文章编号:1671-4598(2019)02-0123-06 DOI:10.16526/j.cnki.11—4762/tp.2019.02.028 中图分类号:TP311 文献标识码:B

基于 B/S 架构的管理系统软件开发

吴晓珊，曹旭东，王 森，魏文龙
（中国石油大学（北京）地球物理与信息工程学院，北京 102249）

摘要：为了优化单位对警用摩托车的管理，使其对车辆调配更加便捷和合理，运用计算机较大的信息存储空间和高性能的处理能力，将摩托车的坐标位置，使用情况，故障情况等车辆基本信息，做出详细的分类和记录，使用户可以直观地在该系统中随时监控车辆信息的变动，并能够及时处理车辆工作中遇到的紧急情况；文章提出了一种基于 B/S 架构警用摩托车管理系统的设计方案，系统采用 WEB 方式实现；服务器通过 TCP 协议获取摩托车终端所采集的数据，并将数据存储到 MySQL 数据库。文章中详细介绍了服务器对数据的存储，WEB 系统的开发和研究，对终端设备的静态数据和动态数据进行分析和呈现。

关键词：B/S 架构；WEB 系统；HTTP 协议；MySQL 数据库

Design of Police Motorcycle Management System Based on B/S Architecture

Wu Xiaoshan, Cao Xudong, Wang Sen, Wei Wenlong
(China University of Patroleum, Beijing 102249, China)

Abstract: In order to optimize the management of the police motorcycle for the unit, make it more convenient and rational for vehicle deployment, using the large information storage space and the processing ability of high performance of computer, making a detailed classification and record according to the vehicle basic information, such as the coordinate position of the motorcycle, vehicle usage, the fault condition, etc, users can directly monitor the change of vehicle information in the system, and timely handle the emergency situation encountered in the vehicle operation. This paper presents a design scheme of police motorcycle management system based on B/S architecture, and the system is implemented in a WEB way. The server obtains the data collected by the motorcycle terminal through the TCP protocol and stores the data to the MySQL database. In this paper, the storage of data, the development and research of WEB system, the static data and dynamic data of the terminal equipment are analyzed and presented in detail.

Keywords: B/S architecture; WEB system; HTTP protocol; MySQL database

0 引言

目前，4G 网络已经全面覆盖，移动互联网也处于高速发展中，高科技信息化技术已经深入到生活的各个方面，随之给我们的工作生活带来了更加便捷、灵活的工作方式。警用摩托车管理系统让用户通过浏览器来直观地获取车辆的地理位置以及基本信息数据，并且能够直接播放车辆的行车记录视频。

警用摩托车管理系统由安装于摩托车上的智能化终端设备、服务器、MySQL 数据库组成。终端采集上传摩托车数据、视频信息；服务器接收终端上传的数据，并将数据存入数据库，并接受前端发起的请求，根据请求调用数据库提取相应信息，打包处理后通过响应的方式交给前端。

1 整体设计

系统整体设计如图 1 所示。系统由安装于摩托车上的智能化终端设备、服务器、MySQL 数据库组成。智能化终端设备完成对摩托车位置、车速故障信息等数据的采集、上传。服务器将终端上传的数据解码存入数据库；接收来自前端的请求，包括数据请求和视频直播请求，收到请求后会根据数据请求的具体内容，从数据库检索数据，并将数据转换成 JSON 格式给前端应答，视频请求则不作处理，以广播的形式直接下发给终端。前端接收用户请求向服务器提取数据，并将数据进行包装，直观地展现给用户。

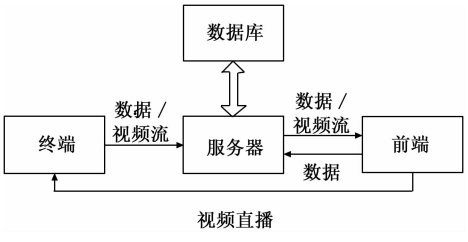

图 1 系统整体设计框图

2 服务器

服务器分为两部分：一部分为 TCP 服务器，与硬件终端设备通信，接收、解析、保存数据；另一部分为 HTTP 服务器，与前端通信，提取、打包、发送数据。

2.1 TCP 服务器

由于摩托车终端数量较多且数据发送频繁，故选用

收稿日期:2018-07-02； **修回日期**:2018-07-27。

作者简介:吴晓珊(1993-)，女，河北任丘人，硕士研究生，主要从事通信与信息系统、计算机软件系统设计方向的研究。

NETTY 框架搭建 TCP 服务器。

Netty 是一个基于非阻塞 IO 的快速开发高性能、高可靠性的网络服务器－客户端架构。Netty 封装了 Java NIO 那些复杂的底层细节，提供简单好用的抽象概念来编程，广泛应用于客户端与服务器长连接、高并发的场景。最近几年，Netty 在计算机互联网行业迅速流行开来，已经成为 Java 通信编程架构的第一选择。

