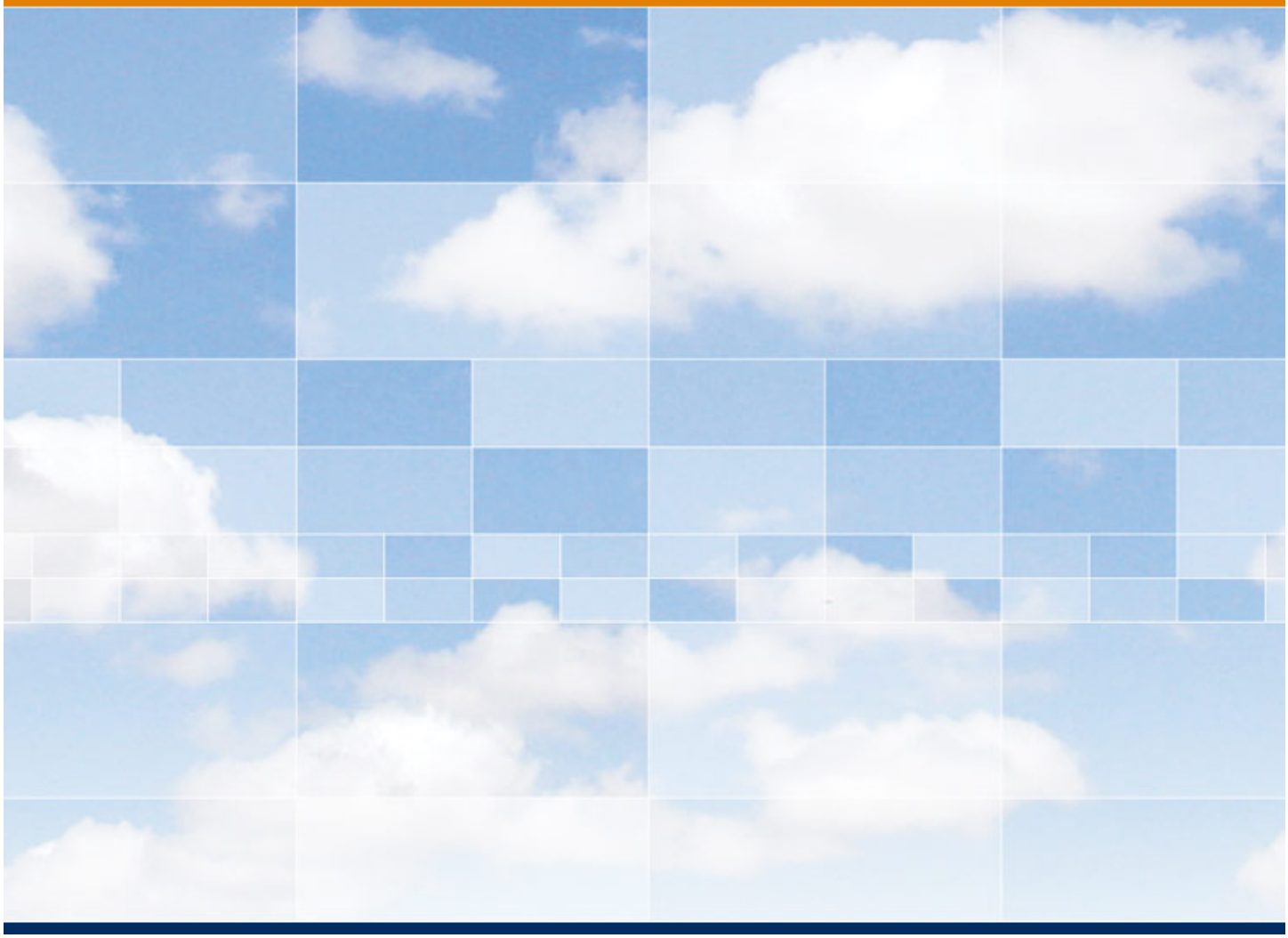
MatrikonOPC



OPC UA - How Deep Does Interface Standardization Go?

Eric Murphy
MatrikonOPC
July 2006

OPC UA - How Deep Does Interface Standardization Go?

With any widely implemented technology, come discussions about its limitations or perceived shortcomings. OPC is no exception. With the recent release of the OPC UA (Unified Architecture), there is a lot of speculation about whether or not it actually solves existing OPC issues.

