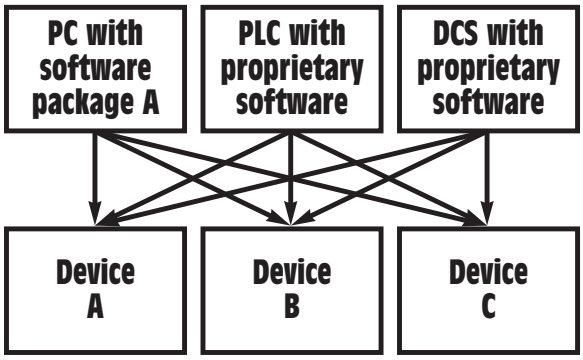
OPC Technical Overview

Introduction *(Statement of the problem)*

The use of microprocessors has proliferated in manufacturing plants, and they often do not work together. Application software should readily communicate with digital plant-floor devices as well as other applications, but this is not often the case. Making these systems work together is the most pressing need of process manufacturers. The problem has become more acute than network connectivity, diverse operating systems, and not-so-open "open systems" that are supposed to facilitate interoperability.

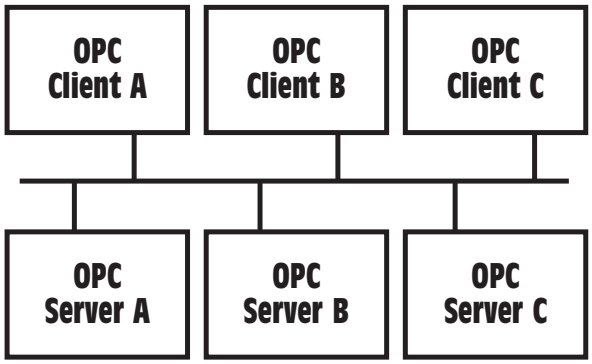
A key reason for this problem is that interfaces are not standard. Proprietary systems that don't communicate among each other are fairly common. Hardware and software choices for process and industrial manufacturers are sharply reduced because their application suppliers provide limited connectivity.



In the absence of any standard, vendors have developed proprietary hardware and software solutions. All process-control and information systems on the market today have proprietary techniques, interfaces, and APIs (Application Programming Interfaces) in order to access the information that they contain. The cost of integrating the different systems and the long-term maintenance and support of an integrated environment can be significant. Custom drivers and interfaces can be written, but the variety increases rapidly because of the thousands of different types of control devices and software packages that need to communicate. This proliferation of drivers exacerbates certain problems, such as inconsistencies among different vendors' drivers, hardware features that are not universally supported, hardware upgrades that can wreck an existing driver, and access conflicts. In the last case, two applications generally may not be able to access the same device at the same time because they use independent drivers. Perhaps worse, a driver may not be available for a particular device application or inter-application combination, limiting End Users' options or forcing them to have one written (and debugged).

For a typical supervisory-control software developer, as much as 25-30% of engineering development time is spent writing drivers. Every time a key supplier comes out with a new controller, all software developers have to write a new driver.

Application software suppliers spend too much money developing and maintaining proprietary interfaces, adding to end-user costs and contributing nothing to solve the problem of getting a variety of systems to work together. Application users are, in a sense, controlled by their application software suppliers.



The solution is having a standard that provides real plug-and-play software technology for process control and factory automation where every system, every device and every driver can freely connect and communicate. Having such a standard makes possible the prospect of totally seamless, truly open and easy enterprise-wide communications between systems and devices, from plant floor to MIS (Management Information System) and beyond.

The name of that standard is OPC (The letters O-P-C originally stood for OLC — Object Linking and Embedding — for Process Control. OLE has since been restructured from object-oriented to object-based and renamed ActiveX). With OPC in place, everything is set to clear away the confusion and high cost of the multiple proprietary servers, drivers, and interfaces that were necessary for systems to communicate in the past. OPC will bring the same benefits to industrial hardware and software that standard printer drivers brought to word-processing.

