

文章编号: 1006-2475(2013)02-0056-05

基于 REST 风格的物联网应用开发模式探究

陈 昊, 李士宁

(西北工业大学计算机学院, 陕西 西安 710072)

摘要: 伴随着网络技术的快速发展, 人类进入了以服务为基础的互联网时代。采用基于 REST 风格的服务, 有效地解决了传统的以 SOAP 为基础的 Web Services 使用过于复杂, 局限于 POST 方法, 很少使用 HTTP 响应代码等方面的问题。本文设计一个基于 REST 的物联网应用系统开发模式, 基于一个具有 REST 风格的物联网业务平台, 将传感器节点等智能感知终端和感知数据看作资源, 使用户可以采用 Django 框架来简便、快速地开发物联网应用。

关键词: 物联网; 传感器; REST; 业务平台

中图分类号: TP393

文献标识码: A

doi: 10.3969/j.issn.1006-2475.2013.02.014

Exploration of Application Development Model for Internet of Things Based on RESTful

CHEN Hao, LI Shi-ning

(School of Computer Science, Northwestern Polytechnical University, Xi'an 710072, China)

Abstract: Service-based Internet era has been formed with the rapid growth of network technology in recent years. Traditionally, SOAP-based Web services rarely use HTTP response codes and exploit HTTP GET and POST methods only, increase complexity of application programming, but RESTful Web services reduce this complexity. This paper introduces a development model for RESTful applications in Internet of Things. Based on RESTful platform of Internet of Things, wireless sensor nodes and data are abstracted as universal resources to provide rapid prototyping using Django framework.

Key words: Internet of things; sensor; REST; business platform

0 引言

随着无线接入技术的快速发展, 通信网络的异构性特征变得更加突出, 如何有效地实现不同网络、不同设备间的互联互通以及获取所需的各类服务, 并构建融合各网络异构性的无线物联网成为业界关注的重点^[1]。同时, 网络发展的最终目的是更好地向用户提供他们所需的各种业务。传感网作为物联网的重要组成部分, 由于物联网业务的拓展受网络能力、终端能力、业务开发模式等多种因素影响, 原有的传感网开发模式已不适用。这使得未来物联网整体架构的设计需要综合考虑终端、运营体系以及应用管理系统开发模式等多方面因素, 以满足物联网业务发展的需求。因此, 有必要对物联网业务环境的体系架构及其新型的开发模式进行前瞻性的研究, 为快速推动物联网业务的发展打下良好的基础。

1 研究背景

1.1 传感网与物联网

传感网是利用各种传感器(光、电、温度、湿度、压力等)加上中低速的近距离无线通讯技术构成一个独立的网络, 是由多个具有有线/无线通讯与计算能力的低功耗、小体积的微小传感器节点构成的网络系统^[2], 简单地理解, 可以认为它将重点解决局域或小范围的物与物的信息交换, 是物联网末端采用的关键技术之一。

物联网^[3-5]是一个基于互联网、传统电信网等信息承载体, 让所有能够被独立寻址的普通物理对象实现互联互通的网络。它具有普通对象设备化、自治终端互联化和普适服务智能化 3 个重要特征。

物联网具有 3 个特性: 全面感知、可靠的传送和

收稿日期: 2012-12-24

基金项目: 国家科技重大专项项目(2012ZX03005007)

作者简介: 陈昊(1988-), 男, 山东济宁人, 西北工业大学计算机学院硕士研究生, 研究方向: 无线传感器网络; 李士宁(1967-), 男, 陕西西安人, 教授, 博士, 研究方向: 移动计算, 无线传感器网络。

智能作用^[6],其控制特性与传感网有很大的不同,主要强调系统的可控性,而传感网则强调系统的可观测性^[7]。

1.2 传感网应用开发模式

如图1所示,传感网^[8]应用整体架构通常按照网络内数据的流向及处理方式分为3个层次:一是感知层,以二维码、RFID、传感器为主,实现对人、物或环境状态识别、感知;二是传输层,通过现有的互联网、广电网、通信网,实现数据的传输、计算和存储;三是应用层,输入输出控制终端,包括电脑、手机、笔记本等终端^[9]。