Prior to OPC UA, the most common complaint has been that existing OPC standards are primarily COM based. OPC UA is a service based, cross platform solution, and no longer so Microsoft centered. This issue aside, the majority of other complaints focus on the specifications not going far enough in their scope of standardization. There are criticisms that OPC does not do enough to mandate security, configuration, and providing a unified address space or defined item 'mapping'. If you consider OPC as the standard for real time data communication, regardless of the data source, it raises an interesting question. Where does the line get crossed from a general interface specification that is open, interoperable and flexible without sacrificing usability, to one that is specialized, rigidly defined and highly integrated?

Most would agree that the existing OPC standards err on the side of caution, and focus specifications at the interface level. If you strip an OPC server to its essence; it represents access to a data source, presents an address space that defines the available data items, provides primitive information about each data item (identifier, data type, accessibility) and has the ability to read and update the value of each data item. OPC does not mandate identifier naming conventions, configuration parameters, and rules for federating applications or detailed security models. However the specifications do not preclude these things either, rather they leave these decisions to the OPC product implementation. Many would argue that it is this simplicity that is one of OPC's strengths, and contribute to its wide adoption. An OPC server data source could be a PLC, DCS, relational database, XML document, Web page, RFID tag, HVAC controller or one of many other possibilities. These have ranging semantics, use cases, and accepted implementation conventions, yet today, all communicate using OPC, despite the things left to implementation. So what incentive is there for vendors to implement features that are not mandated?

There is an argument that truly 'open standards' do not evolve because vendors wish to maintain their proprietary edge, and as a result only implement the base open requirements. However, relying on proprietary knowledge to maintain and grow market share is a short term vision. As users attempt to balance the comfort of a single source supplier with the desire for flexibility, innovation and competitive pricing, they inevitably migrate to open solutions that give them options where they need them. OPC is supported by many competing vendors. Each vendor's competitive edge comes from how features beyond the base specifications are implemented in their product offerings. There is much that specification can not dictate, and is governed by how the implementation is done. There are many excellent OPC products on the market that have implemented these aspects of interfacing very well. Unfortunately, the end user can not rely on them being in *every* OPC product they purchase. Although they have standard based communication this does not necessarily mean they have a *truly* interoperable system. Additional standardization at the system or enterprise integration level is required.

To achieve this vision of OPC across the enterprise, the OPC UA specifications address multiple aspects of interfacing, including a complex data model and unified information semantics, security and redundancy. So, how can a specification mandate detail standards for all these things, regardless of application, data source or use case? Realistically, it can't. The best it can do is provide the base common denominator of these aspects, but more importantly provide a *standard* way to extend the information model and programmatically discover the individual levels of support. This is what OPC UA is designed to accomplish. The result is a specification that is more comprehensive than just the interface details, such that users can expect base level support

for elements like security, redundancy, namespace navigation and richer information models. However, innovation in the implementation will still be a distinguishing factor in OPC UA products, especially in those designed to enhance or federate existing OPC installations. Given this, OPC UA also offers a more comprehensive certification process to ensure the interoperability between the varying degrees of product implementation.

The OPC community cannot dictate the detailed information model for every application. To achieve the reality of *complete* system interoperability, even using an interface specification with flexible query mechanisms, users in the same business space need to agree on some common information semantics. OPC UA has provided the forum for this in the kind of Access Type specifications and industry specific Companion specifications. Collaborations with groups such as EDDL, MIMOSA, ISA and others are a good initial start. The more standards bodies or user groups that contribute UA information models, the wider and more complete the specification adoption will be. OPC UA offers as much as any specification can, without crossing the line to customization. However, regardless of how detailed a standard is, the implementation is what will differentiate products.

To realize the benefits of OPC UA, ensure your issues are known, and understand how OPC UA addresses them, be sure to talk to your OPC vendor, join an OPC user group, or participate in the OPC Foundation.

Eric Murphy, Advanced Architecture System Design Engineer, MatrikonOPC

Eric Murphy, BSc, PEng (Alberta), Eric is a Chemical Engineer with a Process Control specialization and an OPC expert. Eric has been a part of the OPC community since its early beginnings in the mid-1990s. Eric is heavily involved with the OPC Foundation and currently acts as the chair for the OPC Historical Data Access (HDA) working group. Eric is also a member of the OPC Technical Steering Committee (TSC) and an active member of the OPC Unified Architecture (UA) working group.

Visit Eric at his Blog the OPC Exchange **http://blog.matrikonopc.com/** to follow the latest trends and discussions about OPC technology, or visit **http://www.matrikonopc.com** for free downloads.