工控系统中间件研究兼论 OPC

申忠利[1],黄 仙[1],李 晟[2],陈永鑫[3]
(1. 华北电力大学 自动化系,北京 102206;
2. 中国水利电力物资有限公司,北京 100045;
3. 湖南省火电建设公司,湖南 株洲 410077)

摘要: 扼要分析了中间件原理,对 Windows 平台下通信技术的核心思想进行了集中解剖,得出两者共同的核心价值都在于分离。通过对 Windows 平台下通信技术的研究,指出了过程控制对象链接嵌入式技术 OPC(OLE for Process Control)的不足,并根据当前工控系统互联的现状,从中间件角度对工控系统互联的核心技术进行了探讨,对工控系统互联技术的发展方向提出了合理见解,认为开发真正开放的工控系统中间件规范是实现分布式工控系统的有效途径。

关键词: 中间件; OPC; DDE; COM; 通信
中图分类号: TP 273 **文献标识码**: B **文章编号**: 1006-6047(2006)01-0089-04

0 引言

在当前工业监控领域,可编程逻辑控制器(PLC)、分布式控制系统(DCS)、现场控制系统(FCS)等各有不同的通信体系和规范,其内部大都支持以太网互联,而互异系统的互联却不容易,因为没有公共一致的通信基础设施。在采用微软 Windows 平台的工控系统中,基于组件对象模型(COM)技术的过程控制对象链接嵌入式技术 OPC(OLE for Process Control)规范正获得普遍支持,给系统硬件和软件的集成提供了有效的解决方案。支持 OPC 系统,在系统边界对外提供一致的接口,使得互异的系统能够通过系统边界实现互联。

当前我国对 OPC 的研究,大多关注于 OPC 客户/服务器的开发,在实践中起到异种系统互联的作用,推动了我国工控技术的发展。本文从中间件[1]角度出发,通过抓住系统集成的关键——通信,探讨了工控系统体系结构,并通过研究现有 OPC 规范,提出开发真正开放的工控系统中间件[2]规范是实现分布式工控系统的有效途径的观点。

1 中间件技术原理

1.1 中间件基本概念

在网络环境下,将 IT 体系结构概括为资源层、中间件层和应用层 3 层。相应于传统的客户/服务器体系结构,中间件又称为应用服务器。一般认为:中间件是一种软件,它能使处于应用层的各软件之间实现跨网络的协同工作(即互操作),屏蔽了应用软件所涉及的"系统结构、操作系统、通信协议、数据库和其他应用服务"的差异。它类似于单机环境下操作系统与应用软件,是网络环境下分布式应用软件的基础设施,即网络操作系统,如图 1 所示。

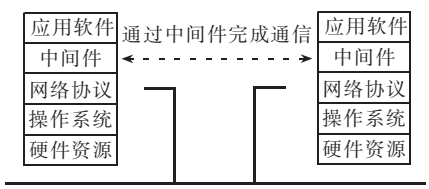

图 1 中间件定义
Fig.1 Definition of middleware

一个完善的中间件必须具有 2 个部分:执行环境软件和应用开发工具。前者屏蔽网络环境的差异,是应用软件实现可互操作性的关键。后者为应用软件的开发提供接口、接口定义语言及其编译器等[3]。目前,较为流行的中间件规范有:OMG 的 CORBA,Microsoft 公司的 DCOM/COM+,SUN 公司的 J2EE 等。这些中间件具有类似的编程模型与通信原理。

1.2 CORBA

OMG 即对象管理组织,从 1989 年建立至今,制订了 CORBA 规范体系,包括通用规范、实时规范、嵌入式规范等,并得到了大约 830 余家厂商和机构的支持。OMG 组织定义了 OMA 对象管理体系结构作为分布在异构环境中的对象之间交互的参考模型。

OMA 由 5 个部分组成:对象请求代理(ORB)、对象服务、公共设施、领域接口和应用接口。ORB 实现客户和服务对象之间的通信交互,而其他 4 个部分是架构在 ORB 之上适用于不同场合的部件。其中,ORB 负责对象在分布环境中透明地收发请求和响应,它是构建分布对象应用、在异构或同构环境下实现应用间互操作的基础。对象服务是为使用和实现对象而提供的基本服务集合。在构建任何分布应用时,经常会用到这些服务,而且这些服务独立于应

