(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 104615748 B
(45)授权公告日 2018.02.27

(21)申请号 201510074748.0

(22)申请日 2015.02.12

(65)同一申请的已公布的文献号
申请公布号 CN 104615748 A

(43)申请公布日 2015.05.13

(73)专利权人 华北电力大学(保定)
地址 071003 河北省保定市永华北大街619号

(72)发明人 孔英会 高育栋 李佩玉 车辚辚

(74)专利代理机构 石家庄冀科专利商标事务所有限公司 13108
代理人 李羡民 达丽娜

(51)Int.Cl.
G06F 17/30(2006.01)

(56)对比文件
CN 202931390 U,2013.05.08,
US 2002116548 A1,2002.08.22,
CN 103092936 A,2013.05.08,
张明月.网页设计与制作研究.《邯郸职业技术学院学报》.2009,第22卷(第2期),第82-83页.

审查员 徐雯晖

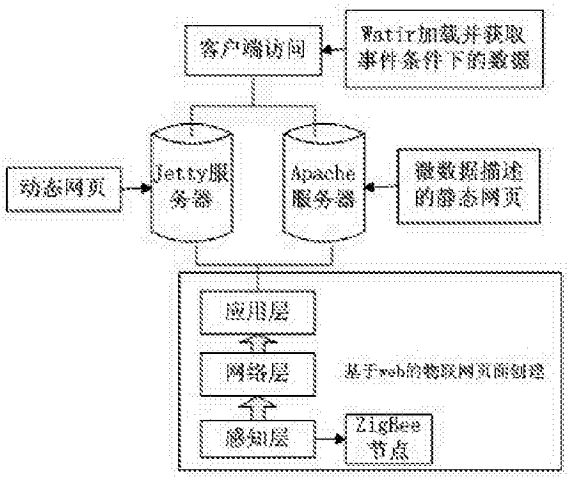
权利要求书1页 说明书9页 附图1页

(54)发明名称

基于Watir的物联网Web事件处理方法

(57)摘要

一种基于Watir的物联网Web事件处理方法，首先创建物联网页面，对终端节点的属性描述到网页上并将网页部署到服务器上，采用自动化测试框架Watir方法加载到物联网动态页面，结合Web页面的结构和内容获取包含终端节点信息的HTML文档并对HTML文档进行解析，对解析后的数据进行预处理后添加事件，将某种条件下的事件数据保存，并显示到页面上。本发明实现了在满足事件条件下，实时抓取物联网多网页动态数据的目的，并且响应速度快、准确率高、可扩展性好，不会出现数据遗漏现象。



CN 104615748 B

1.一种基于Watir的物联网Web事件处理方法，其特征在于：首先创建物联网页面，对终端节点的属性描述到网页上并将网页部署到服务器上，采用自动化测试框架Watir方法加载到物联网动态页面，结合Web页面的结构和内容获取包含终端节点信息的HTML文档并对HTML文档进行解析，对解析后的数据进行预处理后添加事件，将某种条件下的事件数据保存，并将结果数据显示到页面上；

具体操作按以下步骤进行：

A.基于Web的物联网页面创建；

A1.创建包含传感器和无线收发模块的Zigbee网络；

A2.对Zigbee网络中对终端节点的属性描述到网页上并将网页部署到服务器上；

B.采用自动化测试框架Watir方法加载到物联网动态页面；

C.获取包含终端节点信息的HTML文档；

D.解析HTML文档，获取页面文本信息；

E.对解析后的数据进行预处理，细化数据内容；

F.给预处理后的数据添加事件；

G.存储事件结果数据；

H.循环执行步骤BGG；

I.将结果数据显示到页面上；

在所述B步骤中，采用自动化测试框架Watir方法加载事件并获取事件条件下的数据；系统框架的搭建过程为：首先创建物联网Web页面，用微数据来描述每一个传感器的静态属性，用Js文件描述动态属性，并将网页在服务器上进行部署，然后用Web自动化测试框架Watir，在添加事件条件后提取关键和有效的数据并保存；

所述步骤F添加事件的方法包括极值的查找和阈值的添加；其中，极值的查找对象是同种传感器，找出在同一时刻下的最大值或最小值；阈值的添加对象为每个传感器，通过给每个传感器都添加阈值，找出阈值范围内的数据。

2.根据权利要求1所述的基于Watir的物联网Web事件处理方法，其特征在于，步骤A1的具体方法为：采用两个温度传感器DS18B20与两个无线收发模块CC2530相接作为终端节点，第三个无线收发模块CC2530作为协调器，在MSSTATE_LRWPAN协议栈的基础上进行应用开发，组建Zigbee网络。

