Location Transmission

Suxin Yuan, 5090309502
school of electronic information and electric engineering, SJTU
email:samsonyuan1102@gmail.com

June 15, 2012

Abstract

This article is tent to introduce the readers the process of design an android application called ”My Location” which is used to send the phone’s geographic location to specific phone receivers. It also introduce the motivation of this design and what we can do in the future.

1 Introduction

Google has been providing the map service for years, users can share their current locations with their friends via a function called Latitude. However, this service has two drawbacks. First is that each friend on the Internet has the access to your location information, which sometimes you just want specific person have this privilege. This form of sharing also might cause unnecessary information leakage. It is just private. Second, the google service require good internet service, which might uncover in some remote area.

In China, people can call the police not only by telephone, but also by SMS. The message number is area number plus 110. If you call the police by SMS, you have provide your accurate current location. This is inconvenient and sometimes unprocurable.

In order to solve the above problems, I tried to design an android application called ”my location”. It is an android app which can provide the service of locate the phone’s location and send it to others in the form of shot message. It is private, convenient, power saving and quick reacting.

2 Architecture

As figure 1 shows the flow diagram of the android app. Firstly, the sender get the current location; secondly, the sender send the location information and attached information through SMS; thirdly, the receiver receive the shot message containing location data and attached text. lastly, the receiver determine the sender’s current location via google maps or baidu maps.

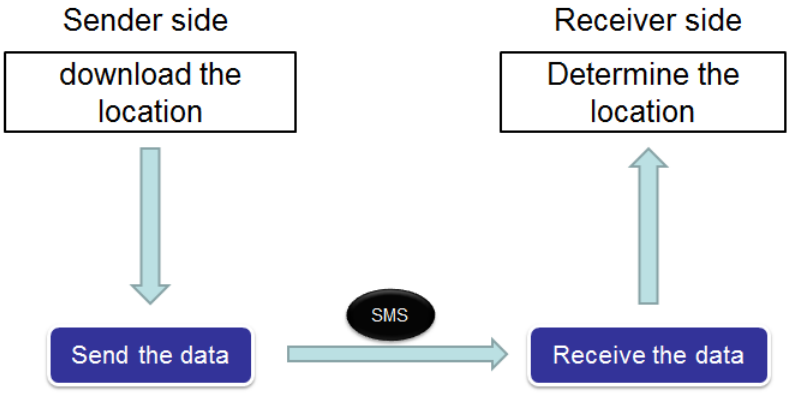


Figure 1: flow diagram of My Location

2.1 Location Achievement

In this section, two basic services is needed:google map service and internet service. The google map service is used to achieve the location through GPS. The internet service is used to achieve the location through WIFI or GSM and 3G network.
The two service has different advantages and disadvantages. The google map service can achieve the accurate location data whose error range is within 20 meters. however, this service might consume more power, which is an important concerning aspect of mobile phone. And since this service has to get access to GPS service, it spend more the 20 seconds to connect with the satellite and download the data. If the sky is covered with dark clouds or there are many tall buildings around, the achievement may be failed. In one word, this service requires strict conditions and cost much. The internet can also serve the location service. opposite of google map service, the Internet can provide the location information quickly(within 5 seconds), only with in covered area of the wireless network. Its drawback is that the data accuracy is not as good as GPS's data. There are differences in the Internet service, the 2G and 3G's accuracy is higher than WIFI.

Considering the characteristics of the two services, we balance the power consumption, the reacting speed and the accuracy.
The two services are respectively corresponding to the following android permission:
android.permission.access_COARSE_LOCATION;
android.permission.ACCESS_FINE_LOCATION;
android.permission.INTERNET;
All these service is provided in ADT and SDK. So we can use them directly.

2.2 Send Location Data

In this section, we will send the achieved location data via SMS.
The location data is a two dimensional coordinate: latitude and longitude. The data is stored in a buffered variable. To declare sender's situation or other information, text is attached after the location data. All this information was stored in a new buffered variable. Then we call the built-in function to send SMS. The information string is cut into several pieces(70 words in a piece) and sent to the receiver according to the phone number one by one.

2.3 Receive Location Data

The receiver will receive no less than two messages. The first one is the sender's location data(Latitude, Longitude). The second one is the attached information. Since the receive service is offered by the receiver's phone, nothing to do with this app, here no further introduction will be list.

