First published in Finnish Society of Automation, Biannual seminar no. XX, 2013

OPC UA Connects your Systems
-
Top 10 reasons why to choose OPC UA over OPC

Andreas Frejborg*, Martti Ojala*, Lauri Haapanen*, Otso Palonen, Jouni Aro****

** Neste Jacobs Oy, Automation Technology, PO Box 310, FI-06101 Porvoo, Finland,*
(tel. 050 458 6712, e-mail: andreas.frejborg@nestejacobs.com, www.nestejacobs.com)
***Prosys PMS Ltd, Tekniikantie 14, 02150 Espoo, Finland*
(tel. 050 410 2500, e-mail: otso.palonen@prosys.fi , www.prosysopc.com)

KEYWORDS: OPC UA, IEC62541, information security, information integration, platform independent

ABSTRACT

This paper discusses the requirements set by the present-day industry on information integration and information security and presents successful solutions based on OPC UA technology. The solutions described are snippets from real-life applications in different process industry installations.

OPC UA, the prominent successor to classic OPC, started its journey in 2004. First industry implementations were available in 2008. Since 2011 it has successfully carried the IEC standard code IEC 62541 [IEC]. Hence, a lot of effort has and will be made by the OPC Foundation [OPC Foundation], solution vendors and makers of development toolkits to assure seamless functionality and efficient communications between different OPC UA solution providers.

OPC UA forces, in comparison to what classic OPC has to offer, solution providers to deliver solutions that are 1) platform independent, 2) incorporates enhanced information security, 3) create real integration from plant-floor to executive-floor, 4) bases on IEC standard, 5) builds on a simplified architecture, 6)bases on a clear, demanding but not technically restricted specification, 7) comprises of a large amount of domain specific add-ons, 8) are scalable, 9) are future-proof and 10) are deployed easily.

This paper describes the possibilities of OPC UA through several example applications. The examples are real-life applications from several process industry domains, ranging from food industry to oil refining and petrochemical industry. Moreover, the paper illustrates the proper system structure for industry standard applications that incorporates future needs.

OPC UA is already the main choice for connecting systems together. It will clearly stand as the backbone for inter-communication between any kinds of automation systems for next decades.

1. PREFACE

OPC UA is by nature platform independent and can, hence, be deployed on different operating systems and environments, from traditional Windows-based PCs to Linux systems and mobile platforms. Platform independence enables wider adoption and more flexible interoperation, as the need for special integration PC computers no longer exists.

OPC UA uses the latest IT security methods and standards to secure data. Enhanced information security through authentication of software and encryption of data are now critical requirements in process control, and OPC UA has those features built-in to the specification. This means in practice that it fits well in the existing corporate IT security infrastructure.

The flexibility of communication achieved with OPC UA means that it is suited to a variety of communication scenarios. All from shop floor to executive level systems. Wider OPC UA adoption will also lead to new categories of software supporting it, widening the scope of applications that can be connected to each other. Integration is easier and more efficient.

OPC UA has been standardized by IEC as IEC 62541 in 2011, ensuring a high level of rigor in the specification and boosting adoption as the confidence in the specification is higher. Furthermore, OPC UA takes into account the different needs of specific industries with its domain specific companion specifications and allows using the power of the extensible information modelling framework to define new industry specific information models.

Integrating OPC UA into an application is relatively straightforward, as in comparison to classic OPC, OPC UA builds on a simplified architecture. As more features are required, the same basic building blocks can be used to

create new functionality without starting from scratch. The specification enables higher level software development kits (SDK) that can also be tested for interoperability. As it is based on a clear, demanding but not technically restricted specification, OPC UA is designed to restrict the possibilities of applications as little as possible. Using the information modelling techniques standardized in the specification, specific application data models can be represented.

OPC UA is very scalable for different environments and use cases. Obviously the full power of the specification is not necessary in smaller applications or in those where resources are limited. OPC UA takes this into account and allows exposing different capabilities in a standard way. OPC UA prepares for evolving needs and technologies by being a modular and extensible specification. For example, new communication standards can be added to UA specifications as they become available. OPC UA is straightforward to deploy as it does not rely on any proprietary security or configuration methods. Network configuration is also done using general IT conventions, so configuring OPC UA in secure networks is much simpler than classic OPC.

2. 10 TOP REASONS

2.1. Platform independent

OPC UA is designed to be platform independent. This clearly separates it from classic OPC, where Windows-specific technologies were used. Today, an increasing percentage of computers are built using an operating system other than Windows. In IT servers Linux has long been the dominant OS [NetMarketShare] and with the emergence of mobile devices such as smart phones and tablets the client end is also changing to non-Windows operating systems. In this landscape, the platform independence of OPC UA simplifies and enables new types of solutions.