对于 NETTY 服务器端而言，属于被动接收请求，服务端 bind 端口采用随机的方式，以避免单台服务器多端口之间的冲突。通过 ServerBootstrap 创建服务器端通讯连接。此外，采用 ChannelGroup 类，既可以自动将被关闭的 Channel 从 ChannelGroup 中删除，还可以统一关闭 ChannelGroup 中的所有通道。

部分代码如下：

```
public void bind() {
    EventLoopGroup bossGroup = new NioEventLoopGroup();//用于接收客户端连接
    EventLoopGroup workGroup = new NioEventLoopGroup();//用于进行网络读写通信
    try {ServerBootstrap b = new ServerBootstrap();
  b.group(bossGroup, workGroup);//绑定两个线程
  b.channel(NioServerSocketChannel.class);
  b.option(ChannelOption.SO_BACKLOG, 1024);
  b.childHandler(new ChildChannelHandler());
    // 绑定端口
  ChannelFuture f = b.bind(53606).sync();
    // 等待服务端监听端口关闭
  f.channel().closeFuture().sync();
  } catch (Exception e) {
    e.printStackTrace();
  } finally { //退出
    bossGroup.shutdownGracefully();
    workGroup.shutdownGracefully();
    }
    }
```

服务端在初始化的时候，由于我们需要对接收的数据进行处理，于是在通道中添加我们自己接收数据实现的方法类，以及 5 分钟判断客户端的在线离线状态。并且对于粘包拆包问题的处理上，采用的了基于换行符的处理方式。

部分代码如下：

```
protected void initChannel(SocketChannel ch) throws Exception {
  System.out.println("IP:" +
  ch.localAddress().getHostName());
    System.out.println("Port:" +
  ch.localAddress().getPort());
    // 5 分钟判断在线离线
  ch.pipeline().addLast(new IdleStateHandler(300,0,0));
    // 半包处理[基于换行符]
  ch.pipeline().addLast(new
  LineBasedFrameDecoder(1024));
    // 字符串编码
  ch.pipeline().addLast(new StringDecoder());
    // 字符串解码
  ch.pipeline().addLast(new StringEncoder());
    // 在管道中添加我们自己的接收数据实现方法
  ch.pipeline().addLast(new MyServerHanlder());
    }
```

在保证服务器能够接收到所有终端的数据并且不丢失的前提下，还要根据终端用户 ID 号，对数据包是否有效进行判断。如果 ID 与数据库中注册 ID 符合则服务器接收数据并存入数据库，如不符合，则拒绝接收。判断用户 ID 流程图如图 2 所示。

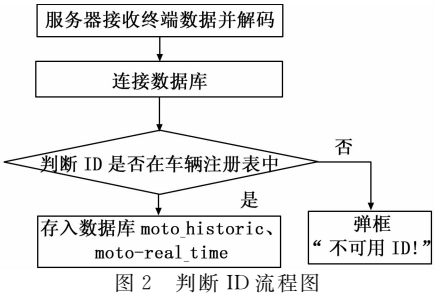

图 2 判断 ID 流程图

对于终端的在线离线状态，将借助 NETTY 提供的 IdleStateHandler 类，如下：

```
public IdleStateHandler(int readerIdleTimeSeconds, int writerIdleTimeSeconds, int allIdleTimeSeconds) {this((long)readerIdleTimeSeconds, (long)writerIdleTimeSeconds, (long)allIdleTimeSeconds, TimeUnit.SECONDS);}
```

即实现心跳机制的类，来处理客户端的连接状态。其中：readerIdleTimeSeconds 为读超时；

writerIdleTimeSeconds 为写超时；

allIdleTimeSeconds 为所有超时。

而根据实际情况以及产品需求，在服务器初始化时，通道中添加的方法为：

ch.pipeline() addLast (new IdleStateHandler (300, 0, 0))；即读超时设置为 5 分钟。系统认为 5 分钟内未发送数据，服务器将断开连接，并将状态记录存储到数据库中，同时不再刷新该终端的实时状态。判断在线、离线状态流程图如图 3 所示。