3.根据权利要求2所述的基于Watir的物联网Web事件处理方法，其特征在于，步骤A2的具体方法为：将温度传感器描述为静态属性和动态属性，并将两种属性描述到Web网页并部署到不同的服务器上，其中静态数据发布到Apache服务器上，动态数据发布到Jetty服务器上；然后通过链接地址，将两个服务器上的网页连接起来。

4.根据权利要求3所述的基于Watir的物联网Web事件处理方法，其特征在于，步骤D中通过“Nokogiri::HTML.parse”方法对获得的HTML文档进行解析，采用CSS Selector来找页面上所需内容的节点，用ID选择器(#id)取回该时刻的页面文本信息。

5.根据权利要求4所述的基于Watir的物联网Web事件处理方法，其特征在于，步骤E中通过利用Ruby的Araay类来创建三个数组以及Ruby的Split方法进行数据的预处理。

6.根据权利要求5所述的基于Watir的物联网Web事件处理方法，其特征在于，步骤H使用rufus-scheduler模块来指定时间执行循环。

基于Watir的物联网Web事件处理方法

技术领域

[0001] 本发明涉及物联网技术领域，特别是涉及物联网中事件的处理方法。

背景技术

[0002] 物联网(Internet Of Things)泛指“物物相连的互联网”，它的核心和基础是互联网，是在互联网基础上延伸和扩展的网络，其用户端延伸和扩展到了任何物品与物品之间。物联网即是把任何物品与互联网相连接，进行信息交换和通信。

[0003] 物联网的体系架构可以分为三层，感知层、网络层以及应用层。其中，感知层为最底层，主要实现对物理世界的智能感知识别、信息采集处理和自动控制，并由通信模块将物理实体连接到网络层和应用层。网络层为中间层，由各种私有网络、互联网、有线和无线通信网、网络管理系统和云计算平台等组成，负责传递和处理感知层获取的信息；各种物联网应用借助无线、有线手段接入网络，在网络互联支持下提供信息的传递、共享和服务的支撑。应用层为最上层，应用层即终端处理层，它是输入输出控制终端，包括计算机、手机等服务器终端，实现对信息的分析、存储、挖掘处理等应用。

[0004] 物联网包含了海量的实体，这些实体每时每刻都会有新的状态，例如，通过传感器感知室内各种状况、温度、湿度、还有光照程度等，感知道路的实时拥堵信息等。要实现对物联网的智能控制以及扩展对物联网的应用，就必须对这些海量实体的实时信息进行数据分析，从这些海量实体返回的实时状态数据中搜寻得到有用的信息。在现阶段的物联网传感器的数据获取和处理上，基于Web的物联网数据管理获得广泛应用。基于Web的物联网系统可以通过嵌入式设备将日常设备和装置上信息全部集成到Web 实现事物的互联，再对智能物体进行访问，传输协议可采用现有的已经被很好地理解和接受的标准，例如URI、HTTP、REST等。

[0005] 事件是指在一个流程中，当达到某种状况或条件而触发某种行为的消息或请求，通常可理解为现实世界中某种状态的改变。物联网事件处理的主要工作是检测事件，即根据已知给定的规则和模式，从基于分散信息系统的大量信息中检测并提取被定义好的事件。对物联网海量数据，找出用户感兴趣的数据、过滤掉其他的冗余数据尤为关键，用户不关心传感器产生的所有数据，只关心对他们有用的数据，比如在“所有温度传感器产生的温度数值中，用户只对最高的温度值产生兴趣，看有没有超过设定的阈值”，“在对智能设备的传感器上，传感器的数据包括温度、湿度和光照，在不同的场合只需要其中的温度或者其他数据”等。

[0006] 目前的网络爬虫或者页面采集工具无法实现动态数据的实时抓取。在对事件的处理上，主要是应用于静态网页，并且没有同时加载添加事件条件下的多物联网页面。其缺点主要表现在以下几个方面：

[0007] 1)大多事件处理的数据来源都是数据库，没有从网页实时地获取数据后添加事件处理。国外研究中具有代表性的复杂事件处理方法及原型系统有SASE、ESPER、RCEDA等，这些原型系统可以用于RFID、GPS和传感器网络等数据上的复杂事件处理，没有针对物联网

Web网页的事件处理技术。

[0008] 2)事件处理在主动数据库中已经做了大量的研究,常用的方法都是基于固定的数据结构,如树、图、自动机或Petri网,没有一种方法是针对半结构化的物联网网页提出的事件处理,更没有方法提出对多个物联网页面同时进行实时事件处理。在对动态页面的处理上,没有方法通过设定时间来循环的抓取网页进行事件条件下的解析。

