

# 基于 Android 的建筑设备物联网系统终端<sup>①</sup>

王少林<sup>1</sup>, 魏仁政<sup>1</sup>, 张来盈<sup>2</sup>, 唐威<sup>2</sup>, 田晨璐<sup>1</sup>

<sup>1</sup>(山东建筑大学 信息与电气工程学院, 济南 250101)

<sup>2</sup>(山东黄金地产旅游集团有限公司, 济南 250101)

**摘要:** 在建筑设备物联网系统环境下设计开发了基于 Android 平台的终端软件. 建筑用电设备物联网系统应用层的软件开发难点在于其子系统(包括电梯、空调、消防、照明等系统)数量众多且接口标准、数据来源各不相同, 不同子系统之间存在大量的异构数据. 本系统在 Android 平台上利用 EventBus 类设计了基于事件驱动的软件架构方法, 有效地降低各个功能模块之间的耦合性; 在 Android 平台上通过调用 Webservice 技术间接访问远程服务器, 解决了大量异构数据整合的问题. 该客户端软件人机界面简洁、美观, 可实现远程实时监控设备状态、修改配置信息等功能, 具有复杂度高、可扩展性强等优势. 该软件已应用于建筑设备物联网系统中, 运行结果证明了该设计是有效、可行的.

**关键词:** Android 平台; 建筑设备物联网; 事件驱动架构; Webservice 技术; 终端软件

## Android-Based Terminal Software for Internet of Things for Building Equipment

WANG Shao-Lin<sup>1</sup>, WEI Ren-Zheng<sup>1</sup>, ZHANG Lai-Ying<sup>2</sup>, TANG Wei<sup>2</sup>, TIAN Chen-Lu<sup>1</sup>

<sup>1</sup>(School of Information and Electrical Engineering, Shandong Jianzhu University, Jinan 250101, China)

<sup>2</sup>(Shandong Real Estate Tourism Group Co. Ltd., Jinan 250101, China)

**Abstract:** This paper presents the design and implementation of Android application for Internet of things for building equipment. There are two key points in the development of Android application for Internet of things for building equipment. One of the key points is that the forms of software interface and the data source are different between the subsystems, the other is the heterogeneous data in subsystems. To reduce the coupling of system architectures and improve the reusability of code, an approach for implementing event-driven is adopted. To access the remote database by Android terminals, Webservice is proposed. Users can monitor the remote devices' status and modify the configuration information anytime they want by using Android terminal. The user interface is very succinct and beautiful. This Android application has already been used in the lab of Shandong Jianzhu University.

**Key words:** Android platform; internet of things for building equipment; event-driven architecture; Webservice; terminal software

建筑设备物联网是将物联网技术应用到智能建筑中, 通过多种通信技术将建筑物内传统的用电设备接入网络, 从而获取到设备的实时状态并实现对设备的远程控制等操作, 最终将采集到的数据信息存储、共享, 为建筑节能、设备诊断等提供数据支持<sup>[1,4]</sup>.

随着网络接入技术的不断发展, 智能手机、平板电脑等手持智能移动终端设备一方面可以作为物联网

系统的数据来源, 另一方面可以成为物联网应用层的一部分, 集成系统底层的功能, 构建起面向各类行业的实际应用. 目前市场上出现的与建筑设备物联网系统密切相关的智能家居系统(比如海尔集团面向个人用户的 U-Home 系列产品)中, 移动终端有着举足轻重的地位. 和智能家居系统不同的是, 建筑设备物联网系统拥有数量更多的底层设备、更加丰富的软件接口

① 基金项目:山东省自然科学基金(ZR2012FM019);济南市科技局高校自主创新计划(201303008)

