工控软件中表格的比较及实现

肖 杰 蔡晋辉 周泽魁 方 骏 浙江大学控制工程系(310027)

Abstract

Due to various purposes of tables in industrial control software,different kinds of realization methods using Visual Basic are presented in this paper.Then the Record Of Operation is illustrated in detail.The authors of this paper firmly believe that this paper will have reference meaning to industrial control software developers.

Keywords:Visual Basic,Industrial Control Software,Record Of Operation

摘 要

以 Visual Basic 为开发工具,针对工业控制上位机软件中各种表格的特点,提出了不同的实现方法。并以操作记录为例加以详细说明。该思想对工业控制软件的编写具有一定的借鉴意义。

关键词:Visual Basic,工业控制软件,操作记录

当前工业实时控制的上位机软件一般由分组显示的工艺流程图、实时数据集中显示表格、历史数据显示及查询表格、历史趋势显示、报警显示和确认处理表格、工艺参数的配置及系统参数配置等部分组成,其中很大的一部分是各种图表。本文经过比较分析控制软件中各种表格不同的功能特点,提出了不同的实现方案,并对一个实现操作记录保存及显示功能的程序具体加以说明。

1 各显示表格的特色比较及相应实现方法

1.1 实时数据表格

实时数据表格将分散在不同流程图上的同一类工艺数据加以集中显示,软件中应该注重数据的安全性、实时性和完整性。在程序实现上主要是合理挑选需要显示的工艺参数,将实时采集到的数据加以显示。既无需考虑磁盘文件的读写,也不需要进行上位机间数据的交换,编写比较简单。而且由于需要显示的参数是事先选定的,数量一定,所以可用行数一定的 MSFlexGrid 控件结合时钟控件实时采样刷新。

1.2 历史数据表格

历史数据表格是对保存于磁盘文件中的工艺数据加以显示的表格。为防止历史数据文件的意外损坏,各上位机一般都会独立将各自采集到的实时数据加以保存,以保证一定的冗余性。历史工艺数据往往以一定时间为单位存放成一个文件,最常用的是将一天中的数据存入一个文件,只需要用一个定长的数组进行记录。例如每分钟存一次采样到的工艺数据,先定义一个每次采样到的工艺数据组成的自定义结构 EveryTimeData,然后定义一个用于保存一天历史数据的数组 EveryDayData[1439]。

```
Type EveryTimeData
        dim a(1000) as integer
        dim b(200) as boolean
.....
End Type
Public EveryDayData(1439) as EveryTimeData
```

使用定长数组使得对历史数据的存放、读取和显示等操作的实现都十分方便。用 VB 中的 Put 语句来实现历史数据的保存:

```
Put #fileHand, Binary, EveryDayData
```

用 VB 中的 Get 语句来实现历史数据对某天磁盘文件的读取:

```
Get #fileHand, Binary, EveryDayData
```

显示只需要定义一个 1440 行的 MSFlexGrid 控件,然后按照数组的顺序逐点对单元格进行赋值。

1.3 报警记录显示表格

报警记录显示一方面具有极强的实时性要求:一旦发生错误,要求立即提醒操作人员;另一方面它又要求具有历史记录功能。一套工艺设备需要报警的设备或工艺参数的数量是一定的,从而决定了当前正在报警的项目的最大值是一定的;但如果以天为单位进行记录并显示时,一天中曾经出现过的报警项目却是不一定的:会有设备出现重复报警现象。在正常的生产情况下,每天的报警记录总条目会局限在一个有限的范围内。由于每条报警记录中不光包括报警开始时间和报警结束时间,往往还需要包含是否正在报警、是否重复记录等标志符号,所以将每条记录的所有相关参数及标志位定义为一个结构,然后用一个足够大的数组来记录一天可能的报警条目是一种合理可行的方法。其文件的存取方式和历史数据文件的存取相似。当然还需要对报警条目超过数组界限这种非常意外的现象加以处理,如对于有 400 个报警点的工艺流程,先定义可以保存 1000 个记录的动态数组,如果报警记录达到 1001 条时就用 ReDim Preserve 语句将数组维数的数目重新定义到 1500。表格依然使用 MSFlexGrid 控件,但需要根据实际使用的记录条目定义 MSFlexGrid 控件的 Rows 属性,并且使用 AddItem 方法往表格中实时加入最新出现的报警条目。报警属

于对实时采集的数据分析得到的信息，而且非常强调实时性，所以各上位机一般都独立对数据进行采样并存放，不需要在各上位机间进行数据传递。

1.4 操作记录表格

操作记录和其他表格数据有着以下非常不同的特点：①操作在每台上位机上单独进行，所以数据必须进行相互传递，以实现信息共享；②主要用于事故的分析和事故责任的查询，实时性要求并不高；③与报警记录需记录开始、结束时间相比，只需记录操作发生点的时间；④各操作事件相互独立不关联，所以不需要任何标志符号；⑤操作事件发生的无规则性，即发生事件的时间上具有偶然性和不确定性，所以每天的操作记录数量波动很大，如果生产分别采用上位机手动操作和自动操作，那就会出现十几倍甚至几十倍的差距。