2.4 Determine Location

After receive the location data, the receiver can determine ,the sender's current location by the location data.
There are two way to acknowledge the location. The fist one is to use the PC's google map service. Google map lab serve the function of LatLng Marker, which can show the location on the map with the Latitude and Longitude input. We can also use the phone's google map or baidu map to acknowledge the location. The operation method is similar to PC termination.

Since this app does't include the built-in service, the function will be introduced in the section of Further to Do.

3 Realization

3.1 Preparation Work

1. install JDK
2. download Eclipse and install it.
3. install ADT Plugin for Eclipse.
   Help > Install Software > Add
   http://dl-ssl.google.com/android/eclipse/
4. configure the ADT Plugin
   window > Preference > Android > Locate Android SDK > Apply > OK
5. create an AVD
   Window > Android SDK and AVD Manager

3.2 Create a Project

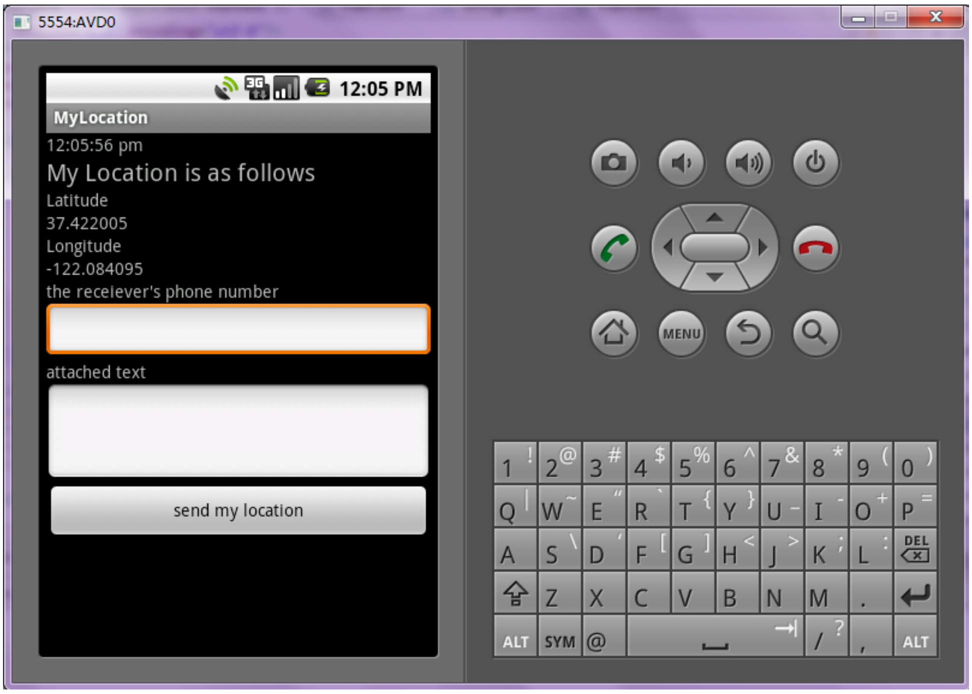
Create a project named ”MyLocation”:
Fill in the project name, the build target, the application name, the package name and the create activity. Then click Finish.

3.3 Manifest

In this section, we will add the uses-permission. Open ”AndroidManifest.xml”, change the label to ”Permissions”, add the permission from the Pulled-down menu.
android.permission.access_COARSE_LOCATION;
android.permission.ACCESS_FINE_LOCATION;
android.permission.INTERNET;