[0009] 3)对Web网页的事件研究主要是针对静态网页,研究内容主要包括消除冗余、有效的整合事件信息、提高事件的准确性等。

[0010] 4)现有的动态网页的解析机制无法满足物联网页面数据获取的实时性,客户端与服务器交互的过程中客户端发送请求、服务器端回应这个过程耗时长,且不稳定,不能实现动态实时抓取,同时容易出现数据遗漏现象。

发明内容

[0011] 本发明针对现有技术的不足,提供一种响应速度快、能够实时准确抓取物联网多网页动态数据的物联网事件处理方法。

[0012] 为解决上述技术问题,本发明所采取的技术方案如下。

[0013] 基于Watir的物联网Web事件处理方法,首先创建物联网页面,对终端节点的属性描述到网页上并将网页部署到服务器上,采用自动化测试框架Watir方法加载到物联网动态页面,结合Web页面的结构和内容获取包含终端节点信息的HTML文档并对HTML文档进行解析,对解析后的数据进行预处理后添加事件,将某种条件下的事件数据保存,并显示到页面上。

[0014] 上述基于Watir的物联网Web事件处理方法,具体包括以下步骤:

[0015] A.基于Web的物联网页面创建;

[0016] A1.创建包含传感器和无线收发模块的Zigbee网络;

[0017] A2.Zigbee网络中对终端节点的属性描述到网页上并将网页部署到服务器上;

[0018] B.采用自动化测试框架Watir方法加载到物联网动态页面;

[0019] C.获取包含终端节点信息的HTML文档;

[0020] D.解析HTML文档,获取页面文本信息;

[0021] E.对解析后的数据进行预处理,细化数据内容;

[0022] F.给预处理后的数据添加事件;

[0023] G.存储事件结果数据;

[0024] H.循环执行步骤B~G;

[0025] I.将数据显示到页面上。

[0026] 上述基于Watir的物联网Web事件处理方法,步骤A1的具体方法为:采用两个温度传感器DS18B20与两个无线收发模块CC2530相接作为终端节点,第三个无线收发模块CC2530作为协调器,在 MSSTATE_LRWPAN协议栈的基础上进行应用开发,组建Zigbee网络。

[0027] 上述基于Watir的物联网Web事件处理方法,步骤A2的具体方法为:将温度传感器描述为静态属性和动态属性,并将两种属性描述到Web网页并部署到不同的服务器上,其中静态数据发布到Apache服务器上,动态数据发布到Jetty服务器上;然后通过链接地址,将两个服务器上的网页连接起来。

[0028] 上述基于Watir的物联网Web事件处理方法，步骤D中通过“Nokogiri::HTML.parse”方法对获得的HTML文档进行解析，采用CSS Selector 来找页面上所需内容的节点，用ID选择器(#id)取回该时刻的页面文本信息。

[0029] 上述基于Watir的物联网Web事件处理方法，步骤E中通过利用Ruby的Araay类来创建三个数组以及Ruby的Split方法进行数据的预处理。

[0030] 上述基于Watir的物联网Web事件处理方法，步骤F添加事件的方法包括极值的查找和阈值的添加；其中，极值的查找对象是同种传感器，找出在同一时刻下的最大值或最小值；阈值的添加对象为每个传感器，通过给每个传感器都添加阈值，找出阈值范围内的数据。

[0031] 上述基于Watir的物联网Web事件处理方法，步骤H使用rufus-scheduler模块来指定时间执行循环。

[0032] 由于采用了以上技术方案，本发明所取得的技术进步如下。

[0033] 本发明从客户端出发，将物联网Web页面放到服务器上，对于任何一个能连网的客户端，通过基于Watir的自动化测试框架，都可以获取事件条件下的物联网页面实时信息并存储，实现了实时抓取事件条件下的物联网多网页动态数据的目的，并且响应速度快、准确率高、可扩展性好，不会出现数据遗漏现象。

附图说明

[0034] 图1为本发明的总体框架图；

[0035] 图2为本发明的流程图。

具体实施方式

[0036] 下面将结合附图和具体实施例对本发明进行进一步详细说明。

[0037] 本发明针对物联网Web应用，设计了基于Watir的物联网Web事件处理系统框架，框架结构如图1所示，系统框架包括Zigbee网络节点、物联网、Apache服务器、Jetty服务器以及客户端，Zigbee网络节点包含温度传感器、无线收发模块CC2530，物联网包含感知层、网络层以及应用层。Zigbee网络节点采集数据信息后上传给物联网网页，物联网网页分别部署到Apache服务器和Jetty服务器，Apache服务器用于展示静态网页，Jetty服务器用于展示动态网页，客户端对Apache服务器和Jetty服务器进行访问，并采用自动化测试框架Watir方法加载事件并获取事件条件下的数据。