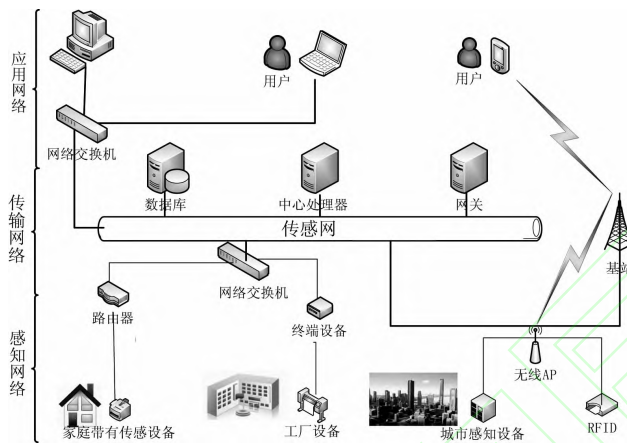


图1 传感网应用架构

对于传感网应用设计来说,需要对整个项目的感知网络、传输网络以及应用网络进行整体架构的设计、部署和开发。基础硬件建设工作不仅繁杂,而且设备资源重复利用率低。物联网中同样存在着这些问题。如何提高物联网应用开发的效率呢?假设感知网络由智能终端供应商进行统一的部署,传输网络由网络运营商进行统一运营。在这样一种状态下,基础硬件的建设完全通过配套设施的形式作为物联网应用开发的辅助,从而减少投入,提高效率。

2 新型物联网应用开发模式

2.1 系统结构

本文提出的新型物联网应用开发模式采用基于REST^[10-12]风格的物联网系统架构,此架构主要由应用管理系统、业务支撑平台和终端网络等3部分组成,如图2所示。

终端网络:在整个物联网体系中,终端网络是由各种具有REST风格,支持即插即用的传感设备和视频设备等终端组织成的网络。应用管理系统开发者不需要关注终端网络的具体实现,只需要通过相应业务平台提供的API接口或者其他第三方管理系统开

发者提供的可进行二次开发的API进行适合自身需求的Web Services^[13-14]开发,这样通过业务平台就能够实现具体的业务需求。

业务支撑平台:从总体上来看业务支撑平台是一个基于REST形式的中间服务平台,具有发现终端网络中节点、统一管理和定义终端网络中节点资源、生成资源描述信息、连接应用系统和终端网络的功能。从内部来看,如图3所示业务支撑平台可以分为3个层次:L1层主要实现资源的发现,对资源的配置,进而将资源转化为服务(数据)发布到L2层的功能;L2层负责应用科学的方法处理,分析服务(数据),并为L3层提供数据,操作接口(API);L3层面向规模最大的第三方应用系统开发者,支持他们开发各类具有REST风格的Web Services。业务平台以上这些功能的实现,几乎完全依赖于应用管理系统开发者,业务支撑平台本身起到管理、维护、运营等功能。

应用管理系统:基于REST风格的Web Service,可以为用户提供友好的、清晰的、易操作的可视化操作界面,用户通过验证登录应用管理系统之后生成授权用户对应功能权限的暴露URL,当用户向管理系统发出请求后,管理系统会根据用户发出的请求反馈对应数据。开发者可以根据自身技术背景来选择开发方式,本文的应用案例是采用Django框架,配合Ajax、CSS3、HTML5等技术来实现的。