2.2 HTTP 服务器

HTTP 用于接受前端的请求，根据请求调用数据库提取相应信息，打包处理后通过响应的方式交给前端，数据交换格式为 JSON，一种轻量级的数据交换格式。

HTTP 服务器接收前端的请求，并对请求进行解析，解析方式为字符串匹配，根据解析出来的条件对数据库执行检索。完成对数据库的操作后将数据打包成 JSON 格式，响应给前端。

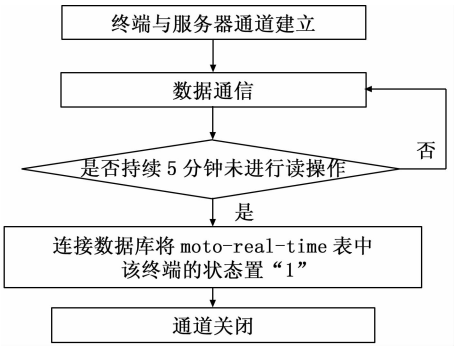

图 3　判断状态流程图

JSON 作为更轻、更便捷的 Web service 客户端格式，目前广泛应用于编程开发中。相较于 XML 格式，它更加便于读取，JSON 中的分隔符限于单引号、小括号、大括号等，而 JavaScript 引擎对数据结构的内部表示正好与这些符号相同，以此简化了数据的访问。此外，它的另一个优点是其非冗长性。传统的 XML 标记会增加数据交换时间，必须包括打开和关闭标记，才能满足标记的依从性，而在 JSON 中所有这些要求只需要通过括号即可满足。所有 JSON 的线上传输效率更高。

部分请求响应 JSON 格式如下：

```
if(包含 all_motor)//实时位置
{
"Motor":
[
{
"ID":"4661",
"data":
{
"longitude":"XXXX",
"latitude":"XXXX",
……}
},
{
"ID":"4662",
"data":
{ ……}
},…… ]
}
if(包含 real)//实时轨迹
{
```

截取命令字段中的 ID，通过 ID 在数据库 moto _ real _ time 表中查找　最新更新时间，根据这个时间以及 ID，在 moto _ historic 表中检索　当天该 ID 的所有数据，数据格式同上。

```
}
if(包含 video)//视频开关
{
```

直接将 video 字段后的命令广播发送给终端。

数据格式：ID＝XXXX/state＝X/channel＝X/

```
}
```

3　MySQL 数据库

MySQL 数据库体积小、速度快，多用户、多线程并且能够处理大量数据[9]，主要以小型应用为主，主要设计目标是实现数据操作速度优化且不影响 SQL 支持性能。应用程序通过 JDBC API 接口引入 JDBC 驱动对数据完成操作，部分程序如下：

```
public static void Table_Create()  {
Class. forName("com. mysql. jdbc. Driver");
try(
  //使用 DriverManager 获取数据库连接，
  //其中返回的 Connection 就代表了 Java 程序和数据库的连接
  //不同数据库的 URL 写法需要查驱动文档知道，用户名、密码由 DBA 分配
Connection conn = DriverManager. getConnection(
  "jdbc:mysql://127. 0. 0. 1:3306/mysql"
  , "root", "123456");
  //使用 Connection 来创建一个 Statment 对象
Statement stmt = conn. createStatement()){
stmt. executeUpdate("create table if not exists motor_register"
//创建表格字段名
  +"(XXX int not null primary key,"
  +" XXX varchar(255),"
……
  +" XXX varchar(255));" );
  }
  }
```

根据数据量分布以及客户需求，设计三个车辆信息表，分别为摩托车登记表、摩托车实时状态表和摩托车历史数据表。登记表和实时状态表中都设置以 clientID 为主键且非空，由于在此两表中 clientID 不可重复存储，此外实时状态表中的 uploadDate 字段是联合主键，为确保实时表格的准确性。三个表格之间的关系如图 4 所示。

表 1　车辆登记表

Field	Type	Null	Key	Default
clientID	Int(11)	NO	PRI	NULL
motobrand	varchar(255)	YES		NULL
motolicense	varchar(255)	NO		NULL
policeID	varchar(255)	NO		NULL
motostatus	varchar(255)	YES		NULL

