Real-time publishing with response:

*Pub/Sub should not be just another fieldbus.*

Date of publication: 26.11.19

Version 1.1

Authors and Supporters:

Paulo Zanini (Weiss GmbH), Wolfgang Doll (3S-Smart Software Solutions GmbH), Heinrich Munz (Munz Endeavors GmbH), Wolfgang Höglinger (Engel Austria GmbH), Gerald Schauer (elrest Automationssysteme GmbH), Johannes Hoos (Festo AG & Co. KG), Etienne Axmann (VDMA), André Mankowski (inIT - TH OWL), Philip Priss (inIT - TH OWL), Friedrich Durand (afag Automation AG), Suprateek Banerjee (VDMA), Stefan Profanter (fortiss), Mohit Agarwal (Atlas Copco)
Summary:

This paper deals with the question of how to implement an interoperable, object-oriented and powerful machine architecture for information exchange and control with OPC UA. The focus here is a lean, scalable and deterministic real-time communication on all hierarchical machine layers - from machine-to-machine down to sensors & actuators. In addition to possible use cases, it will also be shown how a solution approach can look like.

Contents

1. Industry of the future: what are the challenges? .............................................................. 3
2. What does OPC UA offer today? ......................................................................................... 5
3. Why is real-time publishing with response important? ...................................................... 6
4. Requirements & use cases ................................................................................................. 7
5. Proposed Solution ............................................................................................................. 10
6. Conclusion ......................................................................................................................... 12
7. FAQ ................................................................................................................................... 14
8. References .......................................................................................................................... 16
1. Industry of the future: what are the challenges?

The key development in Industry 4.0 is that we start to understand physical assets as digital objects on all hierarchical levels of machines and plants. This leads to seamless and complete object-based machine architectures and will be the enabler for the full range of I4.0 applications. OPC UA plays a decisive role in this context.

OPC UA is the Information and Communications Technology (ICT) standard for industrial automation. In November 2018, more than 20 of the world's leading automation suppliers actively committed themselves to OPC UA and TSN as part of the Field Level Communication initiative (FLC) of the OPC Foundation. The goal of this initiative is to deliver an open, cohesive approach to implement OPC UA including TSN and associated application profiles in automation controllers and field devices. This will advance the OPC Foundation providing vendor independent end-to-end interoperability into field level devices for all relevant industry automation use-cases.¹

The decision for OPC UA as new the ICT has been made. The challenge is now on to exploit the full potential of this new technology in the field level.

The conventional fieldbuses have been doing their job for over 40 years. The large number of existing non-interoperable fieldbuses, the technological discontinuities in the transition (from

The self-description of complex machines and components should be facilitated by the properties of object orientation. States and behavior are combined to objects and are accessible from the outside using methods (encapsulation, information hiding). Similar objects can share certain properties (inheritance). Different implementations can be exchanged via uniform interfaces (polymorphism). These principles of modern IT approaches are painfully lacking.

Over the years, PLCs and automation machines such as machining centers, injection molding machines, robots and controllers for processes like welding, gluing, laser cutting, etc. have become increasingly flexible and efficient. Restricted by the orientation to bits and bytes of fieldbus interfaces, much of their performance cannot be used at all.

Not surprisingly, programming manufacturing systems is often tedious and inefficient, as it lacks code reusability, self-explaining source code and shared libraries. The exploding costs of developing and maintaining modern manufacturing systems under circumstances resembling those of the 1970s led to the development of company-specific frameworks, guidelines and vendor-specific constraints, sacrificing portability and vendor independence.

The representation of physical assets enables component vendors to offer their full functionality into a manufacturer- and platform-neutral, universal and digital resource pool. This will allow machine builders and factory operators to focus their efforts on their actual business, which is making their manufacturing processes more efficient.
2. What does OPC UA offer today?

OPC is the interoperability standard for the secure and reliable exchange of data and information in the industrial automation space and in other industries. It is platform independent and ensures the seamless flow of information among devices from multiple vendors.

Some of the outstanding features of OPC UA are:

- The OPC UA information model describes objects, data fields, states of state machines and methods.
- **Methods** are excellent for determining the point in time for execution, transmission of parameter sets and result values in a consistent way.²
- **State machines** are a core feature in procedural programming languages and are essential in automation engineering. In OPC UA, state machines are a standard entity whose state changes are triggered by method calls and notified by outgoing events.
- In addition, OPC UA provides services such as querying the device information model, reading and writing of data elements, subscribing to change notifications and much more.

