

幻灯片播放无线控制系统模型^①

邢海峰

(内蒙古财经大学 计算机信息管理学院, 呼和浩特 010070)

摘要: 为了使 PPT 演示文稿使用者能够在工作或学习中更加方便高效的控制演讲稿的播放, 提升工作或学习效率、增强效果, 该文提出基于 Socket 数据传输技术的 PPT 播放无线控制系统设计模型, 并采用多种流行的计算机技术实现了该模型. 该系统模型的工作原理是用户使用手机作为控制终端, 通过 Socket 数据传输技术实现手机终端向 PC 机上的服务端发送控制信号, 进而实现跨硬件平台下用户对台式机上 PPT 播放的无线控制. 同时文中提出的设计模型可广泛的用于各类移动终端控制台式机的设计开发中.

关键词: 控制系统; 套接字; Robot; Android 应用开发技术; Activity

Wireless Control System Model of Showing Powerpoint

XING Hai-Feng

(School of Computer Information Management, Inner Mongolia Finance and Economic University, Hhht 010070, China)

Abstract: In order to make the PPT presentation of the user control the speech draft more convenient and efficient in work or study, to enhance the efficiency of work or study, this paper presents the design of the wireless control system model of showing PPT based on socket transmission, and realizes the model using a variety of popular computer technology. The working principle of the system model is that the user transmits the control information encapsulated in the socket to the server by the android mobile phone, and controls the PPT showing in the wireless way. At the same time, the design model presented in this paper can be widely used for the design and development of various mobile terminals to control the personal computers.

Key words: control system; socket; Robot; application development technology for Android system; Activity

PPT(PowerPoint)是日常办公常用软件, 它在各行各业中作为宣传、演讲及展示性工作的辅助工具被广泛使用. 正是该软件的普及使用, 使得各行业中与信息传播工作相关的活动变的方便、高效.

但是, 由于使用该软件播放演示文档时需不断的使用键盘或鼠标来控制播放进度, 这给演示者造成很大的空间局限性, 进而影响到整个宣讲工作的效果. 针对这一问题, 一些硬件制造商设计出了诸如电子教鞭等各种辅助外设来帮助人们改善工作不便, 虽然这解决了用户使用软件过程的不便性, 但也带来了新的问题: 1) 额外的硬件支持增加了软件使用成本; 2) 增加的额外硬件影响便携性.

根据上述的调查分析, 本文提出设计实现既节省

成本又不影响便携性, 并且改善用户使用 PPT 不便性的 PPT 播放无线控制系统模型.

1 设计原理

1.1 模型设计

本控制系统主要设计理念是利用现在流行的移动 Android 平台作为控制端, 通过普遍覆盖的无线 wifi 方式接收信息并传送进入局域网, 然后通过局域网把信号传递到受控端的 PC 机实现 PPT 播放无线控制. 控制系统模型如图 1 所示.

由控制系统模型可知, 该系统模型实现分两个阶段完成: 第一阶段, 当用户需要使用该系统时, 先通过移动终端向被控计算机发出一个连接请求, 当被控

^① 收稿时间:2014-08-30;收到修改稿时间:2014-09-30

计算机接受了该请求后向移动终端返回一个确认连接的请求, 此时, 移动终端和被控端的可靠连接就建立了; 第二阶段: 当移动终端与被控计算机之间的可靠连接建立以后, 用户就可以向被控计算机发送控制信息, 实现对被控计算机的控制。

