Technology Analysis

Is OPC/DX in your Future?

By Dick Caro, CEO, CMC Associates, Acton, MA

The terrorist attacks on the World Trade Center in New York and the Pentagon in Washington, DC, have completely overshadowed another major event on that September 11, 2001 morning – the announcement of OPC/DX by the OPC Foundation and many of its sustaining members. The promise made that fateful day was that the OPC DX (Data eXchange) protocol would enable systems to exchange real-time data independent of the underlying bus technology. To emphasize this point, competitors Emerson, Invensys, Rockwell Automation, and Siemens all announced support for OPC/DX. Since that time, however, there have been no further announcements from the suppliers or the OPC Foundation.

What Is The Problem?

Why was DX needed? To answer these questions, we need to examine the nature of OPC/DA (Data Access) protocol and the features of the underlying "fieldbus" protocols. DA was designed to solve the problems of the HMI (Human/Machine Interface) suppliers who found it necessary to write a custom interface or driver for each type of control or data acquisition system supplying data. Likewise, each supplier of control or data acquisition system found it necessary to write a custom interface or driver for each HMI package they wanted to support. Maintenance of this software through revisions of the HMI software and even generations of the control and data acquisition systems was a nightmare.

OPC/DA solves these problems. HMI suppliers needs to write and support only one interface/driver – the OPC Client unique to their HMI software. Each control or data acquisition supplier needs only to support one interface – the OPC Server unique to their own system. Communications between client and server uses Microsoft Windows COM (Component Object Model) when both client and server are located in the same computer, or DCOM (Distributed COM) when client and server are located in different computers on the same network.

The data model of OPC/DA is based on the needs of the early systems, which is modeled as a generic programmable controller in which all data is mapped into "registers". OPC/DA defines a set of input and output registers with a numbering scheme, and all control and data acquisition suppliers translate from their internal memory to the standardized mapping of OPC/DA. HMI suppliers then only need to populate their real-time database using the OPC/DA data structure produced by the controller or data acquisition system's OPC Server.

While OPC/DA works well for PLCs (Programmable Logic Controllers) in which the I/O and register mapping is simple, it does not provide enough information for process control, variable speed drive control, or many machine controllers. Controllers for these applications operate with algorithms to compute control valve position, torque requirements, or vector positioning. Each of these algorithms has accessible parameters, or attributes. Mapping of attributes and other variables of complex controllers to the OPC/DA data model has become a burden for each of the controller suppliers, and the HMI vendors as well since all of these mappings are unique to the controller or data acquisition system. This is the problem addressed by OPC/DX – to enable a standard way to address attributes of objects in a remote system, the primary problem faced by the user to achieve interoperability of systems.

Foundation Fieldbus calls its objects "function blocks" and defines a rigid set of rules to access the "parameters" or attributes of those objects in its User Layer. For the set of function blocks defined in the Foundation Fieldbus User Layer, there is a well-defined set of DDs (Device Descriptions) available in the Foundation Fieldbus specifications allowing the user to communicate with these function blocks. For each non-standard function block prepared by suppliers, a DD file is supplied so that the user may address the attributes of those function blocks.

Similarly, Profibus defines a set of "profiles" for each major application. Within each application, the ability to perform algorithmic transformations of the data is defined by the supplier who delivers a GSD file containing the attributes of the algorithmic object allowing the user to access that data.

ControlNet, DeviceNet, and EtherNet/IP also define algorithmic transformations of data using sets of EDSs (Electronic Data Sheets) unique to each device. Users may access the attributes of devices using the definitions in these EDSs.

The OPC/DX vision is to replace these diverse interfaces with a DX server allowing device data to be distributed to DX clients. DX technology will be implemented as an extension to OPC/DA but allowing symbolic access to the

DX attributes defined in a set of configuration tables in each device. The format of these configuration tables and the specifications of DX itself are expected in the next few months. Once DX is available and widely supported, it should be possible for control system software with an OPC/DX client to access data on any of the supported networks with an OPC/DX server.

CMC Associates focuses on product and market strategies to help suppliers succeed in their markets by fully understanding future technology trends, end user needs, and their preferences. We follow the news events and prepare news analyses such as this to raise awareness of market and technology trends for our clients. No inside knowledge was used to prepare this document, and it has not been funded by anyone. If you think that this type of analysis can be of value to your company, supplier or end user, please contact the author, Dick Caro at Dick@CMC.us.

Tel: +1.978.635.9449 Mobile: +1.978.764.4728 Fax: +1.978.246.1270
Web: http://www.CMC.us E-mail: info@CMC.us