

OPC 在地方电厂管理信息系统中的应用^①

Application of OPC in LPMIS

戴碧豪 章坚民 (杭州电子科技大学 电子信息学院 浙江 杭州 310018)

摘要: OPC (OLE for Process Control) 规范是一个工业标准,它是基于微软平台的 PC 客户应用程序之间交换实时数据的一种方法,是为了解决不同厂商提供的过程控制设备和商业应用软件之间不能自由通信的问题。介绍了以 OLE/COM/DCOM 技术为基础的 OPC 技术,给出了地方电厂管理信息系统 OPC 服务器和客户端之间的数据通信方案,并用 Microsoft Visual C++ 提供的微软基础类库 (MFC) 实现了客户/服务器模型的编程。

关键字: OPC COM 接口 地方电厂管理信息系统

1 引言

随着计算机技术和控制技术的不断发展,分布式控制系统 (DCS) 以其集中管理、分散控制的优点越来越多地被应用于过程控制工业领域。传统方式开发的监控与数据采集系统 (SCADA),由于不具备开发性,工业控制软件缺乏统一的工业标准,致使不同厂家提供的软件之间进行通信存在巨大困难。另一方面,又由于各厂家提供的软件对控制系统硬件操作的设备驱动程序接口也各不相同,为了对不同的硬件设备都能进行管理,软件开发者必须针对不同的硬件开发相应的驱动程序。而且,当硬件设备升级或修改时,驱动程序也要做相应的修改。此外,在多个应用程序同时访问同一设备时,由于应用软件之间的不兼容,往往对系统的安全构成危害。(OLE For Process Control 过程控制对象链接与嵌入) OPC 规范就是在这样的情况下逐渐成为工业自动化软件的通信标准,它为软硬件之间的数据交换提供了规范的接口。

2 OPC 简介

从本质上讲,OPC 就是 COM (Component Object Model) 技术,像所有的 COM 实现原理一样,OPC 体系结构是客户端——服务器模式,如图 1 所示。从底层来看,把物理设备得到原始数据发送给过程控制系统,

或者把过程控制系统的数送到应用程序;从高层来看,OPC 服务器由三个对象组成:服务器对象、组对象和项对象。OPC 服务器对象,维护服务器自身的信息同时也作为 OPC 组对象的容器;OPC 组对象也维护自身的信息,同时提供一种机制和在逻辑上管理 OPC 项对象;OPC 项对象表示服务器中连接的数据源,OPC 服务器对象和 OPC 组对象是标准的 COM 对象,它们都具有接口和属性,但是 OPC 项对象不是标准的 COM 对象,它不提供接口。所有 COM 对象的访问必须通过接口。因此,OPC 客户端与 OPC 服务器也是通过接口进行连接和通信。在客户端只能看到服务器提供的接口,其中的对象是不可见的。特别是 OPC 项对象,作为一个对象,OPC 客户端是不可见的,同时 OPC 项对象还没有提供访问接口,所有访问项对象必须通过包含该项的 OPC 组对象进行。

3 OPC 客户端与服务器以及终端之间的通信

省级地方电厂管理信息系统是否能按预定目标取得成效,关键在于电厂采集终端能否全面、可行、可信、稳定地采集到规定的“有限、必须”的热电数据。而对那些具备一定自动化或信息系统的电厂,通过 OPC 接口与这些 DCS 或 SCADA 系统的通信,采集到相关数据,是一条经济、可行、可靠的优选途径。