---

² OPC UA methods are comparable to Remote Procedure Calls (RPC), which have been used extensively in computer science.
OPC UA provides two different communication patterns for its one uniform information modeling technology:

1. **Client/Server (C/S)** is a **reliable point-to-point connection** where the OPC UA client can retrieve, change and process services and data from the OPC UA server.

2. **Publisher/Subscriber (Pub/Sub)** is a **cyclical, not confirmed, one-to-many** communication pattern. Communication errors must be detected and corrected on the software layers above the transport protocol, i.e. via the mechanism by which the sender cyclically repeats the messages with the updated content at short intervals. In this way, the receiver can recognize if messages have been lost, e.g. by means of a sequence counter.

For the operation of "real-time publishing with response" it is necessary to set up a point-to-point relationship between the transmitter and receiver via the Pub/Sub protocol. For this purpose, a publisher/subscriber pair can be assigned reciprocally to each node. With this configuration a confirmed communication between exactly two logical partners can be realized.

3. **Why is real-time publishing with response important?**

With the help of the Pub/Sub protocol a vendor independent communication between "large" and also "very small" devices is possible, even under the consideration of strict timing requirements.

---

**Figure 3: Components of an Object**
This is especially important because no additional semantic breaks between C/S and Pub/Sub use cases occur as a result. Services (e.g. method calls) should be available for the Pub/Sub protocol via the same API as the existing Client/Server oriented OPC UA services, just under strict timing constraints.

The principle of "real-time publishing with response" covers all expression possibilities of the OPC UA information models in the areas of automation technology, in which strict timing requirements must be met.

OPC UA has the potential to reach out even simple sensors and actuators, empowering those vendors to embed rich semantics, deliver companion specification compliance and a new dimension of value-adding. However, this is attainable only with minimal footprint.

**Therefore, the manufacturing automation ecosystem deserves the completion of the OPC UA specification covering real-time publishing with response (or method calls) over Pub/Sub.**

4. Requirements & use cases

**VDMA Plastics and Rubber Machinery Association**

In manual operation mode, complex production cells with several subsystems like machines, handling devices and peripheral components offer a lot of possibilities to move all the axes, such as molds, cores, ejectors, robot linear axes or hand axes. To improve workflows at the production floor all these functions should be available on a central operator panel.

Furthermore, the operator could use the robot panel to control the machine equipment or the machine panel to control the peripheral functionalities. Of course, the response time to come to a standstill should be as fast as possible, especially when hold-to-run-buttons are released.

Even if 99% of the method calls were fast enough, it is important to guarantee a worst-case response time. And as there is a lot of equipment in production cells, it is not practical to provide so many cyclic signals (Pub/Sub) for each functionality. So we end up in a modelling mess because real-time method calls are missing right now.

Pub/Sub method calls would enable the usage for synchronizing the production process between the different appliances. While standard methods lose cycle times, the only solution again are cyclic variables. Of course, methods with various arguments would provide a much greater flexibility. There are a lot of examples where methods with requested values like
exchange of the gripper of a robot with the requested gripper number, a conveyor with a
requested runtime or a tempering device with a requested temperature.

Even in the EUROMAP79 standard, robot-machine-interface of the robot have to provide
enable signals which have to be considered by the machine. Real-time methods in place of
cyclical signals would provide the required handshake much more easily.

Use case Safety

A special use case of network-wide method calls are functional safety applications due to the
possibility of atomically indivisible consistency of call and return variables.
The Joint Working Group of the Profinet International association together with the OPC
Foundation has recently released the “Safety over OPC UA Companion Specification”. 3
Most of the functional safety interaction means standardized in this specification are based on
OPC UA Client/Server methods because of the guaranteed consistency of OPC UA methods
but are not deterministic and real-time capable. If Pub/Sub methods as described and
requested in this white paper were available, safety over OPC UA would also immediately work
over Pub/Sub in deterministic real-time.

