

文章编号:1006-2475(2008)08-0085-05

基于 Android 的移动终端应用程序开发与研究

公磊,周聪

(南昌大学信息工程学院,江西 南昌 330031)

摘要:主要介绍了 Google 新近推出的移动设备软件平台 Android 以及基于此平台的移动终端应用程序开发。

关键词:移动设备; Android; 应用程序

中图分类号:TP311 文献标识码:A

Development and Research of Mobile Termination Application Based on Android

GONG Lei, ZHOU Cong

(Information Engineering School of Nanchang University, Nanchang 330031, China)

Abstract: The primary coverage is the Mobile equipment software platform of Google named Android and the development of Mobile termination application based on Android.

Key words: Mobile equipment; Android; application

0 引言

经过多年的发展,第三代数字通信(3G)技术获得了广泛的接受^[1],它为移动终端用户带来了更快的数据传输速率。随着 3G 网络的使用,移动终端不再仅是通讯网络的终端,还将成为互联网的终端。因此,移动终端的应用软件和需要的服务将会有很大的发展空间。Google 为此于 2007 年 11 月推出了一个专为移动设备设计的软件平台——Android。

Android 是一套真正意义上的开放性移动设备综合平台,它包括操作系统、中间件和一些关键的平台应用。Android 最大特点在于它是一个开放的体系架构,具有非常好的开发和调试环境,而且还支持各种可扩展的用户体验,Android 里面具有非常丰富的图形系统,对多媒体的支持功能和非常强大的浏览器^[3]。

Android 平台的开放性等特点既能促进技术(包括平台自身)的创新,又有助于降低开发成本,还可以使运营商能非常方便地制定特色化的产品,因此,它具有很大的市场发展潜力。

1 Android 软件平台的构成^[2]

Android 软件平台主要由五部分构成,它们分别是:关键应用程序、应用程序框架和组件、C/C++ 函数库、Java 程序运行环境、优化了的 Linux 内核。

1. 关键应用程序。

Android 平台内包含一些关键应用程序,如邮件收发客户端程序、短信收发程序、日历、网页浏览器等,而更多有特色的 Android 应用程序还有待于广大开发者共同参与开发。

2. 应用程序框架及组件。

Android 应用程序开发基于框架和组件。Android 本身已在其框架中提供了许多组件供应用程序调用,开发者也可在开发应用程序时顺带开发新的组件,并将该组件放入应用程序框架中,以供自己和其它应用程序调用。

3. C/C++ 函数库。

Android 应用程序框架之下是一套 C/C++ 函数库,它们服务于 Android 应用程序组件,其功能通过组件间接提供给开发者。这些函数库包括:标准 C 函数库、媒体功能库、浏览器引擎、2D 和 3D 图形库及

收稿日期:2008-03-17

作者简介:公磊(1984-),男,江西南昌人,南昌大学计算机应用专业硕士研究生,研究方向:嵌入式系统;周聪(1962-),男,副教授,研究方向:嵌入式系统。

SQLite 引擎等。

4. Java 程序运行环境。

Android 的 Java 程序运行环境包含一组 Java 核心函数库及 Dalvik 虚拟机,它们有效地优化了 Java 程序的运行过程。

5. Linux 内核。

Android 系统平台基于优化了的 Linux 内核,它提供诸如内存管理、进程管理、设备驱动等服务,同时也是手机软硬件的连接层。

2 Android 开发平台的配置和程序开发的基本组件

Android 平台为开发者提供了源码开放的软件开发包(SDK),它包含了使用 Java 语言开发 Android 平台应用的必要工具和组件(API)。进行开发前要先建立合适的开发环境,其中基于 Java 的整合型可扩展平台 Eclipse 是我们的首选。

2.1 搭建 Eclipse 开发环境^[5]

Eclipse 的运行建立在 JRE 之上,因此必须先安装 JRE。可以从 <http://java.sun.com> 下载 JDK5 或 JDK6,然后按提示进行 JDK 的安装;安装完成后,再配置环境变量:如在 Windows XP 或 Windows 2000 的操作系统中,可右键单击“我的电脑”,然后依次选择“属性”\“高级”\“环境变量”,在弹出的窗口的“系统变量”中找到“Path”环境变量,点击“编辑”按钮,在弹出的对话框中的“变量值”一栏输入 JDK 的安装路径即可,最后用同样的方法设置环境变量“Classpath”,至此便完成了 JDK 的安装。

