

基于机务段检修管理系统程序设计

Program Design of Management System on Locomotive Depot Repair

徐 杨 (中南民族大学工商学院计算机科学与技术系,湖北 武汉 430223)

摘 要

结合目前铁路机务段数据信息计算机自动化管理的现状,采用 C++Builder5.0 设计开发了机车检修前台应用程序,后台数据库采用 SQL Server2000 管理,以机车检修管理系统设计为例,介绍其开发过程和所涉及到的问题及解决方法。

关键词:机务段,数据结构,窗体

Abstract

The foreground application programs of the locomotive maintenance and repair are designed and exploited by using C++Builder5.0 and the background database is managed by SQL Server2000.This paper takes the management system design of the locomotive maintenance and repairs for example,and introduces the exploiture process,the problems and solution.

Keywords:locomotive depot,data structure,forms

机务段检修管理系统是机车检修单位不可缺少的部分,本文以机车检修为例,介绍其检修管理系统的一般开发过程,包括后台数据库的建立和维护以及前端应用程序开发两个方面。对于前者要求建立起数据一致性和完整性强、数据安全性好的数据库。而对于后者则要求具备应用程序功能完备,易使用等特点。系统采用 MS SQL Server2000 作为后台数据库,Borland 公司的 C++Builder5.0 作为编译环境。

1 机车检修管理分析

检修生产过程管理系统是机务段微机网络的子系统,该查询系统是专门针对机车运用人员和管理人员的特殊管理性质而设计的,程序设计与 Windows 操作平台有较好的兼容性。通过简洁的菜单提示,以指触显示屏的方式达到调用存储数据,直接在屏幕上生成可视信息。

机车出入段管理是机务段的主要职能。主要负责安排机车的工作日程(机车计划),人员的工作日程(人员计划),机车入段登记(包括:入段时间、整备股道等信息),机车的检修安排,机车的合格证签发。

整个机车出入段管理主要包括以下几个部门:闸楼、运转中心(内勤、外勤)、检测中心(检测、检查)、临修。

根据各个部门的职能以及各部门之间的联系,对整个机车出入段管进行详细的需求分析。

1) 闸楼:机车入段首先要经过闸楼,在闸楼处要进行入段登记,包括:机车型号、入段时间、整备股道等信息。由闸楼将信息反馈到外勤,外勤填写整备日志。机车根据闸楼的分配入段。机车出段时最后经过闸楼,并填写出段登记。闸楼的机车入段信息管理模块主要包括:入段登记、出段登记、整备日志历史记录、机车统计、数据字典几项功能。

2) 外勤:主要根据闸楼录入的信息填写机车计划。管理机车入段信息。包括:外勤与检测中心的数据交换。外勤收到机车入段信息后给检测中心发送检测通知书;外勤与内勤的数据交换;检测中心签发合格证回票到外勤,外勤把机车计划输入司机机车整备日志中。外勤的机车入段信息管理模块主要包括:接收信息、发送信息,存储历史信息几项功能。

3) 检测中心:由检测部分接收外勤的反馈信息,然后派出专业的检修工作人员来进行检修,在检修的过程中记录到检查台帐中去,完成工作。之后由行检部分开始工作,派出行检的专业人员进行行检。行检完毕后进行数据反馈,如果反馈的数据,没

有记录机车的故障(即检测合格)将签发合格证,提票到外勤。如果检测不合格且是一些小问题就与临修点交换数据,由临修点进行修理。修理完后再与检测中心数据交换,由检测中心继续检测,直到机车合格为止。合格后签发合格证,提票到外勤。

4) 临修组:主要是与行检数据交换和接收机车小票,回票给行检。在得到行检部的数据后或是司机的小票后,对提出的故障进行修理,然后把反馈数据与行检部交换。

2 机车检修管理系统设计分析

机务段机车检测管理系统采用多文档用户界面,可以同时多个项目数据的输入和操作,主要有以下功能模块:

1) 用户登录模块:用户使用服务器端系统管理员为其建立的账户进入系统以进行各种操作。

2) 数据输入与维护模块:提供对 13 个机车检测项目的数据输入、维护和查找的用户界面,可以确保数据输入的完整、准确和有效。机车检测项目包括:电机绝缘 TY 检测记录、机车电器、电线路阻值检测、机车电机轴承检测、牵引电机、轴箱轴承检测、大电流导线 TZ 电阻检测记录、增压器转速检测记录、柴油机压缩压力检测、压力波检测、内燃电机线路对地绝缘检测、电力机车高压试验、电力机车主断路器在车检测、电力机车线路对地绝缘检测、电力机车受电弓仪器诊断。

