

# 基于 B/S 结构的实时监测系统在 .net 平台下的设计与实现

Design and Implementation of Real-time Supervisory System  
Based on B/S Architecture Under .net Platform

许南山 刘长华 (北京化工大学信息科学与技术学院 100029)

刁春森 尹光昕 (山东淄博齐鲁石化股份有限公司 255411)

**摘要:**本文分析了实时监测在 B/S 结构下的优势,提出了在 .net 平台下如何设计出速度快、易扩展、操作简单、功能齐全、界面美观的解决方案,介绍了其基本结构和工作原理。并针对出现的刷屏、服务器负荷过重等种种问题给出了一种新的解决方法。

**关键词:**B/S 实时检测 .net

## 1 引言

在连续生产的企业(如化工、电力、钢铁)中,随着生产规模的扩大,生产设备的增加,很多企业都为自己配备了与 DCS、PLC 等设备相连的实时数据库,以方便查看一些重要设备的运行情况。但是这种实时数据库一般都安装在服务器上,如果为企业领导和主要的技术人员的电脑都安装一个客户端管理软件,部署、管理、维修乃至升级都很困难。网络技术的发展为实时监测技术的发展带来了新的契机,B/S 架构应运而生。

B/S 结构,即 Browser/Server(浏览器/服务器)结构。这种结构利用不断成熟和普及的浏览器技术实现原需要复杂专用软件才能实现的强大功能,节约了开发成本,是一种全新的软件系统构造技术。它具有以下优点:

- (1) 能非常容易地实现多用户同时进行监控。
- (2) 开发环境与应用环境分离,便于系统的管理与升级。
- (3) 应用环境为标准的浏览器,降低了进行安装、维护等费用。

## 2 系统设计

### 2.1 开发工具

前台开发工具采用最新的 Microsoft .net 中的 asp.net 这种 Web 应用开发技术,并使用 Visual Basic .net 这种支持对象编程的强类型语言编写控制逻辑。后台使用 SQL Server2000 来存储实时和历史数据以及设备的相关

参数。

### 2.2 基本构思

系统分为两个功能模块:数据采集和数据显示。如图 1 所示:

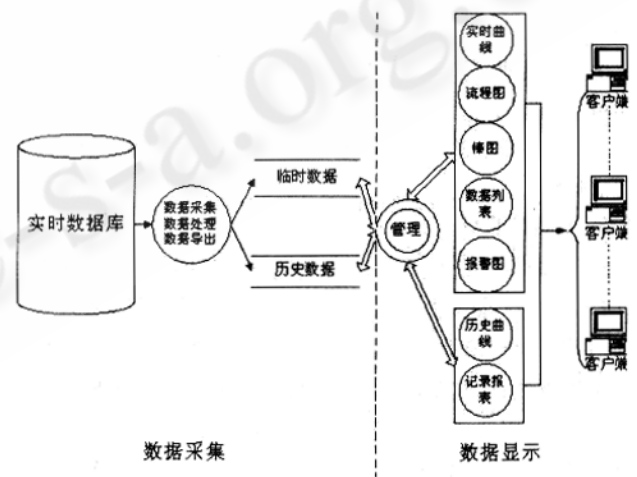


图 1 系统结构图

其中数据采集端负责从实时数据库中获取数据,并将数据插入到 SQL Server 数据库中。数据显示的目标是让用户以一种最简便、最直观的方式来查看自己所关心的设备的运行情况,因此需提供易于操作的图形界面,通过简洁的鼠标“点击”操作,即可实现对各类监测显示页面的切换和方便、快速的查询,并可方便地实现各种监测画面的组态。

数据采集模块安装在 SQL Server 服务器上。如果实

时数据库服务器和 SQL Server 服务器在同一台机器上,则可以很方便的从实时数据中获取数据;如果两个服务器分开,则可在 SQL Server 服务器上安装一个实时数据库的客户端软件,然后通过 ODBC 建一个实时数据库的数据源或通过连接字符串直接与实时数据库相连。根据数据显示的需要,采集到的数据在 SQL Server 数据库中以三种实体存储:一是实时数据表,只用来存储当前的实时数据;二是密集历史数据表,将采集到的数据都插入到此表;三是稀疏历史数据表,用来存储一定时间间隔的数据,便于显示。

数据显示模块采用流程图、棒图、实时曲线、历史曲线、数据列表、记录报表、报警浏览等方式将设备运行情况展现给用户。简单描述如下:

(1) 流程图:以分页方式显示带有实时工况参数的工艺图,工艺图上各设备数据实时刷新。

(2) 棒图:棒上可显示该位号仪表的量程,正常范围,报警范围。可以任意指定位号,构成新的仪表棒图画画。

(3) 实时曲线:以曲线方式跟踪显示生产的实时数据,并能指示曲线上某一点的值。

(4) 历史曲线:以曲线方式显示指定时间范围内的工况参数记录数据,允许选择时间范围,允许对曲线进行浏览扫描。

(5) 数据列表:以表格方式实时、批量显示各位号的当前测量值。

(6) 记录报表:以表格方式显示指定时间范围内的工况参数。

(7) 报警浏览:可随时显示关键数据点的运行状态(正常/报警)。

### 2.3 数据库设计

在流程工业中,一般一个厂由若干车间组成,车间下分工段,每个工段包含若干设备。对不同用户而言,所关心的不一定是所有的设备,比如厂级领导,可能只关心厂里的部分关键设备,而对车间级领导而言,所关心的只能是本车间内的设备,因此,提出了单元的概念。单元分为厂级单元、车间级单元和工段级单元。用户可以根据自己的需要创建自己的单元。实体定义如下:

车间信息表:用于存储车间的基本信息(车间号、车间名称)。

工段表:用于存储各车间内工段的基本信息(工段号,工段名称、车间号)。主键:工段号。外键:工段表\_车间信息表\_FK1(子:工段表.车间号,父:车间信息表.车间号)。

设备位号表:用于存储设备的属性数据(设备号、设备位号、设备名称、单位、量程上限、量程下限、所属工段)。主键:设备号,且为自动编号。外键:设备位号表\_工段表\_FK1

(子:设备位号表.工段号,父:工段表.工段号)。

单元表:用于存储不同级别、不同显示方式的单元(单元号、名称、位置、单元类型)。其中位置表示此单元的级别和所属,字符 F、C、G 分别代表厂级单元、车间级单元和工段级单元,譬如此单元属于车间号为“A”的车间,则位置命名为“CA”;单元类型表示此单元以什么方式显示(流程图、棒图、实时曲线等)。

动态项表:用于存储各单元中包含的设备信息(动态项号、单元号、设备号、xParam、yParam)其中,xParam 和 yParam 表示流程图单元中设备所在的位置坐标。主键:动态项号。外键:动态项表\_单元表\_FK1(子:动态项表.单元号,父:单元表.单元号),动态项表\_设备位号表\_FK2(子:动态项表.设备号,父:设备位号表.设备号)。

实时数据表:用于存储一条实时数据(S1、S2……Si……)。其中 Si 表第 i 个设备的实时值。

密集历史数据表:用于存储较密集时间间隔的各设备的数据(时间、S1、S2……Si……)。

稀疏历史数据表:用于存储较稀疏时间间隔的各设备的数据(时间、S1、S2……Si……)。

## 3 系统实现

### 3.1 数据采集模块

本模块类似于整个系统的一个线程,因此将更新操作放在一个 Timer 时钟里,其时间间隔根据不同企业的需求而定,每一个时间间隔执行一次采集和插入操作。实现代码如下:

```
Private Sub tmrRefresh_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles tmrRefresh.Tick
```

```
    If m = Interval Then
```

```
        GetRTValue()
```

```
        InsertRTTable()
```

```
        InsertDenseTable()
```

```
        m = 0
```

```
    If n = SparseInterval Then
```

```
        InsertSparseTable()
```

```
        n = 0
```

```
    End If
```

```
    m += 1
```

```
    n += 1
```

```
End Sub
```

其中 Interval 和 SparseInterval 是整型变量,Interval 表示刷新时间间隔,SparseInterval 表示插入到稀疏历史

数据表中的时间间隔。GetRTValue()、InsertRTTable()、InsertDenseTable()和 InsertSparseTable 均为子过程, GetRTValue()负责从实时数据库中获取数据存放到变量中, InsertRTTable()负责将数据插入到实时数据表中, InsertDenseTable()负责将数据插入到密集历史数据表中, InsertSparseTable()负责将数据插入到稀疏历史数据表中。

### 3.2 数据显示模块

#### 3.2.1 流程图显示

流程图有两部分组成:工艺流图和设备的表示。工艺流图可以采用 Microsoft Visio、Photoshop 等软件进行绘制,保存成\*.jpg文件。设备的表示可在 Asp.net 环境中使用 Label,显示的信息包括设备位号和实时值。

建立一个流程图页面 Flow.aspx,在其隐藏代码文件 Flow.aspx.vb 中的 Page\_Load 过程里面,首先定义一个 HtmlImage 控件,其"Src"属性为要显示的流程图工艺流程图的存储位置。然后从数据库的动态项表中找到该单元包含的设备,每个设备动态生成一个 Label,并根据存储在数据库中该设备的位置把它置于工艺流程图上。代码如下:

```
lblEquip = New Label
lblEquip.ID = "lbl" & eEquipInfo.设备号
lblEquip.Text = Trim(eEquipInfo.设备位号) & ":" & eEquipInfo.实时数据
```

```
lblEquip.Style("Position") = "absolute"
lblEquip.Style("left") = eEquipInfo.X & "px"
lblEquip.Style("top") = eEquipInfo.Y & "px"
其中 eEquipInfo 是设备类 clsEquipInfo.vb 的对象。
```

为保证显示的信息是最新的,因此页面必须以一定的时间间隔向服务器请求一次数据,在页面的源代码的<Head></Head>标签之间添加下面的语句:

```
<meta http-equiv="refresh" content="20">
```