有了 Java 运行环境,就可以下载 Eclipse ZIP 包,解压后就能直接运行 Eclipse.exe。

在 Eclipse 中安装一个名为 ADT 的 Android 开发工具插件,就可以使用 SDK 进行 Android 应用开发。ADT 安装步骤:在 Eclipse 中依次选择 Help > Software Updates > Find and Install,在弹出的对话框中选择 Search for new features to install 并点击 Next,选择 New Remote Site,在结果对话框中为远端站点命名(如 ADT),并在 URL 栏中输入 <https://dl-ssl.google.com/android/eclipse/>,点击 ok,随后 ADT 站点被添加到搜寻列表中,再点击 Finish 即可;在搜索结果对话框中选择 ADT > Eclipse Integration > Android Development Tools,单击 Next,接受协议,点击 Next 和 Finish;选择 Install All,并重启 Eclipse,ADT 安装完成。

ADT 安装完毕后,还要配置 Eclipse 参数,使其指向 SDK 根目录,过程如下:在 Eclipse 中选择 Window

> Preferences 打开参数设置面板,选择 Android 选项;在 SDK 路径中填入 Android 的安装路径,单击 Apply 和 ok,完成参数设置。至此,就能在 Eclipse 环境下进行 Android 应用的开发了。

2.2 Android 应用程序组件的使用

Android 应用程序以各种组件(API)为基础,API 是 Android 软件平台的重要组成部分,也是开发 Android 应用的关键所在。下面介绍几种主要的 Android API。

(1)当前活动程序 Activity:作为最常用的应用程序组件,几乎所有 Activity 都会通过创建 UI 窗口与用户进行交流。Activity 具有生命周期,它包含运行、暂停、停止等几个状态,状态间的转换可通过调用 onCreate()、onStart()、onRestart()、onResume()、onFreeze()、onPause()、onStop()、onDestory()等方法来实现。要开启新的 activity,可以调用 startActivity(Intent),或调用 startSub-Activity(Intent,int),后者通常和 onActivityResult(int,int,String,Bundle)联合使用,以从一个 activity 获取某结果并返回给打开它的 activity。

(2)图像显示 View:View 是最基本的 UI 构件,它在屏幕中占据一个矩形区域用来显示文本、图像等内容,还可完成图像事件处理。常用的 View 方法有:inflate()(由 XML 文件生成 View)、onDraw(Canvas)(对 View 进行填充)、onMeasure(int,int)(测量 View 的高度和宽度)、onLayout()(完成 View 的布局)、onKeyDown(int,KeyEvent)(View 中有按键操作时被调用)、onFocusChanged(boolean,int)(当 View 得到或失去关注时被调用)、setOnClickListener()(设置 View 点击事件监听器)。View 也包含一些子类,如:TextView(布局文件中的文本显示)、EditText(布局文件中的文本编辑)、Button(按钮)、ListView(显示列表)。

(3)事件 Intent:Intent 抽象描述了想要执行的某种操作,它与 startActivity 一起使用可以开启新的 activity,与 startService 一起使用可以开启后台服务。Intent 最重要的作用还是加载 activity,它是不同 activity 间的桥梁。Intent 也可以被认为是一种概括操作行为的数据结构,它主要包括 action(动作)和 data(数据)两部分,此外它还具有一些附加属性如 category(intent 种类)、type(数据类型)等。常用的 Intent 方法有 intent()(构造 intent)、addcategory()(添加 intent 类型)、getAction()(获取 intent 动作)、getData()(获取 intent 数据)等。

(4)数据存取与共享机制 ContentProvider:Cont-

entProvider 可实现应用程序的数据存取及不同应用程序间的数据共享。ContentProvider 为应用程序提供一些访问方法,包括:Query(Uri, String[], String, String[], String, String, String)(数据信息的查询), insert(Uri, Contentvalues)(在 Contentprovider 中添加新数据), updata(Uri, Contentvalues, String, String[]) (更新 Conntprovider 中的数据), delect(Uri, String, String[]) (删除数据), Gettype(Uri) (获取数据类型)。其中,Uri 标明了 ContentProvider 存储的数据类型,在执行访问方法时都需依据 Uri 作为路径标识来确定要访问哪个 ContentProvider。

(5) 后台服务 Service:Service 是运行在后台的应用程序组件,不直接与用户交互。当前活动程序可以使用 Context.StartService() 开启一项后台服务,还可以通过 Context.bindService() 与后台服务通信。