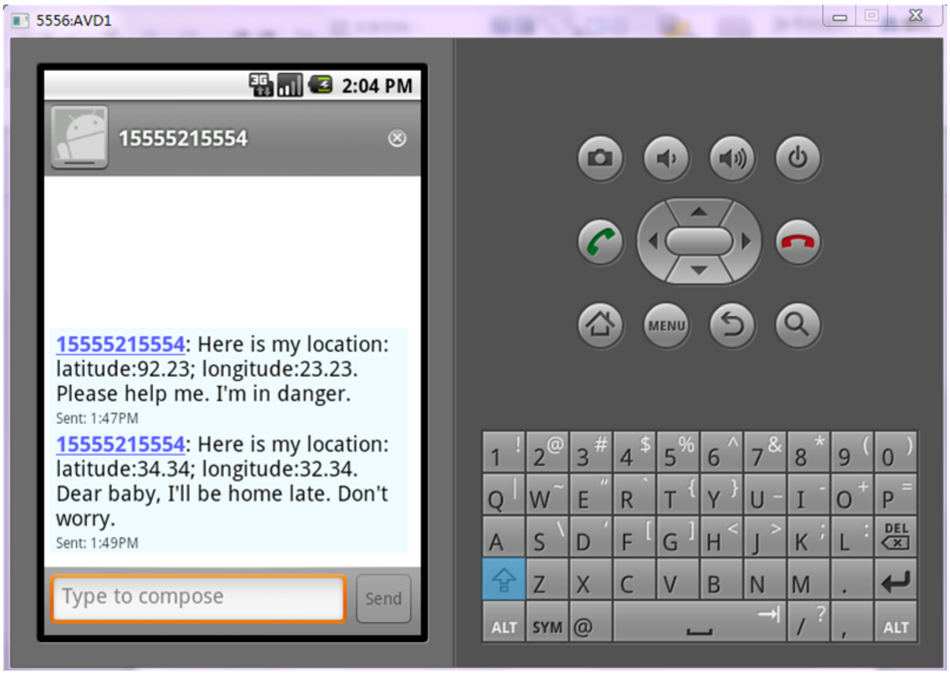
3.4 UI

Open Main.xml, using the Graphical Layout or the xml to edit the user interface. The elements of the UI include a digital clock, a text view of ”My Location”, a text view of ”receiver’s phone number” and ”attached text”, a text input of ”phone number” and ”attached text” and a send button.
The figure of UI is as follows.



12:05 PM
MyLocation
12:05:56 pm
My Location is as follows
Latitude
37.422005
Longitude
-122.084095
the receiever's phone number
attached text

Figure 2: the UI of the app My Location



15555215554: Here is my location: latitude:92.23; longitude:23.23. Please help me. I'm in danger.
Sent: 1:47PM

15555215554: Here is my location: latitude:34.34; longitude:32.34. Dear baby, I'll be home late. Don't worry.
Sent: 1:49PM

Figure 3: the UI of the app My Location

3.5 Location achievement

3.5.1 Definite the LocationListener

Open src/com/demo/android/mylocation/MyLocation.java.

Create a public class Mylocation extends Activity implements LocationListener. Under the class Mylocation, create four methods called ”onLocationChanged”, ”onProviderDisabled”, ”onProviderEnabled” and ”onStatusChanged”.

3.5.2 Achieve the Location using LocationManager

Rewrite function of ”onCreate”, add the function of ”updateState” in it. The function of ”upateState” is used to deal with the action of updating the personal state. firstly, we get the location service provided by the android system,
mgr = (LocationManager) getSystemService(Location_SERVICE)
secondly, use the GPS to achieve the location,
Location location = mgr.getLastKnowLocation(”gps”);
thirdly ,get the location data using the sentence,
location.getLatitude(), location.getLongitude()

3.5.3 Determine the Best Source of Location Services

We could determine the best source of location service according to the power consumption and the accuracy.

when we call the method ”getLastKnowLocation” of ”mgr”, we don’t use GPS directly. We transfer a string called ”best”, which is provided by the method ”getBestProvider” of ”mgr”.

```
Criteria criteria = new Criteria();
best = mgr.getBestProvider(criteria, true);
Location location = mgr.getLastKnownLocation(best);
```

”getBestProvider” has two parameters, one is to determine wether to use the opened source of location service, which is usually typed in ”true”, the other one is additional ”Criteria”.

3.5.4 Location Updates

To make functions like ”onLocationChanged” can be called, we have to add the action of requiring to check the location in the lifetime of ”Activity”. so we add the method of ”requestLocationUpdates” in the function ”onResume” of ”mgr” and the method ”removeUpdates” in the function ”onPause” of ”mgr” to manage the action of ”mgr”.

”requestLocationUpdates” is used to update the phone’s location sate, it will accept four parameters: ”provider”, ”minTime”, ”minDistance” and ”listener”
”removeUpdates” is used to stop updates. It means that when the ”Activity” do not need to use the source of location service, the service may fall to sleep temporarily to save power.

3.5.5 Display the Location

As we mentioned in UI, there are two text-views to display the location: Latitude and Longitude.
We use method ”setMethod” of ”showLatitude” and ”showLongitude” to display the location data.

3.6 Send the Location Data

3.6.1 Gather the Necessary Information

After we achieve the location data, some more information is needed to gathered.