[0038] 系统框架的搭建过程为：首先创建物联网Web页面，用微数据来描述每一个传感器的静态属性，用Js文件描述动态属性，并将网页在服务器上进行部署，然后用Web自动化测试框架Watir，在添加事件条件后提取关键和有效的数据并保存。

[0039] 本发明主要工作包括基于Web的物联网页面创建、服务器部署和Watir加载与事件添加几个方面，具体过程为：首先创建物联网页面，将静态属性保存到Apache服务器，动态属性保存到Jetty服务器上，采用自动化测试框架Watir方法加载到物联网动态页面，结合Web页面的结构和内容获取包含终端节点信息的HTML文档并对HTML文档进行解析，对解析后的数据进行预处理后添加事件，将某种条件下的事件数据保存，并显示到页面上，以满足应用需要。

[0040] 本发明的流程图如图2所示，具体操作包括以下步骤。

[0041] A.基于Web的物联网页面创建

[0042] A1.创建Zigbee网络

[0043] Zigbee技术是常用的一种连接物理实体的通信技术，它基于IEEE802.15.4无线标准研制开发，用于短距离、低速率无线网络通信；其突出优点是应用简单，工作频段灵活，低功耗，低成本，高可靠性，具有自组网和自恢复能力等。

[0044] 本发明采用两个温度传感器DS18B20与两个无线收发模块CC2530相接作为终端节点，另外一个无线收发模块CC2530充当协调器，在 MSSTATE_LRWPAN协议栈的基础上进行应用开发，组建一个Zigbee网络，实现无线组网。

[0045] 无线收发模块CC2530 是德州仪器（TI）目前推出的完整的用于2.4GHzIEEE802.15.4/RF4CE/ZigBee 的第二代片上系统解决方案。CC2530在单个芯片上集成了IEEE802.15.4标准2.4GHz频段的RF无线电收发机，具有优良的无线接收灵敏度和抗干扰性。本发明利用无线收发模块CC2530的射频功能创建zigbee网络。

[0046] 温度传感器DS18B20是美国 DALLAS半导体公司的单总线智能型数字温度传感器，它通过单总线与处理器进行数据传输，主要由64位ROM、温度敏感元件、非易失性温度告警触发器TH和TL、配置寄存器组成。温度传感器DS18B20用来获取环境的温度值。

[0047] 本发明将两个温度传感器分别部署到室内和窗台，传感器的地理特性描述为传感器所处的位置。

[0048] 本发明的终端节点用于采集温度信息，协调器将接收到的温度信息通过串口发送给PC计算机，即将Zigbee网络的传感器数据通过串口显示到网页上。

[0049] A2. 对Zigbee网络中对终端节点的属性描述到网页上并将网页部署到服务器上

[0050] 本发明对传感器的描述分为两类：对静态属性的描述和动态属性的描述。静态属性包括传感器的名称、类型、时间特性等，动态属性包括传感器的动态数据。将两种属性描述到Web网页并部署到服务器上，然后，通过静态网页上的链接地址，链接到动态页面。

[0051] Apache HTTP Server(简称Apache)是Apache软件基金会的一个开放源码的网页服务器，可以在大多数计算机操作系统中运行，支持多种方式的HTTP认证。本发明中将静态数据发布到Apache服务器上。

[0052] Jetty是一种轻量级Web服务器，Jetty 是一个完全由Java实现的、开源的HTTP服务器和 Servlet容器，其运行速度快、更灵活，体现在其可插拔性和可扩展性，更易于开发者对Jetty本身进行二次开发，定制一个适合自身需求的Web Server。Jetty可以用来作为一个传统的Web 服务器，也可以作为一个动态的内容服务器，并且Jetty 可以非常容易的嵌入到Java 应用程序当中。本发明中将动态数据发布到Jetty服务器上。

[0053] 本发明用于测试的两个服务器在一台计算机上，所以两个网址的端口号不同。本实施例中，Jetty服务器的设定端口号为80，Apache服务器的端口号为8080，访问地址为均为本地Ip。

[0054] 在本发明中，通过微数据描述传感器的静态页面，微数据是创建自定义元数据片段，并嵌入到HTML网页里的方法。微数据API通过添加自定义的语义词库来扩展HTML。该技术是一种将机器可读的自定义语义属性嵌入到HTML文档里的标准化途径。微数据表现为可被嵌套的名值对(name-value pairs)。这些名值对的组合被称为条目，每个名值对被称为

一个成员属性。属性itemtype描述定义条目词库名字的URL，Itemid描述条目的全局标识符，Itemprop描述条目的对象属性定义，Itemref拥有一个Itemscope属性，Itemscope属性主要描述一个布尔型的属性，可以创建一组名为条目的名值对。