OPC will bring the same benefits to industrial hardware and software that standard printer drivers brought to word-processing.

OPC Technical Overview

History

OPC (OLE for Process Control) is an industry standard created with the collaboration of a number a leading worldwide automation and hardware software suppliers working in cooperation with Microsoft. The organization that manages this standard is the OPC Foundation. The Foundation has over 150 members from around the world, including nearly all of the world's major providers of control systems, instrumentation, and process control systems. The OPC Foundation's forerunner — a task force composed of Fisher-Rosemount, Rockwell Software, Opto 22, Intellution, and Intuitive Technology — was able to develop a basic, workable, OPC specification after only a single year's work. A simplified, stage-one solution was released in August 1996.

The objective of the OPC Foundation is to develop an open, flexible, plug-and-play standard that allows end users to enjoy a greater choice of solutions, as well as sharply reducing development and maintenance costs for hardware and software suppliers.

The OPC Foundation has been able to work more quickly than many other standards groups because OPC Foundation is simply building on an existing Microsoft standard. Other groups which have had to define the standards "from the ground up" have had a more difficult time reaching consensus as a result of the scope of their work.

Microsoft is a member of the OPC Foundation and has given strong backing to the organization. However, Microsoft has been careful to remain in the background and let the member companies with direct industry experience guide the organization's work.

One of the most valuable aspects of Microsoft's participation is the fact that it hosts an annual OPC Foundation meeting in Redmond, Washington (Microsoft Headquarters) to provide Foundation Members with a preview of coming developments in OLE/COM and other Microsoft technologies. Many Foundation Members are small companies and would not receive that kind of briefing from Microsoft if they were not Foundation Members.

End-Users are encouraged to join OPC Foundation, and several manufacturers actively participate in the specification and technical review process. Both End-Users and Automation Suppliers benefit from having a standard. For every automation system installed today, there is a significant amount of time and money spent on integration. OPC ensures that automation systems can share information and interoperate with other automation and business systems across their plant or factory.

OPC Technical Overview

What is it?

Based on Microsoft's OLE (now ActiveX), COM (component object model) and DCOM (distributed component object model) technologies, OPC consists of a standard set of interfaces, properties, and methods for use in process-control and manufacturing-automation applications. The ActiveX/COM technologies define how individual software components can interact and share data. Backed by Microsoft's NT technology, OPC provides a common interface for communicating with diverse process-control devices, regardless of the controlling software or devices in the process.

The goal of the standard is Plug-and-Play, a concept developed by Microsoft and a number of other companies a few years ago. By using a standard way of configuring computer hardware (and software interfaces) automatically, a device will easily connect to another and immediately work without the need for lengthy installation procedures or complex configuration. Instead of having to learn how to use 100 or more custom toolkits, users will only have to learn one set of tools, because all OPC drivers will work the same way. OPC's purpose is to compel the automation industry suppliers to push all device drivers toward a standard form. Essentially, OPC defines a common interface that permits interface development work to be performed once and then easily reused.

The OPC standard requires hardware suppliers to provide front-line data collection and distribution. They are the most familiar with how to access the device's internal data efficiently. These devices then become OPC servers, providing data to OPC client applications consistently. Application developers can then write code in any language deemed appropriate.

Instead of having to learn how to use 100 or more custom toolkits, users will only have to learn one set of tools, because all OPC drivers will work the same way.

What is COM?

The Component Object Model provides standard interfaces and inter-component communications. Through COM, an application may use features of any other application object or operating system, or allow for software component upgrades without affecting the operation of the overall solution. COM can be used by developers and system integrators to create customized solutions. A binary standard, COM is generic and is the core of DCOM, ActiveX, and OLE technology.

What is OLE?