摩托车登记表用于记录摩托车序列号、摩托车品牌、摩托车车牌、归属警员号、摩托车状态等基础信息，每辆摩托车拥有唯一序列号，只有经注册将信息录入登记表，服务器才会接收 ID 号符合的数据包存入数据库，进而调取

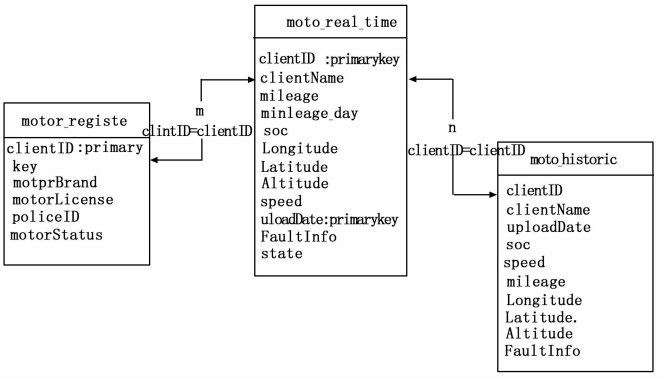

图 4 数据库 E－R 图

车辆状态信息。数据名称以及数据类型属性如表 1 所示。

摩托车当前状态表存储摩托车最新状态，数据包括用户 ID、用户名称、状态更新时间、速度、里程、当天里程在线离线状态等，一个 ID 对应一条记录。当终端被判断离线后，状态变为“1”，再将数据异步存入至历史表。历史状态表存储所有车辆的所有信息，方便前端调取查阅车辆历史轨迹，也可以检索某时间段某辆车的详细数据。当前状态表如表 2 所示。当前状态表如表 3 所示。

表 2 摩托车当前状态表

Field	Type	Null	Key	Default
clientID	Int(11)	NO	PRI	NULL
clientName	varchar(255)	YES		NULL
mileage	varchar(255)	YES		NULL
mileage_day	varchar(255)	YES		NULL
soc	varchar(255)	YES	PRI	NULL
Longitude	varchar(255)	YES		NULL
Latitude	varchar(255)	YES		NULL
Altitude	varchar(255)	YES		NULL
speed	varchar(255)	YES		NULL
uploadDate	varchar(255)	NO		NULL
FaultInfo	varchar(255)	YES		NULL
state	varchar(255)	YES		NULL

表 3 摩托车历史状态表

Field	Type	Null	Key	Default
clientID	varchar(255)	YES		NULL
clientName	varchar(255)	YES		NULL
uploadDate	varchar(255)	YES		NULL
Longitude	varchar(255)	YES		NULL
Latitude	varchar(255)	YES		NULL
Altitude	varchar(255)	YES		NULL
mileage	varchar(255)	YES		NULL
soc	varchar(255)	YES		NULL
speed	varchar(255)	YES		NULL
FaultInfo	varchar(255)	YES		NULL

4 视频直播

视频直播是通过基于 Android 操作系统的终端设备，搭建基于 RTMP 传输协议的流媒服务器，最终在前端流媒体播放器上实现。摄像头开关由前端控制，媒体流通过 RTMP 协议进行传输和播放。

RTMP（Real Time Messaging Protocol），即实时消息传送协议，是由 Adobe 公司提出的一种应用层的协议，用来解决多媒体数据传输流的多路复用和分包问题。并且随着 VR 技术的发展和视频直播等领域的发展，RTMP 协议逐渐热门起来，在高速发展的流媒体市场中得到广泛的应用[10]。

视频直播流程分为以下几步：（1）采集；（2）处理；（3）编码和封装；（4）推流到服务器；（5）服务器流分发；（6）播放器流播放。

当前端发送开启直播命令时，终端摄像头启动采集实时音视频，并编码成 H.264 码流，视频服务器启动建立与终端的链接，然后接收码流并推流至前端播放以及进行实时存储，以 MP4 格式保存至本地。

终端视频格式不是可以直接观看的 MP4 格式，而是封装为 H.264，且每个视频时长 10 秒。由于前端 HTML5 不支持此类视频的播放，需要视频服务器对视频进行实时的格式转换处理，服务器中调用了 FFmpeg 应用程序，将 H264 转换为 MP4，处理后的视频放到指定路径，供前端调用。