Remote Execution of PLCopen compliant Function Blocks

A PLCopen compliant function block (FB) is completely defined via its interface and its behavior
model. The behavior (states, methods, variables) of such function blocks can be automatically
mapped to an OPC UA information model (Companion Specification).
But it also works the other way around. The interface of a function block and its behavior model
can be automatically derived from an OPC UA information model 4. Normally, FBs are used
locally in a PLC but with the suggested mechanism, a FB can be torn apart between a so-
called Caller and a Callee via the network. This leads to a remote execution mechanism of the
FBs.
The source code of two function blocks, called proxy in the Caller and wrapper in the Callee,
can be generated automatically from the information model (see Figure 4).

3 https://opcfoundation.org/markets-collaboration/safety/
4 Please also see PLCopen OPC-UA Client for IEC61131-3
https://plcopen.org/system/files/downloads/plcopen_opcua_information_model_for_iec_6113
1-3_version_1.01.09.pdf
The proxy implements the interface of the original function block but uses a generic transport mechanism that forwards the input/output data and method calls to the "other side". The wrapper encapsulates the original function block, implements the "other side" of the transport mechanism, writes the input data to the original function block, calls the required methods to trigger the internal FB’s state machine and retrieves the corresponding output data. In such a way, function blocks can be executed transparently for the Caller, distributed over a network.5

An essential requirement for the generic transport mechanism is the exact adherence to specific timing limits. These requirements can be met very well with Pub/Sub, UADP and TSN.

![Diagram of Proxy and Wrapper](image)

**Figure 4: Proxy and Wrapper**

**VDMA Integrated Assembly Solutions: Skills**

In order to enable use cases such as autonomous production planning for individual products, flexible manufacturing design, online marketplaces for automation components, and cross-vendor application-specific interoperability is not enough. Interoperability must be obtained across manufacturing steps and across hierarchical layers (sensor/actuator, cell, machine, production line, factory) by means of Product-Process-Resource matching. A skill is the basic entity standardizing process parameters as capabilities in a way that machine operations and motion can be dynamically orchestrated by production line operators supported by problem-solving algorithms instead of being hard coded into PLC programs. The VDMA IAS Group provides a companion specification for skill-oriented control over OPC UA which serves as a reference model for application-specific process descriptions. This model defines a minimal status machine with its natural event-generating, minimal control methods,

---

5 Please also see SOA SPS [https://www.beckhoff.de/default.asp?press/pr1414.htm]
and placeholders for the exchange of input and feedback data. IAS skills require method calls with response and are meant to be used in real-time applications.\(^6\)

Figure 5: Skill orchestration

5. Proposed Solution

To achieve real-time communication between two OPC UA instances, we propose to use the OPC UA Publish Subscribe (Pub/Sub) mechanism which is standardized in Part 14 of the OPC UA specification. OPC UA Pub/Sub is based on UDP or directly on Layer 2 MAC protocol and supports various transportation types. In this paper we present an approach based on the binary encoding via UDP or MAC, called UADP.

The basis for applications based on UADP is the consistent transport of data structures. For real-time applications, the number of transmitted bytes and transmission time is limited, therefore data encoding and transmission via UADP must be as efficient as possible.

Encapsulated in the so-called network messages, one or more data set messages are transmitted cyclically according a global publishing interval. The content of these data sets is

\(^6\) See. Skill-based Engineering and Control on Field-Device-Level with OPC UA
configured by data set meta data records. The different configuration elements can be stored in the address space of an OPC UA server or are available in other forms like for example in form of a NodeSet2.xml file.

**Figure 6: OPC-UA Amendment 6, Chapter 2 - UADP Header Layout for cyclic data exchange**

A mechanism to implement “real-time publishing with response” could be based on the following parts:

- The provider of some services (the **Callee**) becomes the publisher of future communications about the execution status (response) of its actions.
- The client of a service provider (the **Caller**) becomes the publisher of future communications about the execution management (request) of its actions.
- Caller and callee will subscribe to each other's network messages.
- In addition to the data for selecting a desired service on callee site, a request message also transports some data that uniquely identifies the caller and some data that models the service specific parameter set.
• The response message indicates whether a particular caller is currently active for a particular service and which return value for the current service has just been calculated on the callee’s side.
• The connection between caller and callee is established as long as a special marker in the response message is equal to the related marker in the request message.