收稿日期:2005-09-20

用领域[4]。公共设施是为许多应用提供的共享服务器集合。

CORBA致力于提供一个标准的中间件平台,使来自各个不同软件生产厂商的应用程序可以互操作,因而具有一个完备的模型,但导致了复杂性,使得开发困难。不过存在很多通过OMG认证的开源CORBA实现,使得开发自有CORBA简单了很多。CORBA通信原理与DCOM类似,使CORBA组件借助IDL映射,可以与COM组件通信。正是基于这一点,可以创建COM/CORBA兼容的OPC。

1.3 IDA

目前,工业以太网是分布式自动化和现场总线网络通信的一个非常有前景的底层技术平台,但其中间层技术规范还远没形成。IDA(Interface for Distributed Automation)分布式自动化组织为分布式自动化系统定义和开发了基于以太网和IP协议的IDA分布式自动化接口技术标准,其体系结构采用基于IEC61499的分布式系统作为参考模型,提供模块化、分布式和可重用的自动化解决方案,具有较好的开放性和通用性。它规定了IDA系统的结构模型、通信模型、设备公共网页等,其核心技术包括RTPS实时预定/发布通信协议和XML扩展标识语言设备描述。

IDA中间件通过RTPS为应用程序提供了灵活的实时通信关系:发布/预定接收型、可靠发布/预定接收型和客户/服务器型3种通信关系。发布/预定接收型通信关系主要实现缓冲型一对多通信;客户/服务器型通信关系主要用于现场设备间由用户发起的、一对一的排队式、非周期通信。这些优点是DCOM/COM+所不具备的。此外,它还具有跟CORBA一样的优点,可工作于异构型环境,允许即插即用和热插拔,提供确定性通信,遵守实时系统规范,可实现带宽低消耗,支持多点通信等,但没有CORBA复杂。

IDA目前正处于发展阶段,比较成功的应用是Modbus转换器Modbus-IDA。

1.4 Globe

CORBA和DCOM等大多数基于对象的分布式系统主要设计在局域网内实现,不提供跨地域的分布透明性[5]。Globe(Global object-based environment)是基于对象的全局环境的简称,最大特点是支持Internet上用户和对象的广域分布式透明性。一般,工控系统并不需要跨地域分布式计算,但跨地域监测、跨地域信息集成是需要的,一个设计良好的规范应考虑广域环境。

2 Windows平台通信技术

2.1 DDE

Windows操作系统早在Windows 2.x就支持动态数据交换DDE(Dynamic Data Exchange)技术,以及兼容NetBIOS协议的网络环境下的DDE——NetDDE。由于DDE提供了Windows平台下动态数据交换的标准,因而受到了很多厂商的支持,许多软件都提供DDE接口,支持进一步开发,如InTouch,iFix,Ultramax,Matlab,Excel,Winword,组态王、力控等。在当时技术条件下,DDE应用程序成了Windows平台下系统集成的良好工具。

开发DDE应用程序的一种方式是直接利用DDE协议,采用消息机制进行通信,通过直接操纵全局原子表完成数据交换。这种方式的主要缺点是应用程序直接参与通信,程序间耦合性太大。为尽量隔离应用程序的通信过程,微软开发了DDEML.DLL动态数据交换管理库对通信过程进行了大量封装,简化了DDE客户和DDE服务器的开发。对于DDE而言,DDEML.DLL可说是中间件,它一定程度上体现了中间件的核心思想:资源与应用分离。

2.2 COM

随着工控技术的发展,生产规模的扩大,在经济全球化的推进下,现代化生产和管理对信息集成的要求越来越高,采用DDE技术进行数据交换也就显得越来越落后,它过于粗糙,效率低下,在网络环境下不能可靠工作,又由于没有开放的协议可供遵从,导致不同企业的工控系统不能互联,形成了"信息孤岛",阻碍了企业生产效率和管理水平的提高。

Microsoft在1993年正式推出COM,它是一种二进制和网络标准,是软件对象组件重用和通信的一种方式。COM的核心功能体现在二进制重用、进程内和进程外透明通信。支持COM的系统都必须实现COM库功能,包括组件的注册、创建、管理和使用等各个方面。在COM技术中,有2个核心概念:组件和接口。组件是具有一定逻辑功能的可执行二进制代码,接口是对其他软件和组件能使用的组件公用功能的定义,是组件与外界交互的通道。COM接口的实质是连接客户和组件的二进制内存结构,该结构布局的前3项必须是COM规定的3个函数的地址。COM正是通过接口实现它所追求的组件使用方式的一致性,对外界暴露接口函数,管理并维护接口的函数集。接口实现对组件各种技术细节的封装与隐藏,对外界提供透明的功能支持,对组件函数进行抽象与标准化,隐藏各种实现的特殊性。

