By Mike Santori,
Director Industrial Automation Marketing,
National Instruments.

OPC : OLE for Process Control

Industrial automation users are entering a new era in which seamlessly integrated, multivendor control systems will become a reality, and proprietary software and hardware interfaces will become a thing of the past. Dramatic advances in software technology are promising to transform the world of integration headaches imposed by today's proprietary systems into compatible systems and devices. A promising new software standard in industrial automation is OLE for Process Control, or OPC. The goal of the OPC foundation is to establish an open connectivity standard based upon Microsoft's OLE/COM. When manufacturers agree on OLE standards it will be effective as part of an integration framework because OPC provides plug-and-play communication and interoperability between field devices, control systems, and enterprise-wide business applications. The rise of OPC is an example of the effect that general-purpose computer technology is having on the automation industry. As more systems take advantage of the open software and hardware architecture of the PC, manufacturing systems will become increasingly more open, flexible, and lower cost. New technologies based on standards now give us a great opportunity to create an integrated software environment that permits development of reusable, plug-and-play objects that are interoperable across corporate-wide manufacturing and business applications.

Industrial automation users are entering a new era in which seamlessly integrated, multivendor control systems will become a reality, and proprietary software and hardware interfaces will become a thing of the past. Sounds like a dream, doesn't it? Over the last two decades, the industrial automation market has seen a proliferation of proprietary interface standards. Literally hundreds of different and incompatible proprietary interface standards developed by suppliers are required to communicate with today's automation systems and devices. It is only by walking through this maze of interfaces that today's point-based solutions, such as field devices, control systems, and business systems, can be turned into an integrated plant-wide or mill-wide manufacturing system. But dramatic advances in software technology are promising to transform this situation into a world of compatible systems and devices.

TODAY'S SYSTEM CHALLENGES

Lack of software standards limit the effective use of real-time information from today's control systems. Difficulties in integrating the many required software interfaces put a bottleneck on the overall system capability to provide information on demand. Users and systems in today's fast-paced business environment absolutely require timely information in order to make critical production and business decisions. In addition, a lack of standards often means a lack of choice for users. Creating a multivendor system, in which a company can choose and apply the best products and systems for any given application often requires a willingness to invest a significant amount of time and money on system integration. This investment is not just to get the desired automation and control functionality from the system. Much effort is required to ensure that the systems, which are purchased, can share information and interoperate with other automation and business systems in the plant or factory. Today's manufacturing environment often consists of a complex combination of legacy systems and software that makes access to timely, decision-making information difficult. Such systems also impose high recurring maintenance and training costs on the company.

Currently, as a user, your system and information requirements can seem quite straightforward yet in reality are incredibly complicated to achieve. How many times have you wanted to simply transfer some data between two applications or generate a report on how your plant is operating, yet been thwarted by incompatibilities between software applications? Do you wish for the day when you can install software and systems and automatically realize plug-and-play connectivity and interoperability right out of the box? New software technology is bringing about dramatic changes in the automation industry. These technology advances are bringing into focus the reality of plug-and-play software. The industrial automation industry is embracing a new breed of software technology that emphasizes flexibility, openness, and ease of integration. This new technology provides enterprise-wide tools that are scalable, platform-independent, based on object-oriented graphical software development environments, and work in a distributed client/server architecture. Technologies such as Microsoft's ActiveX, based on OLE/COM (Object Linking and Embedding/Component Object Model), Sun's Java language, and Internet/Intranets are all new technologies that are blazing trails to the utopia of open standards and interoperability. These new technologies are just enablers, however. They are not a solution in and of themselves. Suppliers and users must work together to define how to leverage and use these technologies before they effectively create manufacturing and enterprise-wide solutions.