Object Linking and Embedding is used to provide integration among applications, enabling a high degree of application compatibility, even among diverse types of information, OLE technology is based on COM, and allows for the development of reusable, plug-and-play objects that are interoperable across multiple applications (see accompanying OLE Automation definition). It also provides for reusable, component-based software development, where software components can be written in any language, supplied by any software vendor.

What is OLE Automation?

OLE Automation and the underlying COM technologies were designed by Microsoft to allow components (written in C and C++) to be used by a custom program (written in Visual Basic or Delphi). This model provides a precise match for the needs of the process-control industry, with hardware developers writing software components in C and C++ for handling data access from a device. Through OPC, application developers can write code in any language necessary to request and utilize plant-floor data.

What is DCOM?

The Distributed Component Object Model extends COM to networks (remote objects). It is a new, highly optimized protocol, where remote components appear to be local. DCOM was first released for Windows NT 4.0 in August 1996. Microsoft Java and VB Script support DCOM and ActiveX development. Other companies are developing versions of DCOM and ActiveX for non-Microsoft platforms.

What is ActiveX?

ActiveX is an umbrella term of a broad range of technologies that used to be known as OLE Controls, all of which rely on COM. A renaming and restructuring of the OLE Controls technology, it is object-based rather than object-oriented. ActiveX is an open, integrated platform that lets developers and Web producers create portable applications and interactive content for the World Wide Web. It's open, cross platform, and is supported on Mac, Windows, and Unix systems.

What is DDE?

OLE's predecessor, Dynamic Data Exchange, is a method of dynamically moving data among applications in the Microsoft Win32 application-programming interface (API). The DDE protocols send messages between applications that share data and use shared memory to exchange data. Applications can use the DDE protocol for one-time data transfer and for continuous exchanges in which applications send updates to one another as new data become available. Prior to OPC, hardware manufacturers that recognized the need for software connectivity to their hardware was limited in their ability to develop drivers—the choice was between DDE and a list of proprietary DDE derivatives. Selecting any of these had possible effects of limiting the user's choice of software or preventing the acceptance of the hardware.

OPC is analogous to electronic and pneumatic process instruments. In this analogy, process instruments are likened to today's software components. Electronic and pneumatic instruments each have a standard interface — 4-20 mA or 0-15 PSI. Customers could choose the right instrument from the right vendor to meet their unique requirements. The highly integrated digital control products introduced in the eighties and still largely installed today do not offer the End User this freedom of choice.

OPC provides a single, consistent, industry-standard interface that permits software suppliers to focus on adding new features to their software instead of developing long lists of proprietary hardware device drivers. The OPC standard has provided an environment in which device manufacturers will be encouraged to invest in the development of their own OPC servers knowing that the same server can be used by every software, HMI, PLC or DSC vendor. There is an incentive for device manufacturers to bring their inherent knowledge of the industrial networks to OPC server development to ensure that device performance is optimized.

The most obvious benefit is to the HMI vendors who, up until now, have had to invest significant resources in developing and maintaining proprietary drivers to all possible industrial networks.

In short summary, Microsoft's COM is a software architecture that allows applications to be built from binary software components. COM is the underlying architecture that forms the foundation for higher-level software services, like those provided by ActiveX. These services span various aspects of commonly needed system functionality, including compound documents, customer controls, inter-application scripting, data transfer, and other software interactions. For example, ActiveX and COM allow applications to share "objects," such as spreadsheets that are embedded in word-processing documents. When updates to the spreadsheet occur, ActiveX and COM ensures that they are automatically reflected in the word-processing document.

Benefits

Asking what the business benefits of OPC are is like asking what the benefits of plug-and-play technology are to the computer industry. More choices, better access to process data, ease of plug-and-play operation, and efficient utilization of development resources are the main benefits of OPC technology.

OPC brings the value that comes with the use of standards, including reduced training costs, reduced custom development costs, and lower long-term maintenance costs. By design, OPC-compliant products work seamlessly with one another. With this plug-and-play approach, off-the-shelf components can be brought together efficiently to solve immediate requirements. In addition, long-term maintenance and upgrading can be done by removing and replacing individual components in a system without any work needed to "wire up" the new pieces.

