# Android 智能手机天气预报系统设计及实现

摘要：在分析讨论 Android 应用系统设计原理的基础上,提出了 Android 智能手机天气实况预报系统用户界面以及获取并解析城市列表数据的设计方法,给出了在用户界面上呈现列表数据的设计过程,实现了一个简单的 Android 智能手机城市天气实况预报系统。系统在模拟器上通过调试并正常运行。

## 0 引言

为了让智能手机用户能够随时随地查询互联网所提供的服务，一种高效的办法就是将应用系统的功能拓展到手机终端，让手机能通过移动网和互联网访问 Web 网站并处理各项业务。Android 系统是 Google 公司开发的一个开源手机操作系统，它包括了操作系统、用户界面和应用程序，即智能手机工作所需的全部软件。Android 的最大特点是它的开放性体系架构，不仅具有非常好的开发、调试环境，而且还具有各种可扩展的设施，包括丰富的图形组件、多媒体支持功能和强大的浏览器，而且已有许多比较成熟的应用案例。

本文提出了智能手机天气实况预报系统的一种设计实现方法，供开发者参考。

## 1 Android 应用程序设计原理

一个 Android 应用程序，通常由 AcTIvity、Intent Receiver、Service、CONtent Provider 四种功能组件构成。但是，并不是每一个 Android 应用程序都需要用到这四种功能组件，而是只需上述四种功能组件的某些组合。

### 1.1 Activity

Activity 是最基本的 Android 应用程序组件。大多数应用由多个屏幕组成，一个 Activity 通常则是一个单独的屏幕。每一个 Activity 都被实现为一个独立的类，并且从 Activity 基类中继承而来。Activity 类将会显示由视图控件组成的用户接口，并对事件做出响应。Android 使用 Intent 这个特殊类，实现在屏幕与屏幕之间移动。通过解析各种 Intent,从一个屏幕导航到另一个屏幕是很简单的。当向前导航时，Activity 将会调用 STartActivity（Intent myIntent）方法，然后，系统会在所有安装的应用程序中所定义的 IntentFilter 中查找，找到最匹配 myIntent 的和 Intent 对应的 Activity.新的 Activity 接收到 myIntent 的通知后，开始运行。当 startActivity 方法被调用时将触发解析 myIntent 的动作。这个机制具有两点优势：一是能够重复利用从其他组件中以 Intent 形式产生的一个请求；二是 Activity 可以在任何时候被一个具有相同 IntentFilter 的新 Activity 取代。

### 1.2 Intent Receiver

通过 Intent Receiver 实现的应用能够对一个外部的事件作出响应。Intent Receiver 在 AndroidManifest.xml 中注册，也可以在代码中使用 Context.registerReceiver（）进行注册。当一个 IntentReceiver 被触发时，应用不必对请求调用 Intent Receiver,系统会在需要的时候启动应用。各种应用还可以通过使用 Context.

brOAdcastIntent（）将它们自己的 Intent Receiver 广播给其他应用程序。

### 1.3 Service

Service 是一段长生命周期的、没有用户界面的程序。例如媒体播放器这个 Activity 会使用 Context.startService（）来启动一个 Service,从而可以在后台播放音乐。同时，系统也将保持这个 Service 一直执行，直到 Service 运行结束。系统可通过使用 Context.bindService（）方法，连接到一个 Service 上（如果这个 Service 还没有运行将启动它）。当连接到一个 Service 之后，还可以利用 Service 提供的接口与它通信。就媒体播放器来说，还可以进行暂停、重播等操作。

### 1.4 Content Provider

Android 应用程序能够将它们的数据保存到文件、SQL 数据库，甚至是任何有效的设备中。Content Provider 实现了应用数据与其他的应用程序的共享。Content Provider 类实现一组标准的方法，能够让其他应用程序保存或读取此 ContentProvider 处理的各种类型数据。在 Android 中，默认使用 SQLite 作为系统数据库，但使用方法略有不同。Android 中每一个应用程序都运行在各自的进程中，当访问其他应用程序的数据时，需要在不同的虚拟机之间传递数据，这样操作起来会有些困难（正常情况下，不能读取其他应用程序的数据库文件）。ContentProvider 正是解决不同的应用包共享数据的工具。