HUMBLE BEGINNINGS

The most overwhelming trend in automation systems in the past few years has been the move to Microsoft Windows platforms. From the meager beginnings of

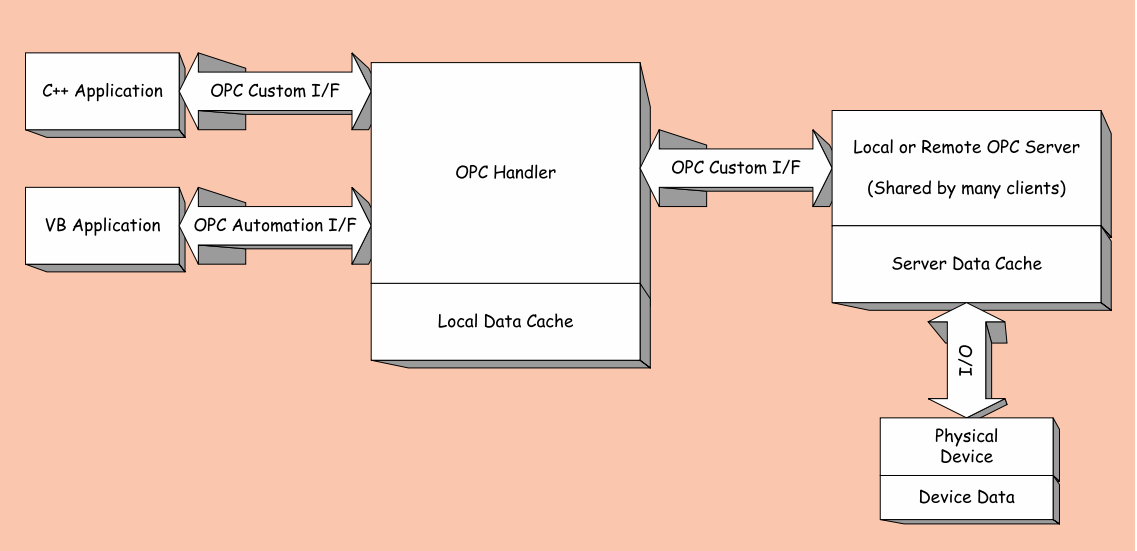

Figure 1.

Windows 2.x as a semi-stable, user-friendly MMI platform, Windows has now evolved in the form of Windows NT into a robust, mission-critical control system platform. In years past, Microsoft's dynamic data exchange (DDE) became a de facto standard communication interface in automation systems. DDE was designed as a simple data passing mechanism for communication between applications such as spreadsheets and word processors. First-generation MMI systems used this protocol, and derivatives such as NetDDE, as a mechanism to share data between devices such as PLCs and MMI applications. Many end-users, however, found performance and reliability limitations when using DDE as a real-time data communication mechanism. Microsoft replaced DDE with a higher performance, more robust, and reliable data exchange technology - object linking and embedding (OLE). With this new standard, the focus can return to concentrating on adding new functionality and engineering new products instead of carrying the burden of reinventing hundreds more proprietary communications interfaces.

STANDARDS ARE PAVING THE WAY

A promising new standard for the industrial automation market is OLE for Process Control, or OPC. An organization, the OPC Foundation, has been created to establish guidelines by which OPC will provide multi-vendor interoperability throughout the plant floor. The OPC Foundation's charter is to establish an open connectivity standard based upon Microsoft's OLE/COM (component object model). OLE/COM is an important part of Microsoft's client/server distributed computing strategy. With its set of interfaces, an end-user can interoperate and communicate between distributed components. However, to effectively use OLE as part of an integration framework, manufacturers must agree on OLE standards. OPC defines a set of OLE/COM interfaces, properties, and methods that extend this technology, making it useful in process and manufacturing automation applications. This initiative has two main benefits. First, OPC will enable control and business applications running on distributed, heterogeneous platforms to integrate at the object level. Second, OPC eliminates the need for different proprietary hardware and software device communication drivers. OPC will provide plug-and-play communication and interoperability between field devices, control systems, and enterprise-wide business applications.