Web sites are commonly running on Linux-based operating systems and increasingly they also use real time data from automation systems. In this kind of situation it is very beneficial to be able to connect to a data source directly using OPC UA, instead of using a Windows-based PC computer as a mediator between the web server and the actual data source. This benefit will get further leverage once JavaScript-based OPC UA clients become available and direct connections from web browsers to OPC UA servers become possible. A JavaScript-based OPC UA stack has been developed at the University of Dresden, but is still in development. Some preliminary results are publicly available in a paper by Hennig, Braune and Damm [Henning et al.].

But platforms are not just about the operating system, there are also deeper differences between platforms. OPC UA can also run on devices that do not have the traditional PC processor architecture (x86 and its relatives), but also on embedded ARM devices and even single-purpose hardware. This will simplify tasks such as data collection, as well as make them cheaper since a full PC installation is not necessary to get OPC UA connectivity to data. This is a clear difference with classic OPC, since it always requires a windows-based PC which also complicates installation and maintenance, and adds a point of failure. System architecture is simplified since all levels and devices can use the same communication protocol.

Platform independence also enabled new types of human-man interface (HMI) solutions, such as tablet interfaces or specialized apps where website-based mobile solutions are not the best choice. Several companies have developed mobile OPC UA sample clients [Android1 & Android2]. One example of a proposed use case is real data integration into electronic plant start-up manuals. In such a solution instructions for maintenance could be accompanied by real-time data of the state of the plant, aiding in maintenance work.

2.2. Incorporates enhanced information security

OPC UA is enabling full security features built-in to the protocol itself, OPC UA specification part 2 and 6 [OPC UA Spec]. The OPC UA binary communication protocol enables full encryption and message signing, which are based on asymmetric X.509 key exchange and an extendable symmetric data encryption. The protocol defines several security policy alternatives, which the application administrator may choose from according to the security requirements of the system where the applications are used. AES encryption (128-bit and 256-bit) is currently available, and must be supported by all standard OPC UA applications [OPC UA Spec].

Alternatively, if the applications are configured to use Simple Object Access Protocol (SOAP) or Hypertext transfer Protocol Secure (HTTPS) messaging, respective security features are available using WS Security or standard SSL Security implementations, respectively.

In addition to the message security, OPC UA also defines how applications authorize access to each other. Basic username and password authentication or alternatively X.509 certificates are also available for user authentication. So the protocol also provides the ability to limit available data and operations based on the application and user that are accessing these.

Because the security features are built-in the protocol, they will always be available when connecting applications in larger networks. This makes it possible to securely communicate via public networks, even without a Virtual Private Network (VPN), which typically offers similar security on top of any communications.

The used security level is always configurable, and can be left out, if necessary. For example, OPC UA implementations in embedded devices are not required to enable security, since it typically goes beyond the capabilities of such devices.

The OPC UA security implementation included in the standard implementations is very efficient and typically does not cause any noticeable extra load or speed degradation.

Comparing to the old DCOM based OPC protocols, which only relied on the hard-to-configure DCOM security, which does not include any encryption capabilities; OPC UA now provides a top-level secure ground on which to base industrial communications.

2.3. Create real integration from plant-floor to executive-floor

Information integration between systems at different levels of operation (Figure 1) has been a challenge during the past decades. With OPC UA (Figure 2) a common information carrier is now available.

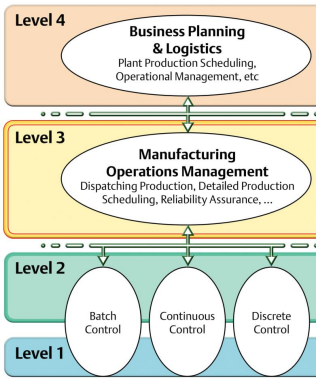

Figure 1, Four levels of ISA-95 [ISA-95]

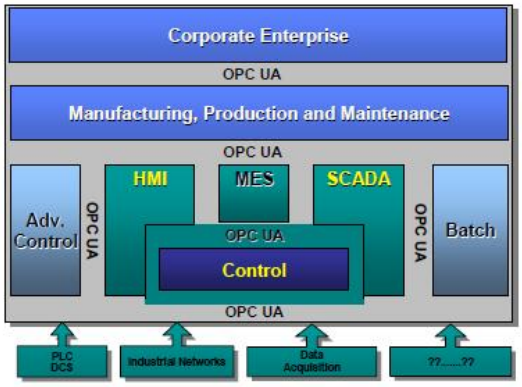

Figure 2, Where OPC UA can be applied, [OPC Foundation]

OPC UA enables together with secure communication and domain specific information models both non-restricted and seamless integration, even between legacy systems (Figure 4). Moreover, due to the fact that the communication can be made secure and narrowed down to a single port it is possible to open some parts of the automation network to direct access from the upper levels of ISA-95.