[0055] 本发明使用Apache2.2将静态网页部署到服务器上。首先，通过Serverroot命令设置保存服务器文件的目录，本实例的保存路径为ServerRoot "D:/Apache Software Foundation/Apache2.2"。其次，配置文件所在位置为D:\Apache Software Foundation\Apache2.2\conf，主要的配置修改如下：①修改Listen为211.82.237.78:8080，其中211.82.237.78为本地IP地址，服务端口号为8080；②修改默认的自动运行文件名称，修改dir_module模块，将微数据编写的静态网页index.html保存到dir_module模块中。通过对Apache的部署，实现对网页的访问。

[0056] 在本发明在对动态页面的处理上，使用JSP编写动态网页，前台数据的展示采用的是JSP进行显示，脚本语言和页面布局采用的是Extjs进行操作。Extjs是一种主要用于创建前端用户界面，是一个基本与后台技术无关的前端ajax框架。

[0057] JSP全名为Java Server Pages，其根本是一个简化的Servlet设计。JSP的概念与微软公司的ASP非常相似，都是在普通HTML文件中内嵌程序语句，避免了开发Servlet时繁琐的HTML输出，可以更好地格式化输出效果。JSP 技术可以让Web 开发人员和设计人员非常容易的创建和维护动态网页。

[0058] 对脚本代码的编写中，具体内容如下：首先设置页面的布局，页面的布局包括panel、button和label，首先定义一个Panel，设置Panel在浏览器中的显示方式，Panel的描述包括Title、Width和Maigin，其中Maigin是外边距，通过在items组件中添加两个button和label来显示传感器数据。在button按钮的设置中，主要是添加button的点击事件，用来获取数值。在handler的处理上，添加自定义的task事件，利用SetText方法将数据显示到label上。

[0059] 在对串口数据的获取过程中，通过定义serialPort类来确定串口设置，包括数据位、校验位和停止位等，用inputStream = serialPort.getInputStream()来读取串口的数据流；inputStream的read方法读取数据流并赋值到readBuffer，通过对每个字节的数据处理，显示出原来的数值，具体处理为每个字节减去0的ASCII码值并处理对应的系数，将值赋值给两个double类型的变量，具体实现如下：

[0060] wendu=(((int)*readBuffer*[2]-48)*10+(int)*readBuffer*[3]-48+((int)*readBuffer*[5]-48)/10.0+((int)*readBuffer*[6]-48)/100.0);

[0061] wendu1=(((int)*readBuffer*[10]-48)*10+(int)*readBuffer*[11]-48+((int)*readBuffer*[13]-48)/10.0+((int)*readBuffer*[14]-48)/100.0);

[0062] 由于在数值的计算和处理中，计算出的结果小数点位数太多，需要进行后续的数据处理。首先通过valueOf 方法将double类型的变量转换为String类型，通过if(data.contains(".") && data.length()>5)判断，当数值长度超过5位数时，用substring方法截取字符串，通过 data.substring(0,data.indexOf(".")+3)只截取小数点后两位；将截取后的数据用renderJson对数据进行格式转换，render 系列方法将渲染不同类型的视图并返回给客户端。Json是一种将数据从服务器传递到客户端的更为有效的办法，转换成Json格式的，值为light和light1，传递给各自的label标签。在JSP文件中，调用

JS文件，将Json变量的数据显示出来。其中，在task事件中，在时间的处理上，用了interval方法，设定interval :3000，每3秒刷新一次数据。

[0063] 本发明通过将两种属性分开部署到两个服务器上，可以方便查看两种属性的状态值，还可以方便的修改其中一种属性；同时，当传感器的数量增加时，两个属性分开部署可以减少对服务器的压力。

[0064] B.采用自动化测试框架Watir方法加载到物联网动态页面

[0065] 本发明用Watir来加载传感器页面并获取数值，根据需要可实现按一定时间间隔连续获取，如每30秒完成一次获取。该方法实时性高，连续性好，能满足各个领域对物联网实时数据的需求。

[0066] Watir是一个用于网页自动化测试的开源工具，在编写脚本过程中，主要通过是使用Watir::IE的一个实例化对象@IE来获得页面上的各种元素，Watir::IE 封装的是一个当前页面的DOM Tree，而不是页面源代码，比如页面如果用Js动态产生一个元素，在Watir中仍然可以访问。Watir是用Ruby语言实现的，Ruby是一种跨平台、面向对象的动态类型编程语言。Ruby语言自然简洁，却依然拥有强大的数据分析和处理能力，主要应用于网络开发语言、数据处理和数据挖掘等方面。

[0067] 采用自动化测试框架Watir方法加载到动态页面的过程具体为：首先，通过“ie = Watir::Browser.new”创建一个浏览器对象实例，从而来模拟浏览器的操作。其次，通过“ie.goto('http://211.82.237.78/')”，采用跳转方式打开预期的页面，同时用span标签下的button按钮来模拟点击按钮，用ie.span(:id, 'button-1011-btnInnerEl').click和ie.span(:id, 'button-1012-btnInnerEl').click点击button让其显示动态的数据。