3 Android 应用程序开发例程

电话簿是一项必不可少的手机终端应用,本文以此为例来说明基于 Android 应用程序开发的基本过程。

3.1 项目的创建

首先,在 Eclipse 中依次选择 File > New > Android Project,在弹出的“New Android Project”对话框的 Contents 部分选择 Create new project in workspace,在“Project name”后输入工程名 Contactnote,在“Package name”后输入 com.google.android.contactnote,在“Activity name”后输入 Contactlist,在“Application name”后输入 Contact,最后点击 Finish,完成电话簿项目的创建。

3.2 项目源程序开发

电话簿源程序主要功能模块框架如下:

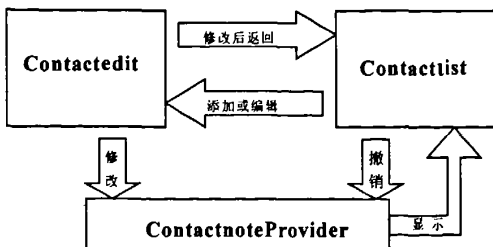


图 1 电话簿功能模块框架

(1) 构建数据存取和共享机制 ContactnoteProvider。

ContactnoteProvider 是电话簿的数据存储及访问组件,作为 ContentProvider 的子类,它必须定义一个

唯一的字符串标识 Uri 来指明自身存储的数据类型。ContactnoteProvider 的 Uri 可以这样定义:public static final Uri CONTENT_URI = Uri.parse("content://com.google.provider.Contactnote/contactnotes")。

ContentProvider 可使用文件存储方式或 SQLite-Database 来存储数据。本文选用了 SQLiteDatabase 并借助基于 SQLite 的类 SQLiteOpenHelper 来构建电话簿存储系统:

```

private static class DatabaseHelper extends SQLiteOpenHelper {
    public void onCreate(SQLiteDatabase db) {
        db.execSQL("CREATE TABLE contactnotes (_id INTEGER PRIMARY KEY," + "peoplename TEXT," + "phonenumber TEXT," + "modified INTEGER" + ");");
        } //创建基于 SQLiteDatabase 的电话簿数据表格 phonenumbers
    }
    
```

Android 平台内置了一套 SQLite 数据存储机制,并包含了一系列管理 SQLite Database 的相关方法,如创建、打开数据库以及执行 SQL 命令等,应用程序可直接使用这些方法来构建私有存储系统,在创建 ContentProvider 时也可以借用这些方法。上面的代码中,ContactnoteProvider 就使用了基于 SQLite 的 execSQL() 方法来构建电话簿的数据存储系统。本文之所以使用 ContentProvider 来存储数据是因为只有它能实现数据共享,而其他存储系统的访问仅限于单个应用之内。

ContactnoteProvider 实现数据共享的前提是能被所有应用程序访问,因此它提供了一些公开的访问接口,应用程序调用这些访问接口时用 Uri 标识确定访问路径。Uri 有两种形式,一种对应的是 ContentProvider 中的所有数据内容,另一种对应了具体的某条(行)数据。以 ContactnoteProvider 为例,com.google.provider.Contactnote/contactnotes 代表整个 ContactnoteProvider 数据的访问路径,而 com.google.provider.Contactnote /contactnotes/# 则代表了具体某条(行)数据的访问路径,其中#指明行地址标号。

ContactnoteProvider 访问接口的实现也可以借助相关的 SQLiteDatabase 方法,以下是实现 ContactnoteProvider 数据更新接口的部分代码,它直接调用了 SQLiteDatabase 的 update 方法:

```

public int update (Uri url, ContentValues values, String where, String[] whereArgs) {
    int count;
    switch (URL_MATCHER.match(url)) {
        case NOTES:
    
```

```
count = mDB.update(" notes", values, where, where-
Args);
break;...
```

调用时只需通过 UriMatcher. match(uri) 将数据更新接口 Uri 类型的路径参数解析为 SQLite Database 方法能识别的数据库表格形式。

(2) Activity 程序的创建与编辑。

当前活动程序 Activity 通常可实现应用程序与用户的交互,本实例包含两个当前活动程序:电话本列表显示程序 Contactlist 和电话本编辑程序 Contactedit。

Contactlist 是电话簿应用程序的入口,程序加载时将它开启。打开电话簿后首先看到的是按某种布局显示的联系人列表,列表内容和后台数据相对应。如果并未向后台数据库中添加任何信息,则列表显示内容为空,但列表的构架仍然存在。下面的代码部分实现了后台数据与列表内容显示的映射:

```
protected void onCreate(Bundle icle) {
super.onCreate(icle);
...
mCursor = managedQuery(getIntent().getData(), PRO-
JECTION, null, null);
ListAdapter adapter = new SimpleCursorAdapter(this, R.
layout.contactlist_item, mCursor, new String[] { Contactnote.
contactnotes, PEOPLENAME}, new int[] { android. R. id.
text1 });
setListAdapter(adapter);
}
```