2.4. Bases on IEC standard

OPC UA is standardized as IEC 62541 [IEC 62541 Specification], which means the protocol is available for examination by anyone. It has already been validated by a neutral organization to fulfil the requirements of a standard. In practice this is accomplished by defining the protocol so that it is not tied to any specific technique, such as DCOM. Instead it defines alternative transport protocols, which can be used to deliver the messages. And in future, new alternatives can be easily added in the specification, without modifications to other parts of the protocol.

Existing standards are also used to extend the information model parts of OPC UA. The base information model includes building blocks for modelling various semantic models. And several existing information models, such as ISA95, BACnet, ISO 61970 and 61850 have or are being added to the OPC UA specification as extensions [OPC UA for ISA-95]. This makes it possible to use OPC UA as a communication protocol for data and information of various systems that use standard information models. This obviously makes it much easier to connect applications from different vendors and different systems together, as both the communication and information models are already standardized.

2.5. Builds on a simplified architecture

Integrating OPC UA into an application is relatively straightforward, compared to classic OPC. When more features are required, the same basic building blocks can be used to create new functionality without starting from scratch. Whereas with classic OPC a new server or client had to be developed for each facet of the protocol (DA, AE, HDA), in OPC UA these all are unified so that OPC UA has to be built in just once, and after that adding new functionalities like accessing history in addition to current values is relatively straightforward.

In new OPC UA installations it is typical to start from classic OPC functionality, which is basic data transfer and possibly eventing functions. Once this basic infrastructure is in place, new use cases such as controlling via OPC UA Methods are identified as a good way to extend the functionality. Since all services in OPC UA use the same communication layer, enabling new features is easy. Adding new features one by one has proven in practice a good way to shift to OPC UA from other communication solutions. A proof-of-concept solution can be expanded feature by feature, reducing re-design and re-development of the system as requirements grow. In OPC UA projects, this piecemeal development strategy has proven to be a good complement to iterative software development methods, where only the necessary functionality is implemented at each step.

Since the architecture is well-defined, higher level software development tools have been developed for it. This cuts down the time required to integrate OPC UA into a product. Higher-level SDKs can also help developers use the more advanced features of OPC UA.

2.6. Comprises of a large amount of domain specific add-ons

The flexibility of OPC UA standard comes from the selected information model which provides a framework for creating and exposing vendor defined information in a standard way. This flexibility extends the usage of OPC UA standard from a mere communication protocol to a wide selection of communication solutions on different application domain specific use cases. Several collaboration projects have been taken place to standardize OPC UA information models for the different application areas, for example:

- OPC UA for Devices (DI) 1.0 released [OPC UA for Devices], preparation for IEC standardization is on-going
- OPC UA for ISA-95 Common Object Model RC 1.00.00 released [OPC UA for ISA-95]
- Field Device Integration (FDI), process of IEC specification release is on-going
- OPC UA for Analyser Devices (ADI) 1.0 released [OPC UA for Analyser Devices]
- OPC UA for IEC 61131-3, PLCOpen 1.0 released [OPC UA For IEC61131-3]

The OPC UA information model allows servers to provide type definitions for objects and their components. The information model supports Object-Oriented Programming (OOP) principles, such as data abstraction, encapsulation, polymorphism and inheritance and thus makes it possible for the vendors to map and publish complex data objects to the OPC UA servers address space.

As an example, a type defined information model for a controller is presented (see Figure 3). It is obvious that a well-defined information model saves time and can, for instance, integrate systems together that when using classic OPC need tailor-made cross-reference tables for connecting data points in different systems.

