

文章编号:1005-0523(2012)05-0044-06

# 基于Web的货车检修工序监测系统的研制

丁振凡<sup>1</sup>, 王小明<sup>1</sup>, 邓建明<sup>2</sup>, 邱新亮<sup>2</sup>

(1. 华东交通大学, 江西南昌 330013; 2. 南昌铁路局, 江西南昌 330001)

**摘要:**介绍了铁路货车检修工序进展监控与管理软件的设计。系统通过Web服务器的串口所连接的无线通信模块与现场的无线手持机通信。主机通过轮询方式查询手持机发送的作业进展信息,利用多线程协调完成通信信息的获取与分析。同时,系统采用B/S模式在浏览器页面能对检修工序进展进行实时监控。在Web服务器端,利用Spring任务定时实现每天工作的启停,以及用Spring编写REST风格的Web服务。在浏览器端,利用AJAX技术实现对Web服务的访问,以及将显示结果在页面上动态显示。Web服务通过Json变换向浏览器传递对象消息。

**关键词:**Spring; 远程在线监测; 任务计划; AJAX; Json; 无线通信

**中图分类号:**TP393

**文献标志码:**A

随着Web应用的普及,基于Web的监控应用越来越广泛<sup>[1-2]</sup>。由于工业数据的实时变化特点,客户浏览器要求能及时跟踪显示数据的变化。为了实现浏览器界面中内容的动态更新,研究者提出了各种办法,总体上分为拉方式和推方式两种,推方式采用Java Applet技术,用Socket通信接收数据<sup>[3]</sup>,要求客户方有一个消息接收线程等待服务器将变化数据推送过来。近年来,出现了富客户端技术Flex,可通过消息服务技术实现实时监控<sup>[4]</sup>。拉方式有的采用ActiveX控件实现数据通信和显示处理<sup>[5]</sup>,近年来流行用客户端的AJAX技术<sup>[6-7]</sup>,利用AJAX结合VML是实现Web图形化监控的一种方法<sup>[8]</sup>。本文采用AJAX实现Web页面对铁路货车的检修进程的实时监控。从而可以让管理员及时了解各道维护台上的工序进展<sup>[9]</sup>,对工序超时情况能及时报警。监测过程的数据自动记录到数据库中,可容易进行各类统计分析和产生报表。

## 1 系统功能和基本构成

系统的功能主要在于通过Web页面实时监控工地上的各工序进展。具体有:

- 1) 车道各台位工作进展监测: 监视分布在各车道上的工序工作状态;
- 2) 工序完成辆份的数量及当前状态监测: 监视各个工序的当前完成维修辆份以及工序的当前状态,用不同图标指示;
- 3) 超时报警消息的查阅: 查询记录在数据库中的报警信息;
- 4) 所有各个工序的功效统计直方图: 以直方图的形式显示当天各工序的功效;
- 5) 当天已维修车型数量的统计直方图: 按车型分类统计给出已维修数量的统计图;
- 6) 当天各台位的维修任务及时限设置: 设置各台位要维修的车型及数量,以及工序完成一个辆份的超时时间限制。

系统的基本构成如图1所示。通过Web服务器串口连接的无线通信模块与手持机进行通信,监管人员通过浏览器界面对工地上6个车道30个工序的维修人员的工作状态进行监控。

收稿日期:2012-10-15

作者简介:丁振凡(1965—),男,教授,研究方向为语义Web、分布式计算。

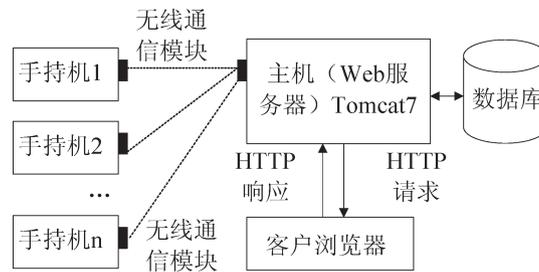


图1 系统的基本组成  
Fig.1 The basic structure of the system

## 2 系统设计

系统采用拉方式进行工作。首先,Web服务端与手持机之间是采用串口通信,服务器的通信线程通过发送轮询信息给手持机来查询每个工序的工作进展。将所有手持机用户的工序当前状态信息用 Current-Status类的静态属性保存,同时写入数据库进行登记。其次,客户浏览器与Web服务器的通信也是采用定时询问方式,客户方通过AJAX定时发送XMLhttp访问请求,根据服务器返回的JSON消息更新页面上的显示,从而实现对远端所有手持机用户工序状态的动态监控。