With these preconditions, a temporally limited, logical connection between caller and callee can be established. A callee can announce its existence and its possibilities by publishing appropriately prepared response messages. A caller can, after having determined that a certain callee is currently available for providing a service, establish a connection to the caller by sending an appropriately prepared request message. One of the next response messages from the callee then informs the caller about the success of its efforts.

6. Conclusion

The digital representation of assets is the very first step to achieve true Industry 4.0 solutions. With the described approach, the object-oriented architecture is also available in real-time communications. There will not be a technological break between data. There will not be a technological break between data accumulation and control of machines. Just as well as there will not be a technological break between real-time or non-real-time communication regarding the use of OPC UA Methods.

OPC UA methods for real-time applications in the field level offers completely new possibilities for solving automation tasks much more directly at the application level. It is the basis for future automation.

It is no longer necessary to map device functions to bits and bytes on fieldbuses and to create own access libraries, after consulting the corresponding bit tables from the documentation of the automation device. Instead, the device services can be called directly by the application programmer. The source code of the access libraries for the automation devices can be automatically generated by using the corresponding OPC UA device model. The automation device comes with its own drivers for control - even in source code.

"IT meets OT" becomes reality thanks to OPC UA and brings added value to OT.
We kindly invite all interested parties to promote our approach and to jointly develop solutions for tomorrow's automation industry.
7. FAQ

What advantages does the Pub/Sub protocol offer over client-server-based communication?

- No full-blown TCP/IP Stack is required (but possible additionally) for communication via the Pub/Sub protocol based on UADP. A much simpler UDP/IP or MAC implementation is enough.
- Unrelated publishers and subscribers usually do not know each other. Therefore, a larger number of subscribers do not further impact the respective publisher and its resources.
- (Embedded) Devices with restricted resources can easily be part of a communication network on basis of the Pub/Sub protocol over UADP because the technical footprint for the Pub/Sub functionality is small.
- With the help of TSN, bottlenecks for high priority real-time messages on the transmission medium are avoided and predefined transport capacities are used. This leads to a strongly deterministic timing behavior.

Why can't we do without TCP/IP based Client/Server services from OPC UA?

Certain services are in very good hands at a central point in the network. Residents' registration offices and key cabinets are a good example. The same applies to an OPC UA network. The administration of certificates, public keys and access points are also trustfully managed there within a server process.

Why is a completely different mechanism like Pub/Sub needed in addition to Client/Server-based communication?

The OPC UA Client/Server pattern is currently only specified on TCP/IP-based communication, while OPC UA Pub/Sub is specified on UDP/IP or MAC optionally under TSN constraints.

By using TCP/IP, all services based on it do not have a strictly deterministic timing behavior. However, this is necessary for industrial automation with hard real-time requirements.
TSN stands for Time Sensitive Networking and was designed to achieve strictly deterministic timing behavior in the sub-millisecond range, comparable to the performance of existing fieldbuses.

Why does the use of the UDP based QUIC protocol for OPC UA Client/Server not meet the requirements for hard real-time characteristics in the industrial environments?

QUIC is a transport protocol on top of UDP, developed by Google\(^7\). It uses mechanisms of TCP and introduces new features not used by other transport protocols. QUIC is optimized to be used for HTTP/2 connections and aims to reduce the end-to-end latency. QUIC works best (compared to TCP) for slow connections with high latency.

The QUIC protocol must fulfill the same requirements for error detection and correction as its predecessor TCP. In order to enable a reliable one-to-one communication connection, QUIC must therefore ensure the correct sequence of the packets and ensure that no packets are lost. By taking the necessary measures, the deterministic timing behavior is strongly affected.

The improvement of the TCP/IP based OPC UA communication using QUIC is certainly a good idea, but this measure does not solve the problems in the field of industrial automation with hard real-time requirements.

\(^7\) https://datatracker.ietf.org/wg/quic/charter/
8. References

Quic Working group: https://datatracker.ietf.org/wg/quic/charter/

OPC UA IAS Demonstrator: https://ias.vdma.org/opcua


OPC UA JWG Safety over OPC UA: https://opcfoundation.org/markets-collaboration/safety/

SoA SPS Beckhoff: https://www.beckhoff.de/default.asp?press/pr1414.htm

Skill-based Engineering and Control on Field-Device-Level with OPC UA (2019): Patrick Zimmermann et al. IEEE (link)