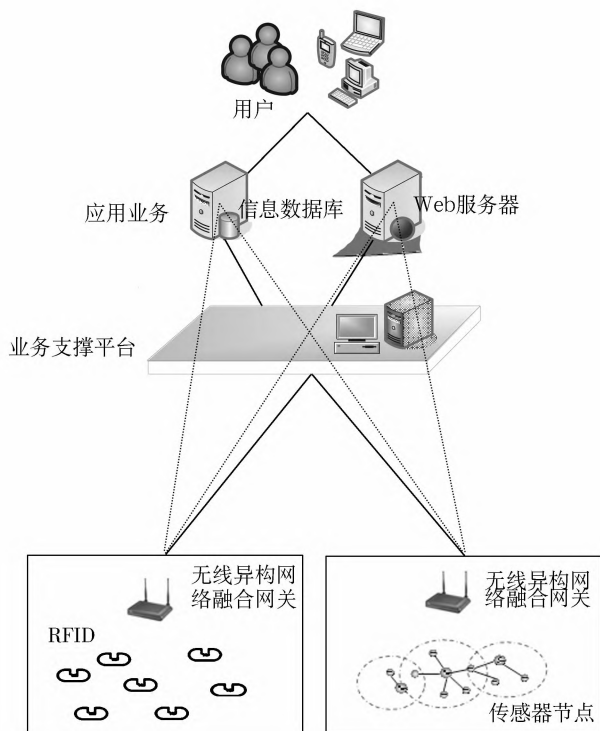


图2 新型物联网应用体系结构图

针对物联网业务集成的复杂性问题^[15],如图2

所示,本文依托具有 REST 风格的物联网开放混搭业务支撑平台,将传感器节点等智能终端和获取的感知数据作为资源,以服务的形式统一注册到业务支撑平台,将传统的一体开发模式拓展为通过对业务支撑平台进行服务注册,经过支撑平台的鉴权和认证等一系列验证,利用支撑平台提供的接口进行开发。

2.2 服务注册

服务注册是按照服务的用途和使用参数以及管理要求,在服务目录中进行登记,并对其属性进行配置的过程。各种类型的智能终端(含传感网网关)和多厂商业务平台,按照统一开放的标准协议接入业务支撑平台。业务支撑平台为具有不同能力的终端提供认证鉴权及业务访问能力,同时能够对接入的终端进行动态调整和控制,保证平台的可靠运行。统一接入运营商的通信网络,为应用系统开发者和终端接入提供统一通信能力的访问渠道,同时为业务平台提供通信能力的接口。

2.3 应用接口

应用接口可分为 3 类:

(1)注册接口(ClientLogin API):应用系统开发者取得业务支撑平台服务授权的功能接口,通过此类平台接口可以将相应的 Web Service 通过业务支撑平台的鉴权认证。

(2)查询接口(SearchServe API):第三方应用系统开发者通过业务支撑平台查询接口,可以在平台 API Pool 里查询可以实现其需求的相关功能的 API。

(3)调用接口(InvokeServe API):通过业务支撑平台的调用接口来调用第三方应用系统开发者业务支撑平台查询接口查询到的相关 API 来实现用户需求功能。

2.4 应用开发流程

应用管理系统的开发本身依托于物联网的整体架构,在完成终端网络的部署后,业务支撑平台作为终端网络和应用管理系统的中间服务平台,统一管理和定义终端网络中的节点资源,为应用管理系统的开发做支撑保障。

开发者在这整个体系中充当了重要的角色,如图 3 所示,同样也分为 3 个层次:LL 层,位于这一层的开发者面向的是平台最底层的资源接口,这些资源就是各个不同的终端网络,LL 开发者的目标就是要对这些网络进行配置,使其具备提供服务的能力;随着服务进入业务支撑平台的 L2 层,ML 级的开发者将对其进行处理,对服务(数据)进行处理,以 API 的方式将这些方法提供到业务平台的 L3 层;HL 层的开发者

面对平台提供的 API 进行各种各样应用的开发。

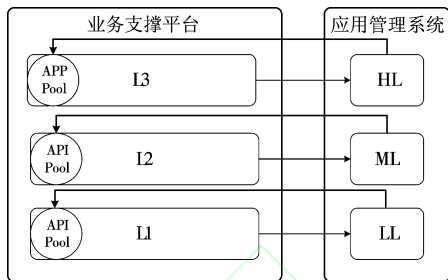


图3 业务支撑平台与应用管理系统结构图

LL 级的开发人员可以称之为系统级的开发人员,他们面向的主要是 L1 层资源列表中所涉及的资源,对这些资源进行处理,编写应用或者提供相应 API 接口。

ML 级的开发者主要关注于业务平台 L2 层服务列表的服务信息,在实际应用中,这些服务信息大多以数据的方式呈现,因而,ML 级开发者所涉及的应用主要目标在于进行数据处理。