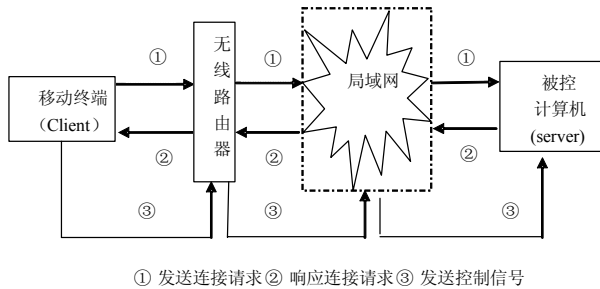


图 1 控制系统模型

由于 Android 移动平台几乎随处可得, 最常见的比如用户手里的 Android 手机就是最佳选择, 所以把用户的手机作为控制 PPT 播放的客户端即经济又不影响便携性, 那么就可以把客户端的控制程序部署到用户的手机中, 而被控计算机端可以开发部署一个负责接受、解析客户端信息, 然后根据解析信息来实现用户控制意图的服务器程序即可。因为用户控制自己的 PPT 播放是 1 对 1 的, 并且需要可靠信号传输, 所以我们可以采用基于 TCP/IP 协议的 Socket 技术来实现客户端及服务端的可靠数据传输。控制系统软件业务流程如图 2 所示。

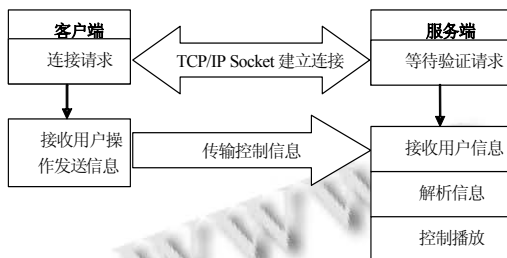


图 2 控制系统软件业务流程

1.2 核心技术

1.2.1 Socket 套接字

Socket^[1]也称作“套接字”, 应用程序通过“套接字”向网络发出请求或者应答网络请求。Socket 有两种实现方式, 即流式 Socket(SOCK_STREAM)和数据报式 Socket(SOCK_DGRAM)。

由于控制系统模型采用流式 Socket, 该“套接字”建立可靠连接是基于 TCP 协议, 所以连接过程分为三

个步骤完成, 分别是服务器监听、客户端请求及客户端确认连接。

当连接建立成功后, 真正的数据传输过程就可以开始了。客户端可以调用 Socket 提供的数据发送接口函数把要传输的数据发送给服务端, 而服务端通过调用数据接收接口函数完成数据接收。

在 Java 语言当中提供了两个类来封装了流式 Socket 的全部接口, 用于支持流式 Socket 的数据传输, 分别是 Socket^[2]类和 ServerSocket^[2]类。ServerSocket 类为服务端提供必要的支持, Socket 类为客户端提供支持。

1.2.2 Robot 类

为了实现服务端 PPT 播放的远程控制, 需要实现程序控制服务端键盘或鼠标的操作。这就需要一种技术支持来帮助程序实现控制键盘或鼠标的操作, 该项键盘或鼠标的程序控制技术在 Java 中早已实现, 它被封装在 Robot 类当中。

Robot 类可以模拟键盘和鼠标等基本输入设备的输入操作, 所以可以通过 Robot 类来控制计算机完成简单的操作。Robot 类常用于测试自动化、自运行演示程序及其他需要控制鼠标和键盘的应用程序。该类的功能是能生成键盘或鼠标的输入事件, 并将事件发送到 AWT 事件队列。但该类生成的事件与 AWT 组件事件又有区别, 该类的事件是在平台的本机输入队列中生成, 例如, Robot.mouseMove 将实际移动鼠标光标, 而不是只生成鼠标移动事件。

1.2.3 Android 应用开发原理

(1) Android 应用程序组成

Android 应用程序就是由组件^[3]组成的, 而组件是可以调用的基本功能模块。Android 系统有四个重要的组件, 分别是 Activity、Service、BroadcastReceiver 和 ContentProvider。

Activity 是 Android 程序的呈现层, 显示可视化的用户界面, 并接收与用户交互所产生的界面事件。Android 应用程序可以包含一个或多个 Activity, 一般在程序启动后会呈现一个 Activity, 用于提示用户程序已经正常启动。

Service 用于没有用户界面, 但需要长时间在后台运行的应用。

ContentProvider 是 Android 系统提供了一种标准的共享数据的机制, 应用程序可以通过 Content

Provider 访问其他应用程序存储在文件系统中的文件,也可以是SQLite中的数据库的私有数据. Android系统内部也提供一些内置的 ContentProvider, 能够为应用程序提供重要的数据信息.