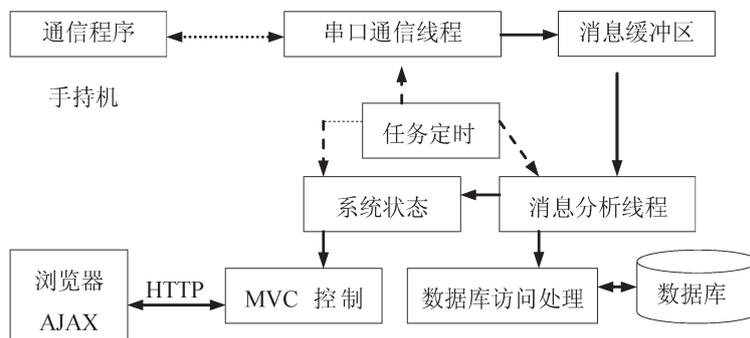


图2 系统的软件模块结构  
Fig.2 System software module structure

以下就Web服务器端的主要模块进行简要说明:

1) 串口通信线程(commService):负责与串口的连接和循环发送轮询消息。并将接收的消息送消息缓冲区。

2) 消息缓冲区(messageBuffer):实现消息的封装及处理。其中包括两个重要的方法,getMessage()方法从消息缓冲区取消息并进行处理;setMessage(String mess)方法将消息送消息缓冲区;由于该类的方法是由线程调用,所以涉及同步处理和对消息队列的加锁访问控制。往消息缓冲区写入消息由串口通信线程在收到一个完整消息后调用。

3) 消息分析处理线程(messageMointor):专门接受处理消息队列中消息的线程。它将循环读取消息缓冲区的消息。它将调用messageBuffer类的getMessage()方法对消息进行分析处理,根据消息内容分析更新系统的状态和进行数据库登记处理。

4) 系统工作状态(CurrentStatus):引入若干Map对象记录系统的状态信息。系统的工作信息将记录到数据库中,同时,为了提高系统的工作效率,通过CurrentStatus类定义静态属性记录系统的当前工作状态信息。例如:

Map<Integer,Platform> alltw:所有台位的工作信息;  
Map<Integer,myPos> allgx:所有工序的工作信息;

其中,myPos记录工序的属性和访问方法,主要属性有工序编号、所在台位、状态信息、完成辆份数量、开工时间、单元开工时间等,Platform记录台位的属性和访问方法。

5) MVC控制器(monitorController):实现要监控信息的数据提取的处理逻辑,接受客户浏览器的访问请求,用Google的Gson工具将结果包装为Json串格式返回给请求者。

6) 数据库访问处理(repairlog):实现业务数据的记录处理,以便为数据统计和数据查询服务。利用JdbcTemplate实现数据库的访问处理,利用Spring IOC实现Bean的数据源连接的注入。

7) 任务定时(TaskService):借助Spring的基于注解的任务定时调度功能规划系统的工作时间,及时地完成每天的初始准备和收尾工作。

### 3 关键技术处理

#### 3.1 用AJAX间隔访问Web服务实现动态监测

##### 3.1.1 用Spring实现REST风格的Web服务

在Web服务端,用Spring编程实现REST风格的Web服务。以工序监控为例,客户方发送访问请求后,Web服务端将返回各工序进展的信息作为响应。由于Web服务与浏览器之间不能直接传递对象,必须采用特殊手段进行包装处理。常用包装形式有XML和Json<sup>[10]</sup>,Json是一种易于被JavaScript解析的对象消息格式,系统利用Google提供的Gson对象的toJson方法进行转换<sup>[11]</sup>。以下控制器代码将存储所有工序的集合对象转换为Json字符串。

```
import javax.servlet.http.HttpServletResponse;
import java.util.*;
import com.google.gson.Gson;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.*;
@Controller
public class monitorController {
    @RequestMapping(value="/lookforx",method=RequestMethod.GET)
    public void display(HttpServletResponse response){
        response.setHeader("ContentType","text/plain");
        Gson gson = new Gson();
        try {
            Collection<myPos> m= CurrentStatus.getAllgx().values();
            // 获取所有工序的状态信息
            String listToJson = gson.toJson(m);//将对象转换为Json格式
            response.getWriter().write(listToJson);
        } catch (Exception e) { }
    }
}
```

myPos为一个封装工序当前状态信息的类,其中含有工序的编号、开工时间、当前所在台位、工程进展状态等信息。通过CurrentStatus类的静态属性记录有各工序的状态信息。值得注意的是,由于响应消息是将字符串返回给客户方,需设置http响应头内容格式为text/plain。

##### 3.1.2 浏览器页面通过AJAX实现工序进展的动态监视

系统通过各种颜色的图片来表示工序的状态,对页面上的显示元素进行有序的标识,以便程序中能循环处理。通过AJAX技术实现内容显示的实时变化<sup>[12]</sup>。以下为工序进展及状态查询的Javascript脚本函数代码,通过Javascript的setTimeout方法实现定时间隔执行,让initProcess方法每隔2秒执行1次,从而完成对