3) 机车管理模块:包括机车型号管理和机车档案管理,用于建立机务段的机车档案。

4) 职工档案管理:建立机务段机车检测者的档案。

5) 查询打印模块:提供 13 个机车检测项目的原始数据报表、机车档案报表、职工档案报表、机车检测情况统计报表、机车检测提示报表共 17 个报表。可以按照常用条件和它们的组合筛选出所需数据。可在一定程度上定制报表的内容和格式。

6) 项目合格标准的设置及检测项目统计:根据机务段的情况,用户可以设置各个机车检测项目的合格标准,系统可以根据设置按照指定格式统计指定时间段、指定项目的检测次数和合格情况。

7) 到期、即将到期和过期检测项目提示:设置:设置各机车型号的各个项目的检测周期和提前提示天数,各机车的各个项目的检测状态和启动时间,系统自动提示的周期。提示:系统根据用户设置采用工具栏字幕、提示表单提示用户。报表:按照用户的设置和机车检测的记录,系统可以生成一个月以内的到期、即将到期和过期机车检测情况的报表。

8) 电线路检测仪的设置和数据导入: 可以设置各支路的合格标准, 可以上传设置到检测仪, 可以从检测仪导入检测数据到系统数据库。

“机车检测管理系统”的软件操作流程如图 1。

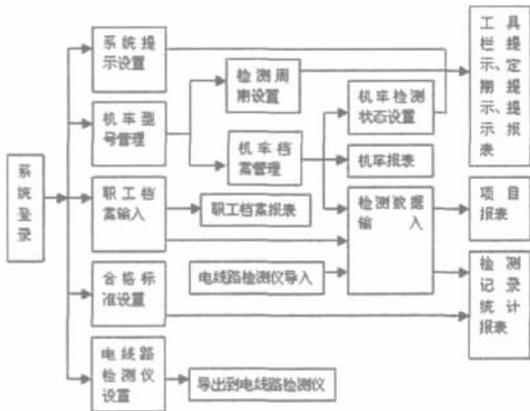


图 1 “机车检测管理系统”的软件操作流程

关于软件操作流程的几点说明:

1) 检测数据的录入: 为了进行机车检测数据的录入工作, 必须按照以下步骤进行: 建立机车型号, 建立机车档案, 建立职工档案, 输入数据。

2) 机车检测提示: 为使系统能够准确、及时地提示机车检测, 应进行如下设置: 建立机车型号, 设置每个型号的每个检测项目的检测周期和提前提示天数, 建立机车档案, 设置每个机车的每个检测项目的启动日期和检测状态, 设置系统提示方式和周期, 系统将按照设置采用工具栏字幕、表单、报表提示应检机车。

3 实例分析

通过对双表—车钩检修记录的制作来举例说明。在管理工具中选择 ODBC, 创建别名 (Alias Name), 这里是 TrainXB_NET。在数据库名称的选择是用 master, LoginID 为 sa, 在测试成功后完成。

可使用查询分析器创建数据表: 这样就创建了 TBLCGCHECK1 和 TBLCGCHECK2 两个表。连接 SQL 服务器, 打开 C++Builder 5.0 会出现登入画面。在 Object Inspector 中可以配置各个控件的属性, 然后进行数据库的连接, 将表中的数据读入可视控件中。因为要浏览显示数据表记录内容, 所以需要在窗体上放置两个 TDBGrid 类型的控件 DBGrid-Master, DBGridDetail 以显示 M/D 关系主从表的相应记录内容; 放置两个 TDataSource 类型的控件 DataSourceMaster, DataSourceDetail 以指明数据源。设置 DataSourceMaster 的 DataSet 属性为 TableMaster, DataSourceDetail 的 DataSet 属性为 TableDetail, 分别指向主从数据表。设置 DBGridMaster 的 DataSource 属性为 DataSourceMaster, DBGridDetail 的 DataSource 属性为 DataSourceDetail。

实现 Master/Detail 关系的关键在于从表关联控件 TableDetail 的 MasterSource 属性和 MasterFields 属性: 前者指向了一个 TDataSource 控件, 该控件 DataSet 属性应该指向 Master/Detail 关系的 Master 表; 后者则指定主表和从表的关联字段, 需要双击打开 ‘Field Link Designer’ 对话框进行设置工作, 选择从表和主表的相应关联字段添加到关联字段中即可。本例中以主表 (TBLCGCHECK1) 及从表 (TBLCGCHECK2) 的 CGID 字段为关联字段, 故设置 TableDetail 的 MasterSource

