

# 一种基于 LonWorks 的通用监视平台

## A General Supervising Platform Based on LonWorks Field Bus

荣治国,林述民(中南大学 信息科学与工程学院,湖南 长沙 410083)

RONG Zhi guo, LIN Shu min (Info. Sci. and Eng. Coll. of Center South Univ., Changsha HNX 410083, China)

**摘要:**分析了 LonWorks 系统体系结构和工作特点,提出了应用 LonWorks 的 DDE 通信方式建立通用的组态监视平台的开发方案,并对应用该方案开发出的 VSC (Visual Supervising Configuration) 系统平台的几个关键技术作了阐述。

**关键词:**现场总线; LonWorks; 动态数据链接; VB 语言

**Abstract:** By analysing the features and architecture of LonWorks systems, provides a new way of designing a general supervising system based on LonWorks' DDE communication, it also gives some technical details of a platform named VSC (Visual Supervising Configuration).

**Key words:** field bus; LonWorks; DDE; VB

中图分类号: TP273; TP311.52 文献标识码: A

### 1 前言

现场总线<sup>[1]</sup> (Field Bus) 是 80 年代末、90 年代初国际上发展形成的,用于过程自动化、制造自动化、楼宇自动化等领域的现场智能设备互连通信网络, LonWorks 现场总线系统是目前各种现场总线系统中比较有代表性的一种,它除拥有现场级控制系统所要求的所有特性外,还具有其他现场总线无法比拟的优点:它的通信协议采用 OSI 全部 7 层模型,控制方式采用命令/状态混合方式,另外系统还具备协议分析能力以及多种通信介质等等<sup>[2]</sup>。目前国内外的各个 LonWorks 系统集成开发商的软件开发重点多以开发下位机节点嵌入式系统和相对应的上位机的监控程序为主。下位机的软件开发应用 Neuron C 进行编程,上位机则以 VB、Delphi 等各种高级语言开发界面作为监控系统,网络定义、程序下装等工作由 Echelon 公司提供的 LonManager 系列工具来完成。

对于 LonWorks 系统的监视系统,虽然利用 VB、Delphi 等快速界面开发工具可以高效的开发出各类不

同应用领域的监视工具,但我们同时也注意到这些系统基本上只存在界面上的差异,并没有完全应用 LonWorks 系统利用网络变量通信的特点。对于同一 LonWorks 系统而言,增加或删除一个或多个结点是经常会发生的,这对于监视系统而言往往就要更改系统的源程序以适应用户的要求。针对这一问题,可以利用 LonWorks 系统利用网络变量进行编程的灵活性,以动态数据链接 DDE (Dynamic Data Exchange) 通信为基础,开发具备组态功能的 LonWorks 监视平台,使用户在无需编程并可以更改 LonWorks 系统网络拓扑结构的情况下以组态的方式自行组建自己的 LonWorks 网络监视系统。

### 2 LonWorks 系统的体系结构和工作特点

LonWorks 现场总线系统的体系结构比较清晰,各层职能分工明确,其基本系统结构如图 1 所示。其中下位机结点主要负责同现场设备进行连接,以便获取现场数据,它的另一个职能是对数据进行加工和处理并相应的发出各种不同的控制信息。结点的上一层是 PCNSS 网卡,它可以通过双绞线等各种通信介质和下位机结点组成各种形式的网络拓扑<sup>[3]</sup>。在 LonWorks 系统中网卡是一类特殊性质的结点,它的主要作用就是连接上位机和下位机,使上位机和下位机可以利用网络变量来交换数据,另外它还可以以消息 (Message) 的形式来完成上位机对下位机的信息传递,使上位机可以象普通结点一样完成整个系统基于事件驱动 (Event Drive) 的工作模式。

中间层的 DDE 数据交换服务器<sup>[4]</sup>的作用是应用 Lon API 把 PCNSS 网卡所采集的 LonWorks 网络上的各种物理数据转换成应用程序可以使用的各个变量,再通过 DDE 特有的共享内存方式分配给各个应用程序。其中 PCNSS 网卡在读取数据的时候和其他结点一样遵循 LonWorks 网络的 LonTalk 标准通信协议; DDE Server 则使用 Microsoft 在 OLE1.0 (Object Linked and Embedded) 中为多任务系统提出的 DDE 通信协

收稿日期:2002-04-23

作者简介:荣治国(1978—),男,湖南人,中南大学计算机应用专业硕士研究生,研究方向为计算机网络与通信,软件工程;林述民,利博赛社保技术有限公司,计算机应用专业硕士。