Asking what the business benefits of OPC are is like asking what the benefits of plug-and-play technology are to the computer industry.

To illustrate the savings, imagine the increase in cost if every household appliance had its own type of wall plug. Eliminating customization drastically reduces the cost of an automation system by saving money during acquisition, installation, and maintenance.

With the introduction of OPC-compliant manufacturing automation products, users are provided their due right to select and implement systems comprised of best-in-class components without the pain of custom interfaces. This user benefit is sometimes referred to as "freedom of choice." For example, both Netscape and Internet Explorer can browse the web equally well, but people use the browser they like best. As a result of this freedom of choice, vendors will need to become more competitive and offer superior products and solutions to maintain their customers.

Besides freedom of choice, the user also has vendor independence, or "freedom from a proprietary lock." If the implemented control system is comprised of modules with proprietary interfaces, any customer who desires to upgrade any component function of the integrated whole is entirely dependent on the vendor. With OPC components, only the module of interest must be upgraded and not the entire system. The requirement to use the original vendor is eliminated. High-priced proprietary solutions (and their expensive after-sale support contracts) will yield to lower cost OPC-enabled alternatives.

Additionally, with Plug-and-Play components, other applications in addition to the original "clients" of the data servers may now request/transmit data without the need of a vendor gateway. The result is more data access to more interested clients with less complication.

Using OPC for connectivity promotes higher quality solutions. OPC interface products are built once and reused many times; hence, they undergo continuous quality control and improvement. The components must constantly prove themselves when products from many vendors interact with them. This approach contrasts the traditional "build once, use once" interfacing approach, and encourages proven, bug-free products.

> *To illustrate the savings, imagine the increase in cost if every household appliance had its own type of wall plug.*

OPC enables companies to build robust and durable automation solutions quickly, choosing the right components for their specific requirements. It decreases the up-front costs and reduces long-term costs of automation and control systems. The beauty of OPC is that the 'client' system—be it in the control room or an executive's office—has to understand only one interface to get data from any device. In the past, if a user wanted to get data from a vendor's proprietary control system, the user had to buy their proprietary application-programming interface. In contrast, OPC presents the data from any control system in the same way. OPC client application can be connected to any vendor's OPC server in the same way and expect the same behavior and information from the server.

Beyond technological elegance, OPC has practical "bottom-line" implications, both for users and vendors. In the past, if a user wanted to mix and match application software systems and devices from multiple vendors, he first had to find out if the software drivers for a given device or system were available. If not, he had to pursue another solution or invest the time and money necessary to develop a custom driver using that vendor's proprietary interface. OPC eliminates the compatibility problem, allowing users to select precisely the devices and systems they want and can afford for a particular application.

Benefits to Vendors

Time Savings *(Eliminate Driver Development)*

OPC server vendors develop one version of their driver that communicates with all OPC client applications. The costly development of I/O drivers will be diminished substantially. The vendor can focus their development resource at communication to the end device, rather than worrying about different client communication schemes.

Increased Served Market through Increased Connectivity and Interoperability

Products will plug together more easily. I/O manufacturers will be able to more readily sell their hardware (one OPC I/O server will replace the need for many specific drivers that can talk to various products). Users will be able to take advantage of the products they want to use.

Focus on Value-Added Activities

Software vendors can focus efforts on adding value to their core SCADA, HMI, and Batch product offerings. It also allows third-party application vendors (such as specialized vertical market packages, advanced alarm handling, and statistical analysis) to work more easily with data from other vendor's products.