收稿时间:2015-05-20;收到修改稿时间:2015-06-23

和更为复杂的通信协议,这些都对基于 Android 的建和数据接口的实现等方面提出了更高的要求。

通过对国内各大 Android 应用商城智能控制类应用的调研可以发现,目前智能家居类的应用下载量较大,功能主要集中在对室内开关设备的控制,另外还有单独针对空调系统和视频监控系统的应。然而集成全部建筑设备物联网子系统的应用程序几乎没有,这主要是因为:①子系统数量众多,软件接口各不相同,且不同系统对实时性、安全性等要求有很大的差异②系统存在大量的异构数据。本文为解决建筑设备物联网系统终端开发所面临的上述问题设计了一种基于事件驱动的软件架构方式,并采用 WebService 技术处理大量的异构数据,取得了明显的效果。

## 1 建筑设备物联网系统架构

### 1.1 系统组成和原理

本系统可根据服务器所处的网络位置分为局域网系统和广域网系统,两种系统的差别之处在于:在局域网系统中智能移动终端设备(以下简称终端)只能依靠局域网中的 WiFi 热点与服务器建立连接,而广域网系统中终端可以通过多种渠道和服务器通信,比如依靠运营商提供的 GPRS 网络、3G、4G 网络等。两种系统的架构方式完全相同,架构图如图 1 所示。

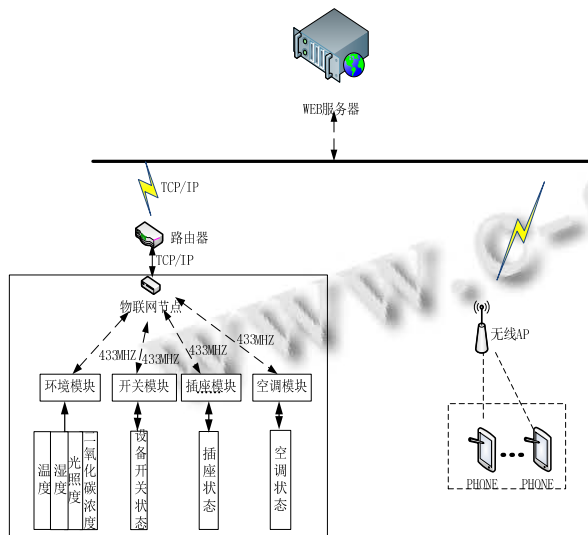


图 1 系统结构图

建筑设备物联网系统由服务器、物联网节点和物联网“感知”模块构成<sup>[2]</sup>。服务器位于建筑设备物联网系统的中心位置,其主要功能是利用其强大的计算能

力对接入的物联网“感知”模块进行实时的监控并且为应用层设备提供服务;物联网节点在建筑设备物联网系统中起到“承上启下”的作用,一方面,它收集附近物联网“感知”模块的状态信息送达通信服务器;另一方面,它接收来自通信服务器的控制命令然后转发给物联网“感知”模块;物联网“感知”模块主要包括环境模块、开关模块、插座模块和空调模块。环境模块起到监测室内环境的作用,根据实际情况可携带温、湿度传感器,热释电传感器,二氧化碳传感器,烟雾传感器的一种或几种;开关模块用于替换日常生活中常见的单联或多联开关,以实现远程监控,插座模块同理;空调模块根据空调厂商提供的协议通过 485 总线与空调室内机通信,可以获取空调当前的参数信息并且对其远程控制。

### 1.2 系统通信方式

建筑设备物联网服务器处于系统的中心地位,本系统中服务器在逻辑上采用三层架构:数据库服务器、Web 服务器和通信服务器。数据库服务器用于存储建筑设备物联网系统中的实时数据和配置信息;Web 服务器是建筑设备物联网 B/S 架构的一部分,用户可以通过浏览器访问发布在 Web 服务器上的网站获取存储在数据库中的设备和用户配置信息;通信服务器是信息的“中转站”,它解析来自物联网节点的实时数据并将服务器的控制命令向下转发。

终端接入 Internet 后通过访问发布在远程 Web 服务器上的 WebService 完成对数据库的增、删、改、查等基本操作,同时根据服务器 WebService 响应更新界面;另一方面,通信程序一直监视数据库 insertTeleCommand 表中的状态,一旦发现状态改变马上通过 Internet 给物联网节点下发控制命令,节点经过解析后通过 433MHz 频率的无线转发给“感知”模块,由此可见物联网节点的功能是使数据在 TCP/IP 协议和无线协议之间相互转换。