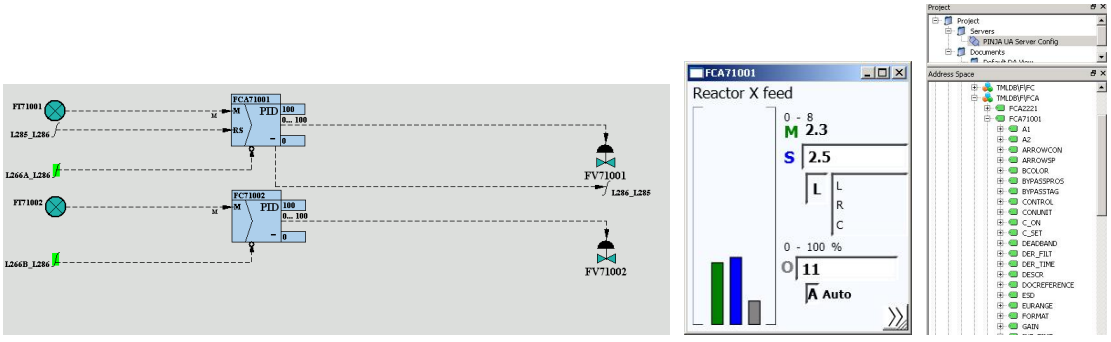

Figure 3, PID controller shown in DCS schematics (left), operating HMI (center) and in a OPC UA tree browser view (right)

2.7. Bases on a clear, demanding but not technically restricted specification

OPC UA is designed to restrict applications as little as possible. While establishing a basic connection and reading some values is straightforward, often OPC UA applications deal with more complex data and tasks. This means that extensibility has been a major design consideration for the specification. OPC UA defines rules for extending the base specification while remaining interoperable. Using the information modelling techniques standardized in the specification, application, vendor or business area specific data models can be represented. This means that the basic data types, device types, alarms etc. can all have additions specified in a standard way, so that all applications remain interoperable with each other, while more data is provided to those applications that are aware of the extensions.

Since the protocol is clearly and strictly defined, it has been possible to create standardized tests and a certification program for applications [OPC Foundation Compliance & Certification]. There is an automatic compliance test tool available from OPC Foundation that can be used to OPC UA products. Once the development of a product reaches a mature state, it can be tested using the automated test tools. If proof of compliance is required, independent test labs provide test services for applications. Overall, the strict testing procedures are meant to ensure that all OPC UA applications interoperate together in practice as well as in theory. This was one of the major downfalls of classic OPC, where implementations of the same specification would vary widely, especially in the less common specifications such as OPC AE and OPC HDA.

2.8. Enables scalable solutions

Where platform independence is about the freedom of deployment, scalability is about the freedom of application scope. These two are closely related, since applications running on limited hardware tend to be more modest in their functionality. Classic OPC applications usually run on systems with desktop hardware. OPC UA applications, on the other hand, can be very different in their scope and functionality, from small embedded servers in intelligent field sensors to large-scale computer systems. Not all OPC UA applications are required to have the same capabilities and the level of support for different functionalities can be expressed to other OPC UA applications [Mahnke et al.]. For example, an OPC UA server on a small embedded device would not have the capability to keep track of changes to values or provide encryption. Meanwhile, machines with abundant resources can provide processor-intensive services such as database queries for client users. These differences can be expressed in the software certificates that applications use, though this capability is still work under progress and not fully specified in the released OPC UA specifications.

In a petrochemical process simulator solution for example, calculations run on computers that have a lot of processing power, while data collection is performed using embedded servers on PLCs with no extra resources available. Different layers of the solution can communicate with each other using a common protocol, even though their system capabilities vary widely

2.9. Solutions that are future-proof

As OPC UA became an IEC standard the leap from “proprietary” classic OPC, where vendors opted to create solutions compatible with their own applications, to standardized communication is noticeable. This encourages end users to demand system vendors and system integrators to switch to OPC UA. Currently all main vendors in the hydrocarbon process industry have an OPC UA solution, or will have a release within a few years.

Moreover the OPCUA standard gives the possibility for the applications to follow the world and pick up new and evolving techniques as they evolve. The standard itself is written in that way that it do not finger point on any platform nor technique to be used as a must. When looking back at classic OPC DA, this was not the case.

When building solutions for the hydrocarbon process industry one must keep in mind the lifecycle of the solution. A normal lifecycle for many of the computer systems built for analysers, process control and advisory calculations is ranging from 12 to 20 years. Therefore it is obvious that when selecting a new technology is must be convincing and have a proven track record together with a wide enough solution portfolio.

2.10. Are deployed easily

As a tenth and final reason, no one wants to do Distributed COM anymore. COM/DCOM used with classic OPC often had a serious impact on the security aspects. Usually, when creating communication between products from different vendors it was more the obvious that security was deemed to be set to merely nothing.

As the solutions are built with modern tools and using programming languages/environments of today, the solutions created are not that tied nor bound to operating system and its specific libraries. Instead the solutions are sandboxed; carrying with them the things they need for operation. For instance, .NET framework is designed so that applications running on different frameworks can run at the same time at the same computer without interfering with each other.