ABOUT OPC

Microsoft's Object Linking and Embedding (OLE), now called ActiveX, Component Object Model (COM) and Distributed Component Object Model (DCOM) technology provided a standard software framework and many available tools including Microsoft's Visual Basic and Visual C++ and Borland's Delphi. DCOM enables an additional level of functionality, for it allows a client computer to use objects located on other networked computers. To OPC, this means an MMI can execute and collect data from OPC servers located on computers throughout a manufacturing facility. Utilizing OLE/DCOM technology for developing software standards to communicate with industrial devices led to the OLE for Process Control (OPC) specifications. Since the device manufacturers understand their hardware, its quirks, and can track changes in their products, these specifications are guidelines for hardware vendors to develop the servers required to communicate with their devices.

As these software drivers enter the market place, you need to understand OLE and DCOM technology and how to take advantage of OPC drivers. OPC is an evolving specification and the proceeding discussion is based on the OPC Specification Version 1.0 dated 8/29/96.

OPC TECHNICAL OVERVIEW

At the lowest object level, an OPC Server consists of an item that represents a physical I/O point on a device network. An item contains properties of the device including: the access path, value, time stamp, quality, and error information associated with it. The access path identifies a recommended communication pathway to the device. Group objects represent collections

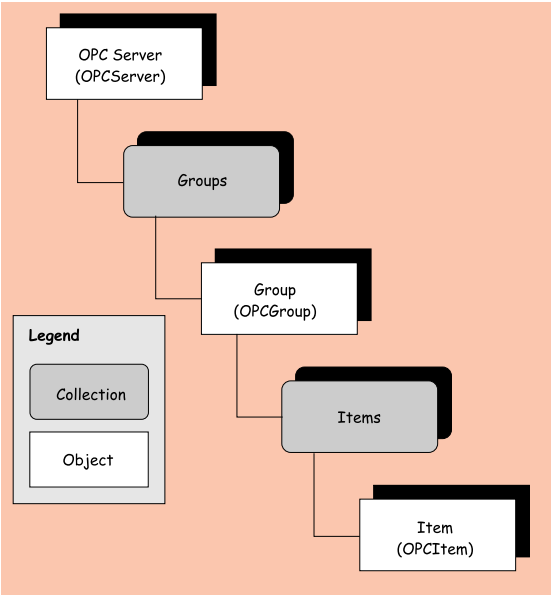

Figure 2.

of items. These items can be on different devices as long as the same server accesses all devices. I/O operations, like read and write, execute on groups. Groups can also perform a repetitive read operation at a specified rate. A server object represents a collection of groups. This hierarchy is represented in Figure 2.

An OPC Server implementation supports two sets of interfaces, the OPC Custom Interface and the OPC Automation Interface.

The OPC Custom Interface consists of the following objects:

The OPC Custom Interface is meant to be implemented by hardware vendors to support specific device and network protocol behavior. The custom interface is easily used by languages that can implement and access the vtable binary interface as prescribed by the OLE/COM specification. In general, client programs which are created in C++ and which wish to attain maximum performance will find it easiest to use the OPC Custom Interface. Use of the OPC Custom Interface from a scripting language such as Visual Basic is difficult if not impossible.

Objects	Basic Description
OPC Server	The main interface to an OPC Server. Contains methods for server level functionality. This includes browsing server address space and the saving and loading of configuration information. The OPC Group interfaces are acquired from here.
OPC Group	Contains methods for the configuration and management of a group of data items (scanning information, update rate, etc.). Groups are maintained by the server on behalf of the attached client. Actual data item definitions are server specific. All OPC Groups have an implementation of DataObject for asynchronous data access.
Enum OPC Item Attributes	Contains methods for accessing attributes of data items configured in OPC groups (current value, status, timestamp, definition, etc.).