## 2 天气实况预报系统的具体实现

基于 Android 平台的智能手机访问服务器的主要技术是数据交互方法。基于 Android 操作系统的天气实况预报系统设计的核心是通过调用通信协议 SOAP（Simple Object AccessProtocol）的接口，从 Web Service 提供商中提取天气预报的数据信息，为客户端服务。其基本功能是当用户运行天气预报系统程序时，在手机屏幕上显示出中国的城市名称，用户单击任意一个城市名，可获得该城市的天气实况预报信息。用户也可以输入城市名称，点击查询获得天气预报信息。CitiesWeather-Forecast 工程目录结构及其源代码文件如图 1 所示。

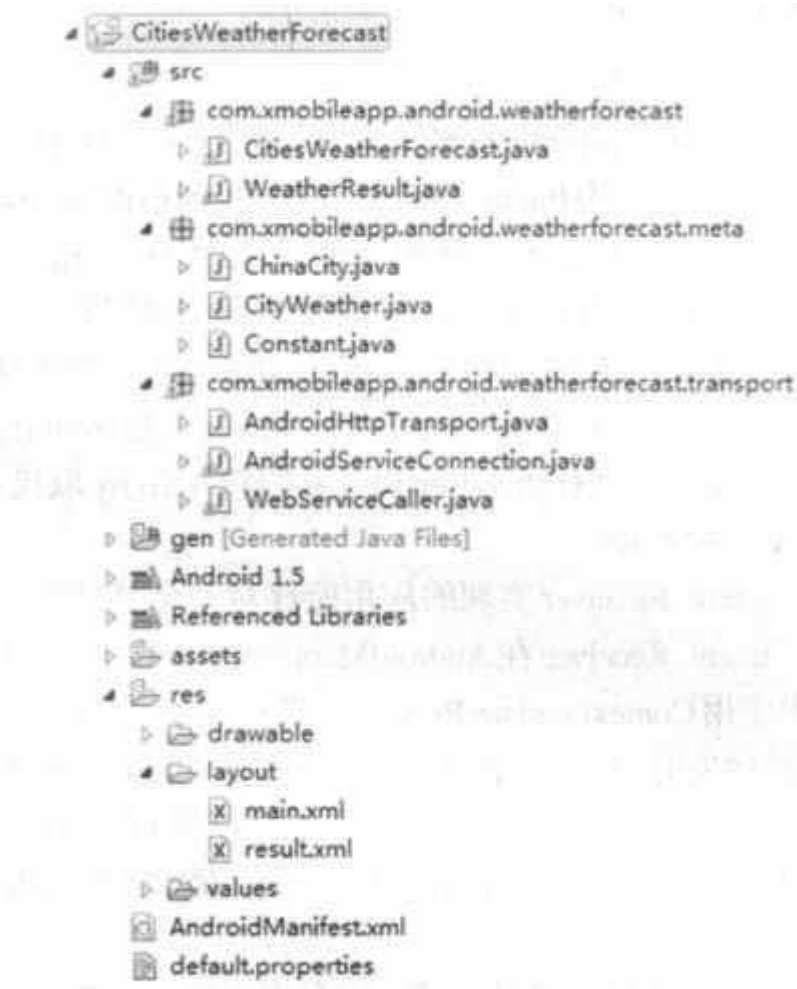


图 1　CitiesWeatherForecast 工程目录结构

2.1 Android 平台的用户界面实现

Android 提供的可变化的用户界面（UI）开发模块是基于 XML 文件的。这些 XML 文件放在工程/res/layout 下面。这个目录可包含所有应用程序所需的非码部分，比如图片、字符串、xml 文件。当要使用到这些资源时，在代码目录中打开 R.java 文件即可。在 XML 文件里编辑界面的代码，实现起来不仅方便，使用时也会更加灵活。

在 Android 程序设计中要用到一些基本的 Android UI 元素，通过使用 Views、View Groups 和 layouts 可为 Activity 创建功能性的、富有直观力的 UI.通常是使用 Android SDK 中提供的一些控件，进行布局、扩展和定制这些控件，并使用 ViewGroups 去组合 Views,创建由相互作用的子控件组成的原子的、重复利用的 UI 元素。也可以创建自己的 Views,来实现显示数据和与用户交互的新途径；或使用一些继承自 View Group 的 Layout 管理器来组织 Android UI 中的单个元素到屏幕上。