议。经过这2层转换,使 LonWorks 网络上的各种数据资源可以被上位机的各种应用程序所使用。当然,这2层转换对组态平台的用户来说都是透明的。

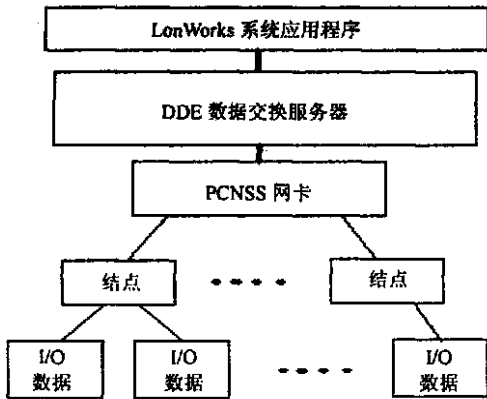


图1 LonWorks 的基本系统结构

对于监视系统<sup>[5]</sup>而言,其工作目的主要是从下位机获取数据,经过加工处理后以适当的形式展现给用户,使用户了解系统当前的工作状态,并对各种异常进行及时处理。因此,上位机的监视平台主要的任务就是获取下位机的网络变量,对数据进行处理之后再以组态的方式呈现给用户;下位机的任务则相对简单,下位机程序只需对各个需要实施监控的 I/O 点数据赋给网络变量,并相应的把网络变量设置成输出状态(其中 Neuron C 使用关键字 Out)就可以了。这样,监视系统就可以在连接 DDE Server 时通过先定位到结点,再定位到结点上的网络变量的方式来找到同 LonWorks 网络连接的各设备上的不同 I/O 数据点了。

### 3 通用 LonWorks 监视系统的设计与实现

通过以上分析我们把采用了通用监视系统的 LonWorks 系统的体系结构作以下简化描述:在这样的一个系统中 I/O 数据以网络变量的形式和上位机的应用程序相关联,上位机对数据的采集就是对网络变量的操作。对于有组态功能的监视系统而言,用户所做的工作就是构建自己的监视画面,然后在组态画面上的相应位置定义和现场设备的 I/O 点所对应的网络变量,再触发程序连接 DDE 数据服务器就可以组建自己的 LonWorks 监视系统了。

下面以应用该方案开发的 VSC(Visual Supervising Configuration)系统为例,简要介绍一下监视系统实现通用性的几个技术细节。VSC 系统平台结构如图 2 所示。

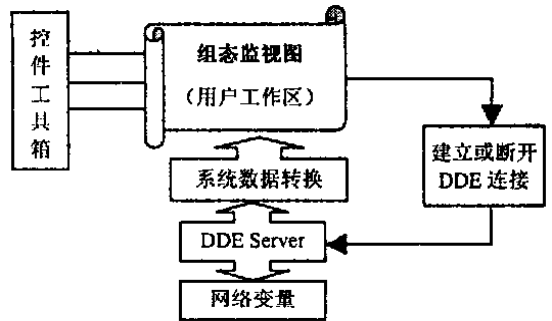


图2 VSC 系统工作模型

VSC 采用 VB6.0 作为开发语言,应用常用的 MDI (Multiple Document Interface) 多文档界面作为系统工作界面,采用多文档界面可以让用户把每一个文档界面作为一个系统来组建,从而帮助用户在构建大规模监视系统时可以把系统分割开进行分屏显示。系统的组态功能通过用户应用工具箱内的各种工具组件在工作区内作图完成。

在这里,我们把 LonWorks 网络上的各个网络变量以网络变量标签的形式给出,并允许用户以结点地址形式区分不同结点上的同名网络变量。表 1 呈现了这种一一对应的逻辑关系。

表 1 一号结点数据映射表

变量名称	类型	量程上限	量程下限	精度	地址	报警上限	报警下限	报警方式	严重报警上/严重报警下
0 压力	DB	0.1MPa	12	2547	0 抖动	2547			
1 水位	DB	0.0m	12	2547	0 抖动	2547			
2 网络变量 2	2547	0	8	2547	0 抖动	2547			
3 网络变量 3	2547	0	8	2547	0 抖动	2547			
4 网络变量 4	2547	0	8	2547	0 抖动	2547			
5 网络变量 5	2547	0	8	2547	0 抖动	2547			
6 网络变量 6	2547	0	8	2547	0 抖动	2547			
7 网络变量 7	2547	0	8	2547	0 抖动	2547			
8 网络变量 8	2547	0	8	2547	0 抖动	2547			
9 网络变量 9	2547	0	8	2547	0 抖动	2547			
10 网络变量 1	2547	0	8	2547	0 抖动	2547			
11 网络变量 1	2547	0	8	2547	0 抖动	2547			

