The Myth and Magic of OPC

Patrick Dixon, DPAS-Inc (www.DPAS-Inc.com)
Michael Velarde, Austin Energy

Abstract

This paper addresses the benefits and shortcomings of control system applications based on OPC (formerly known as OLE (Object Linking and Embedding) for Process Control). A case study at Austin Energy's Decker Power Plant provides the basis for the presentation. It is found that OPC provides great benefit, but the implementation of this standard in the industry is severely lacking in diagnostics to help the end user when problems arise.

Introduction

Many of you have dreamed of the day when a process control system would easily allow third party applications to read and write data in a standard way without a custom, proprietary interface. You remember the days when a vendor with a slick advanced control application spent weeks at your site just to get the darn thing to communicate with your DCS (Digital Control System). You remember two vendors sitting in a room with you trying to convince you that the other guy is to blame for communications problems. You remember staring at a protocol analyzer and digging through code and log files to figure out what is going on. You dream of an end to this nightmare.

Keep dreaming. The day has not quite arrived.

OPC (former known as OLE for Process Control) is a beautiful thing when it works. It is a widely adopted industry standard for communication between applications in a control system. Often times it works without a hitch, and applications automatically install and start talking with your system. It is built upon a widely adopted standard. The fundamental system enabler for OPC, DCOM (Distributed Component Object Model), is a field proven, stable, built-in capability of Microsoft systems which are the predominant platform in process control at the control LAN layer and above. You get that warm and fuzzy feeling that you chose wisely in your selection of system design.

Unfortunately, there are times when OPC shows its uglier side. Communication doesn't work. There is little if any diagnostic information to tell you why. Possible causes run the gamut from something in the application, something in your particular system, or a gremlin within Microsoft Windows.

For example, consider the following message in the system Event Viewer:

```
DCOM        Error   None
10000  NT AUTHORITY\ANONYMOUS LOGON DECKER_OPC
"Unable to start a DCOM Server:
{67D4CF31-FB32-12E3-7DEF-2150DBC90274}.
The error: "Access is denied. "
```

Does this help identify what is wrong and how to fix it?

If you like that, you may also see the following common OPC related error codes when debugging:

0x80040202
0x80070005
0x800706BA

Since most of us have not memorized what these stand for, we have to look these up and see if there is a more helpful explanation. Here is what is provided from one of the latest OPC debugging tools:

0x80040202: Unable to open the access token of the current thread
0x80070005: General access denied error
0x800706BA: The server is unavailable. Maybe the server is not enabled in the Firewall.

Well at least that last one gives you an idea of what to do.

Finally, when things aren't working and you are looking for help, what you are likely to find may not help. The organization that is responsible for ensuring OPC products are able to interoperate with one another is the OPC Foundation. They have a message forum on their website used to get answers on how to solve OPC problems. An example of what you may find is this:

> Posted: Fri Feb 13, 2004 5:15 pm
> We have a customer who wants our client to connect to OPC Servers in two or more different domains. The Client computer will be in a different domain or in a domain of one of the OPC server's computers.
> My question is this: Is it possible for the OPC clients to connect to the remote OPC servers across different domains? If it is, what type of users should the OPC Clients be running under and how should DCOM security be set up to allow access to these users?
> Thanks for any advice.
>
> Posted: Sun Feb 15, 2004 12:46 pm

> To allow OPC communications to work between domains using DCOM you must implement a "trust" relationship between the domains involved. This would allow groups from multiple domains to be granted the proper DCOM access rights for OPC communications to succeed.
>
> Posted: Thu Feb 19, 2004 11:03 am
> Can you tell me if we need to set up a trust in both directions? I can see that the OPC Server domain needs to trust the OPC Client domain. Does the OPC Client Domain also need to trust the OPC Server domain? I ask this because we use Callback to pass all the information and so there is a DCOM connection in both directions.
>
> Posted: Thu Feb 19, 2004 3:30 pm
> Yes the trust must be in both directions because of the OPC callbacks.

For those more experienced and familiar with OPC, you know that this is not quite accurate and can lead you down the wrong path. Of course this is a message board and there in no guarantee anything posted on the Internet is accurate, however this is often about the only source available when you need help.

The purpose of this paper is to present experience with an OPC installation as an example to plea for better diagnostic support of this technology. Experience at the Austin Energy Decker Road power plant will be used to illustrate an example of an OPC project and the real world challenges inherent in such projects.