这样本页面就会每隔 20 秒请求服务器一次。到此为止,看上去数据更新的问题就已解决,但是由于浏览器每次刷新的时候,都会重新生成一次本页面,所以就会出现“刷屏”的现象。为解决这个问题,使用了“隐藏页面”,即定义一个框架集,如下所示:

```
<frameset cols="0, *">
  <frame name="hFrame" src="隐藏页面">
  <frame name="dFrame" src="显示页面">
</frameset>
```

其工作原理如图 2 所示:

让隐藏页面每隔一定的时间请求一次数据库服务器,从数据库中获取设备的实时数据,同时将显示页面中设备的

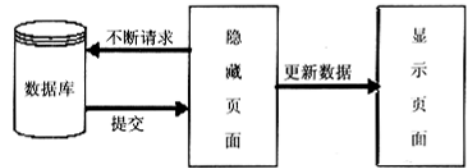


图 2 页面显示工作原理图

信息更新。更新代码如下:

```
Response.Write("< Script > parent.dFrame.
document.all("&" & "lbl" & eEquipInfo.设备号 & ").In-
nerText=" & Trim(eEquipInfo.设备位号) & ":" & eE-
quip Info.实时数据(drRTValue) & "</Script>")
```

#### 3.2.2 棒图的显示

棒图也由两部分组成:底图和液柱。棒图的底图不能由绘图软件绘制,因为每个设备量程范围不同,单位也不一样,由它们自由组态所构成的底图也就不一样,因此在请求棒图页面的时候,需调用设计好的绘图程序为所请求的单元绘制底图。这时,就会出现另一个问题:绘图程序是放在服务器端还是客户端。如果放在服务器端,则需调用.net所提供的诸如 Bitmap、Graphics 等绘图类进行绘制,并将生成的图形文件发送到客户端。从这里不难看出,如果大量用户同时访问服务器,就会造成服务器过重的负荷,因此,绘图程序要在客户端实现。客户端代码的实现,采用了 JavaScript 和 Vml(Vector Markup Language),其中用 Vml 的 Rect 对象绘制棒图的柱,用 Line 对象绘制棒图的刻度,用 Text-box 对象显示棒图诸如量程、刻度等信息。例如实现绘制棒图的柱的代码如下:

```
var newRect = document.createElement("<v:rect
'柱的属性'></v:rect>");
group1.insertBefore(newRect);
```

其中 group1 是一个 Vml 的 Group 容器,它让上面所提到的所有 Vml 元素使用共同的坐标系。

棒图的液柱也是一个 Rect 对象,在底图生成的时候同时创建,并置其 Height 属性为零。液柱高度的调整需根据设备实时值的变化不断调整,因此也要不断的访问服务器。为避免“刷屏”现象,也采用了“隐藏页面”技术。液柱高度调整的代码为客户端代码,放在显示页面的源文件中。代码如下:

```
function drawBar(barID,h, isAlert)
//barID 为液柱 ID,h 为液柱高度,isAlert 表示是否报警
{
  document.all(barID).style.height = h;
  if (isAlert == 1)
```

```
document.all(barID).fillcolor='red';  
else  
document.all(barID).fillcolor='green';
```

隐藏页面向服务器请求到实时数据后,换算成液柱的高度(h)和设备的状态(isAlert),调用上述代码,进而将设备的实时情况以棒图的形式展现给用户。调用代码如下:

```
Response.Write("<Script>parent.rightFrame.  
drawBar("& 液柱号 &","& 高度 &","& 设备状态 &");  
</Script>")
```

### 3.2.3 其他显示方式

实时曲线、数据列表和报警浏览的显示均采用“隐藏页面”技术,使显示页面既能将实时状态呈现给用户,又不出现“刷屏”现象。

实时曲线和历史曲线的生成和棒图类似,均采用 Javascript 和 Vml 来画底图和曲线,从而可以大大减小了服务器的压力。其中,曲线用 Vml 的 PolyLine 对象绘制。

数据列表和记录报表的显示用 Datagrid 控件,通过它自带的或者自定义的样式,简洁、美观的将批量设备的信息展现给用户。

报警浏览的显示相对简单,只需为每个设备动态生成一个 Label。如果判定设备处在报警状态,就置其颜色为红

色,否则为绿色。

## 4 结束语

B/S 架构下实时监测系统使管理者能够快捷、直观、形象的查看自己所关心设备的运行情况。管理者可以不受地点、时间的限制,只要计算机能够上网,就可以通过浏览器进行查看,为决策提供了快速、准确的依据。在 .net 平台实现的基于 B/S 架构的实时检测系统已成功地应用于多个化工企业。

### 参考文献

- 1 黄理、曹林有、张勇等,ASP.NET/XML 深入编程技术[M],希望电子出版社,2002。
- 2 (美) Francesco Balena. Programming Microsoft Visual Basic.NET[M],李珂等译,清华大学出版社,2003。
- 3 Tony Bain, Denise Gosnell. Visual Basic.NET and SQL Server 2000: Building an Effective Data Layer[M], Wrox Press Ltd, 2002。
- 4 修乾,基于 Web 的 .NET 应用程序开发[J],计算机工程,2003,29(21):185-187。