属性为 DataSourceMaster, 指向主表 TBLCGCHECK1, 部分程序如下:

```
//-----
#include <vcl.h>
#pragma hdrstop
#include cgcheckUnit1.h
//-----
#pragma package(smart_init)
#pragma resource *.dfm
TForm1 *Form1;
//-----
__fastcall TForm1::TForm1(TComponent* Owner)
: TForm(Owner)
{
    SYS_Item[0]= 车钩距轨面的高度中修为 835- 885mm ;
    SYS_Item[1]= 车钩全开位尺寸 220- 245mm ;
    SYS_Item[2]= 车钩锁闭位尺寸 110- 127mm ;
    SYS_Item[3]= 钩舌销与钩耳套孔的径向间隙: 短 1.0- 3.0mm
    长 3.0- 5.0mm ;
    SYS_Item[4]= 车钩锁闭位锁铁上移量尺寸为 5- 22mm ;
    SYS_Item[5]= 钩尾框扁销孔长度原形 106+3mm, 中修不大于
    115mm ;
    SYS_Item[6]= 钩体扁销孔尺寸原形 44+2*110, 中修不大于
    49*118mm ;
    SYS_Item [7]= 缓冲器自由高 二号缓冲器为 514mm MX 型
    缓冲器为 568mm ;
}
//-----
//这是可以实现自动生成项目表的数据//
void __fastcall TForm1::Button1Click(TObject *Sender) //插入表
1//
{
    QrNET1->Close();
    QrNET1->SQL->Clear();
    QrNET1->SQL->Add( begin transaction );
    QrNET1->ExecSQL();
    try
    {
        int cgid;
        bool find=false;
        QrNET1->Close();
        QrNET1->SQL->Clear();
        QrNET1->SQL->Add ( insert tblcgcheck1(F_DATE,
        F_ORDER,F_CHECKTIME,F_XTRAINNO,F_STRAINNO,F_CHECKER,
        F_BRIEF,F_TSZ) values (:F_DATE,:F_ORDER,:F_CHECKTIME,:
        F_XTRAINNO,:F_STRAINNO,:F_CHECKER,:F_BRIEF,:F_TSZ) );
        QrNET1->ParamByName ( F_DATE )->AsDateTime=
        DateTimePicker1->Date;
        QrNET1->ParamByName ( F_ORDER )->AsInteger=
        Edit5->Text.ToInt ();          QrNET1->ParamByName (
        F_CHECKTIME )->AsDateTime=StrToDateTime (MaskEditf_check-
        time->Text);
        QrNET1->ParamByName ( F_XTRAINNO )->AsInte-
        ger=Edit2->Text.ToInt();
        QrNET1->ParamByName ( F_STRAINNO )->AsInte-
        ger=Edit3->Text.ToInt();
        QrNET1->ParamByName ( F_CHECKER )->AsInte-
        ger=Edit4->Text.ToInt();
        QrNET1->ParamByName ( F_TSZ )->AsInteger=Ed-
        it5->Text.ToInt();
        QrNET1->ParamByName ( F_BRIEF )->AsString =
        Memo1->Text;
```

```

QrNET1- >ExecSQL();
//读出上面填入的数据//
QrNET1- >Close();
QrNET1- >SQL- >Clear();
QrNET1 ->SQL ->Add ( select f_cgjid from
tblcgccheck1 where F_DATE=:F_DATE and F_ORDER=:F_ORDER
and F_CHECKTIME = :F_CHECKTIME and F_XTRAINNO = :
F_XTRAINNO and F_STRAINNO=:F_STRAINNO and F_CHECK-
ER=:F_CHECKER and F_BRIEF=:F_BRIEF and F_TSZ=:F_TSZ or-
der by f_cgjid desc );
QrNET1- >ParamByName ( F_DATE )- >AsDateTime=
DateTimePicker1- >Date;
QrNET1- >ParamByName ( F_ORDER )- >AsInteger=
Edit5 - >Text.ToInt (); QrNET1 ->ParamByName (
F_CHECKTIME )- >AsDateTime =StrToDateTime (MaskEditf_check-
time- >Text);
QrNET1- >ParamByName ( F_XTRAINNO )- >AsInte-
ger=Edit2- >Text.ToInt();
QrNET1- >ParamByName ( F_STRAINNO )- >AsInte-
ger=Edit3- >Text.ToInt();
QrNET1- >ParamByName ( F_CHECKER )- >AsInte-
ger=Edit4- >Text.ToInt();
QrNET1- >ParamByName ( F_TSZ )- >AsInteger=Ed-
it5- >Text.ToInt();
QrNET1- >ParamByName ( F_BRIEF )- >AsString =
Memo1- >Text;

