

OPC 跨平台通信的实现方法

吴鑫, 宫亮, 杨煜普

(上海交通大学自动化系, 上海 200240)

摘要: 随着工业自动化的发展, 工控领域中常常会遇到异构平台之间的用于过程控制的 OLE(OPC)通信问题。提出 3 种方法, 分别利用 EntireX DCOM, Java 以及 XML OPC, 在 Linux 下开发 OPC 客户端。使用 Linux 上的 OPC 客户端访问 Windows 平台上的 OPC 服务器, 从而实现 OPC 的跨平台通信。描述各种方法的关键技术并给出相关代码, 并对所有实现方法进行比较, 分析它们在工业应用中的前景。

关键词: 用于过程控制的 OLE; XML DA 协议; Python 语言; Java 技术; Linux 平台

Implementation Method of OPC Cross-platform Communication

WU Xin, GONG Liang, YANG Yu-pu

(Dept. of Automation, Shanghai Jiaotong University, Shanghai 200240)

【Abstract】 With the development of industrial automation, communication between OLE for Process Control(OPC) applications in heterogeneous platforms becomes a frequent issue in industrial process control area. This paper describes three methods, using EntireX DCOM/Java/XML OPC respectively to develop OPC clients under Linux. OPC client on Linux platform is able to communicate with OPC server on Windows platform, which is a successful implementation of OPC cross-platform communication. Key technologies in each method are discussed and corresponding code is also included with explicit explanation. A comparison is made between all implementations, based on which their further application in industrial environment is evaluated.

【Key words】 OLE for Process Control(OPC); XML DA protocol; Python; Java; Linux platform

1 概述

用于过程控制的 OLE(OLE for Process Control, OPC)是为了解决现场管理层和过程管理层之间的通信标准而设计的规范, 在工业自动化中有着非常广泛的应用。随着工业自动化的发展以及嵌入式系统在工业过程控制领域中的引入, 数据采集与监视控制(Supervisory Control And Data Acquisition, SCADA)系统常常需要从多种过程控制设备中获取数据, 其中很多设备都是基于 Windows 平台的, 但仍有一部分设备是基于非 Windows 平台(如 Linux)的。为了最大限度地利用现有资源为工业现场服务, 需要把所有的操作系统平台整合到同一个系统中。但 OPC 规范曾经都基于微软公司的分布式组件对象模型(Distributed Component Object Model, DCOM)技术, 也就意味着 OPC 只能应用于 Windows 平台。因此, 如何让异构平台(如 Windows 平台与 Linux 平台)中的应用程序协同工作, 即如何实现 OPC 的跨平台通信, 成为工控领域中的新课题。

本文基于 EntireX DCOM, Java 以及 XML OPC, 提出了 3 种方法, 很好地解决了 OPC 的跨平台通信问题。

2 EntireX DCOM 技术

DCOM 在 OPC 中的应用无疑是非常成功的, 唯一的缺陷是 DCOM 具有平台依赖性, 仅适用于 Windows 平台。因此, 如果能够把 DCOM 从 Windows 平台移植到其他平台上, 将会很好地解决 OPC 对 Windows 平台的依赖性, 从而可以直接利用 DCOM 的网络通信技术实现 OPC 跨平台通信。

事实上, 微软很早就开放了 DCOM 技术。1996 年 11 月, 微软将 DCOM 移交给开放组(Open Group), 公布了 DCOM 的

代码。这就使得 DCOM 技术在非 Windows 平台(如 Linux)上也得到了实现。这种 DCOM 的实现是以库文件的形式安装在目标平台上的, 利用这些 DCOM 库文件可以开发 OPC 应用程序^[1]。DCOM 在 Linux 下有多种实现版本, 其中 Software AG 公司与微软合作, 最早推出了 DCOM 在 Linux 下的实现版本, 即为现在的 EntireX DCOM 软件开发包。利用它, 运行在 Linux 平台上并启用了 DCOM 功能的客户端程序, 可以与 Windows 平台上的 DCOM 服务器通信。

EntireX DCOM 还提供了一个 Visual Studio AddIn 工具, 可以把 Windows 平台上已有的 OPC 客户端程序导入到 Linux 中^[2]。导入过程结束后, 在 Linux 下(Fedora Core 1 平台)运行 OPC 客户端程序, 可以正常访问 Windows 平台上的 OPC 服务器, 从而实现了 OPC 的跨平台通信。