### 1.3 建筑设备物联网系统终端软件开发技术路线

目前,建筑设备物联网系统中包括电梯、空调、消防、安防监控、门禁、可视对讲系统在内的建筑用电设备运行模式各异,难以形成包容各种设备的网络通信接口标准,即使是同种设备(如中央空调)因为不同生产厂家采取的技术路线不同而难以形成较为统一的传感及控制方式。在这种情况下若不采用统一的网

络通信接口标准和数据传输格式将会大大增加软件开发难度和维护成本, 基于上述考虑本文选用跨平台的数据交换技术 WebService 作为客户端和服务端之间的通信方式.

在建筑设备物联网系统终端软件开发的需求分析中提出了一定的实时性要求, 比如说用户需要看到此时此刻家中接有大功率负载的插座是否处于关闭状态, 或者用户远程对空调下发开机命令后需要及时知晓空调设备是否动作, 即底层接收到数据后 UI 需要实时显示. 上述需求对于小规模软件开发通过实现回调接口就能满足, 但是建筑设备物联网系统子系统数量众多, 接口标准各不相同, 若都采用实现回调接口的方式则各模块间的耦合度将会大大提升, 不利于软件的维护和功能扩展, 基于此本文采用了基于事件驱动的软件架构方式.

## 2 关键技术探讨

### 2.1 WebService 技术

WebService 是一种基于网络的应用之间通信的标准化方法, 它使用开放的、标准化的协议为网络中的应用提供服务, 并且不依赖操作系统或者编程语言<sup>[3,6]</sup>. WebService 的关键技术包括 XML、SOAP、WSDL 和 UDDI, 其中 XML 起到描述数据信息的作用. 基于 XML 的 WebService 技术可以对建筑设备物联网系统中存在大量异构数据格式进行有效的转换和整合. 另外, 由于 XML 文档包含了数据和数据的格式, Android 终端接收到 XML 文档之后就可以直接根据 XML 的标签理解数据的含义并以恰当的方式显示在用户界面上.

本系统终端软件通过 WebService 与服务器通信的工作流程如图 2 所示. 终端软件数据访问层根据业务逻辑层的需要生成 SOAP 请求, 并将其内嵌到 HTTP 中发送到建筑设备物联网 Web 服务器, WebService 会接收到该 SOAP 请求并通过调用相应的方法做出 SOAP 应答, 应答同样以 HTTP 的方式被送回到客户端.

采用 WebService 技术之后无论何种格式、何种来源的数据都会以 XML 的形式被移动终端所接收, Android 终端 app 需要做的仅仅是利用 XmlPullParser 类对接收到的 XML 数据进行解析, 然后按照与上位机之间规定好的协议解析收到的数据而不需要考虑具体的数据来源. 同理, 用户在终端界面上的操作也由

程序按照协议转换成控制命令, 通过 WebService 发送到服务器. 这样实际用电设备采集的数据无论是模拟量、开关量亦或者是函数数据及音视频数据都不会被终端单独考虑, 且终端发出的控制命令也不会因为被控对象的不同而改变数据格式.