首先通过 managedQuery (getIntent().getData(), PROJECTION, null, null) 来获取指向后台数据的指针 mCursor, getIntent().getData() 和 PROJECTION 共同决定了哪部分后台数据将被显示;然后,使用 SimpleCursorAdapter() 来定义后台数据到前台显示的映射方式,其中 R.layout.contactlist_item 定义了列表的布局,Contactnote.contactnotes. PEOPLENAME 指明显示内容为联系人姓名,android. R. id. text 指明姓名显示为文本;最后,通过 setListAdapter(adapter) 来激活映射。

光有列表显示还不足以实现 Contactlist 与用户的交流,真正的交互是通过创建可选菜单来完成的,而每个菜单选项都对应于用户的某种操作。Contactlist 使用 onCreateOptionsMenu() 来创建菜单:

```
public boolean onCreateOptionsMenu(Menu menu) {
super.onCreateOptionsMenu(menu);
menu.add(0, INSERT_ID, R.string.menu_insert).set-
```

```
Shortcut(3, a);
Intent intent = new Intent(null, getIntent().getData());
intent.addCategory(Intent. ALTERNATIVE_CATEGORY);
menu.addIntentOptions ( Menu. ALTERNATIVE, 0, new
ComponentName ( this, ContactList. class ), null, intent, 0,
null);
return true;
}
```

可选菜单中,“添加”菜单项 Insert 是被静态创建的,不管显示内容是否为空,该菜单项都存在。而对于那些在显示内容非空时才有意义的菜单项,可以用方法 menu.addIntentOptions() 来创建,这是一种动态创建方式,它创建的菜单项对应了某种事件 intent。使用前先用 intent.addCategory (Intent. ALTERNATIVE_CATEGORY) 将 intent 的类型设为 ALTERNATIVE_CATEGORY 或 SELECT_ALTERNATIVE_CATEGORY,以表明这些 intent 及其对应的 action 是可以变化的。

电话簿程序应该具备响应内外事件的能力,如电话拨打程序或短信发送程序向电话簿索取号码时引发的事件。Contactlist 必须先获取事件的 action 和 data,然后依据 action 进行分类处理。action 包括 VIEW_ACTION(向用户显示数据)、MAIN_ACTION(开启入口程序)、EDIT_ACTION(提供编辑入口)、PICK_ACTION(获取数据并返回)等不同类型。假如短信发送程序想从电话簿中获取号码,那么它会发出一个 action 类型为 PICK_ACTION 的 intent, Contactlist 接收并确认后执行 setResult (RESULT_OK, url.toString()) 将号码返回。如果是程序内部菜单触发的编辑事件,那么 Contactlist 会使用 startActivity (new Intent (Intent. EDIT_ACTION, uri)) 开启 Contactedit 完成编辑工作。

Contactedit 是电话簿中另一个交互程序,由 Contactlist 的添加或编辑事件引发,它为用户提供了联系人姓名和号码编辑窗口,用户完成编辑并确认后,Contactedit 调用 ContactnoteProvider 的 insert() 或 update() 方法将用户输入的值保存到电话簿数据库中去,然后返回 Contactlist,这时 Contactlist 显示的是更新后内容。

3.3 代码的运行

完成工程的创建和源代码的编辑后就开始运行程序。在 Eclipse 平台中选择 Run > Open Run Dialog 选项,打开运行对话框。然后,在对话框左边的 type filter text 下选择 Android Application > contactnote,再

点击对话框中的 Run 按钮,应用程序就会被加载到仿真器上并开始运行(如图 2)。

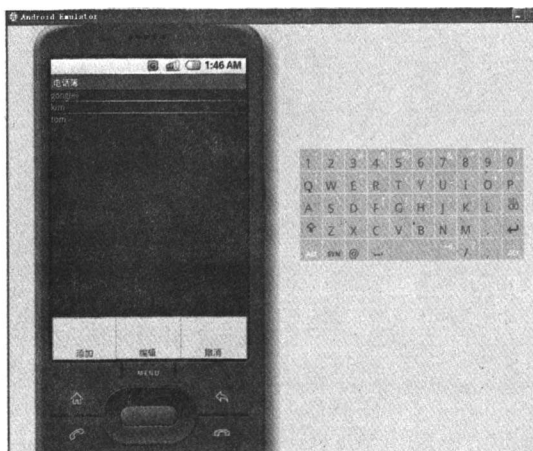


图 2 电话簿程序仿真运行

4 结束语

Android 手机应用程序开发需要着眼于它的整个体系架构,是一项复杂的工程。本文分析了 Android 体系的层次架构,并在此基础上介绍了 Android 应用程序开发的相关技术和基本的开发流程。Android 平