3 Linux 下 OPC 客户端的开发

如果硬件厂商提供了基于 Windows 平台的 OPC 客户端源代码, 那么利用 EntireX DCOM 就可以很方便地将客户端移植到 Linux 平台上, 无需进行再次开发。但是很多情况下, 只能获得打包后的安装程序, 无法提取程序的源代码, 因此, 不能进行代码移植。这时可以选择直接在 Linux 下开发 OPC 客户端。

Java 是一种纯粹的面向对象的编程语言, 利用 Java 编写

基金项目: 国家“973”计划基金资助项目(2004CB720703)

作者简介: 吴鑫(1985-), 男, 硕士, 主研方向: CGSE-ES 网络分布式监控系统的分析与设计; 宫亮, 硕士; 杨煜普, 教授、博士生导师

收稿日期: 2009-01-15 **E-mail:** wuxin1@gmail.com

的程序可以不加修改地运行于多种平台，具有良好的可移植性与平台无关性。同时，DCOM 在 Linux 下的实现包含 Java 版。因此，此处选用 Java 语言在 Linux 下开发 OPC 客户端^[3]。

3.1 Utgard工程

Utgard 是一个开放源码的工程，提供了利用纯粹的 Java 语言所编写的 OPC 库文件。利用 Utgard 的库文件，可以开发符合 OPC 数据访问(Data Access, DA)2.0 规范的客户端。

Utgard 由 opencada-opc-dcom 和 opencada-opc-lib 2 个库组成。其中，opencada-opc-dcom 库使用 j-Interop 实现了 DCOM 的交互功能，j-Interop 是实现了 DCOM wire protocol (MSRPC)的一个开源类包。opencada-opc-lib 库则是简化了的 OPC 接口程序，由 4 个 Java 包(package)组成，每个包都提供了一些 Java 类供调用，如 ConnectionInformation, Server, BaseBrowser 以及 TreeBrowser 等，如图 1 所示。这些类是用 Java 开发 OPC 客户端的基础。

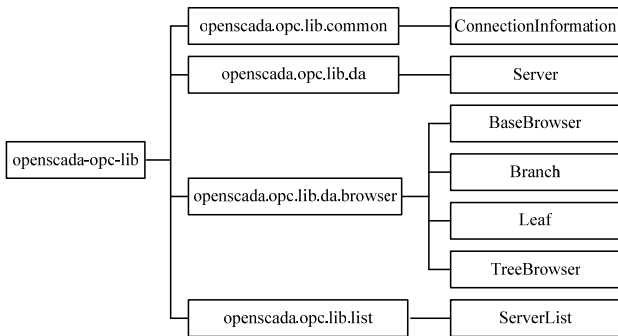


图 1 opencada-opc-lib 库结构

3.2 Linux下OPC DA客户端的创建

OPC DA 客户端程序中的 main 方法代码如下：

```

public static void main ( String[] args ) throws Throwable
2  { ConnectionInformation ci = new ConnectionInformation ();
    ci.setHost ( args[0] );
4  ci.setDomain ( args[1] );
    ci.setUser ( args[2] );
6  ci.setPassword ( args[3] );
    ci.setClsid ( args[4] );
8  Server server = new Server ( ci );
    try
10 { server.connect ();
        BaseBrowser flatBrowser = server.getFlatBrowser ();
12 if ( flatBrowser != null )
        {for (String item: server.getFlatBrowser ().browse ( "" ) )
14     {System.out.println ( item );
        }
16 }
        TreeBrowser treeBrowser = server.getTreeBrowser ();
18 if ( treeBrowser != null )
        { dumpTree ( treeBrowser.browse ( ), 0 );
20     }
        browseTree ( "", treeBrowser, new Branch ( ) );
22 }
    catch ( JIException e )
24 { e.printStackTrace ();
        System.out.println ( String.format ( "%08X: %s",
e.getErrorCode ( ),

```

```

26     server.getErrorMessage ( e.getErrorCode ( ) ) );
    }
28 }

```

该程序可以连接到指定的服务器，并打印出服务器中的数据元素。第 2 行调用 ConnectionInformation 类，创建连接的信息。第 3 行~第 7 行设定了具体的连接参数——主机、域名、用户名、登录密码及服务器的 ClsID。第 8 行调用 Server 类，建立了一个新服务器。第 10 行连接到该服务器。第 11 行~第 16 行调用 BaseBrowser 类对服务器进行基本浏览，并打印出所有组名。第 17 行~第 22 行调用 TreeBrowser 类对服务器进行树型浏览，并依次打印出每组内的所有元素。第 23 行~第 28 行可以在程序出错时打印出错误信息。