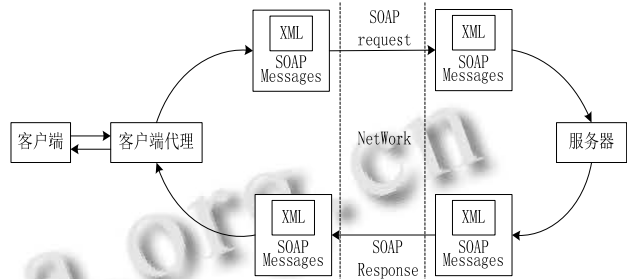


图 2 WebService 工作流程图

### 2.2 基于事件驱动(Event-Driven)的 Android 编程方法

根据建筑设备物联网移动终端设备的需求分析, 客户端界面需要得知当前“感知”设备的状态信息, 比如环境模块中的人员数量信息 n. 为了获取 n 的具体值, 在本系统中需要通过 WebService 与服务器通信, 具体通信过程是通过将 SOAP 请求绑定到 HTTP 协议之上进行数据交换的, 而 HTTP 连接是一种“请求-响应”的方式, 即客户端需要先与服务器建立连接然后才能发出获取数据 request, 服务器 response 后马上断开连接, 因此这是一种“短连接”方式且服务器不会主动向客户端发送数据. 按照传统的编程方法, 程序需要定时向服务器发出查询请求, 根据服务器的响应信息实现回调接口完成用户界面更新. Android 平台下的一种实现方式是工作线程读取到数据后利用 Handler 类的 sendMessage 方法将数据发送到 UI 线程, 然后在 UI 线程中调用 handleMessage 方法获取数据并显示. 这种方法在规模比较小的项目中是可行的, 但是建筑用电设备的特点各不相同, 不同系统(如空调、消防、安防、视频等)运行模式各异, 产生了不同的软件接口, 因此若采用实现回调接口的方法就必须另外定义一个 Handler 管理类, 这会导致代码的耦合度大大提高, 不利于后期的扩展和维护. 考虑到以上问题, 本次设计采用了在 Android 平台上基于事件驱动的软件架构方法.

事件驱动可以定义为程序执行流程由动作触发的事件决定. 事件驱动的架构可以降低模块间耦合程度. Android app 开发常常采用 MVC 的架构方式, 但是传统的 MVC 架构模式在遇到大型项目时就需要手动在

各个层次之间添加数据接口,上述情况中若采用 MVC 架构程序员需要为子线程逐个编写管理 Handler 的接口类,这样各层之间耦合性就会变高,且后期维护代价很大。基于事件驱动的编程往往包含一个事件总线,分别由不同的模块发送或监听事件,任何模块在不了解内部逻辑的情况下,只要监听到有事件到来就可以执行动作。显而易见,这种架构的简洁性和可维护性都将大大提高。

在 Android 系统中有很多基于事件驱动的实例,比如 Activity、Service 等组件在其整个生命周期中的各种状态。现以 Activity 组件为例简要说明如下:当 Activity 组件被创建时, onCreate 方法就会被调用;按下 Home 键当前 Activity 会被终止,系统就会调用 onPause 方法,随后调用 onStop 方法;当系统内存紧张 Android 会回收 Activity 占用的内存,与此同时 onDestroy 方法会被调用,即 Android 系统会根据监听到的事件调用执行相应的 onEvent 方法。但是 Android 系统不是完全由事件驱动的,即它不可能监听所有的事件,比如在 Android app 开发过程中由后台线程读取的数据、其他组件如 Service 传来的数据等事件都必须通过程序员自定义管理类或回调接口来实现监听。

本文提到的建筑设备物联网客户端的软件架构便是采用 Android 系统中常用的基于 EventBus 的事件驱动架构方式。EventBus 是一种发布者/订阅者(Subscribe/Publish)的设计模式,可以实现组件、服务和线程之间最大松耦合。EventBus 的结构图如图 3 所示。图中事件总线(EventBus)是其他实体之间通信的通道,app 在运行过程中所发生的一切动作都可被看做事件。订阅者可以通过向 EventBus 类注册获取监听事件总线的权限,一般包含有一个或多个 onEvent 方法。发布者可以向事件总线发送事件。

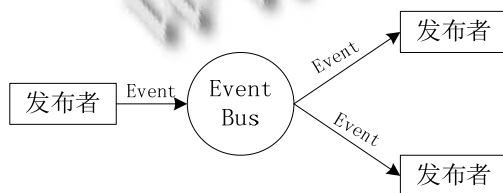


图 3 Web 信息抽取流程

### 3 建筑设备物联网客户端软件的实现

#### 3.1 开发环境

Android 开发目前常用的 IDE 有两种: Eclipse 和

Android studio. Eclipse 本身是一个使用 java 语言开发的开源 IDE, 开发 Android 应用时需要安装 Google 提供的 ADT 插件, 并下载相应的 Android SDK 版本, 本系统在 Android4.0 版本上开发和测试. Android studio 是由 Google 公司在 2013 年推出的一项全新的基于 IntelliJ IDEA 的 Android 开发环境.