```

```

QrNET1- >Open();
if(QrNET1- >RecordCount!=0)
{ cgid=QrNET1- >FieldByName ( f_cgjid )- >AsInte-
ger; //读出 cgid//
find=true;}
QrNET1- >Close();

```

程序流程如下:



4 结束语

系统功能基本符合用户需求, 能够完成车辆基本信息的输入和统计, 以及各类相关报表的打印, 并提供部分系统维护功能, 使用户方便进行数据备份和恢复、数据删除。同时也考虑到了单位编码以减少数据冗余的问题, 对于数据的一致性的问题也通过程序进行了有效地解决。

参考文献

- [1] 杨彬, 薛松.C++Builder 开发应用指南[M].机械工业出版社, 1999
- [2] 杨宗志.C++Builder 数据库程序设计[M].清华大学出版社, 2001
- [3] 周之.SQL Server 2000 参考大全[M].清华大学出版社, 2003

[收稿日期: 2006.10.9]

风河公司取得 FSM 实验室 Hard Real- Time Linux 技术

风河系统公司日前宣布, 风河已取得由 FSM 实验室(Finite State Machine Labs, Inc, FSMLabs)开发的设备软件业界唯一的商用级“硬”实时(hard real-time) Linux 技术——RTLinux。风河公司此次获得了该项技术的全套知识产权, 包括专利权、著作权、商标注册和其他相关产品的所有权。按照购买协议中的有关规定, 风河公司还将取得今后在嵌入式应用领域的 RTLinux 用户 runtime 收益权。随着购买协议的生效, 风河公司将把 hard real-time 技术全面集成到其业界领先的基于 Linux 的设备软件平台中, 为风河 Linux 用户带来更好的性能体验。

通过此次投资购买, 风河公司加强了在 real-time 技术方面的专业实力, 进一步拓宽了公司的市场。风河公司面向多样化的垂直细分市场, 针对电子设备制造商的各种特定需求, 提供了一系列丰富的 real-time 特性。目前, 风河能够为设备软件行业用户提供三种 real-time 解决方案:

第一是 VxWorks, 这是一套完全的“确定过程式”(deterministic)实时操作系统(RTOS), 能够实现单个实时操作系统中多实时任务的并行运行;

第二是 RTLinux 技术, 它全面支持 hard real-time 应用, 同时能将标准 Linux 作为一个独立进程运行;

第三是软件 real-time 解决方案, 这是标准 Linux 操作系统的一个开源程序补丁。

风河公司战略开发副总裁 Vincent Rerolle 表示: “对于我们广大的 Linux 用户而言, hard real-time 技术将进一步拓宽他们在关键任务级别的设备软件开发项目, 增强 Linux 的生存能力。作为风河 hard real-time 技术基础的延续, RTLinux 将完美地融入风河设备软件平台产品系列, 进而提高风河 Linux 产品的价值。此次购买活动也充分体现了风河的 DSO 发展战略——为客户提供各种灵活多样的选择, 以及同时支持 Linux 和 VxWorks 平台的创新的产品和技术。”

RTLinux 技术可以确保 hard real-time 级别的响应时间, 能够支持较为广泛的系统应用, 包括单内核功能型手机、宽带 IP 通信、机器人技术和工业控制等。RTLinux 由 FSM 实验室开发, 该技术一经推出就被普遍认为是目前业界最佳、最成熟的可用 hard real-time Linux 解决方案之一。RTLinux 技术提供了一个与标准 Linux 内核共存的 hard real-time 运行模块, 较好地结合了高性能响应下的 hard real-time 特性和标准 Linux 操作系统下的文件管理、网络通信和其他操作系统功能等常规特性。FSM 实验室将根据风河的授权, 继续致力于 RTLinux 软件在企业应用领域的商用化。

随着 RTLinux 与风河全球领先的 Linux 设备软件平台的结合, 电子设备制造商将能够获得一个成熟、可靠的全新技术, 用于各种复杂多样的基于 Linux 的下一代应用的开发, 满足它们对 hard real-time 特性的需求。风河公司将推出集成 RTLinux 技术的 Wind River Platform for Consumer Devices、Linux Edition, 适用于各种高性能、hard real-time 的解决方案, 例如功能型手机、各种需要高容量流媒体的数字图像应用以及包括车辆碰撞系统在内的各类车载应用等。集成 RTLinux 技术的 Wind River Platform for Consumer Devices、Linux Edition 将把各种基于 Linux 的高速包交换设备软件应用提升到一个全新的高性能级别, 例如需要 hard real-time 特性的高速 IP 包交换路由系统等。

(由美国风河系统公司北京办事处供稿)