

文章编号: 1005-8451 (2011) 01-0046-03

# 基于组件技术的微机监测上位机软件系统

侯胡的, 李永燕, 杨云国

(上海铁大电信设备有限公司, 上海 200070)

**摘要:** 微机监测系统主要用于铁路信号设备运行状态和运行参数的监测。分析现有微机监测上位机软件系统的架构, 并对这些架构所存在的问题进行分析。采用软件组件技术重新设计后的微机监测软件系统一定程度上解决了这些问题, 同时还具有一些自己的特点。本文阐述重新设计后的微机监测软件系统架构、设计难点及其不利之处, 对这种架构的应用前景进行了展望。

**关键词:** 微机监测; 组件对象模型; 软件; 多线程

**中图分类号:** U284.7

**文献标识码:** A

## Upper Computer Software System of micro computer monitoring based on component technology

HOU Hu-di, LI Yong-yan, YAN Yun-guo

(Shanghai Tiedate Telecommunications Equipment Ltd., Shanghai 200070, China)

**Abstract:** Computer Monitoring System mainly used in monitoring the running status and arguments of railway signale equipments. This paper analyzed the architecture of existing Upper Computer Software System of Micro Computer Monitoring System and its problems. The component technology based on upper Computer Software System of computer monitoring resolved these problems in some degree and also have own characteristics. This paper explained the architecture of redesigned Micro Computer Monitoring Software System and its difficulties and some faults.

**Key words:** micro computer monitoring; component object model; software; multithread

微机监测系统, 主要用于集中监测铁路信号设备的运行状态和运行参数, 目前已经广泛应用于我国铁路行业。微机监测系统组成部分主要有数据采集系统和上位机处理软件系统。而现有的上位机软件系统主要由3部分组成: 站机、服务器和终端程序。其中站机位于车站机械室内, 通过与下位机软件的通信, 获取、汇总并分析数据采集机采集到的信息, 同时上传到服务器。服务器用于转发站机数据到各个终端, 同时存储历史数据。终端则提供查询数据并进行综合显示的功能。

### 1 微机监测软件系统架构

该系统2种软件架构见图1和图2。

图1表示每一个车站都设有一台监测主机, 上面装有站机程序。所有站机程序通过使用相同的网络应用层协议连接到电务处、电务段2个服务器上, 各家终端只连接自己的服务器。

图2表示每一个车站也都装有一台监测主机,

上面装有站机程序。电务处终端要查看的各个站机送来的信息, 都是经过电务段服务器转发给电务处服务器, 然后再由电务处服务器发送到各个终端; 反之, 电务处终端的查询命令也是先要经过电务处服务器发送到电务段服务器, 再由电务段服务器转发到各个车站。

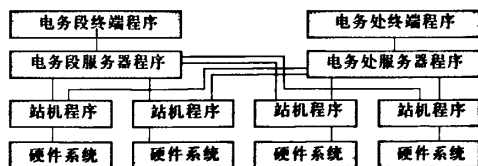


图1 微机监测软件架构模式1

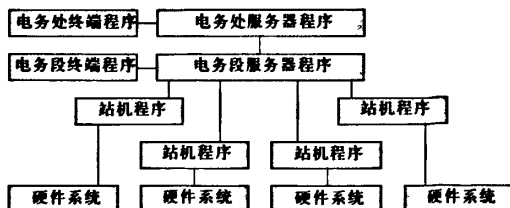


图2 微机监测软件架构模式2

上面2种方式, 都有以下一些缺点:

收稿日期: 2010-05-20

作者简介: 侯胡的, 工程师; 李永燕, 工程师。

(1) 系统运行效果很大程度上依赖于服务器的稳定性, 服务器出现问题时, 整个网络看不到数据。

(2) 站机层面,所有的设备驱动都在一个程序模块类,可重用性差。各个厂家的设备接口众多,缺乏统一的接口协议,修改程序影响范围大。

(3) 终端软件调看数据如日曲线的数据量少, 且功能单一。

(4) 与其他既有系统, 如电务信息管理系统 CSMIS 的交互不方便, 采用数据库交互的方式效率较低。

(5) 与信号专家诊断系统的交互不方便, 采用 socket 通信的方式较为麻烦。

## 2 面向对象的组件技术与问题

这里的组件技术是指微软公司提出的 COM/DCOM 技术。

COM (组件对象模型) 是一种以组件为架构单元的对象模型, 这种模型使系统中各个组件能够以二进制接口的方式进行交互。该模型具有 3 个特性: 语言无关性、进程透明性和可重用性。

DCOM (分布式组件对象模型) 是 COM 的无缝扩展, 它支持不同计算机上的组件对象与客户对象之间的相互通信。对于客户程序而言, 这些组件程序所在的位置是透明的, 编程人员不用考虑不同机器上程序之间的通信的具体细节, 而是关注于系统业务本身的实现上。

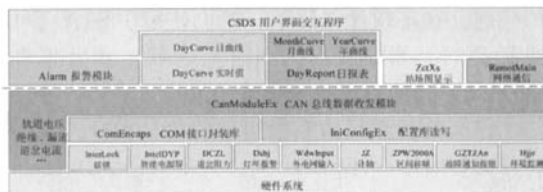
有了组件技术,就可以将一些可重用模块如微机联锁、智能电源屏等 I/O 驱动程序做成组件,然后在这些组件的基础上对现有微机监测软件系统进行重新架构。