2.3 COM通信原理

COM是另一种不同于DDE的Windows平台下进程间通信技术,它有2个核心概念:单元和封送。单元是微软COM技术特有的概念,是COM对象的生存环境,它可以是单线程单元或多线程单元。当COM客户和COM对象不在同一单元时,其通信过程将利用系统提供的封送机制完成。如果COM客户和COM对象不在同一单元,但在同一台计算机

上,则采用轻量级过程调用 LPC(Lightweight Procedure Calls)方式的封送机制,而如果 COM 客户和COM 对象处在不同计算机上,则采用远程过程调用RPC(Remote Procedure Calls)方式的封送机制。采用封送机制时,与 COM 客户直接通信的对象已不是 COM 服务器对象,而是 COM 服务器对象在 COM 客户地址空间的代理,由该代理通过通道对象与处在服务器地址空间的服务器存根对象通信,该存根对象才直接与服务器对象通信。该过程如图 2 所示。

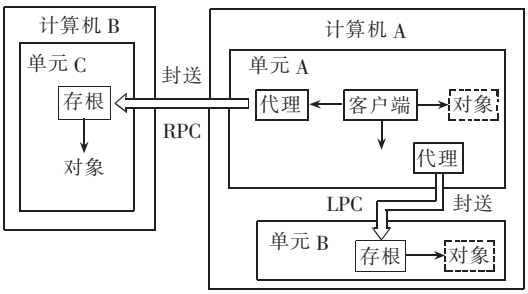

图 2 COM 通信原理

Fig.2 Communication principle of COM

LPC 又称本地过程调用。它由 Windows 操作系统在内核地址空间维护一个服务器通信端口、客户通信端口、命名/匿名连接端口以及用于内存映射的共享内存区。服务器创建连接端口以监听连接请求,客户发起连接请求。连接建立后,客户和服务器都得到一个通信句柄用于通信,数据通过由内核维护的客户和服务器共享内存进行交换,从而完成通信过程。不难看出,它与常规的套接字通信过程极为类似。但是,Win 32 API 并没有提供可用于LPC操作的函数,要在自己的应用程序中使用 LPC,就得调用 NTDLL.DLL 中的函数,但这些函数没有文档化,微软声明可能在不同版本操作系统上更改,使得用户在某个平台上直接开发的 LPC 程序在另一个版本的系统上就不能运行。

微软 RPC 是一种创建高效客户/服务器通信的技术,与DCE RPC 兼容,是 DCE RPC 的扩展。RPC 程序需要借助 MIDL 微软接口定义语言及其编译器实现标准代理/存根,通信过程需要 RPC运行库支持。RPC是微软进程间通信机制的一种,它可使用微软另外 3 种进程间通信机制:命名管道、NetBIOS 和套接字与远程系统通信。DCOM/COM+采用对象调用模型,默认通信方式是同步调用:调用对象的用户在接收到回应之前一直处于阻塞状态。为了提高同步调用的性能,微软引入了一种特殊的 COM 对象——可连接对象支持回调。更进一步的扩展是支持异步调用,也就是支持异步 COM。异步 COM 要求用户和对象处在运行状态,通信是暂时的。DCOM/COM+还通过提供 QC (Queued Component)队列组件支持持久性异步通信。

但是,OPCDA 的异步通信并没有利用异步COM 提高性能,而是采用连接点对象实现,增加了客户的负担,也降低了通信性能[6]。

3　DDE 与 COM 的核心区别

DDE 与 COM 的核心区别在于通信过程的规范性。

DDE 应用程序的一个特点就是应用程序可以而且也必须直接参与通信过程的管理,如会话管理和事务管理。此外,由于微软直接在函数级对通信协议进行描述,以及一些数据格式问题,使得 DDE 客户和服务器有较大的耦合性,一个 DDE 应用程序难以与多个厂家的 DDE 应用程序可靠通信。用户参与通信管理导致的另一个问题是通信性能的不可靠。由于可以手工控制通信资源,设计精良的程序在特定条件下可以减少资源消耗,获得较高的通信性能,但设计欠佳的应用程序除了本身不能良好运行外,还可能导致对方崩溃,这是很难接受的缺点。