[0068] C.获取包含终端节点信息的HTML文档

[0069] 在完成Watir加载定位之后，通过“ie.html”方法获取框架下的HTML文档，HTML文档中包含Zigbee网络中终端节点的所有信息。网页的部分HTML文档如下：

[0070]
```
<span id="button-1011-btnInnerEl" class="x-btn-inner x-btn-inner-center"
unselectable="on">点我获取当前室内温度</Span>
<label id="light" class="x-component x-component-default" for=""
style="margin:20px 20px 20px 20px;">室内温度值为:21.69 ° C</label>
<span id="button-1012-btnInnerEl" class="x-btn-inner x-btn-inner-center"
unselectable="on">点我获取当前窗口温度</Span>
<label id="light1" class="x-componsnt x-component-default" for=""
style="margin:20px 20px 20px 20px;">窗口温度值为:21.29 ° C</label>
```

[0071] D.解析HTML文档

[0072] Ruby的插件Nokogiri是一种HTML、XML、SAX阅读解析器，许多关于Ruby解析XML、HTML的插件有很多，其中最出名的的有Hpricot和Nokogiri。Nokogiri的速度比目前应用广泛的Hpricot要快许多。经过Benchmark测试表明，Nokogiri在加载XML文档的速度是Hpricot的7倍，在XPATH搜索的速度是Hpricot的5倍，而在CSS选择器的搜索上是Hpricot的1.62倍。因此Nokogiri被认为有可能取代Hpricot的新一代Ruby的解析库。

[0073] 本发明中，Nokogiri通过“Nokogiri::HTML.parse”方法对获得的HTML文档进行解析。Nokogiri 提供了 XPath 及CSS Selector方式来寻找文档里的节点，但是CSS locator比XPath locator速度快，并且CSS Selector能非常精准的定位到测试的Elements。CSS Selector可以粗略分成几类基本的类型：ID选择器(#id)、Class选择器(.class)、类型

(type)选择器(p)等。这些都是单一的选择器，可以在应用中把它们组合起来，如:div#id，div#p等。本发明采用CSS Selector 来查找页面上所需内容的节点，用ID选择器(#id)取回该时刻的页面文本信息，解析后的数据内容为数据预处理做准备。

[0074] E. 对解析后的数据进行预处理

[0075] 本发明首先将解析后的数据进行预处理，利用Ruby的Araay类来创建三个数组，通过doc.css('#light').text[i]和doc.css('#light1').text[j] (其中i和j为数字在light和light1标签内对应的下标)，将解析后的数字逐位保存到数组中。同时，将解析出标签内的完整内容也都保存到数组中wendu = [doc.css('#light').text,doc.css('#light1').text]。利用Ruby的Split方法，通过“:”来分割Label标签内的内容，将数字部分通过wendu[0].split(":")[1]整体取出并赋值给一个字符串类型的变量。

[0076] 本发明通过利用Ruby语言对网页进行解析，将解析后的文档进行进一步的数据处理，将数字和内容分的更细、更精确，为后续添加事件处理做好准备工作。

[0077] F.给预处理后的数据添加事件

[0078] 事件方法包括极值的查找和阈值的添加，其中，极值的查找对象是同种传感器，找出在同一时刻下的最大值或最小值;阈值的添加对象为每个传感器，其中，阈值分为最大阈值和区间阈值，通过给每个传感器都添加阈值，找出阈值范围内的数据。对事件的处理过程如下：

[0079] F1.对最大值的处理，由于两个温度传感器都在一个环境中，数值相差不是很大，故本发明将两个传感器的温度值通过逐位作比较，找出最大值。具体的实现过程为:首先两个传感器的最高位作比较，如果最高位大小不一样，跳出循环，将最高位大的传感器数值输出，如果最高位相同，比较第二位的大小，找出最大值，依次按照这种规则作比较，直到找出最大值。在最小值的查找过程中，应用的方法类似于最大值的查找。

[0080] F2.对温度的阈值处理，分为上限阈值和区间阈值，在本发明中阈值的添加均为整数。

[0081] F21.对上限阈值的处理，设定上限阈值为32°C，由于温度传感器的部署在室内，数值为两位数且变化不大，本发明使用逐位法与阈值作比较，用to_i方法将解析后的数字转换为整数。首先与阈值第一位作比较，如果小于第一位，则将数据显示出来。如果等于第一位，则继续比较第二位，如果小于第二位，将数据显示出来，否则用 “超出阈值”字样来表示，具体的实现使用条件判断语句，将两个条件通过或连接起来，条件为:shuzi[0].to_i<3||(shuzi[0].to_i=3&&shuzi[1].to_i<2)。