两款 IDE 都非常适合用于 Android 开发, 由于 Android Studio 发布时间不长, 在调试过程中难免出现 BUG 且网络上可参考资源相对较少, 因此本次基于 Android 平台的建筑设备物联网系统客户端的设计采用 Eclipse 开发环境.

#### 3.2 系统数据交互

智能终端设备通过 Webservice 与远程服务器通信并实现对数据库的增、删、改、查等操作, 通信过程如图 2 所示. 由于 Android SDK 中并没有提供调用 Webservice 的库, 因此本系统在开发过程中引用第三方开源类库 ksoap2. ksoap2 是一个体积小、轻量级的库文件, 非常适用于移动终端设备. 目前在 Github 上发行的 3.4.0 版本是 ksoap2 的最新版, 该版本修正了之前网络信号弱时出现的内存溢出的 BUG. Android 平台上调用 ksoap2 需要导入相应的 jar 包.

使用 ksoap2 调用 Webservice 的具体实现代码如下:

```

String nameSpace="http://tempuri.org";//命名空间
//xxxxxx 是 Webservice 查询数据库方法名称
String methodName = "xxxxxx";//可从 WSDL 获得
String endPoint = "xxxxxx";
String soapAction="http://tempuri.org/xxxxxx";
SoapObject rpc = new SoapObject(nameSpace, methodName);
//为方法设置需要传入的参数, str1 是方法名 str2 是值
rpc.addProperty("str1","str2");
//本软件使用 SOAP 版本为 VER12
SoapSerializationEnvelope envelope =
new SoapSerializationEnvelope(SoapEnvelope.VER12);
envelope.bodyOut = rpc;
// 本系统中 Webservice 是使用 .Net 4.0 平台发布的
envelope.dotNet = true;
envelope.setOutputSoapObject(rpc);
HttpTransportSE transport =
new HttpTransportSE(endPoint);
try {
transport.call(soapAction, envelope);
  
```

```

SoapObject object = (SoapObject) envelope.bodyIn;
//获取返回的结果
String result = object.getProperty(0).toString();
} catch (Exception e) {
e.printStackTrace();
}

```

Android 客户端调用 Webservice 时, 需要预先知道发布 Webservice 的 Web 服务器地址, 以便向此地址发送 SOAP 请求, 然后向被调用的 Webservice 方法传入参数以及向 ksoap2 库提供 Webservice 的版本信息等, 这些相关信息都可以通过在浏览器地址栏键入 Web 服务器提供的地址查看 WSDL 文档, 在此文档中就可以查看上述程序中需要的参数信息, 比如 SOAP 版本、可使用的名称、返回的数据类型等。

### 3.3 客户端 UI 设计

本系统 app UI 设计在满足功能需求的前提下, 尽量保证界面简洁、美观<sup>[7,8]</sup>, 主要包括欢迎界面、登陆界面、模式选择界面、门禁监控界面和控制列表界面等, 另外还有自动更新和修改配置界面。部分界面截图如图 5 所示。其中控制列表界面包含了建筑设备物联网系统的主要功能模块, 如图 4 所示。