客户端程序运行于 Fedora Core 1 平台，此处利用 Softing 公司提供的范例服务器(Softing OPC Demo Server，安装于局域网内 Windows Server 2003 平台)来测试客户端程序。该服务器实施了 OPC Da 规范 1.0A, 2.05 等。其中，DA 服务器第 1 层命名空间包含 9 个节点，每个节点又都分别包含一些不同的元素。程序运行后，Linux 平台上的客户端程序与局域网内的 Windows 平台上的服务器建立连接，并准确地输出 DA 服务器的所有节点及其下属元素等信息，OPC 跨平台通信成功。

4 OPC XML-DA的引入

文中描述的 2 种方法都是采用基于 DCOM 的 OPC DA 协议进行 OPC 客户端的开发。利用 DCOM 可以很好地实现 OPC 的跨平台通信，但它难免会有一些历史局限性。随着工控技术的发展以及 OPC 新规范的发布，可以尝试利用新的 OPC 规范实现跨平台通信。

2003 年 7 月，OPC 基金会发布了 OPC XML-DA 规范。这一规范不再使用 DCOM 作为传输层协议，而改用 Web 服务。要实现 Web 服务，对象只需要支持 HTTP 和 XML。使用 Web 服务的各种应用程序通过 SOAP 进行通信。SOAP 与 XML 结合，把消息封装成适合通过 HTTP 协议传输的形式。HTTP 可以穿过防火墙，OPC XML 也可以在因特网中传输，这使得 OPC 的跨平台通信成为可能。

4.1 OPC XML-DA定义的服务类型

如表 1 所示，OPC XML-DA 规范定义了 8 种服务，每种服务都包括一个请求(Request)和一个响应(Response)。通过对这些服务的定义，提供了访问工业现场数据的标准接口^[4]。

表 1 OPC XML-DA 定义的服务类型

服务	描述
Browse	在服务器的命名空间里搜索所有可获取的项目(item)的名字(标记名)
GetProperties	返回一个或多个项目的相关信息
GetStatus	返回关于服务器、版本、当前模式、运行状况等信息
Read	返回一个或多个项目的值、品质和时间戳
Subscribe	指定一个客户希望持续更新的项目列表
SubscriptionCancel	删除在前一个 Subscribe 调用中指定的项目列表
SubscriptionPolledRefresh	返回自前一个 SubscriptionPolledRefresh 调用以来，在项目列表中数值发生变化的所有项目
Write	向一个或多个项目中写入新值

4.2 基于PyOPC的OPC XML-DA客户端的创建

目前存在很多工具包，可以用来快速开发 OPC XML-DA 应用程序。这些工具包可以生成底层协议与上层应用程序之间的接口，并且支持 XML 消息的创建以及 XML 消息到 HTTP 的映射。但是大多数开发工具包都无法免费获得，而且它们

中的大部分都是基于微软的 .Net 框架, 具有平台依赖性。

PyOPC 是用 Python 编写的 OPC XML-DA 框架, 可用来快速创建基于 OPC XML-DA 的客户端和服务端^[5]。PyOPC 及其所有下层技术都是开放源码的, 而且 Python 编程语言可运行于大部分的系统平台上(如 Windows, Linux 等), 因此, 可以利用 PyOPC 在 Linux 下开发 OPC 客户端程序。

4.2.1 PyOPC 的基本概念

任何一个基于 OPC XML-DA 的框架都需要支持一些特定的技术, 尤其是需要构造能够处理并发请求以及操作 SOAP 和 HTTP 协议的服务器。尽管 Python 本身有着大量的库文件, 但这些并不能实现上述技术, 因此 PyOPC 采用了另外 2 个用 Python 编写的框架。PyOPC 的这些基本技术如图 2 所示。

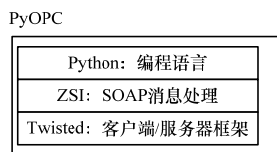


图 2 PyOPC 框架的基本技术

Python 是一种面向对象的解释性的程序设计语言, 具有脚本语言中最丰富和强大的类库。ZSI(Zolera Soap Infrastructure)是一个 SOAP 框架, 可以解析以及串行化 SOAP 消息。Twisted 是一个异步的客户端/服务器框架。

4.2.2 PyOPC 的类层次结构

PyOPC 引入了一些类的概念, 遍及整个 PyOPC 框架。这些类相互继承, 形成了一个类层次结构, 如图 3 所示。

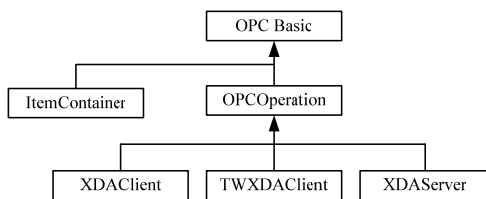


图 3 PyOPC 框架的类层次结构

OPC Basic 类实现了其他继承类所使用的基本功能; OPCOperation 类负责产生和解析所有 OPC XML-DA SOAP 消息; ItemContainer 类用来描述 OPC 对象; XDAClient 类可以创建简单的 OPC XML-DA 客户端。