^① 基金项目:2007 年浙江省经贸委、浙江省电力公司工程项目

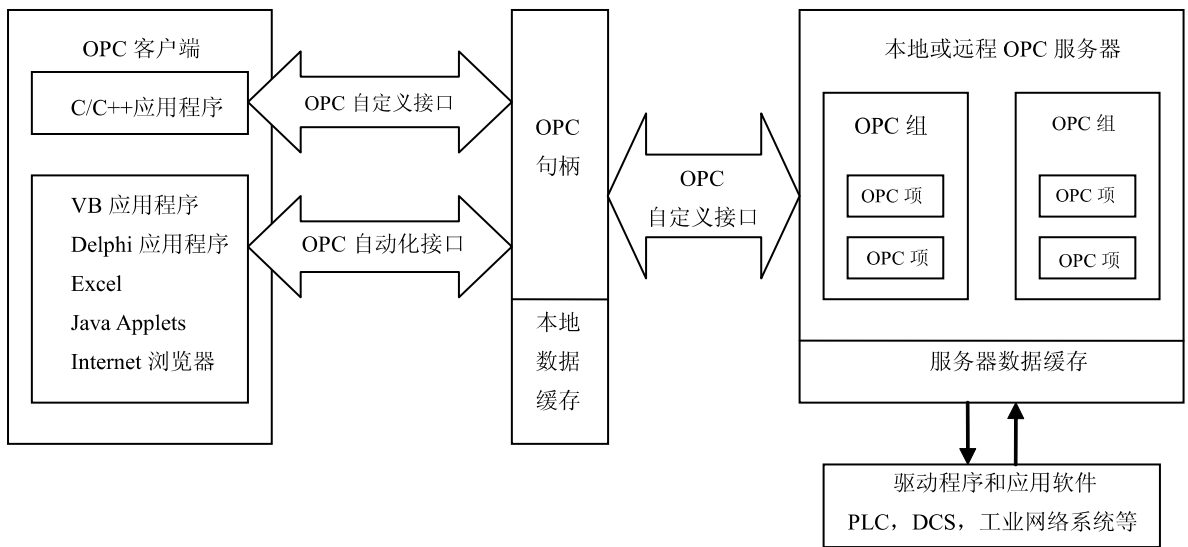


图 1 基于 OLE/(D)COM 的 OPC 体系结构

3.1 地方电厂管理信息系统 OPC 服务器的调研

OPC 数据服务器主要实现的是数据采集功能。OPC 的数据访问方法主要有同步访问、异步访问和订阅 3 种,其中异步访问和订阅这两种数据访问方法涉及到服务器端与客户端的双向通信。在 OPC 数据存取规范 1.0 版本中,这两种方法是用通报连接机制来实现的;在 OPC 数据存取规范 2.0 版本中,这两种方法是通过可连接对象机制来实现的。这两个机制都是 COM 中实现双向通信的机制。目前大多数正在开发的 OPC 服务器都遵循 2.0 版本。

OPC 服务器数据主要从地方电厂管理信息系统中的 DCS 系统中取得。然而,客户端要连接到服务器就必须要知道 OPC 服务器的计算机名,OPC 服务器程序的 ProgID,以及服务器中建立的项对象的名称,该项名称可以到服务器程序中的 CTag 类中的成员变量找到。

3.2 OPC 客户端代码的实现

Visual C++ 是一种功能强大的 Windows 应用程序可视化软件开发工具。VC 支持面向对象的设计方法,并可以使用功能强大的微软基础类库 MFC (Microsoft Foundation Class)。并且由于微软公司在市

场上的垄断地位,用 VC 开发出来的软件稳定性好、可移植性强,软件和硬件相互独立,而且 OPC 也是微软公司专门开发出来的通信规范。所以用 VC 去开发 OPC 程序是一个非常好的选择。主要的开发步骤如下:

3.2.1 包含 OPC 头文件。

开发 OPC 客户应用程,除了需要 OPC 接口外,还需要在程序中包含 OPC 标准库文件,可以从 OPC 基金会网站 (网址: www.opcfoundation.org) 下载这些文件:

```
#include "opcda_i.c"           OPC 数据存取接口
#include "opcda.h"             OPC 数据存取 2.0 头文件
#include "opccomn_i.c"         OPC 公共接口定义
#include "opccomn.h"           OPC 公共头文件
```

3.2.2 初始化 COM 环境。

```
CoInitializeEx( NULL, COINIT_MULTITHREADED );
```

3.2.3 连接服务器。

(1) 初始化服务器计算机名,以及 OPC 服务器程序的 ProgID,并初始化 OPC SERVER 接口。

```
m_arrMultiQI[ MQI_IOPCSEVER ]. piID =
&IID_IOPCServer;
```

```

m_arrMultiQI[ MQI_IOPCCOMMON ]. pIID =
    &IID_IOPCCommon;
m_arrMultiQI[ MQI_IOPCCONNPT ]. pIID =
    &IID_IConnectionPointContainer;
m_arrMultiQI[ MQI_IOPCITEMPROP ]. pIID =
    &IID_IOPCItemProperties;
m_arrMultiQI[ MQI_IOPCBROWSE ]. pIID =
    &IID_IOPCBrowseServerAddressSpace;
m_arrMultiQI[ MQI_IOPCPUBLIC ]. pIID =
    &IID_IOPCServerPublicGroups;
m_arrMultiQI[ MQI_IOPCPERSIST ]. pIID =
    &IID_IPersistFile;
    
```

(2) ProgID 转换成 clsid。

```

CLSIDFromProgID ( m_strProgID, &clsid );
    
```

(3) 调用 CoCreateInstanceEx() 函数。

```

hr = CoCreateInstanceEx (
    clsid, // CLSID
    NULL, // 不需要聚合或包含
    CLSCTX_SERVER, // 连接 local 和 remote
    &CoServerInfo, // 远程计算机名称
    sizeof( m_arrMultiQI ) / sizeof( MULTI_QI ), // 需要查询的 IIDS 数量
    m_arrMultiQI ); // 需要查询的指针数组
    
```

在调用此函数之前,必须要初始化以上接口。如果该函数调用成功,则说明客户端已经和服务器进行了连接。

3.2.4 客户端和服务端之间的数据通信

(1) 建立客户端的组对象。

组对象的逻辑模型如图 2 所示。调用 IOPCServer 接口的 AddGroup() 函数来把组对象加入到服务器对象当中。

```

hr = m_pIserver -> AddGroup (
    pszName, // 组名字
    pGroup -> IsActive(), // active state
    pGroup -> GetUpdateRate(), // 更新速率
    (OPCHANDLE) pGroup, // our handle to this group
    &Bias, // 时区
    &Deadband, // 死区百分比
    pGroup -> GetLanguageID(), // language ID
    &hServer, // 对于此组返回的服务器句柄
    &dwRevUpdateRate, // 返回的更新速率
    IID_IUnknown, // 请求 IUnknown 指针
    &pUnknown );
    
```