台具有极大的开放性和兼容性,相信广大开发者可以借助 Android SDK 提供的强大技术支持,开发出丰富多彩、极具个性的移动应用。

参考文献:

- [1] 3G 继续稳步快行[J]. 中国通信业,2007(2):50-51.
- [2] Code Home. Android-An Open Handset Alliance Project [EB/OL]. <http://code.google.com/android/what-is-android.html>,2008-01-01.
- [3] 邸烁. Android 揭秘和未来发展趋势[EB/OL]. <http://publish.itpub.net/zt/android/index.html>,2008-03-17.
- [4] 百度空间. Google Android 里头到底有什么? [EB/OL]. <http://hi.baidu.com/lujwc81/blog/item/12777c0754bc b8c97b8947c3.html>,2008-01-28.
- [5] 叶达峰. Eclipse 编程技术与实例[M]. 北京:人民邮电出版社,2006.
- [6] 张超,王阿川,王智. 基于 J2ME 和 J2EE 的手机软件的研究[J]. 黑龙江科技信息,2007(3):21,201.
- [7] 刁扣锁. 短信息在嵌入式 Linux 系统中的设计与实现[D]. 东南大学硕士学位论文,2005.
- [8] 刘向辉. 基于 Java 的移动终端中间件的研究与设计[D]. 华东师范大学硕士学位论文,2005.
- [9] 高博. 基于 J2ME 的移动通信技术的研发与应用[D]. 沈阳工业大学硕士学位论文,2006.

(上接第 84 页)

```

LoginBean.closeStmt();//关闭 Stmt 对象
LoginBean.closeConn();//关闭 Conn 对象
//使用 session 对象的 putValue()方法,把用户名 name1
赋给变量 username
session.putValue("username",name1);
% >
//如果登录成功,使用 javascript 语句直接进入 index_ok.
jsp 文件中
<script language = javascript >
window.location = "index_ok.jsp? mark = 成功登录"
</script >
<% |
//如果有错,则输出提示信息“用户名或者密码错误!”
else
{
rs.close();
LoginBean.closeStmt();
LoginBean.closeConn();
% >
<script language = "JavaScript" >
alert("用户或者密码错误!");
</script >

```

5 结束语

随着 Web 技术的发展,人们生活水平的提高,人们生活方式也发生了巨大的改变。网上购物的出现使人们可以足不出户,看遍世界,消费者可以用自我服务的方式来完成交易,而企业通过网站可以展示产品,与用户进行交流沟通,从而吸引顾客,扩大客户群。网上购物将对社会的生产和管理、人们的生活方式、法律制度和教育文化带来巨大的影响。

参考文献:

- [1] 杨学瑜,王志军,刘同利. JSP 入门与提高[M]. 北京:清华大学出版社,2004.
- [2] 刘荣华,张扬,聂颂,等. Oracle 数据库管理与应用系统开发[M]. 北京:人民邮电出版社,2005.
- [3] 鲁晓东,李育龙,杨健. JSP 软件工程案例精解[M]. 北京:电子工业出版社,2005.
- [4] 豆腐. JSP, ASP +, ASP, PHP 运行速度测试报告[EB/OL]. <http://www.itlearner.com/Article/2006/3038.shtml>,2006-05-05.