Benefits to Users

Time Reduction through Lower System Integration Costs

OPC eliminates the need for costly custom software integration. OPC provides plug-and-play software and hardware components from a variety of automation software, device, and system suppliers. Process and manufacturing companies can easily integrate applications into corporate-wide automation and business systems, something that has been virtually unachievable in the past. OPC-compatible components greatly reduce system integration costs because all software and hardware components adhere to a single, standard interface that's being adopted around the world. Automation suppliers are providing hardware devices with integrated OPC servers that are replacing proprietary device-driver software. The driver connection between hardware and software from different vendors has historically been the number one headache in system integration. OPC offers the opportunity to ease the pain and shorten the application development cycle. This gets automation projects up faster, which saves time for new projects and brings the benefit of automation to the process sooner.

Ease of Integration with Plug-and-Play *(Connectivity)*

PC technology is extending beyond hardware I/O to more complex control and business systems. DCS, SCADA, HMI, plant scheduling, maintenance, and other

manufacturing applications supporting the OPC standards enable the open exchange of information between cooperating applications across the manufacturing enterprise. This allows the manufacturing customer to focus his efforts on value-added business activities versus system integration problems.

The business benefit is that we have interoperability between clients and servers. The end users reap the benefit by being able to pick and choose which components are appropriate for their installation and be assured that the components will play together.

Ease of Connectivity and Interoperability of Custom Applications

Customers may develop simple Microsoft Visual Basic applications to exchange data with any OPC server or to use their favorite OPC client application to exchange data with any OPC server. The secondary benefit is that client applications, with full access to the plant floor, can be written with little or no knowledge of the industrial network. Standardization has provided the stability necessary to encourage applications from a much wider range of software vendors and service providers.

Eliminate Proprietary Lock of Legacy Vendors

OPC client applications can focus their development on the application functionality, rather than device connectivity. Previously, customers were limited to choosing among the client applications that supported communication to the devices in their installation.

With OPC, customers are no longer bound to a single vendor. If a plant has a legacy installation, End-Users do not need to stick with the same vendor. All client applications have the same connectivity to the same set of devices. What vendors have in their repository of implemented connections will no longer be a factor in the customer's decision to purchase an OPC client application.

If a device vendor develops a new product, it is up to the device vendor to provide the OPC interface. It is not the responsibility of every software supplier to invest thousands of dollars in new drivers.

Freedom of Choice to Pick "Best in Breed Products"

With the interoperability OPC provides, End-Users can choose software or hardware from different vendors and know that their components will seamlessly work with one another. In return, vendors will need to become more competitive to maintain their customers' loyalty, benefiting End-Users.

OPC technology makes it possible for Systems Integrators and End-Users to select optimum, vendor-independent components when designing an automation system. This "open systems" approach enables engineers to pick products that meet their exact requirements, as opposed to modifying the requirements to match the system, regardless of the vendor.

Access to Data by Anyone in the Automation Hierarchy

Another benefit of OPC is access to process-related data at every level of the enterprise. No longer is this strategic data restricted to the plant floor. Visual Basic access via the OPC Data Access Specification permits plant data to flow upstream to the business applications. Armed with the right data, decision-makers are better equipped to make strategic and "Just In Time" decisions to improve business efficiency.

Widespread adoption of OPC will also result in greater sharing of information across multiple applications simultaneously. As more applications are OPC-enabled, the same information may be distributed to multiple applications (such as maintenance, inventory, operator displays, and document management) using a combination of OPC and DCOM—such that business processes may be coordinated simultaneously.

Ease of Use — Auto-Configuration of Tags

Effectively designed OPC components are also very easy to use, requiring very little configuration. OPC servers do not require the user to configure tags at all; the server can automate this configuration, making an OPC installation a turnkey solution.

Reduced Troubleshooting and Maintenance Cost

OPC offers a standard that once learned minimizes the need to be an expert on every protocol.

Add/Delete without System Shutdown

Items can be added and deleted without shutting down the server. This is far superior to many proprietary drivers that require the driver be stopped before points can be added. An example of the use of this feature: points can now be added through a database front end—defining points consistent with the server's syntax. Data will be returned immediately on an OPC Server.

