D. Hästbacka, P. Kannisto, and A. Kätkyniemi, "Interoperability of OPC UA PubSub with Existing Message Broker Integration Architectures", in 48th Annual Conference of the IEEE Industrial Electronics Society, 2022, accepted for publication.

IECON 2022, Brussels, Belgium - https://iecon2022.org

Interoperability of OPC UA PubSub with Existing Message Broker Integration Architectures

David Hästbacka, Petri Kannisto, Antti Kätkytniemi
Tampere University
Tampere, Finland
{david.hastbacka, petri.kannisto, antti.katkytniemi}@tuni.fi

***Abstract*—Interoperable communication technologies are of key importance in production systems with increasing needs for data in their adoption of data-driven methodologies and new, emerging applications. OPC UA PubSub defines an alternative to the traditional client-server communication with a publish-subscribe model for data to cater to scalability and data-driven cloud application needs. In this paper, the OPC UA PubSub model is compared to some other message broker and communication technologies and integrated with an existing message based integration model for evaluating the interoperability. A case example is presented where data payloads and information security practices are integrated using an adapter approach.**

***Index Terms*—OPC UA PubSub, Interoperability, Industry4.0**

I. Introduction

Industrial production systems are dependent on communication, and the need for data is increasing with data-driven operations and the adoption of artificial intelligence (AI) methods, e.g. to optimize operation and minimize use of resources. The need is evident within Industry 4.0 integrations and the implementation of cyber-physical systems (CPS) using Internet of Things (IoT), cloud computing and AI [1]–[3].

Communication technologies are of key importance in these systems of systems when both the amount of data and the number of data sources are on the rise. Ultimately, one would want plug and play of components [4] regarding both hardware and software. Integration from device to device and from edge to cloud is required when digitalizing operations. Doing this efficiently for software systems requires interoperability. Interoperability has multiple definitions, but this paper considers it the ability of entities to both communicate and act together, realizing loosely coupled integration [5]. Depending on the definition, interoperability can occur in multiple levels, such as technical and semantic (the focus of this paper) as well as organizational and legal [6]. For interoperability and Industry 4.0, the development and adoption of standardized data models and services as well as adoption of industrial communication standards are seen as high-priority challenges and actions [3].

Industrial process monitoring and control systems may have a lifetime of decades, and modernization often involves communicating with legacy systems as well. Development and standardization within Open Platform Communications (OPC) has been one of the most significant efforts in advancing communication practices to integrate the control systems, devices and monitoring applications of various vendors. The Unified Architecture (UA) standard specifies platform-independent communication protocols, information models and security practices in order to have interoperable communication between the systems and components of various manufacturers [7]. One of its newer additions, the PubSub specification of Part 14 [8], defines an alternative to the traditional client-server model with a publish-subscribe (PubSub) model to cater for better scalability in data-driven applications.

Outside of industrial and high-dependability applications, these message-oriented communication models, decoupling the producers and consumers, e.g., using a publish-subscribe model and/or a message broker, have been used for long. The typical use cases include IoT applications as well as Internet and cloud-based systems composed of multiple components, such as microservices processing data. The examples of broker-based technologies include Advanced Message Queueing Protocol (AMQP) and Message Queuing Telemetry Transport (MQTT), for which OPC UA PubSub also defines support. Despite a similar base communication technology, the integration of existing applications with the new OPC UA PubSub communication is not as straightforward as one might think.

In this paper, we discuss how OPC UA PubSub opens up OPC UA to a broader set of application integrations and consider the integration with existing broker-based communication architectures. A case study example is presented where an integration in both ways is implemented between OPC UA PubSub and an existing AMQP message broker architecture for a plant-wide process control application. The research questions can be summarised as follows:

- How does OPC UA PubSub stack up against other message-broker-based integration models?
- Can OPC UA PubSub be adopted for and integrated with an existing message based integration architecture to further enhance its interoperability?

Section II presents related work and background to the need for new integrations. Following is an introduction to OPC UA PubSub in Section III and a comparison with message-oriented communication models, including AMQP, MQTT and User Datagram Protocol (UDP) in Section IV. Section V presents

This work received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 723661 as well as from Academy of Finland under grant id 322676 "Distributed Management of Electricity System" (DisMa).

our example case and implementation with OPC UA PubSub and AMQP communication. The results and discussion from our comparison and example case are provided in Section VI before concluding the paper in Section VII.