5 实现效果

警用摩托车管理系统前端主要包括系统登录界面、车辆注册界面、实时监控界面、信息检索界面、车辆信息界面五个部分。

5.1 系统登录及注册

警用摩托车登录系统界面部分用于保障系统安全，设置使用权限，用户只需要输入提前设置的用户名、密码便能够成功登录该管理系统，然后进入主界面。

警用摩托车系统注册界面，用于录入摩托车基本信息，包括摩托车序列号、摩托车品牌、摩托车车牌、警员号、摩托车状态信息等，限制摩托车序列号位数字，各栏均不为空，否则会提示错误，注册不成功。注册成功后，会将信息写入数据库 motor _ register 表，只有在 motor _ register 表中录入信息的车辆，在界面中才可以查询其信息。

5.2 实时监控界面

实时监控界面如图 5 所示，主要功能为在地图上直观显示所有车辆位置，同时在最标点标签上显示车辆 ID 以及电量信息等简要信息。

5.3 信息检索界面

信息检索界面如图 6 所示，可通过输入 ID 号查找某辆车信息，显示实时轨迹并播放实时视频。视频窗口上方显示该车辆的状态，包括实时速度、里程等。具体操作是在

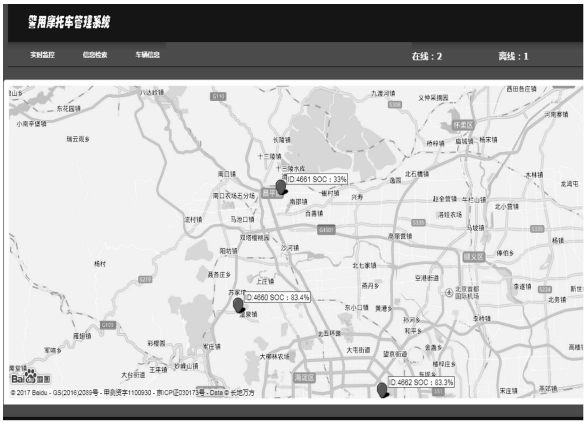

图 5　实时监控界面图

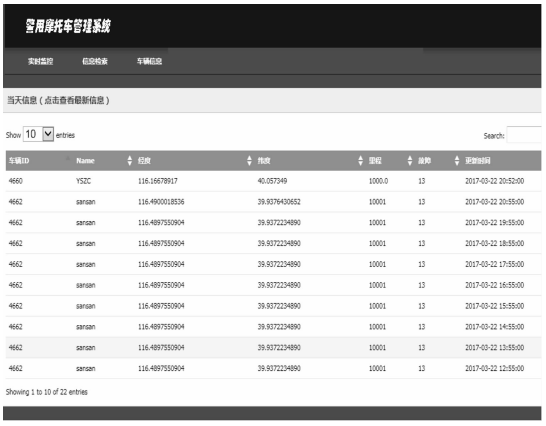

图 7　车辆信息界面图

ID 输入框内输入车辆 ID，单击 ID 输入框后的确定按钮，跳转查看该车辆的信息，若输入 ID 不存在，则会弹框提示。选择好起止时间后，单击时间选择框后的确定按钮，可以查看该车辆在某个时间段的轨迹以及录像。

图 6　信息检索界面图

5.4　车辆信息界面

车辆信息界面默认显示所有车辆最新信息如图 7 所示，信息包括车辆 ID、摩托车品牌、经纬度、里程、故障信息以及更新时间等。可根据表头某一项信息对表中元素进行排序。表格右上方 Search 框可输入车辆 ID，检索某一车辆的所有信息，点击表格标题栏可切换显示所有车辆当天所有信息。

6　结束语

在 4G 网络全面覆盖和互联网产业的迅速发展的背景下，使人们的工作方式越来越多样化，利用网络的便捷设计的警用摩托车管理系统，方便了单位实现内部车辆合理调配以及有效排查车辆故障获取车辆基本信息，可以极大地提高公安人员的执法效率。本文设计的软件主要是基于 B/S 架构，实现了警用摩托车基本数据采集、数据存储管理和行车记录视频直播等功能。