基于Android的移动终端应用程序开发与研究

作者: [公磊](#), [ZHOU Cong](#), [GONG Lei](#), [ZHOU Cong](#)
 作者单位: [南昌大学信息工程学院, 江西, 南昌, 330031](#)
 刊名: [计算机与现代化](#) **ISTIC**
 英文刊名: [COMPUTER AND MODERNIZATION](#)
 年, 卷(期): 2008, "" (8)
 被引用次数: 0次

参考文献(9条)

1. [3G继续稳步快行](#) [期刊论文]-[中国电信业](#) 2007 (02)
2. [Code Home Android-An Open Handset Alliance Project](#) 2008
3. [邸烁 Android揭秘和未来发展趋势](#) 2008
4. [百度空间 Google Android里头到底有什么?](#) 2008
5. [叶达峰 Eclipse 编程技术与实例](#) 2006
6. [张超, 王阿川, 王智 基于J2ME和J2EE的手机软件的研究](#) [期刊论文]-[黑龙江科技信息](#) 2007 (03)
7. [刁扣锁 短信息在嵌入式Linux系统中的设计与实现](#) 2005
8. [刘向辉 基于Java的移动终端中间件的研究与设计](#) [学位论文] 2005
9. [高博 基于J2ME的移动通信技术的应用](#) 2006

相似文献(4条)

1. 期刊论文 [丛秋波, 刘洋 MIPS:推动MIPS架构进入移动市场](#) -[电子设计技术](#)2010, "" (4)
 多年前, 当sandeep Vij还是Xilinx公司通用产品部的总经理兼公司副总裁时, 本刊记者曾与之有过工作来往, 应算旧识. 今年初, 悉闻他新任了MIPS公司总裁兼首席执行官. Xilinx是一家很有激情和创新的公司, Sandeep Vij在半导体行业也积累了20多年高级管理和营销的丰富经验, 他接手MIPS后将会给该公司带来哪些新的理念? 借sandeep Vij日前访华之际, 本刊记者就此进行了专访.
2. 期刊论文 [龙真 统一未来手机操作系统Google和33个巨头的联袂游戏](#) -[IT时代周刊](#)2007, "" (22)
 在万众期待中, 人们渴盼已久的Gphone并没有如约而至, 取而代之的却是此前业内人士并不看好的平台战略. 11月6日, Google宣布将与33家终端和运营企业组建开放手机联盟 (Open Handset Alliance), 并共同开发名为Android的开放源代码的移动操作系统. 据悉, 开放手机联盟旨在开发多种技术, 大幅削减移动设备和服务的开发和推广成本. 而Android则是一个完全整合的移动软件系统, 包括一个操作系统、中间件、便于用户使用的界面以及各类应用. 采用该系统的首台手机将于2008年推向市场.
3. 期刊论文 [韩群 Google手机战略曝光产业链各方反响不一](#) -[通信世界A](#)2007, "" (42)
 一直以来, 关于“Google手机”的传言不断在业内流传, 但究竟Google是直接发布手机终端, 还是发布一个手机平台, 一直没有定论. 11月5日, 谜底正式揭晓. Google联合多家合作伙伴正式成立“开放手机联盟 (Open Handset Alliance)”, 面向全球, 推出基于Linux的手机操作系统“Android”, 该平台是完全意义上的开放性、综合性移动设备平台, 将加快新型移动设备的研发速度. 专家表示, 该举措将对产业格局产生重大影响, 对此, 产业链各方反响不一.
4. 学位论文 [马志强 基于Android平台即时通信系统的设计与实现](#) 2009
 即时通信系统产生以来, 这种以网络为基础的、与其它在线用户进行信息交互的实时方式, 以其方便快捷的特点, 受到了对消息反馈即时性要求很高的商业和服务行业的青睐. 传统的即时通信应用大部分是在PC端实现的, 但随着无线传输网络的发展, 各种智能操作平台的推出提供了硬件基础, 在移动设备上的即时通信系统的研究成为了3G业务新的亮点. 本文为解决移动设备与PC端即时通信以及通信协议缺乏统一标准的问题, 分析了当前的研究现状, 设计并实现一个Android平台上使用Jabber协议的移动即时通信系统.
 本文采用软件工程的管理方法, 对项目的需求进行了分析, 完成了功能用例建模, 从HTTP传输、Jabber协议解析到客户端的各个模块部分进行分析设计, 提出了本系统的体系结构和整体架构设计方案, 阐述了系统实现应用的关键技术, 建立了可扩展的会话模型, 采用松散耦合的方式设计、实现了Android平台上基于Jabber协议的移动即时通信系统.
 经过测试运行, 结果表明, 系统完成的用户注册登录、好友管理、分组管理以及信息交互等即时通信功能能够顺畅运行, 解决了与不同通信协议对接进行交互的问题, 获得了良好的用户体验, 已用于实际工程项目中.

本文链接: http://d.g.wanfangdata.com.cn/Periodical_jsjyxdh200808026.aspx

授权使用: 华中科技大学 (hzkjdx), 授权号: 448652be-abf4-4730-bc2f-9e1400e5a4c4

下载时间: 2010年10月19日