COM 技术有着规范的通信过程,它有通信过程的标准实现,该标准实现由 COM 基础设施自动完成,大部分应用程序只需利用该标准实现就能获得很高的通信性能。在符合 COM 规范的条件下,COM 技术也支持通信过程的手工实现,以优化通信性能。COM 的一个核心思想是“接口与实现分离”[7]。COM 的通信过程也是如此。在 COM 技术中,对通信协议的描述全部是在接口层而不是函数层,COM 客户和 COM 服务器通过接口进行通信,即便是手工实现,也须遵循由接口所描述的通信规范。在“接口与实现分离”技术的支持下,COM 完全支持“资源与应用分离”的思想,彻底封装通信过程,产生全新的编程模型,保证 COM 通信的规范性,大大降低应用程序间的耦合性,显著提升了应用程序的通信性能。

出于商业利益的需要,COM 并没有将“资源与应用分离”的思想贯彻到底,使得 COM 严重依赖于 Windows 平台,依赖于 DCOM/COM+,从而使OPC 实际上成为微软定义的规范。当 OPC 基金会、OPC 产品提供商等宣扬 OPC 的优点时,那其实不过是任何一个公开的规范所共同具备的优点,而 OPC 平台依赖性的缺点却被有意忽略了。

3　Windows 平台工控系统中间件 OPC

OPC 是基于 COM 技术的,而 COM 组件须依赖其基础设施 —— DCOM/COM + 中间件才能运行。因此,严格地讲,OPC 并不属中间件,而属于基于中间件的应用软件。但是,对于工控系统互联而言,OPC 有着与中间件类似的功能,它隔离了上层应用软件与工控系统底层资源的差异,使支持 OPC的互异系统可以互联。因此,在此意义上可视OPC 为工控系统的中间件。其作用如图 3 所示。

由图 3 可见:正如前言所述,OPC 是通过互异系

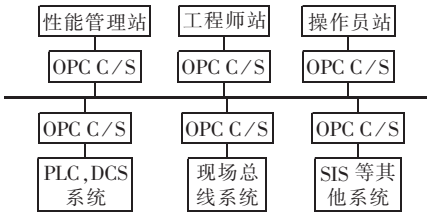

图3 基于OPC客户、服务器的互异系统互联

Fig.3 Interconnection of isomerous systems based on OPC client and server

统的边界互联的,是“系统与系统之间”的中间件,而不是一个系统内部的中间件。符合同一规范的系统内部,如图3中的现场总线部分,相互之间是完全互联的,没有通信障碍,3个现场总线设备通过共同的现场总线规范实现互联。OPC的优点在于它是一个开放性的规范,使得人们有共同遵守的前提。其次,OPC具有技术支撑,它建立在COM基础上,有微软和其他很多国际大公司的支持。如果要建立完全的分布式系统,让所有不同网络协议的系统和设备都能透明互联,则须彻底实现分布式中间件思想,建立开放的、与平台无关的工控系统中间件规范,则图3将无需OPC,如图4所示。

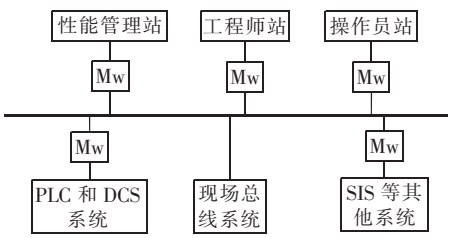

图4 基于中间件的工控系统体系结构

Fig.4 Architecture of industrial control system based on middleware

在图4中,Mw表示中间件。与图3的直观比较在于现场总线系统部分,单个设备支持中间件互联。没有标识的表示该设备不是或无法实现中间件。不同工控系统都实现相同规范的中间件,所有互联的系统、设备,对于应用而言就是资源,不但在物理上、而且在逻辑上都实现了分布式透明互联。这样的中间件可以有2种方式:一种是建立在现有各种工控体系的通信规范上,由这些厂家在现有规范上增加中间件层,它类似于OPC,但应用范围更广,如能在各种嵌入式平台,工业操作系统及PLC,DCS,FCS上实现;另外一种实现方式就是建立全新的工控系统中间件规范,如前文所述IDA,形成新的体系。2种方式相比,前一种受到的阻力应该少很多,即使得不到现有厂商的支持,但也不至于反对。

4 结论

4.1 OPC的不足

当前采用COM技术的OPC规范,因为没有充分利用COM的异步通信功能,既增加了软件开发的难度,又没有获得更高的性能。在OPC通信性能不能满足要求的情况下,应放弃COM标准技术,而采用平台提供的更底层的通信技术。