Synchronous and Asynchronous Device Writes *(not possible before OPC)*

Synchronous and asynchronous device writes, with acknowledgment, is superior to previous DDE drivers which presented huge problems to application developers. On some DDE drivers, an application would attempt to write a value to the PLC. However, before the value actually got to the PLC, the value would be overwritten by a polled read by the driver.

Non-Obsolescence

One of the benefits of using standard technology like COM, DCOM, and ActiveX is that current OPC clients will not be obsolete when new functionality is added to the server. It's very easy to extend the OPC server by adding new COM interfaces while keeping all the existing COM interfaces backward compatible. This feature is very important to End-Users.

OPC Technical Overview

Frequently Asked Questions

Who should care about OPC?

You should care about OPC if your applications are largely run in personal computers and you are involved with solving plant integration problems. As a critical mass of servers and OPC-enabled applications become available, OPC is likely to become an important part of your plant integration tool set. The OPC specification promises a future without proprietary interfaces that will greatly benefit both manufacturing customers and automation suppliers.

How is OPC going to improve my bottom line?

From a business perspective, the use of OPC for connectivity promises to reduce the cost of automation, control, and integration solutions. By using OPC-compliant products, significant savings can be achieved through shorter development efforts and a wide choice of vendor hardware and software solutions.

For every automation system installed today, a significant amount of time and money is spent ensuring that the system can share information with other systems and devices. OPC will save the customer time and money by eliminating a lot of the system integration problems caused by lack of open standards that exist between automation devices, systems, and manufacturing software.

After an automation system is installed, OPC will not improve business bottom line directly; however, OPC will provide a common method to access real-time information. The real key to improving the bottom line is to distribute and use the information throughout the business' value chain.

With the introduction of DCOM, how does OPC handle problems such as the remote server being disconnected?

DCOM provides built-in features that ensure OPC clients and servers have a robust and reliable mechanism to exchange real-time information across the network. DCOM also handles retries and time-outs between an OPC client and a remote OPC server and tries to re-establish communications if they are disconnected.

One of the strong points of OPC is that it leverages other standard software technology like Microsoft ActiveX, DCOM, and Windows NT. Microsoft DCOM technology makes distributed, client/server networking transparent to the OPC application. DCOM makes the underlying network communication protocol transparent to the OPC client/server. DCOM may send OPC messages using a variety of transports such as UDP, TCP/IP, and IPX, using the same OPC application using DCOM.

Can OPC implement safe shutdowns?

If you are asking if OPC can be used to implement a safety shutdown system, the answer is no. If you are asking if OPC can implement safe shutdowns of a system, if certain conditions exist, the answer is yes.

OPC does not directly specify any type of shutdown mechanism or requirements of a client/server to provide this type of functionality. OPC servers and clients can exchange messages that contain shutdown commands. These commands can be interpreted and implemented by the control logic running in the software and hardware control devices that are connected to OPC.

Acknowledgments

Parts of this OPC Overview were extracted from a Special OPC Supplement of Start Magazine published in their January/February 1998 issue.The entire OPC Supplement can be viewed at:

<www.startmagazine.com/v2n1opc0.htm>

Contributors to the Start Magazine Supplement included:

John Baier
Rockwell Software

Ed Bassett
Start

Bob Bloom
Intellution

Tom Burke
Rockwell Software

Al Chisholm
Intellution

Mark Davidson
Foxboro

Mark Edmond
Start

Brandon Ekberg
Rockwell Software

John Ham
ObjectAutomation

Ted Hannah
SST

Don Holley
National Instruments

Chris Kellogg
Iconics

Chris Martyniuk
Matrikon Systems

David Rehbein
Fisher-Rosemount

Steve Rubin
Intellution

Marcus Schmidt
Microsoft

Paul Shinohara
USDATA

Nizar Somji
Matrikon Systems

Dan Sussman
Start

Randy Thompson
FactorySoft