HL 级的开发者的产品主要是面向用户需求来进行针对性开发。HL 级开发者通过调用应用系统需求中涉及的对功能 API 进行应用开发,他们面向的是具体的应用场景,比如果园监测,游泳馆环境监测,智能教室监测等。

3 应用案例

3.1 项目背景

陕西是苹果大省,在全球苹果生产中占有重要位置。项目面向苹果种植精准管理需求,优化无线感知终端设计,将感知终端部署在果园中,终端之间自组织成网络,通过即插即用方式接入运营商的融合网关。以杨凌和洛川为示范基地,感知终端根据作物生长长期的需求,以适合的采样周期采集监测区域的环境参数。同一区域的终端以协作方式将监测数据汇聚到融合网关,通过混搭业务支撑平台将数据转发到用户的应用管理系统。通过应用管理系统的功能模块对获取的数据进行分析和处理。

3.2 网络结构

本系统由 3 部分构成:面向农业智能感知传感器节点——用于采集空气温度、土壤湿度和光照强度等数据;数据服务提供者——智能感知终端将节点服务抽象为资源,周期性地产生数据事件,将监测数据通过业务平台推送到应用平台中的数据库服务器;应用管理系统——应用开发者通过业务平台,或者调用数据库服务,获得查询的历史数据,或者访问节点资源,获取果园当前监测数据,根据项目需求采用 Django

框架来设计和实现基于 REST 特性面向农业的应用管理系统。系统结构如图 4 所示。

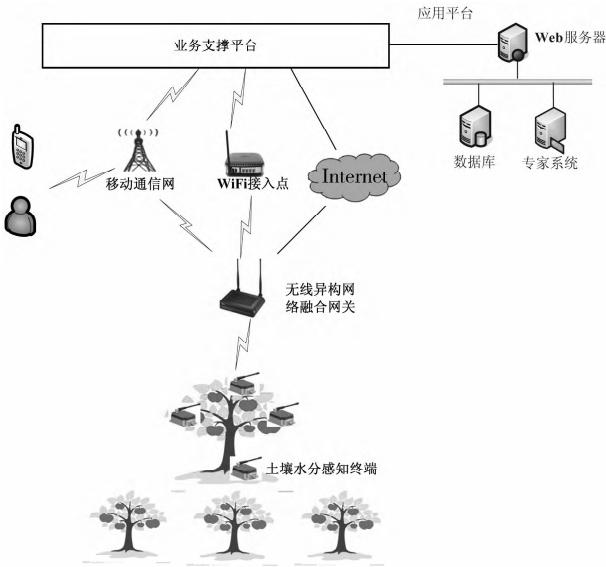


图 4 案例系统结构图

3.3 设计与实现

基于 REST 风格的物联网是以资源为中心,任何事物,只要具有被引用的必要,它就是一个资源。在 REST 的世界里,整个 Web 被看作一组资源的集合。每个资源必须至少有一个统一资源标识符 URI,URI 既是资源的名称,也是资源的地址^[16]。REST 中的资源是数据和表现形式的组合,以资源为核心的设计思想是 REST 的核心所在。