BroadcaseReceiver 是用来接受并响应广播消息的组件, 不包含任何用户界面, 可以通过启动 Activity 或者 Notification 通知用户接收到重要信息.

(2) 组件生命周期

所有 Android 组件都具有自己的生命周期, 是从组件建立到组件销毁的整个过程. 在生命周期中, 组件会在可见、不可见、活动、非活动等状态中不断变化. 本节仅介绍控制系统中使用的 Activity 生命周期^[4], 如图 3 Activity 生命周期所示.

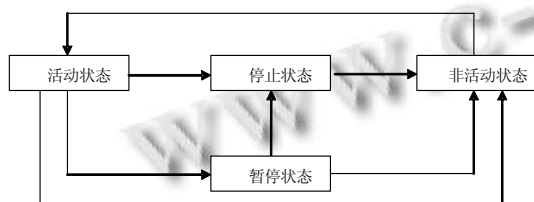


图 3 Activity 生命周期

Activity 生命周期指 Activity 从启动到销毁的过程, Activity 表现为四种状态, 分别是活动状态、暂停状态、停止状态和非活动状态.

活动状态: Activity 在用户界面中处于最上层, 完全能被用户看到, 能够与用户进行交互.

暂停状态: Activity 在界面上被部分遮挡, 该 Activity 不再处于用户界面的最上层, 且不能够与用户进行交互.

停止状态: Activity 在界面上完全不能被用户看到, 也就是说这个 Activity 被其他 Activity 全部遮挡.

非活动状态不在以上三种状态中的 Activity 则处于非活动状态.

(3) 组件的回调函数

Android 系统的 Activity 组件提供了 7 个回调函数来控制该组件的创建、状态变化及销毁, 分别为: onCreate、onStart、onRestart、onResume、onPause、onStop、onDestroy.

(4) Android 应用程序间消息传递

Android 系统中采用称为意图的 Intent 类来实现不同应用程序间的信息传递. Intent^[5]可以划分为显式意图和隐式意图两种. 显式意图: 调用 Intent.SetCompo-

nent()或 Intent.setClass()方法指定了组件名或类对象, 显式意图明确指定了 Intent 应该传递给哪个组件或类对象. 隐式意图: Android 系统会根据隐式意图中设置的动作(action)、类别(category)、数据(URI 和数据类型)找到最合适的组件来处理这个意图.

2 系统模型实现

2.1 客户端实现

2.1.1 界面设计

客户端的界面有两个: 第一个为客户端连接主界面, 该界面向用户提供连接服务端的操作接口, 客户端程序首先启动该界面; 第二个为客户端控制界面, 该界面向用户提供控制服务端行为的操作接口, 当连接成功时进入该界面. 界面设计方式有三种: 所见即所得, 适合设计简单界面; 通过配置 xml 文件设计界面; java 代码设计界面. 但在正式的开发过程中建议使用 xml 方式.

客户端主界面主要包含四个控件, 两个普通文本框视图框用于提示用户操作, 一个可编辑框接收用户输入服务端 IP 地址, 一个按钮来实现获取 IP 地址及将该地址传向控制界面操作; 根据客户端主要任务, 客户端控制界面主要包含 6 个按钮控件, 分别对应“向前翻页”、“向后翻页”、“从当前页播放”、“重新播放”、“结束播放”及退出程序六个操作. 这里不再给出相关 xml 配置文件, 运行效果详见文中图 4 所示.