OPC

OPC (original acronym for OLE for Process Control) is a standard. It is not a product, it is not software code. It is a standard presented to developers of control systems and software applications for enabling communications between process control systems and process control applications. It exists because there are features inherent in process control applications that are not sufficiently addressed by other existing communications standards such as OLE, DDE, and others. It provides communications at the application layer, not for field I/O and hardware devices. That is the realm of Fieldbus, Profibus, Modbus, and others.

OPC is built on and relies upon the Microsoft Windows architecture. It specifically is dependent on DCOM (Distributed Component Object Model), a Microsoft mechanism for inter-application communication on a network. OPC built on DCOM does not work on Unix, Linux, Apple, or other non-Windows systems, although plans for portability are being designed in future enhancement of the standard.

On one side of OPC communications is an OPC server application. This application "exposes" data items in a process control system or application. By "exposing" the data, it makes it available to any

OPC client application. The OPC client can browse these data items in realtime and connect to them. The client can perform reads and writes once it has connected to the data item. OPC has provisions to make data items read-only and to limit "exposure" of specified data.

In addition, OPC has a security mechanism that relies on Microsoft system administration. Accounts and passwords in the operating system are used to control access to data. Any OPC server that receives a request from an OPC client must authenticate that the client has access to read/write the data. The account and password of the requesting client application must match a user/account in the machine where the server resides. If there is no match, the server will not allow the client to connect.

For more detailed information on OPC, please refer to the many books available such as "OPC Fundamentals, Implementation, and Application" by Iwanitz and Lange.

Austin Energy

Austin Energy's Decker Road power plant has twin 400 mega-watt generating units (furnace, boiler, turbine). The facility is 40 years old and had been operating on panel boards and vintage control systems. In recent years the systems have been replaced and upgraded to the latest technology. It is desired to make use of this technology to open up the system for advanced control schemes and to reduce the effort to provide reports. There are 2 criteria for system design:

- Allow applications to easily read/write data without the high maintenance and cost of custom communication drivers
- Security! This is a critical provider of power to the city of Austin and the Texas grid. This control system will not be connected in any way to the outside world (Internet). Also, functions such as plant operation, system configuration, and programming should only be accessible to the appropriate plant personnel.

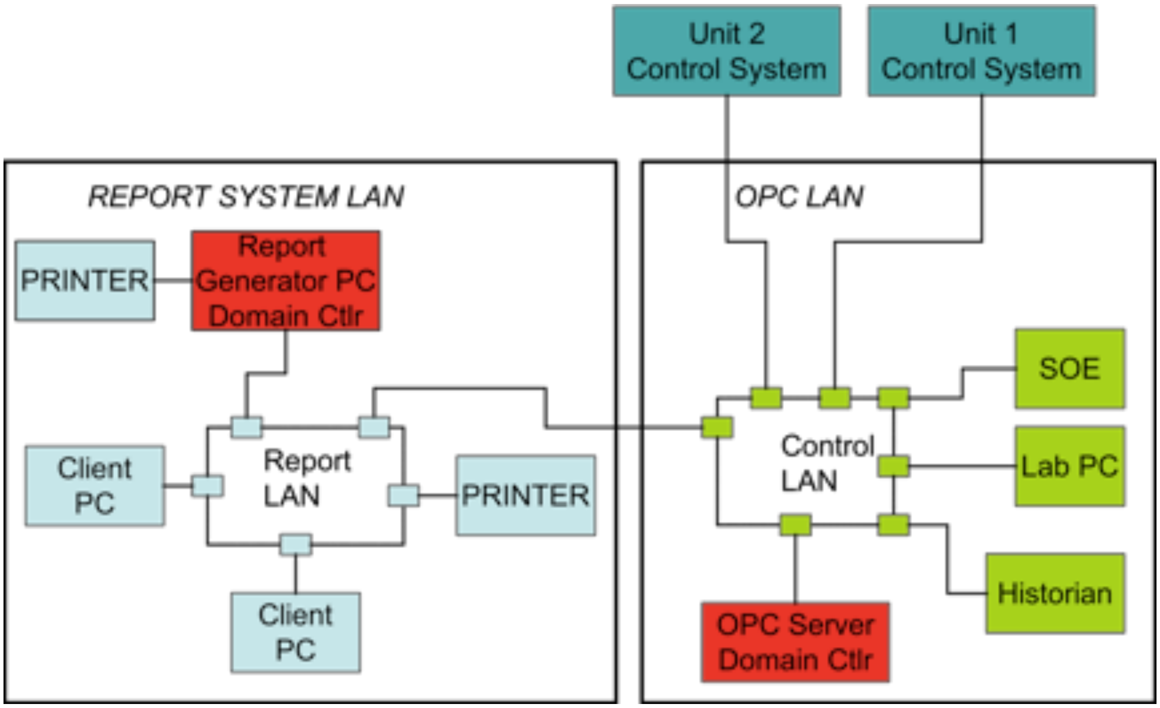
This would seem to be an ideal opportunity to make use of OPC compliant systems and applications.

