이드 플랫폼을 기반으로 하는 기존의 일정관리 시스템을 간소화시키고 타이머 형식의 알람방식을 사용하는 스마트폰용 애플리케이션 스마트 알람을 제안한다. 제안된 애플리케이션 콘텐츠 스마트 알람의 특징은 다음과 같다. 첫째, 간편하게 자신의 하루 일정을 입력하고 관리 할 수 있다. 기존의 일정관리 프로그램은 일정을 관리 하는 측면에서는 매우 뛰어난 효율성을 보이나 빈번하게 바뀌는 일정이나 급작스럽게 생겨나는 일정에 대해서 입력하기 번거롭고 귀찮은 한계성이 있었다. 본 논문에서 제안한 방법은 간편하고 손쉽게 일과에 대해서 수정과 추가가 가능하다. 둘째, 알람의 방식이 타이머를 기반으로 하는 상대시간 입력 방식이다. 일정이 순차적으로 시간을 짜입새 있게 이용해야하는 직업군에서는 절대시간을 생각하는 것 보다는 간편한 일정 이후의 상대시간을 고려하는 것이 더 편한 방법이다. 예를 들어 연구실에서 어느 작업의 분석 결과를 기다린다고 하자. 1차 분석이 끝난 뒤 2차 작업을 할 수 있고, 그 이후 다음 작업이 순차적으로 이루어진다면, 또한 작업의 진행 시간이 일정하지 않다면 각각 일정의 알람시간을 입력하는 것은 상대시간 입력이 더 간편하다. 현재 고

정된 일정 입력 인텍스를 사용자가 필요한 만큼 이용 할 수 있도록 수정하고 입력된 상대시간을 계산해서 발생시각을 보여 줄 수 있도록 개선, 수정할 계획이다.

참고 문헌

- [1] Bum Kyo Oh, Tae Hwan Kang, Beongku An, "Android Based Booky Content," Journal of IWIT, vol.10, no. 2, pp.53-59, April 2010.
- [2] Rodriguez NEC, Juareg JIG, Savage R, Fernandez MRL, "Design and implementation of an android," IEEE MWSCAS'09, August 2009.
- [3] Xianhua Shu, Zhenjun Du, Rong Chen, "Research on Mobile Location Service Design Based on Android," WiCom'09, September 2009.
- [4] Gozavez J, "First Google's android phone launched, EEE Vehicular Technology Magazine," vol.3, issue 3, pp.199-202, March 1987,
- [5] Conti JP, "The androids are coming," Engineering & Technology, vol.3, issue.9, pp.72-75, May-June 2008.
- [6] <http://www.openhandsetalliance.com>
- [7] <http://developer.android.com/sdk/index.html>
- [8] <http://www.eclipse.org/downloads>
- [9] <http://java.sun.com/javase/downloads>
- [10] <https://market.android.com/>

※ 이 논문은 2011년도 정부(교육과학기술부)의 재원으로 한국연구재단의 기초연구사업 지원을 받아 수행된 것임(2011-0025907).

저자 소개

오 진 우(준회원)

- 2012: 홍익대학교 컴퓨터정보통신공학과(학사)
 - 2012-현재: (주)페타즈, 연구원
- <주관심분야: 안드로이드, 홈네트워크, 모바일 무선네트워크, 이동통신>

안 병 구(중신회원)



- 1988: 경북대학교 전자공학과 (BS)
 - 1996:(미)Polytechnic Univ., Dept. of Electrical and Computer Eng., Brooklyn, New York, USA. (MS)
 - 2002:(미)New Jersey Institute of Technology(NJIT), Dept. of Electrical and Computer Eng., New Jersey, USA.(Ph.D)
 - 1989-1994: Senior Researcher, RIST, Republic of Korea
 - 1997-2002: Lecturer & RA, New Jersey Institute of Technology (NJIT), USA.
 - 2003-present: Professor, Hongik University, Republic of Korea.
 - 2005-2011: Marquis Who's Who in Science and Engineering was listed.
 - 2006-2011: Marquis Who's Who in the World was listed.
 - 2012-현재: 대한전자공학회 컴퓨터소사이어티 회장
- <주관심분야: Wireless Networks, Ad-hoc & Sensor Networks, Multicast Routing, QoS Routing, Cross-Layer Technology, Cooperative Communication, Network Coding, VLC, Android, Patch Antenna, Bioinformatics>