根据操作记录以上的特点，使用集合技术来动态保存数据、WinSock 控件执行数据发送、MSFlexGrid 控件来显示一天的数据、以天为单位建立数据文件进行存盘无疑是十分合适的选择。下面就以该表格为例，加以比较详细的说明。

2 操作记录表格的具体实现方法

2.1 采用集合进行记录

集合是一种将一系列相关的项构成组的方法。它的优势在于可以非常方便地插入、删除和检索该集合中的项。可见它非常适用于操作记录这样条目数量不确定、需要不时进行增减的情况。但是集合对象中的项不能采用用户自定义类型，这就是在报警记录中采用了可以变大小的动态数组技术，而没有选用集合的主要原因。对于内存中的集合，需要定时进行数据存盘、数据发送给其他上位机和及时清空等工作，避免一台上位机内存中的记录集合积累过多，而其他上位机又长期得不到该机操作信息的问题。实现过程如下：

先定义一个全局的公共集合变量，作为操作记录的内存缓冲区：

```
Public NowMemoryOperationRecord As New Collection
```

然后再定义一个全局过程，用以实现向该集合加入最新操作记录。传入的参数中：OperatorName 代表操作员名称；OperationExplication 代表操作说明；AddExplication 是一可选项，代表操作补充说明，如计算机名称等。将该三个参数和当前操作时间以用逗号分割的字符串形式加入到 NowMemoryOperationRecord 集合。如果发现集合中的数量已经达到规定的上限，那就调用另一个过程 SaveOperationRecord 将集合中的项目存盘，并清空集合。程序如下：

```
Public Sub AddOperationRecord (OperatorName As String,OperationExplication As String, Optional AddExplication As String = " ")
Dim Addstr As String
.....
Addstr=CStr(Now) & "," & OperatorName & "," & OperationExplication & "," & AddExplication
NowMemoryOperationRecord.Add Addstr
    If NowOperationRecord.Count > UpperLimit Then
SaveOperationRecord (Date)
    End If
End Sub
```

对于需要记录的操作，只需要调用 AddOperationRecord 过程，传入必要的参数。SaveOperationRecord(FileName As Variant)是一个以存盘文件名作为传入参数的过程，它主要实现的功能是将 NowMemoryOperationRecord 集合中的所有操作记录字符串存储到指定存盘文件，然后用 Remove 语句将 NowMemoryOperationRecord 集合清空。

2.2 利用 WinSock 控件执行数据发送

利用 VB 中的 WinSock 控件可以与其它上位机建立连接，并通过用户数据报协议 (UDP) 或者传输控制协议(TCP) 进行数据交换。UDP 协议是一种无连接协议，两台上位机之间的数据传输类似于传递邮件：从一台计算机发送到另一台计算机，但是两者之间没有明确的连接。在应用程序某个任务完成时需要通知其它计算机，在发送的数据量不大的情况下，UDP 协议是比 TCP 协议更适宜的选择。要在两台上位机中间发送数据，需要完成以下的四步(在连接的双方)：①将 WinSock 控件的 Protocol 属性定义成 UDPProtocol，RemoteHost 属性设置为另一台计算机的名称或点格式下的 IP 地址字符串。②将 RemotePort 设置为与另一台计算机的 LocalPort 属性相同的端口。③调用 Bind 方法，指定使用的 LocalPort。④用 SendData、GetData 语句实现数据的传送与接收。

在使用 UDP 协议的时候，可以任意地改变 RemoteHost 和 RemotePort 属性，同时始终保持绑定在同一个 LocalPort 上。使用这一特性可以非常容易地实现对多台上位机的数据发送：

```
For I = 0 To ComputerNum-1
If NetIP(I) <> Winsock1.LocalIP Then
Winsock1.RemoteHost = NetIP(I)
Winsock1.sendData SendOperationRecord
End If
Next I
```

ComputerNum 代表联网的上位机数量，NetIP(ComputerNum-1) 是一个包含所有上位机 IP 地址的数组。SendOperationRecord 代表要传送的字符串，可以有两种不同的方法实现。

第一种方法：SendOperationRecord 是集合中所有条目的组合：

```
SendOperationRecord=" "
For I = 1 To NowOperationRecord.Count
    Recordstr = NowMemoryOperationRecord.Item(1)
    NowMemoryOperationRecord.Remove 1
    Print #FileNum, Recordstr
```

```
        If I = 1 Then
            SendOperationRecord = Recordstr & "(" & ComputerName & ")"
        Else
            SendOperationRecord =SendOperationRecord & vbCrLf & Recordstr & "(" &_ ComputerName & ")"
        End If
Next I
```