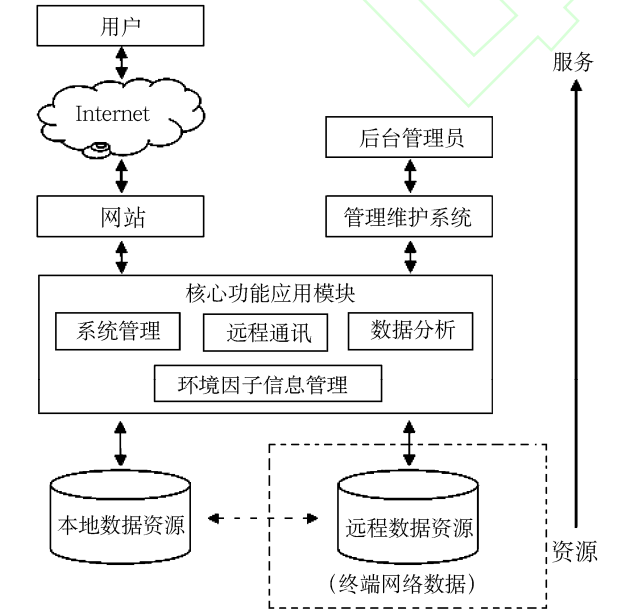


图 5 案例系统逻辑示意图

应用管理系统主要由网站系统、后台维护系统、本地数据资源和远程数据资源等 4 部分组成。从功能的角度,将应用管理系统的模块划分为传感器节点

管理、传感器网络管理、环境因子信息管理、图形化分析、用户管理、日志管理、短信收发以及预警管理等。系统逻辑示意图如图 5 所示。

本文采用 Django 框架来开发具备 REST 风格的应用系统。该系统通过网络实时显示果园相应传感器获取到的果园中作物生长环境因子,提供给用户或专家,用户或专家通过监控中心系统可以实时准确地得到作物的生长因子相关分析结果,及时对出现的问题采取相关措施。

具体步骤如下:

(1) 规划数据集。

作为简化的版本,本应用主要针对从业务平台获取到的示范基地信息、果园信息、传感器信息和作物生长因子等 4 项数据做相应处理。通过业务平台的查询接口(SearchServe API)查询 4 项数据对应的 API 分别为 LocationData、OrchardData、SensorData、EnvironmentalData,通过调用接口(InvokeServe API)调用查询到的 API 将远程数据直接与本地 Web 服务中功能处理模块结合来满足对应的业务需求。“示范基地信息”中包含示范基地名称、简介和 ID;“果园信息”中包含关注度、内景图和 ID;“传感器信息”中包含传感器的 ID、位置、类型以及工作状态;“作物生长因子”中包含时间以及各类传感器获取的光照、温度、土壤湿度等示范基地作物生长环境数据。

(2) 创建数据模型。

基于本实例中的实体,可以有两种对数据的处理方式:一种是直接处理,即直接由功能模块完成远程数据的分析和处理;另一种是间接处理,即可以本地创建数据库,将获取的远程数据资源先保存到本地服务器,然后进行数据的分析和处理。考虑到更好地展现 REST 风格,本文主要讨论对远程数据的直接处理。

(3) 定义资源并分配 URI。

鉴于面向果园的具有 REST 风格的应用管理系统中需要暴露很多资源,从简化模型的角度考虑,本文将其中的示范基地,布置传感器的果园,果园的区域等这 3 个关注点抽象成 3 种资源来分配 URI:

- /location/: 显示示范基地(如杨凌或者洛川等)
- /location/number/: 显示相关 ID 编号的果园列表信息
- /location/number/number_A/: 显示果园相关区域作物生长因子信息

可以通过 Django URL 配置文件把这些 URI 映射到资源上。

url.py 中的部分内容如下:

```
from django.conf.urls.defaults import *
from wsn-web.views import *
```

```
urlpatterns = patterns('',
    (r'^location/[-w-]+/$', location_list),
    (r'^location/[-w-]+/orchard/$', orchard_list),
    (r'^location/[-w-]+/orchard/([\w-]+)/$', orchard_area_detail), )
```

(4) 设计视图。

Django 在处理模型-视图-控制器 (Model-View-Controller) 模式方面, 为了实现一个资源在统一接口幕后的行为, 把代码放在视图 (View) 里面。以“示范基地列表”资源只读视图为例简单介绍相关代码。

“果园列表”资源只响应 GET 请求, 部分代码如下:

```
views.py
from django.shortcuts import get_object_or_404, render_to_response

from wsn-web.models import Job
def index(request):
    object_list = Job.objects.order_by('-pub_date')[:10]
    return render_to_response('location/location_list.html', {'object_list': object_list})

def detail(request, object_id):
    job = get_object_or_404(Job, pk=object_id)
    return render_to_response('location/location_detail.html', {'object': job, 'location': job.location})
```