2.1.2 功能实现

(1) 入口类定义

由于客户端是 Android 手机平台, 所以需要定义应用程序运行入口类, 并把应用程序运行入口类注册到主配置 xml 文件中, 注册配置代码如下:

```

<activity android:name=".MainActivity"
  android:label="@string/app_name">
  <intent-filter>
  <action android:name="android.intent.action.MAIN" />
  <category
  android:name="android.intent.category.LAUNCHER" />
  </intent-filter>
</activity>
  
```

该入口类继承了 Android 组件 Activity, 主要完成连接主界面显示, 为建立与服务端的 Socket 连接准备格式正确的 IP 地址, 通过 Intent 对象跳转到客户端控

制 Activity. 该类的定义及核心接口函数实现如下:

```
public class MainActivity extends Activity {
//成员变量定义省略
public void onCreate(Bundle savedInstanceState) {
第 1 步: 创建主界面, 初始化 IP 地址编辑框及连接按钮
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
initView();
```

第 2 步: 为按钮绑定鼠标单击事件, 该事件中实现 IP 格式合法检测, 建立 Intent 对象跳向客户端控制 Activity

```
btnConnect.setOnClickListener(new View.OnClickListener() { public void onClick(View v) {
String ipstr = edtTxtIP.getText().toString();
if (!chkIP(ipstr)) {
toastText("IP 格式不正确!");} else {
Intent put_intent =
New Intent(MainActivity.this,
ClientActivity.class);
put_intent.putExtra("ipAddress", ipstr);
startActivity(put_intent);}}});}
```

(2) 客户端控制类定义

客户端控制类同样也继承了 Android 组件 Activity, 也需要注册到主配置 xml 文件中, 注册配置代码如下:

```
<activity android:name=".ClientActivity" />
```

该类实现客户端的主要业务, 即向服务端发送建立 Socket 连接请求, 获得用户指令并发送到服务端, 类定义及核心功能接口函数实现如下:

```
public class ClientActivity extends Activity {
//成员变量定义省略
protected void onCreate(Bundle savedInstanceState)
{
第 1 步: 创建客户端控制界面, 对六个控制按钮初始化, 获取服务端 IP 地址
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_client);
//获取控制按钮开始
start = (Button) this.findViewById(R.id.btnStart);
.....
exit = (Button) this.findViewById(R.id.btnExit);
```

//获取按钮结束

```
Intent get_intent = getIntent();//从 Intent 中获得主
连接 Activity 传递的信息
```

```
ipaddress = get_intent.getStringExtra("ipAddress");
```

第 2 步: 建立与服务端的连接

```
try {socket = new Socket();//建立连接开始
```

```
SocketAddress sAdd = new
```

```
InetSocketAddress(ipaddress,SERVER_HOST_PORT);
```

```
int timeout = 5000;
```

```
socket.connect(sAdd, timeout);
```

```
CtoS = new ObjectOutputStream
```

```
(socket.getOutputStream());
```

```
StoC = new ObjectInputStream
```

```
(socket.getInputStream());//建立连接结束
```

```
} catch (Exception e1) {连接失败处理...}
```

第 3 步: 为控制按钮定义及绑定事件

```
start.setOnClickListener(pptClickListener);//为按钮
绑定事件
```

```
.....
```

```
exit.setOnClickListener(exitListener);
```

```
private OnClickListener exitListener = new
```

```
OnClickListener() {...};
```

// 实现控制事件

```
private OnClickListener pptClickListener = new
```

```
OnClickListener() {
```

```
public void onClick(View v) {
```

```
Poster poster;//自定义类, 封装控制信号
```

```
switch (v.getId()) {判断控制键被按下
```

```
case R.id.btnStart://处理开始播放
```

```
poster = new Poster(SHIFT_F5);
```

```
try { CtoS.writeObject(poster);} catch
```

```
(IOException e) {e.printStackTrace();}
```

```
break;
```

```
case R.id.btnStop:.....//处理停止播放
```

```
case R.id.btnRestart:.....//处理重播
```

```
case R.id.btnForward:.....//处理向前翻页
```

```
case R.id.btnBack:.....//处理向后翻页
```

```
default: break;}}};
```

```
private void conFail() { ...// 连接失败后, 控制按钮变灰不能使用}
```

```
private void reDraw(Button button, int id) { ...// 重画按钮}
public void toastText(String message) {...//打印提示信息}}
```