从中间件角度而言,OPC软件只能运行在DCOM / COM + 环境下,对其他中间件标准支持甚少,使得工控系统不能与采用其他中间件的系统通信,阻碍了工控系统与其他系统的融合。

从OPC规范的发展及其所采用的技术看,在“.NET是更好的COM”这一商业推销策略的驱使下,鉴于现在已经有.NET版本的OPC实现,如果OPC基金会再推出.NET版本的技术规范也不足为奇,这将使OPC的软件平台都必须是.NET平台,即便是嵌入式设备,如果想支持OPC,也不得不使用Windows CE.NET。

OPC的互联功能只停留在互异系统的边界,实际上它保护了每个系统的单独发展,从而阻止了工控系统的进一步开放,阻碍了工控技术的发展。

与新兴的IDA相比,它不能一网到底直达就地设备级,它没有专门的实时规范。

4.2 中间件解决方案

在目前还没有专门的工控系统中间件规范的情况下,可以采用OPC规范解决现存问题,满足现有需要。可以利用MIDL与CORBA IDL的映射实现CORBA平台上的OPC[8]。

更重要的是,应彻底支持“资源与应用分离”的思想,借鉴OPC规范体系①②,根据工控系统特点,开发兼容我国自有操作系统的开放的工控系统中间件规范,并在该工控系统中间件基础上建立新的工业数据交换技术规范,致力于将这些规范上升为国家标准,上升为国际标准。

参考文献:

[1] BRITTON C. IT体系结构与中间件——建设大型集成系统的策略[M]. 刁联旺,李彬,译. 北京:人民邮电出版社,2003.

[2] 阳宪惠. 开放工控系统的中间件——OPC技术[J]. 自动化博览,2002(2):6-8.
YANG Xian-hui. The middleware in open industrial control system—OPC technology[J]. **Automation Panorama**, 2002(2):6-8.

[3] 张云勇,张智江,刘锦德,等. 中间件技术原理与应用[M]. 北京:清华大学出版社,2004.

[4] 王千祥. 应用服务器原理与实现[M]. 北京:电子工业出版社,2003.

[5] van STEEM T. 分布式系统原理与范型[M]. 杨剑峰,常晓波,李敏,译. 北京:清华大学出版社,2004.

[6] BOX D,BROWN K,EWALD T,et al. Effective COM[M].

① OPC Data Access Custom Interface Standard v2.05A.

② OPC Data Access Custom Interface Specification 3.0.

(下转第95页 continued on page 95)

(上接第 92 页 continued from page 92)

余蒲澜,译. 北京:中国电力出版社,2003.

[7] BOX D. COM 本质论[M]. 潘爱民,译. 北京:中国电力出版社,2001.

[8] 任学军. 基于 CORBA 的 OPC 技术研究与应用[D]. 西安:西北大学,2003.
REN Xue-jun. Research and implementation of OPC technology based on CORBA[D]. Xi'an:Northwest University, 2003.

(责任编辑:李育燕)

作者简介:

申忠利(1976-),男,湖南邵东人,硕士研究生,主要从事计算机数据通信技术研究和智能算法研究(**E-mail**:www-lieon@126.com);

李 晟(1977-),男,江苏盐城人,部门副经理,主要从事电厂计算机控制系统调研工作;

陈永鑫(1976-),男,湖南株洲人,主管工程师,主要从事火电热工控制系统的调试工作。

Study on middleware and OPC in industrial control system

SHEN Zhong-li[1], HUANG Xian[1], LI Sheng[2], CHEN Yong-xin[3]

(1. Dept. of Automation, North China Electric Power University, Beijing 102206, China;
2. China National Water Resources & Electric Power Materials & Equipment Co., Ltd., Beijing 100045, China; 3. Hu'nan Province Thermal Power Construction Co., Ltd., Zhuzhou 410077, China)

Abstract: Principle of middleware is analyzed in brief, and the kernel idea of communication technology on Windows platform is studied together. Their common kernel value rests with separation. Based on the research of communication technology on Windows platform, insufficiencies of OPC(OLE for Process Control) are concluded. According to the status quo of interconnection of industrial control systems, its kernel technology is discussed from the view of middleware. The author provides views for its development trend and regards the establishment of real open middleware standard as an effective way for the realization of distributed industrial control system.

Key words: middleware; OPC; DDE; COM; communication