本文完成的主要工作有以下两大部分内容，即后台服务器和前端浏览器。

6.1　后台服务器

系统服务器车辆基本信息通讯部分主要由 NETTY 框架、HTTP 协议和 MySQL 数据库搭建而成，行车记录视频由 RTMP 搭建而成。实现了接收终端发送的 protobuf 数据并解码存入已设计好的数据库，数据库的设计满足每次访问数据库的效率最优。HTTP 协议用于接收前端请求数据并作出响应，根据请求提取数据库相应信息，数据将以 JSON 格式发送给前端 [8]。行车记录视频通过搭建基于 RTMP 传输协议的服务器实现视频流推送到 RTMP 服务器，再将视频发送给前端进行直播。此外，终端录制的视频可以通过服务器转化为 MP4 格式保存至本地。

6.2　前端页面

主要应用 HTML、CSS 和 JavaScript 技术设计网页，可实现在地图上展示车辆位置以及车辆基本信息，可针对某辆车某时间段行车轨迹进行查询和观看行车记录视频。

本文运用了目前最流行的框架和直播技术，设计的警用摩托车管理系统软件，最终通过测试成功实现以上所介绍的功能。为办公人员提供了便捷的管理服务，大大优化了工作效率。

参考文献：

[1] 刘　鹏，宋　为，万　俊. 基于 C/S 与 B/S 架构的科研项目管理系统 [J]. 软件导刊，2010，9 (1)：110－111.

[2] 苏东震，陈　明，史忠植. 基于 B/S 架构的数据挖掘原型系统的设计与实现 [J]. 微电子学与计算机，2008，25 (12)：131－133.

[3] Li Q S. Design for Property Management System Based on B/S Architecture [J]. Advanced Materials Research，2014，2863 (846).

[4] Ling Z，Qiao Y S，Xiao X L. The design and implementation of the colleges network examination system based on B/S architecture [J]. Applied Mechanics and Materials，2014，3082 (543).

[5] 龚　鹏，曾兴斌. 基于 Netty 框架的数据通讯服务系统的设计 [J]. 无线通信技术，2016，25 (1)：46－49.

[6] 滕阳阳，胡 栋. 基于 Netty 的 HTTP 协议栈的扩展设计与实现 [J]. 无线通信技术，2017，26 (3)：38－42.
[7] Velan P，Jirsík T，Celeda P. Design and Evaluation of HTTP protocol parsers for IPFIX measurement [M]. Springer Berlin Heidelberg：2013－06－15.
[8] 卢少军，过丹婷，刘守印. HTTP 协议与 JSON 格式在 ZStack 中的实现与应用 [J]. 电子测量技术，2016，39 (11)：100－104.
[9] 李荣国，王 见. MySQL 数据库在自动测试系统中的应用 [J]. 计算机应用，2011，31 (S2)：169－171，175.
[10] 王 艳. Android 系统中 RTMP 流媒体直播的设计与实现 [J]. 电视技术，2017，41 (1)：64－67.
[11] 雷霄骅，姜秀华，王彩虹. 基于 RTMP 协议的流媒体技术的原理与应用 [J]. 中国传媒大学学报（自然科学版），2013，20 (6)：59－64.
[12] 韦立梅，张淑荣. 基于 HTML＋CSS＋JQuery 的网站开发简述 [J]. 电脑与电信，2017 (9)：69－70，76.
[13] Wei S Y，Xhakaj F，Ryder B G. Empirical study of the dynamic behavior of JavaScript objects [J]. Software：Practice and Experience，2016，46 (7).
[14] Yue C，Wang H N. A measurement study of insecure Java Script practices on the web [J]. ACM Transactions on the Web (TWEB)，2013，7 (2).
[15] Anonymous. Research and markets：beginning HTML，XHTML，CSS，and JavaScript：essential update to the key Web authoring standards [J]. M2 Presswire，2010.
[16] Gardner P A，Maffeis S，Smith G D. Towards a program logic for JavaScript [J]. ACM SIGPLAN Notices，2012，47 (1).

