

工业控制中实时数据传递新技术^①

陈松乔 张瑞瑛

(中南工业大学计算机系, 长沙 410083)

摘 要 采用面向对象机制, 论述了 Windows 环境下动态数据传递的基本原理, 介绍了主要动态数据交换事务以及在工业控制中动态数据传递的流程。并采用 Borland C++ 语言给出了工业控制中动态数据传递的基本实现。

关键词 Windows 环境 动态数据交换 工业控制

在工业控制中, 往往需要用到实时数据传递, 例如, 将通过数据采集系统采集的实时数据及时地传递给数学模型、专家系统或其它应用程序, 而且可能需要将一批数据同时传递给多个应用程序。在 DOS 环境下, 应用程序之间只能通过文件等数据存储方式来传递数据, 其缺点是速度慢、效率低, 而且不能同时进行传递和实时数据处理, 给用户造成了很大不便。作者根据 Windows 提供的动态数据交换机制 (DDE) 很好地解决了这个问题。DDE 可以协调 Windows 应用程序之间的数据交换, 向用户提供了一种更加集成化的 Windows 工作环境。为此, 本文将重点阐述 DDE 机制以及 DDE 在工业控制上的实现。

1 Windows 环境下动态数据传递的基本原理

1.1 动态数据交换 DDE

由于 Windows 系统是基于消息的结构, 所以传递消息是在应用程序之间自动传递数据的最佳方法。DDE 便是这样一种消息协议, 参与动态数据交换的两个应用程序负责一个 DDE 会话, 发送数据的应用程序是服务器应用程序, 接收数据的应用程序是客户应用程序

(以下简称服务器和客户)。DDE 用三级名字, 即服务名、主题和项名来标识数据单元。服务名是客户想要与服务器建立会话时, 服务器需要响应的字符串; 主题名用以识别逻辑数据关联的字符串; 项名则是用来标识数据的字符串。每个 DDE 会话由应用程序名和主题唯一确定, 只有这两者都分别相同时, 服务器才能与客户建立会话。会话开始后, 客户可以和服务器建立多条永久数据连接。数据连接是一种通信机制, 通过它, 服务器可以在数据被更改后通知客户。在数据连接或 DDE 会话终止前, 这个通知过程将一直工作。DDE 具有两种永久数据连接: 暖连接和热连接。在暖连接中, 服务器通知客户数据发生了改变, 但只有当客户请求时, 它才把新的数据传递给客户。而在热连接中, 一旦数据改变, 服务器就立即将更改后的数据传递给客户。

1.2 动态数据交换管理库 DDEML

DDEML 是应用程序用来共享数据的动态连接库。不必直接发送、传递和处理 DDE 消息, 应用程序可以用 DDEML 提供的函数管理 DDE 会话, 即两个应用程序之间的数据交换。DDE 应用程序不再使用原子和指示共享数据内存对象的指针, 而是创建和交换串句柄以及数据句柄。DDEML 为服务器应用程序提供登记服务名的机制, 这些名字可以向全系统中的

① 收稿日期: 1995- 04- 17; 修回日期: 1995- 09- 18

其它应用程序广播,以便这些应用程序与服务
器应用程序连接。

1.3 动态数据交换事务

服务器与客户建立了会话关系以后,就可
以相互发送事务,以便传递数据和做出相应
的事务处理。服务器和客户必须分别提供一
个回调函数来处理 DDE 事务。DDEML 具有16种
DDE 事务,本文介绍其中主要的6种事务:

- 建立会话事务 XTYP __CONNECT
- 会话开始事务 XTYP __ADVSTART
- 数据请求事务 XTYP __ADVREQ
- 数据传递事务 XTYP __ADVDATA
- 会话终止事务 XTYP __ADVSTOP
- 会话取消事务 XTYP __DISCONNECT

1.4 动态数据交换流程

由于在工业控制中多为实时数据传递,即
服务器定时采集数据,并即时通知客户,而
不是等待客户自己来索取数据(客户根本
不知道数据何时改变,应该何时去索取),以
免造成数据的丢失。因此,本文采用热数
据连接方法。具体过程如下:

1.4.1 准备工作

服务器与客户分别登记应用程序,分别
创建服务名、主题和项名的句柄,并获取
回调函数。服务器还应登记服务名。

1.4.2 会话连接

客户调用函数 DdeConnect,并向服务器
发送 XTYP __CONNECT 事务。服务器接收
到该事务后,分别匹配服务器和客户的服务
名句柄以及主题句柄,如若相同则连接成
功,否则失败。

1.4.3 数据传递

客户调用函数 DdeClientTransaction 并
指定 XTYP __ADVSTART 后,向服务器发
送该事务。服务器收到该事务后,若返回
非零值,则协商环建立成功。当服务器数
据改变后,服务器调用函数 DdePostAdvise,
并向自身发送 XTYP __ADVREQ 事务。服
务器每当接收到该事务时,就调用 DdeCreate
DataHandle 函数,并向客户发送 XTYP
__ADVDATA 事务,

以通知客户数据已发生了改变。此时,客
户接收到该事务并调用函数 DdeGetData,
获得更新后的数据。

1.4.4 终止取消会话

客户调用函数 DdeClientTransaction 并
指定 XTYP __ADVSTOP 后,向服务器发
送该事务。服务器收到该事务后,若返回
非零值,则取消协商环。客户调用函数
DdeDisconnect,并向服务器和客户发送
XTYP __DISCONNECT 事务。服务器与
客户释放所有句柄。至此,会话取消。