[0082] F22.对区间阈值的处理，设定阈值范围为24°C到30°C。通过Ruby的区间Range方法来添加区间范围，使用“…”表示一个半闭半开的区间，用to_i方法处理数字，截取小数点后的数字，通过if((24…30) === str.to_i)添加阈值区间。当数值在此范围时显示数据，不在此范围是显示“超出阈值”字样。

[0083] 在后续的扩展中，当网页上有多个传感器的数据时，可以给每个传感器都设定相应的阈值。本发明通过给预处理后的数据添加事件，给同种传感器找极值，给每个传感器添加阈值，能够高效、实时、准确的获取数据。

[0084] G.存储事件结果数据

[0085] 本发明通过Ruby中的文件操作来存储数据，File类是IO类的子类，本发明通过

File类创建文档，用来存储数据。即通过“File.new("d.txt","w")”方法实现创建txt文档，其中‘w’意思是表示以写入的方式打开文件，d为文档名称，在本发明中，文档名称为filewendu。通过 file.puts方法将获取的数据保存到filewendu.txt中。同样可以创建几个文件实例，经过事件处理后的数据，可以分开保存并呈现给用户。

[0086] H.循环执行

[0087] Rufus-scheduler 是Ruby的一种可定时插件，可以让程序在指定的时间内运行。

[0088] 本发明使用rufus-scheduler模块来指定时间执行程序，通过scheduler.every ’10s’设定时间，每10秒执行步骤B~G，更新filewendu.txt中的数据，保证数据是最新的。

[0089] I.将数据显示到页面上

[0090] 本发明将数据保存到文本当中，通过本地的网页调用，将结果显示到网页上。同时本发明也可以将物联网事件处理结束后的实时实体数据链接并存入数据库，能满足物联网海量数据量的要求，同时也易于后续对于数据处理分析工作的进行。

[0091] Watir处理多个添加事件条件下的物联网页面实验

[0092] 为了说明本发明提出方法的有效性，设计了一组实验，实验中需要的实验环境和实验内容如表1所示，客户端的配置与相关应用软件如表2所示。

[0093] 表1

[0094]

实验对象	对象①	对象②	对象③
实验对象描述	某大学现代无线通信实验室（物联网实验平台）的智能家居系统	某大学现代无线通信实验室（物联网实验平台）的用电信息采集系统	自行搭建的物联网实验系统
显示内容	商户和门口两个实体，每个实体的监测内容包括温度、湿度和光照，六个传感器采用 Push 的方式将感知到的信息实时推送到网页上	汇总各房间的用电数据至网页，包括 8 个房间的用电数据采集-电压、电流、功率、电能量采集等	两个温度传感器上传数据至网页
刷新方式及时间	异步更新，各个位置的数据更新周期均为 5 秒	异步更新，每 15 分钟刷新一次	异步更新，各个位置的数据更新周期均为 3 秒

[0095]

链接地址	http://211.82.236.83/sh/	http://211.82.236.83/dianli/system/	http://211.82.237.78/
处理内容	同种传感器的数据找出最大值，对六个传感器分别添加上限阈值和范围阈值	以其中一个房间的三相电压为例，给每个电压添加阈值，当电压超过阈值时，显示警告信息	两个温度数值的最大值以及两个传感器分别添加上限阈值和范围阈值

表2

[0096]

PC 硬件配置	Intel Core i5-3210M （2.50GHz）内存：4G 硬盘：500G
PC 操作系统	Windows 7 旗舰版 32 位操作系统
PC 网络环境	某大学校园网
应用软件	Watir Gem Nokogiri Ruby

[0097] 实验过程主要分为三步：Watir对单个网页在事件条件下的处理，Watir对同一平台两个网页事件条件下的处理，Watir对不同平台多网页事件条件下的处理。具体实验过程如下：

[0098] 首先，Watir对单个网页的数据处理，实验对象为③。

[0099] 本次实验中，循环周期设为10秒，实验中使用Ruby的Time类来确定Watir加载和解析需要的时间，通过Time.now方法，直接获取当前的时间，在程序的开始和结束加上时间戳，该时间可以精确到秒，通过两个时间的差值来确定加载并解析的时间在2秒左右。其中，通过Time类的to_f方法，在数据解析前后加上时间戳，可确定解析时间大概在6毫秒左右。在准确率的研究上，连续10次抓取页面数据，数据都能准确的保存，准确率达到100%。