Project

The design is to have a base control system for each generating unit running independently. On top of these systems is a fiber optic ring network, which ties in a plant historian and allows for specific data communication between the 2 systems. Also a Sequence of Events (SOE) recorder is connected on this network and provides inputs to the control systems. An OPC bridge also runs on this system and provides the bridge for all communications on the network. This ring is setup as a Microsoft Windows domain.

The last layer is an office network. This allows managers to access realtime data from the system and generate reports. It is imperative that office personnel not be able to make changes that will effect operations.

The network diagram looks like this:



In order to do this, the office network is setup as a separate domain. In Windows terminology, the plant domain and the office domain could either be separate domains in the same forest, or separate domains in separate forests. Windows 2000 forces a two-way transitive trust between 2 domains in the same forest. A two-way trust enables users from either domain to access resources in the other domain. This is not desired; we want the office to get able to get data, but not write (read-only). Therefore, we want to setup a relationship where the office system trusts the plant system.

Implementation

Both base control systems have an OPC server application that “exposes” process data to OPC clients. The plant fiber optic LAN connects by Ethernet to the base system node that hosts this OPC server application on each system. On the plant LAN is a server class machine running Windows 2000 server. This machine is the domain controller for the plant process control LAN. The base system nodes that connect to the plant control LAN are not members of the domain; they are members of their own base control system network. A machine cannot be a member of 2 domains at the same time.

This can present a problem for OPC based system design. Often an OPC client must be in a domain that has a trust relationship to the server in order to be granted access and connect. In our case were we able to connect with limited functionality. The ability to browse available data items in the server was prohibited without an established trust relationship. We were able to connect to data items by hand-typing the pathname in the client.

At this point we are able to read data from both control systems through OPC. The historian can store data from both systems.   For some specific cases we are able to read data from one control system and write it down to the other.

The next step is to connect the office LAN and read process data for reports. Since we do not want office personnel to make changes in the control system, we intend to implement the office LAN as a separate domain in a separate forest from the plant process LAN. To place the office LAN in the same forest under a different domain would result in Windows automatically setting a 2-way trust between the domains. Due to security concerns we do not want this. Therefore, our choice is the only way to go.

We were able to setup the office LAN with its own domain and forest. We then tried setting up the one-way trust. Through several iterations we were able to get this setup. We were able to verify that the plant control LAN could access files on the office network, but the office network had no access to the plant LAN

Our OPC client was unable to get good data from the OPC server in this configuration. We were able to connect and browse data items, but every data item would show “Bad data”. We thoroughly checked every consideration and were not able to find a way to get good values. There is no diagnostic that will tell you what the origin of the problem is. We ended up re-installing the office LAN server to act as a separate domain in the same forest. This yielded the same results. Lastly we made the Office LAN server a member of the plant LAN domain and we were able to get good data. In order to address security, we decided to implement this through creating several user accounts in the office domain with restricted access. We also located the OPC client application outside of the office area.

What is most lacking in OPC is diagnostics information. If a client and server cannot communicate due to Microsoft security, there is no indication to the user that this is the problem.

Current status

This system has been in place for over 4 years. Over the course of this time there have been interruptions of data communications occasionally. Various individuals have performed maintenance, but there is a general fear and reluctance to touch it.

The report LAN is not being used due to intermittent communication problems.

A process historian was installed on the control LAN. Austin Energy attempted to use OPC to bring in data points to the historian. This worked very well in some instances, and failed in others. There have actually been intermittent interruptions in communications that have caused loss of historical data. Experts were brought in to ensure that the OPC server was set up correctly. After re configuring the server, data flow will start off good and then fail for unspecified reasons. Austin Energy is currently looking into different options to harness these data points for the historian.

It is interesting to note that security vulnerabilities have been associated with DCOM (Distributed Component Object Model), which OPC relies on heavily. ISA TR99 states "communications like DCOM that dynamically open a wide range of ephemeral ports should be avoided." It is also interesting to note that current OPC standardization is moving away from the Microsoft Windows based architecture and protocols like DCOM.

Conclusion

OPC is a conceptual improvement upon custom communication drivers in process control systems. What is needed to make it a complete solution is diagnostic information that will help the user debug problems when they arise.