在表格中,需要把网络变量的原始数据用如下公式转换成实际值:

$$y = \frac{x(y_2 - y_1)}{2^t} + y_1$$

其中 y:显示给用户的系统实际值;

y1:工程量程下限;

y2:工程量程上限;

t:数据采集精度(如 8 位、12 位等);

x:采集数据的数字量。

通过上表可以看出下位机结点和用户在上位机的组态监视图中各个数据点的对应关系。在组态图中,我们只要把这些网络变量标签定义在一些可以用不同

形式动态显示数据控件上就可以完成对组态图的数据定义了。在应用 VB 的开发过程中我们把这些用于动态显示数据的各种控件以 ActiveX 的形式保存在组态平台的工具箱内,常用的数据显示控件包括数值显示、棒状图、曲线、液位、阀门、温度计等等。此外,为了表示通用的系统流程,我们还另外加入了管道、锅炉、水箱、气缸等常用图形,以及可以从外部导入各种常用格式图形的 ActiveX 控件。这些不同类型的控件在组态平台中以分组的形式列出,组态系统可以对其进行放大、缩小或简单的透明处理,从而方便用户组态。

在 VB 中要实现上述方案还有以下几个问题:

(1) ActiveX 的动态加载与删除。在 VB 中动态加载控件有 2 种方案可以实现。

第一种方法是 VB6 中的标准做法,这种做法把 VB 系统中的所有组件作为 Control 对象来看待,对于组件的加载和删除通过分别调用 Control.Add() 和 Control.Remove() 2 个方法来实现。VSC 系统使用的就是这种用法,我们把系统工具箱内的各控件以控件数组的形式保存,在新建系统时为每一个对象初始化一个实例,用户每次添加控件时创建一个该实例的副本,并通过修改 Index 属性把该副本设置成拥有惟一标识的系统对象。以下代码是这种做法的一个举例:

```
Private Sub LoadControl ()
Set oControl = Controls.Add ( VB.CommandButton ,
MyButton )
    添加一个名为 MyButton 的命令按钮
oControl.Left = 100
oControl.Top = 100
oControl.Visible = True 使控件可见
End Sub
```

在这种模式下的事件编程关键是建立一个控件数组,再通过各控件数组的 Index 属性来决定对不同控件的响应。

第二种方法是利用 VB 中的类编程模型,使 VB-ControlExtender 对象与 Event Info 相结合,再利用系统提供事件陷阱捕捉来完成所增加控件的事件编程。例如下面的代码:

```
Dim WithEvents oControl As VBControlExtender
Private Sub LoadControl ()
Set oControl = Controls.Add
( VB.CommandButton , MyButton )
oControl.Visible = True
End Sub
Private Sub oControl-Object Event (Info As Event Info)
```

```
Select Case Info.Name
Case Click Click 事件
    可以添加处理 Click 事件代码
MsgBox 您按了 MyButton !
Case Else 其他事件
    Handle unknown events here.
End Select
End Sub
```

上面的代码中 Dim WithEvents oControl As VB-ControlExtender 是带事件声明,声明之后就可以在代码窗口的左上角的对象下拉框中发现该对象出现了,也就是说该对象有了事件或方法了,它的事件有 Drag-Drop, DragOver, Lost Focus, Got Focus, Object Event 和 Validate,其中 Object Event 是通用的事件捕捉,可以通过他结合 Select Case 语句来进行其它各种通用事件的捕捉和处理。

以上的 2 种方案都可以实现控件的动态加载和删除。相比之下,第二种方法的灵活性较高,但同时对各种事件处理也相对复杂。对于组态平台来说,系统提供给用户的工具箱内的各种控件已经固定,也就是对于各种未知控件的处理在系统中并无太大意义,而系统的工作重点应放在对控件的各种事件的处理上。另外,应用可代索引的 Index 属性还可以方便集合的操作。因此,我们选择第一种方案作为系统处理控件操作的解决方案。

(2) 各个控件在加载、删除和修改时须对它们所对应的网络变量的逻辑关系进行相应的修改和调整以保证系统正常运行并可以实现在线修改。为达到这一目标,我们通过应用 VB 中集合(Collection)的方法来实现这种动态关系的确定。即首先建立可以表达这种逻辑关系的一个类(Class),然后每次对控件进行操作时初始化一个类的对象,并把对控件的各种操作表示成这个对象的一种状态,再依据对象状态把它加入到集合或从集合中删除。这样在连接 DDE 数据源时所有的数据交换只在这个集合的范围内操作。这样做的另一个好处是在打开或保存系统组态图时,我们也只要对该集合进行操作就可以了。