Firstly, by using the method ”getText” of ”phonenum”, we get the receiver’s phone number ”phone”. Secondly, by using the method ”getText” of ”attached”, we get the attached information ”attach”.
All this data is then transferred to string. And the location data and the attached infor-

mation is gathered in a buffer string called ”text”.

3.6.2 Send the Information

Since all the necessary information is gathered , SMS can be used to send the information to the specific receiver.

Button ”sendsms” is the trigger point, once the button is pressed, the action of sending SMS is called.
We use the method ”OnClickListener” to realize this function. ”OnClickListener” is defined in the method ”setOnClickListener” which is belong to ”sendsms”.
In the ”onClick” method within ”OnClickListener”, the action of sending SMS is defined. Since the sending SMS class is served by the android system, we can use it directly. Firstly, the SMS service is called in use by Initializing class ”SmsManager” to ”manager”. Then the imformation is divided into pieces which are string arrays by using the method divideMessage of ”manager”. At last, the method ”sendTextMessage” of ”manager” is called to send the ”text”. Once the SMS is sent successfully, a toast is called to inform the user.

4 Problem & Solution

In the process of writing the App, I had trouble in transmitting data. firstly, I use the string as the type of variable, it works in the simulation terminal on PC. However, it will crash when downloaded in the real phone and took into action. I tried several times, and can’t find way to solve this problem. Then, I realized that there are differences between the simulation terminal and the real phone, they store and transmit the data using different mechanism. So I change the type of transmitting data to bufferstring which is advanced compared with string. The problem is solved perfectly.

I also met some problems in the use of ”button”, since the button is a trigger point, it has to be defined and initialized, at first, I did not acknowledge this and could not realize this function. By reading the reference book carefully, I get to know the mechanism of button and realize the function successfully.

5 Further to Do

In this project, I just wrote the app of sending the current location of users, but I did not realize the receiving part. Receivers have to use the google map or baidu map to determine the intuitive location.
Another weak point of this app is that users have to type in the receiver’s phone number. Actually, few people can remember all friends or family’s phone number.

The above two point may make the app not convenient enough. In the future, we can make some utilization on these. Firstly, add in the function of searching contact stored in the phone. secondly, add the overlay of map to determine the environment of the location without using the other tools.

6 Acknowledgement

Many thanks to Professor Wang, he give us great lectures and introduce us the front new idea of wireless communication.
Also thank Dr. Tian, he give me much help on solving the problem of class and give us vivid class.

7 Code

7.1 Mainfest

```
¡?xml version=”1.0” encoding=”utf-8”?¿
¡manifest xmlns:android=”http:// schemas.android.com/apk/res/android”
package=”com.demo.android.mylocation”
android:versionCode=”1”
android:versionName=”1.0” ¿

¡uses-sdk android:minSdkVersion=”7” /¿

¡application
android:icon=”@drawable/ic_launcher”
android:label=”@string/app_name” ¿
¡activity
android:name=”.MyLocation”
android:label=”@string/app_name” ¿
¡intent-filter¿
¡action android:name=”android.intent.action.MAIN” /¿
¡category android:name=”android.intent.category.LAUNCHER” /¿
¡/intent-filter¿
¡/activity¿
¡/application¿
¡uses-permission android:name=”android.permission.ACCESS_COARSE_LOCATION”/¿
¡uses-permission android:name=”android.permission.ACCESS_FINE_LOCATION”/¿
¡uses-permission android:name=”android.permission.INTERNET”/¿
¡uses-permission android:name=”android.permission.SEND_SMS”/¿

¡/manifest¿
```