Nowadays, the life of OPC based communication has become increasingly harder, as systems are more and more networked together but at the same time secured from each other with hard-core firewalls, and/or techniques like VLAN (Virtual LANs) and DMZ’s (Demilitarized Zones). Classic OPC with its DCOM protocol restricts the computer-to-computer communication to a set of general and standards ports, like TCP port 135. As this is the main port for many serious network attacks, it is usually prohibited between any segments involving both automation and the office network. One can of course use tunnelling software, but with the availability needs of the hydrocarbon process industry, for example, it is not that convenient solution. Instead - by using OPC UA - with it is own protocol and hence, fully definable ports, it is possible to create as secure solutions as the deployment demands.

3. SUMMARY

With current classic OPC based installations at their mind, customers might ask themselves – is this worth the effort? For sure it is! As OPC UA is backwards compatible it is fairly easy to switch part of the system to use OPC UA technology, including security and information model features (Figure 4).

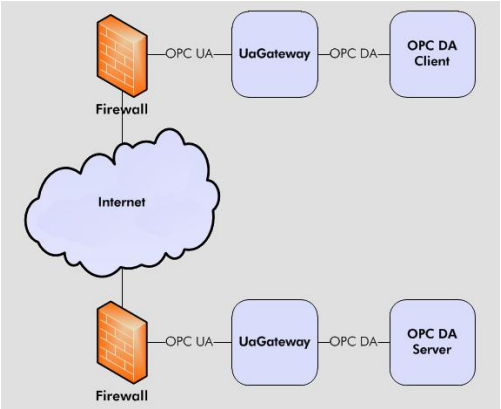

Figure 4, With OPC UA classic OPC can simply be bridged over firewalls

OPC UA is very scalable for different environments and use cases. Obviously the full power of the specification is not necessary in smaller applications or in those where resources are limited. OPC UA takes this into account and allows exposing different capabilities in a standard way. OPC UA prepares for evolving needs and technologies by being a modular and extensible specification. For example, new communication standards can be added to UA specifications as they become available. OPC UA is straightforward to deploy as it does not rely on any proprietary security or configuration methods. Network configuration is also done using general IT conventions, so configuring OPC UA in secure networks is much simpler than classic OPC.

To conclude, OPC UA is not only the communication technology of today, it is the solution.

REFERENCES

[Henning et al.] Hennig S., Braune A., Damm M.: JasUA: A JavaScript Stack enabling web browsers to support OPC Unified Architecture's Binary mapping natively, 2010 IEEE Conference on Emerging Technologies and Factory Automation (ETFA), 13-16 Sept. 2010. DOI: 10.1109/ETFA.2010.5641005

[Mahnke et al.] Mahnke W., Leitner S., Damm M.: OPC Unified Architecture. Springer 2009. ISBN 978-3-540-68898-3

[Android1] Prosys OPC UA Android Client, available at http://www.prosysopc.com/opc-ua-android-client.php

[Android2] UaExpertMobile for Android, available at http://www.unified-automation.com/home.htm

[OPC Foundation Compliance & Certification] OPC Foundation Compliance & Certification website, http://www.opcfoundation.org/Default.aspx/Compliance-Certification/Compliance.asp?MID=Compliance

[OPC UA Spec] OPC UA Specification, Parts 1-13., version 1.02, http://www.opcfoundation.org/

[IEC 62541 Specification] IEC 62541 Specification, Parts 1-13, version 1.02, IEC 62541-[1-10] ed1.0, http://iec.ch

[OPC UA for Devices] OPC UA For Devices 1.00 Companion Specification, http://www.opcfoundation.org/

[OPC UA for Analyser Devices] OPC UA For Analyser Devices 1.00 Companion Specification, http://www.opcfoundation.org/

[OPC UA For IEC61131-3] OPC UA For IEC 61131-3 1.00 Companion Specification, http://www.opcfoundation.org/

[OPC UA for ISA-95] OPC UA for ISA-95 Common Object Model 1.00, http://www.opcfoundation.org/

[IEC] International Electrotechnical Commission, http://www.iec.ch/

[OPC Foundation] OPC Foundation, http://www.opcfoundation.org/

[NetMarketShare] NETMARKETSHARE, Market share statistics for Internet Technologies, Trend of operating systems in use, http://www.netmarketshare.com/operating-system-market-share.aspx?qprid=9&qpcustomb=0

[PLCopen] PLCopen – for efficiency in automation, http://www.plcopen.org/