代码中, 视图是由 location/location_list.html 字符串进行命名的。该视图是使用名为 object_list 的示范基地列表的上下文呈现的。所呈现的视图字符串随后被传递到 HTTPResponse 构造器中, 后者通过这个框架被发送回请求客户机那里。

XHTML 格式的示范基地列表: (部分代码)

```
<head>
<title> Location List </title>
</head>
<body>
<ul class="location">
<li><a href="/Yangling">Yangling</a></li>
<li><a href="/Luochuan">Luochuan</a></li>
...
</ul>
</body>
```

至此, 一个采用 Django 框架开发的基于 REST 风格的应用管理系统已经成型, 然后通过业务支撑平台所提供的接口, 将硬件或数据资源与系统功能模块相结合, 最后将应用管理系统所提供的功能通过统一的 URI 接口提供给用户使用。

4 结束语

基于 REST 风格的物联网作为各种技术的融合而成的新型技术体系, 其发展还有待更深入地探讨和

研究。本文依托国家科技重大专项, 对整个系统架构进行了初步的讨论, 以果园应用场景作为应用案例, 采用 Django 框架实现基于 REST 风格的应用管理系统。

参考文献:

- [1] 黄怡, 崔春风. 移动泛在网络的发展趋势[J]. 中兴通讯技术, 2007, 13(4): 1-3, 33.
- [2] 王志良, 王粉花. 物联网工程概论[M]. 北京: 机械工业出版社, 2011.
- [3] 刘云浩. 物联网导论[M]. 北京: 科学出版社, 2010.
- [4] 王震, 刘智斌. 基于物联网的养老院管理系统设计与实现[J]. 计算机与现代化, 2012(6): 77-80.
- [5] Luciana Moreira Sá De Souza, Patrik Spiess, Dominique Guinard, et al. SOCRADES: A Web service based shop floor integration infrastructure[C]// Proceedings of the 1st International Conference on The Internet of Things. 2008: 50-67.
- [6] IBM 商业价值研究院. 智慧地球赢在中国[EB/OL]. http://www.ibm.com/smarterplanet/global/files/cn_zh-cn_overview_decadeofsmart_wininchina.pdf, 2010-09-30.
- [7] 徐刚, 陈立平, 张瑞瑞, 等. 基于精准灌溉的农业物联网应用研究[J]. 计算机研究与发展, 2010, 47(z2): 333-337.
- [8] 王殊, 阎毓杰, 胡富平, 等. 无线传感器网络的理论及应用[M]. 北京: 北京航空航天大学出版社, 2007.
- [9] Jan Robert Nopper, Michael Ten Hompe. Analysis of the relationship between available information and performance in facility logistics[J]. Logistics Research, 2009, 1(3-4): 173-183.
- [10] 林彩霞, 仇建伟. 基于 REST 的图形标绘服务系统技术研究[J]. 计算机与现代化, 2012(10): 76-81.
- [11] [美] Leonard Richardson, [美] Sam Ruby. RESTful Web Services 中文版[M]. 徐涵, 李红军, 胡伟译. 北京: 电子工业出版社, 2008.
- [12] Fielding R T. Architectural Styles and the Design of Network-based Software Architectures[D]. Irvine, California: University of California, 2000.
- [13] Priyantha N B, Kansal A, Goraczko M, et al. Tiny Web services: Design and implementation of interoperable and evolvable sensor networks[C]// Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems. 2008: 253-266.
- [14] Pautasso C, Zimmermann O, Leymann F. Restful Web services vs. "big" Web services: Making the right architectural decision[C]// Proceedings of the 17th International World Wide Web Conference. 2008: 805-814.
- [15] Wilde E. Putting Things to REST[R]. Technical Report UCB iSchool Report, UC Berkeley, 2007.
- [16] 潘冰. 面向资源的 RESTful Web 应用研究[J]. 微计算机应用, 2010, 31(7): 38-43.