在一个 Android 应用中，用户界面由 View 和 View Group 对象构建。View 与 View Group 有很多种类，而它们都是 View 类的子类。View 对象是 Android 平台中用户界面的基本单元。View 类是 widgets（工具）类的父类，它们提供了诸如文本输入框和按钮之类的 UI 对象的

完整实现。View Group 类是 Layouts（布局）类的父类，它们提供了诸如流式布局、表格布局以及相对布局之类的布局架构。

View 对象是一个数据体，它的属性存储了用于屏幕上一块矩形区域的布局参数及内容。并负责它所辖的这个矩形区域之中所有测量、布局、焦点转换、卷动以及按键/触摸手势的处理。作为一个用户界面对象，View 同时也担任着用户交互关键点以及交互事件接受者的角色。天气实况预报系统程序功能比较单一，其 UI 也相对比较简单，只要有一个输入框、一个查询按钮以及一个显示所有中国城市信息的列表即可。为了建立 Android 平台的用户界面，首先要在 Package Explorer 窗口中展开 Layout 后新建。xml 文件，用来对窗口界面进行布局，主要有系统运行的主配置文件 main.xml 和 result.xml.

Layout 是一类特殊的 ViewGroup 控件，它们本身没有任何可显示内容，存在的惟一原因就是其中的内部结构，能够更好地摆放它的子控件。比如 Linearlayout,可将子控件按水平或垂直方向按顺序排列下去；Tablelayout,可以将子控件按照表格的形式，一枚枚放置好；Relativelayout 更灵活，可以设定各个控件之间的对齐和排列关系，适合定制复杂的界面。有了 Layout 的存在，控件和控件之间不再是割裂地存在，而是更有机地结合在一起，设定起来也更为方便。在本实例的 main.xml 文件中主要是添加一些界面的布局设置，例如如下代码：

```
<LinearLayout

android:orientation="horizontal"

android:layout_width="fill_parent"

android:layout_height="wrap_content"

>

android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:text="@string/cityText"/>

<Button android:id="@+id/searchBtn"

<Button android:id="@+id/searchBtn"

android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:text="@string/searchBtn"/>
```

</LinearLayout>

上述代码生成了一个 Linearlayout,定义了一个 EditText 和 Button,并且设置了相关的参数。在 Android 中，控件最重要的大小属性是 width/height,开发者可以指明控件的大小，控件的宽、高以及在屏幕中的显示位置等，可以设定成为 fill_parent 和 wrap_content.另外，还可以设置相应的文本信息，并可通过 android:text="@string/cityText"/>进行引用（string 文件存放在工程的 res/values 中）。按照这种方法，还可以依据需要定义相应的布局控件，譬如定义用来呈现天气实况信息的 UI 等。

2.2 创建中国城市列表数据

根据天气预报系统功能需求分析，系统启动后，应在主界面上呈现出中国国内所有城市的列表数据，因此需要创建中国城市列表数据，包括获取、解析城市列表数据，并在用户界面上呈现出来。为此，需要解决的问题是手机终端从 Web 服务器获取了所有类型的数据之后，将这些数据进一步交给 Android 手机终端 View 组件[4],在手机界面上显示给用户。

[1] 获取城市列表数据

Android 没有提供 Web Service 的组件库，但可以通过修改在 PC 或 J2ME 上使用的开源 Web Service Java 库作为 AndroidWeb Servic 库使用。kSOAP2（http://ksoap2.sourceforge.net）是一个在智能手机上的 SOAP Web Service 客户端包，可用于资源受限制的 Java 环境如 Applets 或 J2ME 应用程序。在 GoogleCode 上有一个项目 ksoap2-android（http://code.google.com/p/ksoap2-android/）可用在 Android 平台上实现轻量级的 SOAP 库，即使用 ksoap2-android 的 API 来调用远端 Web Service 的服务。