1.5 系统示意图

动态数据交换的系统示意图见图1。

图1 系统示意图

2 DDE 在工业控制中的实现

2.1 传统数据传递方法

传统数据交换中往往是用文件来传递
数据,即服务器将数据存入文件,而客户
再从此文件中取出数据。这样做速度慢、
效率低,无法保证实时数据的准确性。

Windows 中还提供了另一种数据交
换机制,即裁剪板交换机制。它用一个
公共缓冲区存放数据句柄,服务器通过
此句柄向客户传递数据。以上两种方法
都需要客户不时地主动请

求服务器发送数据, 而客户不知何时数据会发生改变, 何时应去请求新的数据, 这样就可能造成数据的损失和时间上的浪费。

2.2 DDE 的实现

本文用 Borland C++ 语言, 采用面向对象机制进行编程。以“株洲冶炼厂沸腾焙烧专家系统”为例, 此系统需要对40个工艺参数点进行实时监控和处理。这40个点分别为温度(16点)、风量(8点)、水位(8点)、料量(4点)和SO₂浓度(4点)。基本算法如下:

2.2.1 服务器程序(class TServer):

服务器主要负责采集数据, 处理建立会话

事务、会话开始事务、数据请求事务、会话终止事务和会话取消事务, 建立/取消会话和协商环, 以及将数据传递给客户。服务器设置时钟, 每一秒采集一次数据, 并将数据存放在数组 szItems 中, 其句柄为 hszItem。该程序详见表1。

2.2.2 客户程序(class TClient)

客户负责申请建立会话和协商环, 接收数据和处理会话取消事务。所接收的数据存放在数组 szData 中, 其句柄为 hData。该程序详见表2。

表1 服务器程序(class TServe)

```
// 初始化
void TServer ...SetupWindow()      { // 获取回调函数 Callback
    lpfn= MakeProcInstance(( FARPROC) TServer ...CallBack, GetApplication()-> hInstance);
    // 登记应用程序
    DdeInitalize(&idInst, (PENCALLBACK)lpfnCallBack, 0, 0L);
    // 创建各句柄, "Server"、"Topic"和"Item"分别为服务名、主题和项名
    hszService= DdeCreateStringHandle(idInst, "Server", CP __WINANSI);
    hszTopic= DdeCreateSttringHandle(idInst, "Topic", CP __WINANSI);
    hszItem= DdeCreateStringHandle(idInst, "Item", CP __WINANSI);
    // 登记服务名
    DdeNameService(idInst, hszService, NULL, DNS __REGISTER);
    // 设置数据采集间隔为一秒
    SetTimer(HWindow, 1000U, 1);
}
void TServer ...WMTimer(TMessage&)
{ 采集数据到 szItems;
    DdePostAdvise(idInst, hszTopic, hszItem);
}
TServer ... Tserver()
{ 释放所有句柄;
}
// 回调函数
HDDDEDATA FAR PASCAL TServer ...CallBack( WORD wType, WORD wFmt, HCONV hConv,
                                           HSZ hsz1, HSZ hsz2, HDDDEDATA hData, DWORD, DWORD)
{ switch( wType )
  { case XTYP __ADVREQ: // 创建并返回数据句柄
      return DdeCreateDataHandle(idInst, &szItems,
                                sizeof(szItems), hszItem, wFmt, 0);
    case XTYP __ADVSTART: return (HDDDEDATA) 1; // 返回1, 建立协商环
    case XTYP __ADVSTOP: return 0; // 返回0, 取消协商环
    case XTYP __CONNECT: 分别匹配服务器和客户的服务名句柄以及主题句柄;
        if 相同 then return (HDDDEDATA) 1;
        else return 0;
    case XTYP __DISCONNECT: 有关参数清零;
        break;
    default : break;
  }
return NULL;
}
```

表2 客户程序(class TClient)

```
// 初始化
TClient ~SetupWindow()
{ 准备工作; // 同服务器
  // 请求连接
  hConv= DdeConnect(idInst, hszService, hszTopic, NULL);
  // 请求建立协商环
  DdeClientTransaction(NULL, 0, hConv, hszItem, CF __TEXT, XTYP __ADVSTART, 1000, NULL);
}
TClient ~TClient()
{ DdeClientTransaction(NULL, 0, hConv, hszItem, CF __TEXT, XTYP __ADVSTOPT, 1000, NULL);
  释放所有句柄;
}
// 回调函数
HDEDDATA FAR PASCAL __export TClient ~CallBack( WORD wType, WORD, HCONV hConv,
        HSZ, HSZ, HDEDDATA hData, DWORD, DWORD)
{ switch(wType)
{ case XTYP __ADVDATA: // 获取数据
    DdeGetData(hData, &szData, sizeof(szData), 0);
    case XTYP __DISCONNECT: 有关参数清零;
        break;
default:          break;
}
return NULL;
}
```

3 结论

据传递。

DDE 方法不同于前两种传统数据传递方法, 它不需要客户主动向服务器请求数据, 一旦数据改变, 服务器会通知客户, 并将改变后的数据及时准确地传递给客户, 客户只需接收数据。这种方法非常适合工业控制过程中的数

参考文献

1 巴特尔著. 中国计算机用户. 1994. (12): 45- 48.
2 陈润泰等著. 检测技术与智能仪表. 1990: 223.

NEW TECHNOLOGY OF DATA EXCHANGE
IN INDUSTRY CONTROL

Chen Songqiao, Zhang Ruiying
Department of Computer Science,
Central South University of Technology, Changsha 410083

ABSTRACT Objecting Object(OO) mechanism was adopted to discuss the basic principle of Dynamic Data Exchange(DDE) under Windows environment. The DDE routines and the process of DDE in industry control was also introduced, and the basic implement of DDE in industry control with Borland C++ language was given.

Key words Windows environment dynamic data exchange industry control

(编辑 何学锋)