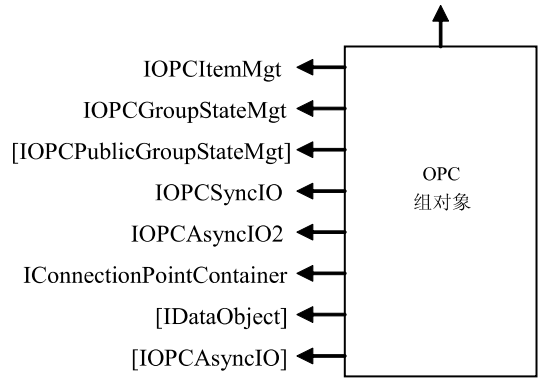


图 2 OPC 组对象的逻辑对象模型

(2) 建立客户端的项对象。

初始化服务器提供的 ITEM 名称,调用 IOPCItemMgt 接口的 AddItems() 函数来发送增加 ITEM 的请求,调用成功之后就可以进行客户端和服务端之间的读写操作了。

```

hr = m_pIItemMgt -> AddItems (
    dwCount, // Item 个数
    pItemArray, // item 定义的结构数组
    &pResults, // 结果数组
    &pErrors ); // 错误数组
    
```

(3) 异步刷新数据。

对于 OPC 客户端和服务端之间的数据刷新,我们采用异步读写机制,调用 IOPCAsyncIO2 接口的 Refresh2() 函数,然后该函数又会触发回调函数,即 IOPCDataCallback 接口的 STDMETHODIMP OnDataChange() 函数。调用成功之后,OPC 客户端就已经成功取到了 OPC 服务器上所需的数据。

```

hr = m_pIAsync2 -> Refresh2 (
    dwSource, // 源 (device or cache)
    dwClientTransID, // 客户产生的事务 ID,会在回调中收到
    &dwCancelTransID );
    
```

```
// 服务器产生的事务 ID,用于取消这次操作
STDMETHODIMP OnDataChange (
    DWORD dwTransID,
    OPCHANDLE hGroup,
    HRESULT hrMasterQuality,
    HRESULT hrMasterError,
    DWORD dwCount,
    OPCHANDLE * phClientItems,
    VARIANT * pvValues,
    WORD * pwQualities,
    FILETIME * pftTimeStamps,
    HRESULT * pErrors);
```

3.2.5 客户端与终端之间的数据通信

终端每隔 5 分钟发送一条报文请求数据,OPC 客户端收到报文之后,判断该报文自带的时间点在客户端上是否存在数据。如果存在,客户端会把数据通过串口发送到终端;如果数据不存在,则客户端会返回数据未准备好到终端,终端收到此报文后到下一分钟会再发送此时间点的数据请求,直到数据取到为止。终端把采集到的数据存在自己的内存里,并定时 5 分钟通过 GPRS 把报文中送到主站。地方电厂管理信息系统数据采集总的流程如图 3 所示。

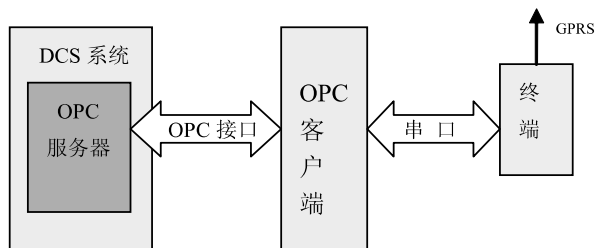


图 3 地方电厂管理信息系统数据采集流程

4 结束语

OPC 技术规范把硬件供应商和应用软件开发者分离开来,使得双方的工作效率都有了很大提高。软件开发商无需了解硬件的实质和操作过程,就能访问 OPC 数据服务器中的数据,尤其是开发商在已使用了组态软件进行实时监控的过程控制系统基础上,用 C++ 等高级语言开发系统时,大大简化了过去从设备传输数据的复杂过程。

本文的研究课题,是浙江省经贸委主持、浙江省电力公司实施的的对地方电厂的能量生产过程进行监视,对地方电厂的节能减排和以热定电等国家政策行为的信息直接数据采集。然而,本文提出的这种 OPC 数据采集方案既满足了地方电厂管理信息系统的数据需求又形成了工程项目经济需求的数据集成方案,在浙江省某个热电厂也投入了使用,这是非常可行而又值得借鉴的。

参考文献

- 1 The OPC Foundation. <http://www.opcfoundation.org>
- 2 OPC Foundation. OPC Data Access Custom Interface Specification Version 2.05, 2001.
- 3 侯俊杰. 深入浅出 MFC (第二版). 武汉: 华中科技大学出版社, 2001.
- 4 Dale Rogerson. COM 技术内幕. 北京: 清华大学出版社, 2002.
- 5 潘爱民. COM 原理与应用. 北京: 清华大学出版社, 2003.
- 6 马增良, 兰斌. OPC 数据访问服务器实现机制研究. 计算机工程与应用, 2003, 21: 65 - 67.