因此需要在 CitiesWeatherForecast.java 的 onCreate（）中进行数据获取的初始化工作。通过定义一个方法 public List<String>getAllCitiesNAMEs（）可获得中国城市名称列表。也就是说，采用它提供的获得全球各个国家城市的方法 GetCitiesByCountry, 通过 http://www.webservicex.net/globalweather.asmx 上的 WebService 可提供天气预报服务。譬如，通过 Constant.java 中的 public class Constant{},设置 SOAP Action 要调用的方法名、命名空间以及 Web Service URL 值，其源代码为：

public static final String SOAP_ACTION=

"http://www.webserviceX.NET/GetCitiesByCountry";

public static final String METHOD_NAME=

"GetCitiesByCountry";

public static final String SOAP_ACTION2=

"http://www.webserviceX.NET/GetWeather";

public static final String METHOD_NAME2="GetWeather";

public static final String NAMESPACE=

"http://www.webserviceX.NET";

public static final String URL=

http://www.webservicex.net/globalweather.asmx;

然后在 WebServiceCaller.java 中实例化一个 SoapSerializationEnvelope 对象，设置 SoapObject 的命名空间、方法名、参数等；并通过实例化一个 AndroidHttpTransport 对象来调用 WebService,并获得 xml 字符串数据，其代码段如下：

AndroidHttpTransport androidHttpTransport=

new AndroidHttpTransport （Constant.URL）;

try {

androidHttpTransport.call（soapAction, envelope）;

Object result=envelope.getResponse（）;

xmlStr=result.toString（）;

} catch（Exception e） {

e.printStackTrace（）;

}

[2] 解析数据列表

Android 操作系统对 xml 字符串数据的操作功能很强，提供了 dom、sax 以及 xmlpul 3 种方式。Android SDK 提供了 android.sax 包以方便 SAX Handler 的开发，可用来解析 xml 结果字符串。在本设计实例的 CitiesWeatherForecast.java 文件中，解析 xml 数据时，先通过 RootElement root=new RootElement（"NewDataSet"）获得 xml 数据的根节点；然后再寻找其子节点直到找到 City 子节点，并设置一个监听器 setEndTextElementListener（）来获得 City 子节点的值；最后使用 org.xml.sax 的 SAXParser 来解析 xml 数据，把数据存入 List<String>并返回：

SAXParserFactory factory=SAXParserFactory.newInstance（）;

SAXParser parser=factory.newSAXParser（）；

XMLReader xmlreader=parser.getXMLReader（）；

xmlreader.setContentHandler（root.getContentHandler（））；

InputSource is=new InputSource

（new StringBufferInputStream（xmlStr））；

xmlreader.parse（is）；

[3] 在 UI 上呈现城市列表数据

当获得 List<String>类型的数据之后，就可以把它绑定到 main.xml 定义的 ListView 组件上，然后使用 ListActivity 呈现列表数据。在 CitiesWeatherForecast.java 中的代码段为：

List<String> cityList=getAllCitiesNames（）；

setListAdapter（new ArrayAdapter<String>（this,

android.R.layout.simple_list_item_1, cityList））；

即使用方法 setListAdapter（）把一个 ListActivity 填充进去。

2.3 城市天气实况预报系统的调试

城市天气实况预报系统的测试结果如下。在 Eclipse 的 Package Explorer 窗口中用鼠标右键选择 CitiesWeatherForecast 工程名，在弹出的窗口中选择"Run As"→"Android Application"安装该应用程序到 Android 模拟器并启动它。然后单击所要查询的城市名称列表项，稍等片刻便会显示出该城市的天气实况信息。也可以在文本框中直接输入所要查询的城市名称，单击"Search"同样会获得相应城市的天气预报信息。

3 结束语

Android 平台具有极大的开放性和兼容性，并且本身大量使用了开源代码库，深为开发人员所喜爱。Android 智能手机应用程序的开发涉及了它的整个体系结构，是一项非常复杂的工程。本文在介绍基于 Android 平台的应用程序设计原理的基础上，提出了 Android 用户界面设计、获取并解析城市列表数据的一种方法，给出了在用户界面上呈现数据的原理与设计过程，最后通过模拟器进行了应用程序的调试。当然，基于 Android 平台的开发技术还需要进一步完善，需要在日后的工作中不断探索、研究，以建立实用的城市天气实况预报系统。

作者：刘枫　　来源：《计算机时代》