### 3 基于 COM/DCOM 的微机监测软件架构

### 3.1 系统架构

采用组件技术后,微机监测软件系统是用用户界面交互程序和实时值、日报表等模块以及各个 I/O 驱动模块。在基于组件的微机监测软件架构里,服务器程序可以不用,以前的终端程序也就相当于一个用户界面程序去连接远端的硬件 I/O 驱动模块。

具体的模块划分和系统结构如图3。联锁、智能电源屏等 I/O 驱动模块位于软件系统的最低层。所有这些模块和 CAN 总线数据收发模块 Can-ModuleEx (通过 CAN 协议收集轨道电压、绝缘漏流、道岔电流等数据) 组成一个软件虚拟层, 这个虚拟软件层由许多组件组成。这些组件对外提供接口, 通过这些接口, 就可以获取缓存在这些组件中的数据。另外, 对数据的处理过程如报警、实时值、日报表等也可以编写成独立模块, 这样可以针对不同的应用场合使用不同的模块。



**图 3 微机监测系统模块划分和系统结构**

模块与模块之间则是采用队列数据结构进行数据的交互，交互的接口使用自定义结构体。比如变化开关量在idl文件中的定义如下：

```
typedef
[
    uuid(7E42E7AD-2C3F-48ae-B5EE-
E1979493E6A6),
    helpstring(" 变化开关量结构 "),
]
struct COM_DigitChangeQueueTemplate
{
    [helpstring(" 分机号 ")]BYTE FenJiNo;
    [helpstring(" 开关量序号 (最高位表示
变化状态) ")] WORD DigitNo;
}COM_DigitChangeQueueTemplate;
```

然后在程序中使用封装好的队列类取出数据：

```
COM_DigitStatusQueueTemplate *pDigit,  
OutQueue(DigitStatusQueueNo,pDigit);
```

对于在有多个微机监测厂商的环境里,如电务段里往往会有多个厂商的软件和设备,在新的软件架构下就不需要另外再设立各自的服务器,而是只要将用户界面主程序直接连接到相应的车站机中的相应的 I/O 模块即可。

重新架构后的微机监测系统架构如图4。

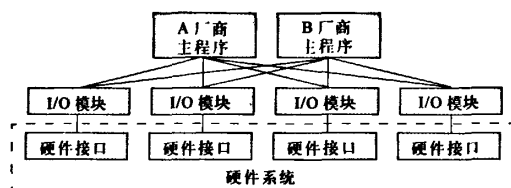


图4 新微机监测软件架构图

### 3.2 多线程与组件的关系

由于微机监测软件系统采用了MFC多线程方式实现，因此不可避免的牵涉到多线程调用组件的问题。在本软件系统中，我们使用了STA（单线程套间）启动线程，同时通过列集和散集将COM对象的指针传递到各个线程。

(1) 在主线程里获取组件的接口指针，比如将开关量组件指针pDigit通过列集封装成IStream接口指针pStream；

```
IStream *pStream = NULL;
```

```
static IDigitComp *pDigit;
```

```
HRESULT hr = ::CoMarshalInterThreadInterfaceInStream(
    __uuidof(IDigitComp), pDigit, &pStream);
```

(2) 通过散集在各功能模块线程中内部使用和释放该接口指针：

```
DWORD WINAPI DigitProcMain(LPVOID pWndView)
```

```
{
```

```
IStream *pStream = (IStream*) pWndView;
```

```
HRESULT hr = ::CoGetInterfaceAndReleaseStream(
    pStream, __uuidof(IDigitComp), (void**) &pDigit);
```

```
    其他逻辑处理过程 .....
}
```

### 4 微机监测采用组件技术的难点

铁路上对信号设备的监测具有采集点众多，数据量较大，各个模块之间关联紧密，实时性要求高等特点。

微机监测系统已经开发了10多年，铁路内有数千个微机监测系统使用的都是老的系统架构，在这基础上又有一系列应用层通信协议，而且每

个微机监测厂家又有自己的一些特色功能，这些都使得微机监测的功能变得比较复杂。

### 5 微机监测软件使用组件技术的优势与不利

采用组件技术的优势：

(1) 所有的驱动程序都被独立出来，如同在硬件层之上行成一个与硬件无关的虚拟层，在此基础上可以形成统一接口协议。

(2) 由于这些组件继承了自动化接口IDispatch，因此可以采用ASP或者JSP进行编程，通过Web界面与微机监测系统交互。

(3) 一定程度上增强程序的稳定性，因为当出现异常时，可以隔离到个别模块，其他模块仍然可以照常运行，从而增强软件系统的整体稳定性。

(4) 更容易添加扩展功能，如信号故障诊断分析等。

(5) 可以不用专门的终端软件，甚至服务器软件。

(6) 不再需要下载配置库，因此站机配置的改变会实时反映出来。

采用组件技术的不利之处：

(1) 软件开发和运行环境的配置较为复杂，软件调试不如以前方便。

(2) 网络中各个主机须有相同的用户名和密码，或者需要配置网络的工作域。

### 6 结束语

重新设计后的微机监测软件系统能够使各厂家统一接口协议，甚至制定类似于OPC (OLE Process Control) 的统一接口协议。由硬件厂家提供I/O驱动程序模块和接口。通过连接到不同的组件上，可以实现双网冗余系统。

参考文献：

- [1] 赵相荣，李 萍. TJWX-2000型信号微机监测系统[M]. 北京：中国铁道出版社，2001.
- [2] 潘爱民. COM原理与应用[M]. 北京：清华大学出版社，1999.
- [3] Christopher Tavares. 深入解析ATL[M]. 2版. 北京：电子工业出版社，2007.