4.2.3 基于 PyOPC XDAClient 类的 OPC XML-DA 客户端创建

基于 PyOPC XDAClient 模块的客户端代码如下:

```

from PyOPC. OPCContainers import *
2 from PyOPC. XDAClient import XDAClient
def print_options(ilist,options):
4 print ilist; print options; print address='http://advosol.us/
XMLDADemo/XML_Sim/OpcXmlDaServer.asmx'
6 xda = XDAClient (OPCServerAddress=address,
ReturnErrorText=True)
8 print_options(xda.GetStatus( ))
print_options(xda.Browse( ))
10 print_options(xda.Read([ItemContainer(ItemName='simple_
item', MaxAge=500)], LocaleID='en-us'))

```

该代码显示了基于 PyOPC XDAClient 的客户端首先获得

服务器的状态, 并浏览到根目录, 最后读取了一个条目。

在程序中, 第 1 行和第 2 行导入必需的 PyOPC 模块。第 3 行定义了一个简单的函数, 用来打印 ItemContainer 的对象列表(ilist)和全局选项(options)。第 6 行创建了一个客户端对象: 作为全局选项, 指定了服务器地址并通过“ReturnErrorText=True”指出客户端要求返回详细的错误描述信息。第 8 行~第 10 行显示了 3 种不同的 OPC 操作。这些操作的返回参数是 ItemContainer 的对象列表和全局选项(一个 Python dictionary 容器), 这些返回参数由函数 print_options 处理。

客户端程序运行于 Fedora Core 1 平台, 此处利用 Advosol 公司提供的网上 XML-DA 服务器来测试客户端程序。该服务器地址为“http://advosol.us/XMLDADemo/XML_Sim/OpcXmlDaServer.asmx”, 运行于 Windows Server 2003 平台。程序运行后, 客户端正确显示了服务器的状态、服务器 ItemContainer 中的所有元素以及指定条目的信息, 跨平台通信成功。

5 比较评估

尽管 OPC 基金会在 DCOM 之后发布了很多新的 OPC 协议, 但目前工业过程控制领域中应用最广泛的仍然是基于 DCOM 的 OPC 产品。同时, DCOM 技术也不再仅仅局限于 Windows 平台, 在非 Windows 平台(如 Linux)上也得到了实现。这就使得传统的基于 DCOM 的 OPC 协议不再具有平台依赖性, 可以用于实现 OPC 的跨平台通信。

XML-DA 作为新兴的 OPC 协议, 在通用性与市场占有率上无法与 DCOM 相抗衡, 现在仍算不上主流协议。但随着 SCADA 系统对 OPC 技术的跨平台性能和 Internet 特性要求越来越高, XML-DA 固有的基于 Web 服务的优势将逐渐得到体现。越来越多的设备供应商和软件开发商开始提供对 XML-DA 的支持, 如西门子公司的 WinCC 软件已经集成了 OPC XML-DA 服务器, Softing 等厂商也提供了各种开发 OPC XML-DA 应用程序的工具包。可以预见, 在不久的将来, OPC XML-DA 必将得到更为广泛的应用。

6 结束语

本文描述的方法实现了 OPC 通信中多平台的融合, 不同平台上的 OPC 应用程序也能够协同工作, 使得 SCADA 系统中 Linux 平台或嵌入式系统的引入成为可能。未来将研究如何在 Linux 下开发 OPC 服务器。

参考文献

- [1] Beharrell M, Barillère R. OPC Evolution Toward UNIX[C]//Proc. of the 10th International Conference on Accelerator & Large Expt. Physics Control Systems. Geneva, Switzerland: [s. n.], 2005.
- [2] Software A G. DCOM on Non Microsoft Platforms[EB/OL]. (2001-01-24). <http://www.opcfoundation.org/>.
- [3] 焦青松, 吴应良. OPC 客户端的 Java 实现方法[J]. 微计算机信息, 2006, 22(10): 60-62.
- [4] 彭 瑾, 王锦标. 新的 OPC 规范——OPC XML-DA[J]. 冶金自动化, 2004, 28(1): 6-9.
- [5] Himmelbauer H. A Python Framework for the OPC XML-DA 1.0 Standard[EB/OL]. (2006-11-07). <http://pyopc.sourceforge.net/>.

编辑 顾逸斐