2.2 服务端实现

服务端程序主要任务有两个，首先利用 ServerSocket 来创建监听客户端连接请求，当连接建立之后，通过流套接字方式接收客户端的控制信息，解码信息及利用 Robot 类的接口函数实现对 PPT 播放的控制，该类的定义及核心接口函数实现如下：

```
public class Methods extends JFrame implements ActionListener {
    /*类成员变量及界面初始化、显示及相关其它操作在此不给出*/
    ....
    public void action() throws IOException,
    AWTException,ClassNotFoundException,
    InterruptedException {
```

第 1 步：初始化“套接字”，创建 Robot 类
ServerSocket server = new ServerSocket
(SERVER_HOST_PORT);
Robot robot = new Robot();
Socket socket;

第 2 步：监听客户端连接请求
do { //各种提示信息省略
 socket = server.accept();//监听连接请求
 CtoS = new ObjectInputStream
 (socket.getInputStream());
 StoC = new ObjectOutputStream
 (socket.getOutputStream());

第 3 步：接收控制信息，解析并实现控制
do {
 Poster poster = (Poster) CtoS.readObject();
 key = poster.getKey();
 switch (key) {
 case SHIFT_F5://实现播放
 robot.keyPress(KeyEvent.VK_SHIFT);
 Thread.sleep(20);
 robot.keyPress(KeyEvent.VK_F5);
 Thread.sleep(10);
 }
 robot.keyRelease(KeyEvent.VK_F5);

```
robot.keyRelease(KeyEvent.VK_SHIFT);
Thread.sleep(10);break;
case F5:.....//实现重播
case LEFT:.....//实现上一页
case RIGHT:.....//实现下一页
case ESC:.....//停止播放;}//一次控制结束
} while (key != -1);//一次连接结束
CtoS.close();
StoC.close();
socket.close();
第四步：一次连接结束，返回到第二步
} while (key == -1);
server.close();}}
```

3 系统部署测试

3.1 系统部署

手机客户端：首先确定手机 Android 平台版本为 2.3 或以上，因为客户端程序开发基于 Android 2.3 版本的开发包，然后下载该系统客户端生成的 apk 文件，直接点击安装即可。程序运行如图 4 所示。



图 4 手机端运行结果

把服务端程序编译生成 jar 包等直接复制到 C 盘根目录下，直接点击运行 jar 包。程序运行如图 5 所示。



图 5 PC 端运行结果

3.2 系统测试

首先保证局域网当中的无线 wifi 畅通, 打开手机的无线连接且查验连接成功, 然后启动 PC 端的服务程序, 打开需要播放的 PPT 演示文档, 最后打开手机终端的控制程序, 输入 PC 的 IP 地址确定, 进入控制页面. 测试结果如表 1 所示.

表 1 测试结果

手机按钮	幻灯动作	结果
上一页	向上前一页	正确
下一页	向后翻一页	正确
退出	退出手机控制程序	正确
开始播放	全屏播放	正确
重新播放	从第一页开始播放	正确
暂停播放	停止播放	正确

4 总结

本文使用当前流行的开发技术设计实现了能够对 PPT 播放进行无线控制的系统. 该系统不但解决了用户控制 PPT 播放过程的不便性, 而且无需额外硬件设备支持(wifi 覆盖普及, 本系统可与其它网络服务共用

wifi, 不需专门硬件), 节约使用成本, 同时手机作为客户端不影响便携性. 该系统的应用将对众多 PPT 使用者提升工作或学习效果形成积极的推动作用. 另外, 本文中提出的系统模型可广泛的应用到各种类似的控制系统中, 对诸如手持终端控制台式系统的控制系统都适用.

参考文献

- 1 Stevens WR. Advance Programming in the UNIX Environment. Addison Wesley, 1992: 450-452.
- 2 魔乐科技软件实训中心. Java 从入门到精通. 北京: 人民邮电出版社, 2010: 405.
- 3 董晓刚. 浅析 Android 系统的四大基本组件. 中国电子商务, 2013, (1): 39.
- 4 关晶鑫, 李永全. Android 中的 Activity 生命周期. 电脑知识与技术, 2013, (11): 2713-2715.
- 5 Cowley. The relevance of intent to human-Android strategic interaction and artificial consciousness. Robot and Human Interactive Communication, 2006: 480-485.