7.2 main.xml

```
¡?xml version=”1.0” encoding=”utf-8”?¿
¡LinearLayout xmlns:android=”http://schemas.android.com/apk/res/android”
android:layout_width=”fill_parent”
android:layout_height=”wrap_content”
android:orientation=”vertical”¿
```

```
¡DigitalClock
android:id=”@+id/digitalClock1”
android:layout_width=”wrap_content”
android:layout_height=”wrap_content”
android:text=”DigitalClock” /¿
¡TextView
android:layout_width=”fill_parent”
android:layout_height=”fill_parent”
android:textSize=”20sp”
android:text=”My Location is as follows”
/¿
¡TextView
android:layout_width=”fill_parent”
android:layout_height=”fill_parent”
android:text=”Latitude”
/¿
¡TextView android:id=”@+id/showlatitude”
android:layout_width=”fill_parent”
android:layout_height=”wrap_content”
android:text=””
/¿
¡TextView
android:layout_width=”fill_parent”
android:layout_height=”fill_parent”
android:text=”Longitude”
/¿
¡TextView android:id=”@+id/showlongitude”
android:layout_width=”fill_parent”
android:layout_height=”wrap_content”
android:text=””
/¿
¡TextView
android:layout_width=”fill_parent”
android:layout_height=”fill_parent”
android:text=”the receiever’s phone number”
/¿
¡EditText android:id=”@+id/phonenum”
android:layout_width=”fill_parent”
android:layout_height=”fill_parent”
android:numeric=”integer”
android:text=””
/¿
¡TextView
android:layout_width=”fill_parent”
android:layout_height=”fill_parent”
android:text=”attached text”
/¿
¡EditText android:id=”@+id/attached”
android:layout_width=”fill_parent”
android:layout_height=”fill_parent”
```

```
android:minLines="3"
android:text=""
/¿
¡Button android:id="@+id/sendsms"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:text="send my location"
/¿
¡/LinearLayout¿
```

7.3 myLocation.java

```
package com.demo.android.mylocation;

import java.util.ArrayList;
import android.app.Activity;
import android.location.Criteria;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.Bundle;
import android.telephony.SmsManager;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

public class MyLocation extends Activity implements LocationListener
/** Called when the activity is first created. */

private LocationManager mgr;
private String best;
public StringBuffer locationSms = new StringBuffer();
@Override
public void onCreate(Bundle savedInstanceState)
super.onCreate(savedInstanceState);
setContentView(R.layout.main);
updateStat();
Button sendsms = (Button)findViewById(R.id.sendsms);
sendsms.setOnClickListener(new OnClickListener()
public void onClick(View v)
// TODO Auto-generated method stub
EditText attached = (EditText)findViewById(R.id.attached);
String attach = attached.getText().toString();
String content = locationSms + attach;
EditText phonenum = (EditText)findViewById(R.id.phonenum);
String mobile = phonenum.getText().toString();
SmsManager manager=SmsManager.getDefault();
ArrayList¡String¿ texts=manager.divideMessage(content);
for(String text:texts)
manager.sendTextMessage(mobile, null, text, null, null);
Toast.makeText(MyLocation.this, ”send successfully”, Toast.LENGTH_LONG).show();

);


public void updateStat()
// TODO Auto-generated method stub
mgr = (LocationManager) getSystemService(LOCATION_SERVICE);
Criteria criteria = new Criteria();
best = mgr.getBestProvider(criteria, true);
Location location = mgr.getLastKnownLocation(”gps”);
TextView showlatitude = (TextView)findViewById(R.id.showlatitude);
TextView showlongitude = (TextView)findViewById(R.id.showlongitude);
if(location != null)
locationSms.append(”This is my location right now:”+”Latitude:”);
locationSms.append(Double.toString(location.getLatitude()));
locationSms.append(”;Longitude:”);
locationSms.append(Double.toString(location.getLongitude())+”.”);
showlatitude.setText(Double.toString(location.getLatitude()));
showlongitude.setText(Double.toString(location.getLongitude()));
elseshowlatitude.setText(””);
showlongitude.setText(””);



public void onLocationChanged(Location location)
// TODO Auto-generated method stub
TextView showlatitude = (TextView)findViewById(R.id.showlatitude);
TextView showlongitude = (TextView)findViewById(R.id.showlongitude);
if(location != null)
locationSms.append(”This is my location right now:”+”Latitude:”);
locationSms.append(Double.toString(location.getLatitude()));
locationSms.append(”;Longitude:”);
locationSms.append(Double.toString(location.getLongitude())+”.”);
showlatitude.setText(Double.toString(location.getLatitude()));
showlongitude.setText(Double.toString(location.getLongitude()));
elseshowlatitude.setText(””);
showlongitude.setText(””);



public void onProviderDisabled(String provider)
// TODO Auto-generated method stub


public void onProviderEnabled(String provider)
```

```
// TODO Auto-generated method stub


public void onStatusChanged(String provider, int status, Bundle extras)
// TODO Auto-generated method stub


@Override
protected void onResume()
super.onResume();
mgr.requestLocationUpdates(best, 6000, 1, this);


@Override
protected void onPause()
super.onPause();
mgr.removeUpdates(this);
```

References

Wireless Communicaitons and Network, Jon W.Mark and Weihua Zhuang, peason Education,Inc.

Google Android Development, gasolin.