[0100] 其次，Watir对同一平台的两个网页处理：实验对象为①和②。

[0101] 实验中使用本地的Watir框架同时加载同一平台的两个页面并解析网页内容，循环周期设为10秒，将两个网页的结果保存到一个文本中；同样，通过添加前后的时间戳确定运行时间，在5秒的时间内加载并解析完成两个网页，其中，通过Time类的to_f方法，在数据解析前后加上时间戳，解析的时间大概在10毫秒左右。而且在连续抓取10次的情况下，准确性达到100%。

[0102] 最后，Watir对不同平台的三个网页处理：实验对象为①、②和③。

[0103] 在本次的实验中，循环周期设为10秒，连续抓取十次，加载并解析三个网页，通过添加时间戳，确定对三个网页的加载并解析完成大约需要7秒。其中，通过Time类的to_f方法，在数据解析前后加上时间戳，解析的时间大概在12毫秒左右。在准确性的研究上，在十次中都能抓取到正确的数据，准确性为100%。

[0104] 由以上实验看出，基于Watir的物联网事件获取方法能按照预先的设定的时间连续抓取以保证获取最新的数据。在同一客户端对三个系统的由少到多逐次测试，随着同时加载网页数量增加，加载并解析的时间也会随之上升，但经测试目前多数计算机终端能在几秒内完成，且准确率一直保持的很稳定，可以100%完成，满足对物联网Web网页实时性事件处理的要求。

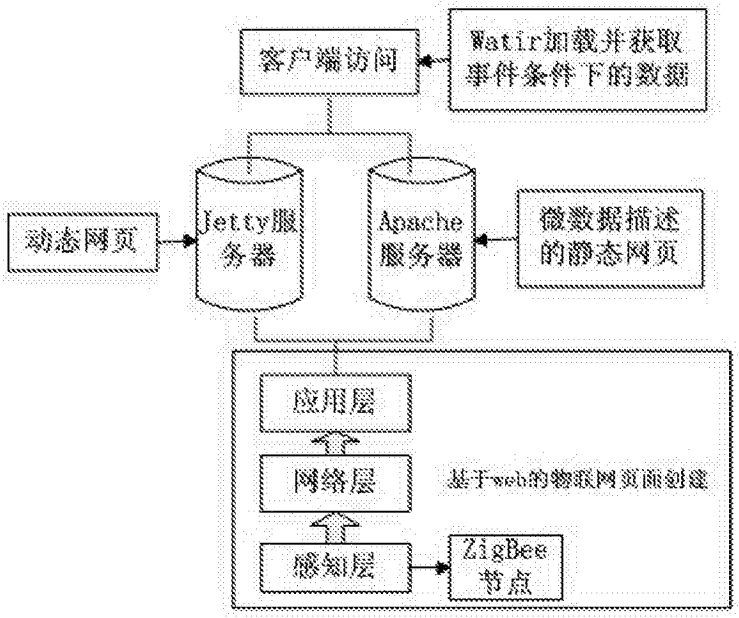

客户端访问
Watir加载并获取事件条件下的数据
动态网页
Jetty服务器
Apache服务器
微数据描述的静态网页
应用层
网络层
基于web的物联网页面创建
感知层
ZigBee节点

图1

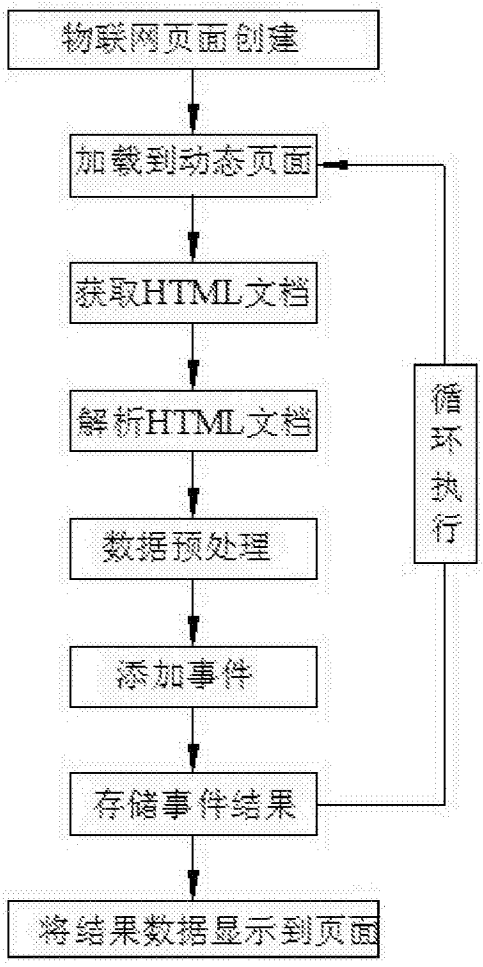

物联网页面创建
加载到动态页面
获取HTML文档
解析HTML文档
数据预处理
添加事件
存储事件结果
将结果数据显示到页面
循环执行

图2