ComputerName 是本机的计算机名称，用于在其它上位机中显示该操作事件发生的地点。

接受方的 WinSock 控件当发生 DataArrival 事件时，只需要调用 GetData 方法接收数据，然后再用 Print 语句添加到相应的磁盘文件中去。

另一种似乎更加有效的方法是在每个操作事件发生、记录加入到集合时实现发送数据，即在 AddOperationRecord 过程中实现数据传递；为防止上位机系统时钟的不一致，操作时间应该以接受方的为准，所以 SendOperationRecord 中不包括时间信息：

```
SendOperationRecord =OperatorName & "," & OperationExplication & "," & AddExplication & "(" & ComputerName & ")"
```

接收方的 WinSock 控件当发生 DataArrival 事件时，只需要先调用 GetData 方法接收数据，再将接收到的字符串配上接收时刻的时间，最后将记录添加入 NowMemoryOperationRecord 集合。

两种方法各有优缺点：第一种方法将一定时间内收集的数据集中一次发送，优点是防止了网络或系统短时间的错误引起发送不成功造成数据遗失、接受方处理数据容易；缺点是显示其他上位机的操作记录不够及时、文件写操作比较频繁。第二种方法能及时发送操作记录、并减少了文件写操作次数；但是增加了 winsock 发送和接受事件的频率。具体采用哪种方法，可以根据实际需要决定。考虑到操作记录并不需要实时察看的特点，在实际工作中笔者采用了第一种方法。

2.3 使用 MSFlexGrid 控件来显示一天的数据

VB 中提供的 Microsoft FlexGrid (MSFlexGrid) 控件可以方便地显示网格数据，同时也提供了对网格数据进行各种操作的功能：高度灵活的网格排序、合并和格式设置等功能，网格中可以包含字符串和图片。操作记录显示表格可以利用该控件非常容易地实现。我们将窗体中的 MSFlexGrid 控件的列数定义为五列，表头分别显示："序号"、"操作时间"、"操作人员"、"操作说明"、"操作说明补充"等内容，由于每个单元格要显示的内容长度不一致，所以应该将 AllowUserResizing 属性设置成为 flexResizeColumns，使用户可以用鼠标自由调整各列的宽度。为了防止在宽度调整中，由于总宽度不足露出控件的背景，可以利用 ColAlignment 属性将前三列定义成单元格的内容居中对齐，而将最后一列定义为单元格的内容左对齐，并将宽度定义成一个足够大的数值。

要显示历史上某天的所有数据时，只需从数据文件中读数据到显示数据的集合 ShowOperationRecord 中，然后将集合 ShowOperationRecord 中的数据加以显示。

```
Do While Not EOF(FileHand)
    Line Input #FileHand, Recordstr
    ShowOperationRecord.Add Recordstr
Loop
```

集合中每一条项目正好是表格中一行，但如前所示，每条记录是以逗号分离的字符串来表示的，所以显示时还需要使用 Split 函数，将字符串转化成一个下标从零开始的一维数组。

当天操作记录的显示，由于关系到存放在 NowMemoryOperationRecord 集合中的数据，情况稍微麻烦一点。需要先从磁盘文件中读取当天已经存盘的记录到集合 ShowOperationRecord 中，然后将 NowMemoryOperationRecord 集合中的条目也加到 ShowOperationRecord 集合的末尾，最后加以显示。在显示表格的同时还需要不断查看是否有最新的本机操作记录或其他上位机传送来的操作记录，及时修改 MSFlexGrid 控件的长度，并加以填充显示。

考虑使用方便，在显示表格中，除需要提供指定日期数据文件显示功能外，一般还提供前一天、后一天、当天数据文件显示等快捷工具键和当前表格中操作人员、操作事件、操作时间等的自动查询与排序功能。

2.4 以天为单位建立数据文件

以天为单位存储数据，就是将一天的数据存放在一个文件中。操作记录的 NowMemoryOperationRecord 集合并不包含一天中所有的记录，它只是包含了最近的一个时间段内产生的，还没有存盘的数据。该时间段的长度的确定应该综合考虑以下因素：操作工作的频繁程度、显示其它上位机操作记录的实时性要求、内存的允许大小、防止由于断电等原因造成内存数据的遗失、最大程度减少对硬盘数据的读、写等。文件任何 I/O 操作之前先用 Input 或 Append 方式打开，或用 Print 语句往文件中加入最新的操作记录，也可用 Line Input 语句从文件中读取历史记录。用文本方式存放的一个优点是操作历史记录文件可以用 NOTEPAD.EXE、Word 等工具进行浏览和查询。读写文件采用 put、get 语句对数据数组进行操作。

3 结束语

使用本文方法实现的程序已经在天台石梁、宁波金狮等多家啤酒厂的发酵过程计算机控制系统中得到了应用，实践证明是可行的。

参考文献

1 王若菁.巧用 MicrosoftFlexGrid 控件.计算机应用，2000(8)

[收稿日期：2003.3.13]