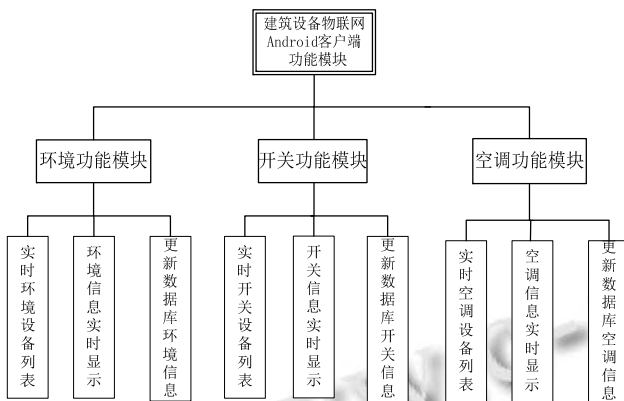


图 4 Android 客户端主要功能模块图

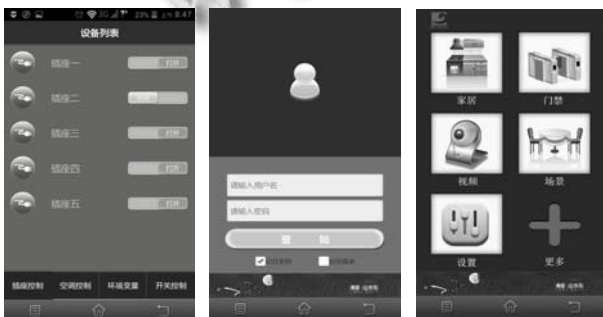


图 5 Android app 运行截图

控制列表 Activity 是由环境功能模块、开关功能模块和空调功能模块三个 Fragment 组成, 用户可以通过左右滑动屏幕选择相应的界面。Fragment 和 Activity 之间可以通过 Android 的 Bundle 类或者通过在 Fragment 内实现回调接口进行通信。三个模块界面相似, 都是通过 ListView 显示在线设备, 通过 Switch 控件改变“感知”设备的状态, 界面上的设备开关状态都对应实际状态。模块界面如图 5 所示。

### 4 结语

基于 Android 的建筑设备物联网系统客户端是山东建筑大学山东省智能建筑技术重点实验室开发的建筑设备物联网系统的重要组成部分, 目前该系统已集成到学校智能建筑信息集成实验室供学生实验时使用。本文设计和实现的客户端软件属于实验室教学演示环节的一部分。目前, 本文设计并实现的客户端软件已作为山东掌控物联网科技有限公司针对某高档住宅小区开发的建筑设备物联网系统的一部分投入使用。

本文设计并开发了基于 Android 平台的建筑设备物联网系统客户端。根据建筑设备物联网系统存在大量异构数据的现象利用 Webservice 技术作为客户端与远程服务器通信的桥梁; 为了降低模块之间的耦合性, 提高软件的可维护性和可扩展性本文采用基于事件驱动的软件架构方法, 在 Android 系统中利用事件总线 EventBus 类实现这一架构。本文实现的客户端软件已在商业化的使用过程中证明了相关理论和技术在工程应用中的可行性。

### 参考文献

- 1 鹿曼. 基于 Android 的智能家居控制系统的设计与实现[硕士学位论文]. 济南: 山东建筑大学, 2013.
- 2 申斌, 张桂青, 胡华, 等. 建筑设备物联网平台下谐波监测系统的设计与实现. 计算机应用与软件, 2013, 30(5): 36-40.
- 3 王少林, 王越, 申斌. 基于 SOA 的建筑设备物联网体系架构研究. 计算机技术与发展, 2014, 24(1): 196-199.
- 4 谢秀颖, 彭伟, 张桂青, 等. 基于电力载波技术的建筑设备物联网节点研制. 计算机工程与设计, 2013, 34(5): 1651-1656.
- 5 林小捷. 基于 Android 自动化测试平台的研究与实现[硕士学位论文]. 广州: 华南理工大学, 2013.
- 6 李莉, 高峰. Webservice 技术综述. 信息系统工程, 2014, (1): 36-36.
- 7 章玉宛. 浅析 UI 设计中字体设计. 美与时代: 城市, 2014, (3): 74-74.
- 8 郑卉, 邓晶. 移动软件开发中的 UI 设计. 中国科技信息, 2014, (9): 153-154.