The OPC OLE Automation Interface is meant to provide any OLE automation controller (e.g. Visual Basic and Delphi) with similar capabilities to access and configure process control data as the OPC Custom Interface. The OPC Automation Interface is completely independent of the OPC Custom Interface aside from using it as a client. In fact, a single common version of the OPC Automation Interface should work for all clients and servers. It is expected that a common implementation of the automation interface with be provided by the OPC Foundation.

OPC specifies the following OLE Automation Interfaces on objects, which may be used by any automation controller (e.g. Visual Basic and Delphi).

Object	Basic Description
OPC Server	Object with methods and properties for server level functionality (maps to OPC Server custom interface). An automation object with this interface must be created before you can get references to the other OPC automation objects.
OPC Group	Object interface with methods and properties allowing group (collection of OPC Items) functionality. The OPC Group object contains methods and properties that allow group level functionality (scanning information, update rate, etc.). Maps to OPC Group custom interface
OPC Item	An OPC Item object contains methods and properties that allow item level functionality (current value, status, timestamp, definition, etc.).

THE EMERGENCE AND ACCEPTANCE OF THE PC

OPC is an example of the effect that general-purpose computer technology is having on the automation industry. The advancement and widespread acceptance of the PC has demonstrated that suppliers and users can work together to create an industry-wide standard. There are standardized computer buses, such as ISA Plug and Play and PCI, standardized I/O ports such as RS-232 and Ethernet, and standardized software operating systems such as Windows NT. With these standards, millions of users share common applications and exchange information seamlessly. As more systems take advantage of the open software

and hardware architecture of the PC, manufacturing systems will become increasingly more open, flexible, and lower cost. This will allow end users to focus on their key success factors, such as better quality and improved time-to-market, rather than on the integration headaches imposed by today's proprietary systems. Suppliers will benefit from the lower costs of technologies whose development costs are spread across millions of PC systems rather than just tens of thousands of automation systems. Vendors also benefit from the ability to focus on their differentiating value-added capabilities rather than on developing enabling technologies such as OPC.

New technologies based on standards now give us a great opportunity to create an integrated software environment that permits development of reusable, plug-and-play objects that are interoperable across corporate-wide manufacturing and business applications. This capability has already proven valuable to business users, where spreadsheets, word processors, and databases can share information across company boundaries. The promise of plug-and-play control system components is within close reach-. Interchangeable hardware and software components based on standards like OPC are a great step forward to achieving this goal.

Mike Santori is the Director, Industrial Automation Marketing at National Instruments. Santori oversees the direction of National Instruments products as they are applied to industrial automation applications. His focus covers all products as they apply to these applications, with special focus on industrial communications products, such as PC-based interfaces for fieldbus networks. He was previously the Software Development Manager, an R&D position in which he managed the company's application software products, LabVIEW and LabWindows, as well as the software development for VXI and DSP products. Santori has been with National Instruments since he graduated from Texas A&M University in 1986 with a B.S. in electrical engineering. He has published numerous technical articles and conference papers, especially in the areas of fieldbus, application software, and test program development.

ADVERTISEMENT INDEX

COMPUTER PRODUCTS	65	OR INDUSTRIAL COMPUTER	85
DIGITAL EQUIPMENT CORP.	29, 89	QNX SOFTWARE SYSTEMS	2
EMBEDDED SYSTEMS 98	53, 69	RADISYS CORP.	25
GREEN HILLS SOFTWARE	49	REAL-TIME CONSULT	21
GROUPIPC	103	RTUSI	99
IMAGINATION SYSTEMS INC.	57	SYSCOMMS 98	33
INTEGRATED SYSTEMS INC.	115	SYSTRAN CORP.	45
NATIONAL INSTRUMENTS	15	US SOFTWARE	93
		WIND RIVER SYSTEMS	116

To reserve your advertisement,
contact Alexander Teetaert at:
Tel: 32-2-520.55.77 or Fax: 32-2-520.83.09
or Email: info@realtime-info.be

RT consult