服务器数据的轮询访问。在 `initProcess` 方法内,首先,通过 `XMLHttpRequest` 对象发送 URL 访问请求,通过 `XMLHttpRequest` 对象的 `responseText` 属性得到服务端的响应消息。然后,调用 JavaScript 的 `eval` 函数将服务器返回的 JSON 文本转化为对象形式。最后,根据对象数据项的值,利用 DHTML 技术更改页面元素的 HTML 内容,从而实现对相关对象的实时监测显示<sup>[12]</sup>。

```
function initProcess(){
    myurl="lookforgx";//服务器的URL
    xmlhttp.Open("GET",myurl,false);// xmlhttp为创建的XMLHttpRequest对象
    xmlhttp.send();//发送URL访问请求
    var str = xmlhttp.responseText;
    var jsonobj=eval("(" + str + ")");//将Json消息转化为对象形式
    for (i=0;i<jsonobj.length;i++){
        var gx=parseInt(jsonobj[i].gx);//获取工序编号
        document.getElementById("t"+gx).innerHTML="" + jsonobj[i].amount;
        //在页面中相应工序位置显示当前完成的辆份
        document.getElementById("img"+gx).src="images/" + jsonobj[i].status + ".gif";//显示工序的状态指示图片
        document.getElementById("img"+gx).alt="台位:" + jsonobj[i].pos;
        //鼠标移上时将显示工序所在台位
    }
    setTimeout("initProcess()",2000);//每隔2秒更新显示
}
```

图3为系统的工序监控界面。每个工序的当前状态通过不同颜色的图标指示。工序完成的检修数量通过数字显示。

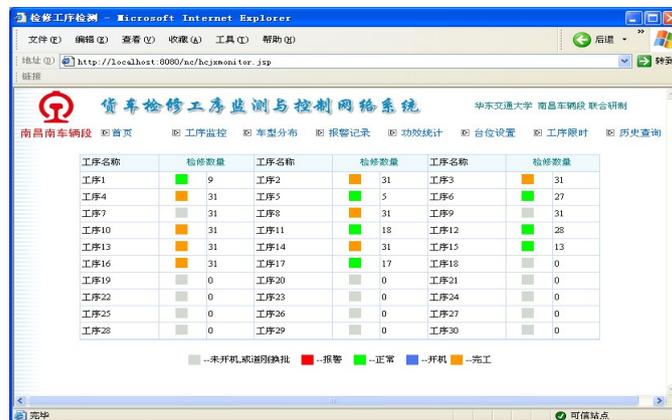


图3 监视工序的状态及进展

Fig.3 The process status and progress monitoring

### 3.2 利用Spring任务定时控制线程的工作

为了让系统在非工作时间处于“休息状态”,尤其是通信线程,在休息时间发送轮询信号是没有意义的。有必要对系统的线程和工作状态进行特殊处理。Spring的任务调度为应用系统中特定任务的执行安排提供了有效的支持。Spring的 `TaskScheduler` 可通过 `@Scheduled` 注解进行定义,其任务施加执行对象可以是任何一个方法,给任务编写提供了很大方便。注解中通过 `cron` 参数指定任务执行的时间点。以下为每天6点启动消息接收线程的实现代码:

```
import org.springframework.scheduling.annotation.Scheduled
import org.springframework.stereotype.Service;
@Service
```

```

public class CommTask {
    @Scheduled(cron="0 0 6 * * ?") //每天6点启动
    public void begin() {
        if (! CurrentStatus.monitormessage.isAlive()) {
            CurrentStatus.monitormessage.start();//启动消息接收线程
        }
    }
}

```

值得一提的是, Spring的任务定时要用到 AOP(面向切面编程)的功能, 因此在应用工程的类路径中引入 aopalliance.jar 包。

### 3.3 利用移动通信技术实现手持机与主机的通信

Web 服务器方通过 Java 串口编程与手持机通信。移动通信模块连接在串口上, 实现无线信号和串口数据的转换, 因此, 实际的数据通信变成串口通信。在手持机端用户界面简单, 用户通过点击按钮发送自己的状态进展消息, 向主机报告在哪个台位完成了一份维修任务。

#### 3.3.1 主机与手持机的消息交换

主机采用轮询机制询问每个工序和检测道的工作状态信息。监管人员在完成一道工序后将发送信号给主机。主机每次在轮询一个手持机时发送“ask, 手持机号”的消息。只有匹配编号的手持机才给予应答。