(上接第 113 页)

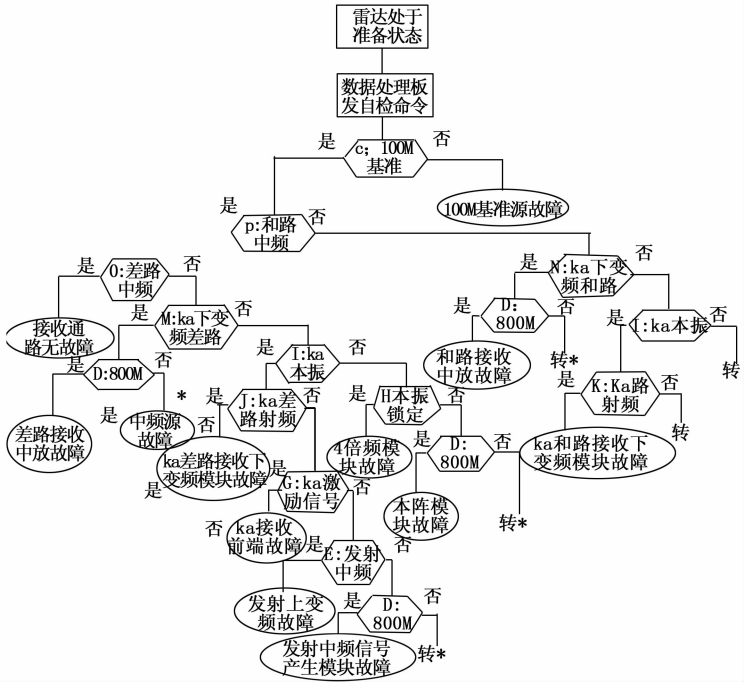

图 9 接收和路诊断隔离流程图

找出产品测试性设计的薄弱点，进行优化和改进，最终使得产品满足指标要求。

5 结论

复杂电子产品的测试性与后期的维修性和保障性密切相关，测试性设计是否优良直接影响产品的综合性能。本文以某雷达系统为例分析测试性设计的基础条件和要求，并对测试性验证试验后暴露的测试性问题进行了分析和改进，最终回归试验得到了满足指标要求的量化结果及合理可行的诊断序列。结果表明，该测试性设计改进优化方法，能够在对产品进行尽可能少的硬件更改情况下大大提升测试性水平，同时为类似处于设计阶段的复杂电子设备测试性设计提供有用参考。

参考文献：

[1] 田 仲，石君友. 系统测试性设计分析与验证 [M]. 北京：北京航空航天大学出版社，2003.
[2] GJB2547A－2012 装备测试性通用要求 [M]. 中国人民解放军总装备部，2012.
[3] 贲 德，韦传安，林幼权. 机载雷达技术 [M]. 北京：电子工业出版社，2006.
[4] Tsai Y T，Hsu Y Y. A study of function－based diagnosis strategy and testability analysis for a system [J]. Journal of Mechanical Engineering Science，2011，Vol. 0 Part C：1－11.
[5] Wang H X，Ye X H，Wang L. Research on optimizing the fault diagnosis strategy of complex electronic equipment [J]. ComSIS，2010，7 (1)：223－230.
[6] 林典雄，李 岩，张 勇. 航空装备综合诊断的现状与发展设想 [J]. 航空工程进展，2011，2 (3)：349－354.
[7] 温熙森，胡 政，易晓山，等. 可测试性技术的现状与未来 [J]. 测控技术，2000，19 (1)：9－12.
[8] 连光耀，黄考利，陈建辉，等. 复杂电子装备智能测试性设计技术 [J]. 兵工自动化，2006，25 (8)：71－73.
[9] 苏永定. 机电产品测试性辅助分析与决策相关技术研究 [D]. 长沙：国防科技大学，2004.
[10] 于功敬，厚 泽，王振华. 装备测试性设计与诊断策略优化技术研究 [J]. 电子测量技术，2012，35 (7)：8－11.
[11] 连光耀，黄考利，赵常亮. 复杂电子系统测点与诊断策略优化方法 [J]. 系统工程与电子技术，2004，26 (11)：1739－1742.
[12] 畅 玲，党红肖. 数传接收机系统可测试性设计 [J]. 火控雷达技术，2015，44 (2)：78－82.
[13] 柳新民，邱 静，刘冠军，等. 诊断间歇故障降低 BIT 虚警 [J]. 测试技术学报，2004，18 (4)：369－372.
[14] 杨 鹏，邱 静，刘冠军. 测试不可靠条件下的诊断策略优化方法 [J]. 仪器仪表学报，2008，29 (4)：850－854.
[15] Liu G，Li F. A new optimization selection for test process of equipment manufacturing based on fused algorithm [A]. International Conference on Advances in Mechanical Engineering and Industrial Informatics (AMEII 2015) [C]. 2015：402－405.