#### 4 结 论

VSC 系统作为通用 LonWorks 监视系统的一个解决方案,具备使用方便、通用性强等特点。经实际检验,它在我院的过程控制实验室的压力、温度、液位等模拟实验系统中运行良好,基本上可以实现通用监视

# 用 Java 编写 HTML 文件分析程序

## Compiled Analytical Program of HTML File by Java

薛莹(潍坊学院 公共计算机教学部,山东 潍坊 261041)

XUE Ying(Department of Common Computer Teaching, Weifang University, Weifang SD 261041, China)

**摘要:**从实践的角度说明了 Java 语言中输入流类 StreamTokenizer 在编写 HTML 文件分析程序中的应用,并给出了利用 StreamTokenizer 流分析 Web 页面的例程。

**关键词:**超文件标示语言文件;Java 语言;输入流类 StreamTokenizer

**Abstract:** Explains the application of compiling analytical program of HTML file by Java in input stream class StreamTokenizer, and gives an example of analyzing the Web page layout by StreamTokenizer.

**Key words:** HTML file; Java; input stream class StreamTokenizer  
中图分类号: TP312/A 文献标识码: A

### 1 引言

Web 服务器的主要工作是对 HTML 文件中的各种标记(Tag)作出正确的分析,而编程语言的解释程序也是对源文件中的保留字进行分析后再做解释。在实际应用中,常常会需要对某一特定类型的文件进行关键字分析,例如,要将某个 HTML 文件下载并同时下载与之相关的 .gif、.class 等文件,这时就要对 HTML 文件中的标记进行分离,以便找出所需的文件名及目录。在 Java 之前,这类工作需要从文件中的每个字符进行逐个分析,从中找出所需要的内容,这样不仅编程量大,而且容易出错。而利用 Java 的输入流类 StreamTokenizer 进行 HTML 文件的分析,效果比较好。这里,要实现从已知的 Web 页面下载 HTML 文件,在对其进行分析后,下载该页面中所包含的 HTML 文件(在 Frame 中的)、图像文件和 Class(Java

收稿日期:2002-03-04

系统的开发目的。

### 参考文献:

- [1] 刘鑫. lonworks 现场总线网络产品的开发与应用[J]. 电子技术应用,1998(10):2-3.
- [2] 阳宪惠. 现场总线技术及应用[M]. 北京:清华大学出版社,1999.

Applet) 文件。

### 2 StreamTokenizer 类

StreamTokenizer(令牌化输入流)<sup>[1]</sup>的作用是将一个输入流变成令牌流。令牌流中的令牌实体有 3 类:单词(即多字符令牌)、单字符令牌和空白(包括 Java 和 C/C++ 中的说明语句)。

StreamTokenizer 类的构造器为: StreamTokenizer (InputStream in)。

该类有一些公有实例变量: ttype, sval 和 nval, 分别表示令牌类型、当前字符串值和当前数字值。如果我们需要取得令牌(即 HTML 中的标记)之间的字符时,应访问变量 sval。而读下一个令牌的方法是调用方法 nextToken()。nextToken() 的返回值是 int 型,共有 4 种可能的返回值:

StreamTokenizer. TT. NUMBER: 表示读到的令牌是数字,数字的值是 double 型,可以从实例变量 nval 中读取。

StreamTokenizer. TT. WORD: 表示读到的令牌是非数字的单词(其他字符也在其中),单词可以从实例变量 sval 中读取。

StreamTokenizer. TT. EOL: 表示读到的令牌是行结束符。

如果已读到流的尽头,则 nextToken() 返回 TT. EOF。

开始调用 nextToken() 之前,需要设置输入流的语法表,以便使分析器辨识不同的字符。方法 whitespaceChars(int low, int hi) 定义没有意义的字符范围。

出版社,1999.

- [3] 杨育红. LON 网络控制技术及应用[M]. 西安:西安电子科技大学出版社,1999.
- [4] 赖昊,薛君英,何熙文. 基于 lonworks 技术的现场控制系统[J]. 微处理机,1999(3):1.
- [5] 冯显勇,陈辉堂. 基于 lonworks 技术的远程监控系统的设计与实现[J]. 微型机与应用,2000(1):3.