手持机没应答, 主机在超时后, 将自动询问下一手持机, 如此循环往复。手持机发送信号反馈工序和车道的工作状态。手持机的应答消息中包括: 消息识别码, 工序或车道号, 工作进程, 所在台位等。

通信消息的设计与处理是项目较为复杂部分, 在消息接受处理过程中涉及各类状态变量的改变和数据记录登记, 如: 工序的开工时间、超时等信息取决于对收到消息的分析处理。这里消息分: 车道新批开始、工序进展消息、工序请求关机等。

#### 3.3.2 服务器端的 Java 串行通信编程

服务器端的串口通信线程从串口接收数据, 将数据放入消息缓存。消息处理线程在缓存中有消息时将从 wait 中醒来, 然后读取消息进行解析处理。服务端采用 Java Comm 工具包实现对串口通信的编程处理<sup>[13]</sup>。Java 串口通信的基本过程是: ① 装载驱动程序; ② 打开串口, 进行串口的初始化; ③ 通过串口的输入输出流实现数据读写访问。串口通信编程处理的难点是保持通信双方收发消息的同步, 数据消息格式的设计与多线程规划是提高通信处理效率的关键。本系统采用的无线传输, 在通信距离远时还需要添加中继进行转发。

## 4 结束语

系统通过手持机与主机之间的无线通信收集工地操作员发送的工序进展信息。利用 AJAX 技术实现与 Web 服务器的通信及页面内容的动态显示。管理人员可以通过 Web 界面对工地的维修进展进行实时监测, 对各道工序的工作状态能进行记录、报警和统计。该系统的设计思路也可用于其他类似的 Web 远程监控系统中。

### 参考文献:

- [1] 吕海东, 高际邦. 基于 DWR 反向 AJAX 的 Web 监控系统[J]. 微型机与应用, 2010, 29(13): 60-62.
- [2] 徐永晋, 陶万强. 基于 Web 的智能居家生活系统监控软件关键技术及实现[J]. 仪表技术, 2010(3): 11-13.
- [3] 范云芝, 陈树平. 利用 Socket 实现基于 Web 的远程监测系统[J]. 陕西工学院学报, 2005, 21(1): 53-55.
- [4] 赵振华, 廉东本. 基于 Flex 的 WebGIS 框架的研究与应用[J]. 计算机系统应用, 2011(11): 146-149.
- [5] 熊亮, 张锴. 一种新型接触网开关综合监控系统的设计与实现[J]. 华东交通大学学报, 2012, 29(4): 86-91.

- [6] 丁振凡. AJAX技术在网络考试分析中的应用[J]. 华东交通大学学报, 2007, 24(5): 74-76.
- [7] 张曼琳, 顾容, 徐梦. 基于Ajax技术的高校在线考试系统的设计与实现[J]. 中国教育信息化, 2012(5): 44-48.
- [8] 丁振凡. 利用AJAX结合VML实现Web图形化监控[J]. 电脑编程技巧与维护, 2012(12): 84-85.
- [9] 窦慧东. 铁路货车轮对检修工序控制系统[D]. 西南交通大学, 2005: 5-6.
- [10] 丁振凡. 基于AJAX结合Spring MVC的信息访问服务模式研究[J]. 计算机时代, 2012(6): 25-29.
- [11] 丁振凡. Spring REST风格Web服务的Json消息封装及解析研究[J]. 智能计算机与应用, 2012, 2(2): 9-11.
- [12] 丁振凡. Web编程实践教程[M]. 北京: 清华大学出版社, 2011.
- [13] 丁振凡, 王小明, 邓建明, 等. 基于Java的串口通信应用编程[J]. 微型机与应用, 2012, 31(13): 84-86.

## Development of Truck Maintenance Procedures Monitor System Based on Web

Ding Zhenfan<sup>1</sup>, Wang Xiaoming<sup>1</sup>, Deng Jianming<sup>2</sup>, Qiu Xinliang<sup>2</sup>

(1. East China Jiaotong University, Nanchang 330013, China; 2. Nanchang Railway Bureau, Nanchang 330001, China)

**Abstract:** The paper discusses the software design of the train repair procedure monitoring and management. In the monitor system, wireless communication is applied for the data transmission between the wireless handsets and communication module connected with the web server serials. Then the host computer inquires work progress from handsets through polling, and the acquisition and analysis of communication information is fulfilled through multi-threaded coordination. Meanwhile, the system realized by B/S mode carries out the real-time monitoring maintenance process in the browser page. On the Web server, the Spring is used to program daily task start and stop in REST style Web service. While AJAX technology is adopted to achieve access to the Web service and the dynamic display of the results on the page, the object message is transmitted to the browser by Json transform.

**Key words:** Spring; remote online monitor; task schedule; AJAX; Json; wireless communication