II. Data integration in process industry and related research

Industrial processes may contain multiple sub-processes and chained entities along a supply chain. Linked processes may cause bottlenecks, and it may be necessary to coordinate distributed production. For example, some local optimization might affect a later stage of processing, e.g. use of raw materials or resources, and impact the global optimum in terms of used resources or environmental impact.

To improve interoperability, the communicated data should adhere to information models based on open standards. Industry-accepted standards are among the key factors of successfully implementing Industry 4.0 [3], [9]. A proposed conceptual framework for data-driven smart manufacturing implies the exploitation of Internet and cloud technologies as data is collected and processed [10]. In today's globally networked processes, Internet connectivity is often mandatory, even for components on lower levels typically not exposed to the Internet. This sets requirements on information security but also on cyber security [11] associated with a physical risk.

There are only a few related studies involving the new OPC UA PubSub. In previous research, an OPC-UA-PubSub- and MQTT-based configuration tool and a PubSub implementation that can be integrated into other OPC UA applications have been developed [12]. The OPC UA MQTT tool introduced enables the flexible configuration of publishers and subscribers [13]. An interesting development is the OPC UA synergy platform [14] that includes a PubSub server and is developed with the ISO/IEC 30131 IoT reference architecture in mind. In their work they consider HTTP, AMQP and UDP communication for interoperability with different needs. The applicability of OPC UA PubSub on factory automation use cases has been evaluated by [15] outlining different combinations of OPC UA communication to assess applicability.

OPC UA PubSub also targets to real-time capabilities with Time-Sensitive Networking (TSN). In [16], an approach is proposed to combine non-real-time OPC UA servers with real-time OPC UA PubSub without the loss of real-time guarantees for the publisher, and an open source implementation is presented. For field level communication requiring hard real-time interaction, a simulation model based on TSN and OPC UA PubSub has been proposed by [17]. [18] vision a skill-based, vendor-independent plug-and-produce architecture based on OPC PubSub over TSN.

For traditional OPC UA, [19] propose a mechanism for the auto-configuration of OPC UA systems. They claim that the method allows the automatic configuration and deployment of an OPC UA server only from the information provided by industrial devices giving OPC UA servers the ability to become first-class plug and play systems thus improving flexibility, adaptability, and scalability.

Middleware solutions for industrial environments have been studied with the aim to develop a common reference architecture for agile manufacturing control systems [20], [21]. Another interesting approach for interoperable systems of systems is Arrowhead Framework [22]. The Arrowhead approach provides primarily the service infrastructure including discovery, orchestration, and authorization but does not specify how information should be conveyed between application systems. In comparison, OPC UA has a strong information modeling aspect to it in addition to well-defined interfaces, which also applies to a large extent to the PubSub part.

III. OPC UA PubSub spanning beyond industrial control systems

OPC UA [7] is a platform-independent standard for systems and devices to communicate. OPC UA specifies both protocols and information model with data semantics. A traditional OPC UA application consists of a server with an address space for the client to browse programmatically. The address space is made up of a hierarchy of nodes with references between each other. When the nodes and the references make use of a standard OPC UA information model, e.g. OPC UA for Devices [23] or OPC UA for Asset Administration Shell (AAS) [24], integration into different systems is straightforward. OPC UA also embeds models for information security including certificate-based authentication and authorization as well as the possibility to sign and encrypt the exchange.

The PubSub specification [8] changes the communication model as there is no longer any client or server role. In fact, any application can be a PubSub participant without being an OPC UA client or a server. Instead, participants are either publishers or subscribers without knowledge of who, if any, are interested in the messages being produced.

There are currently two models to implement OPC UA PubSub: brokerless and broker-based mode. The first uses UA Datagram Protocol (UADP) for frequent transmission in a one-to-one or one-to-many configuration directly between the participants. This brokerless model is based on multicast and relies on network infrastructure devices. The second model uses a broker, which is the focus of this study. When the broker receives the data from the publisher, it conveys it to interested subscribers. Subscribers interact with the broker indicating what data they are interested in without necessarily knowing who will send it. The specification includes definitions for AMQP 1.0 and MQTT 3.1.1 brokers.

In the broker mode, it is a broker's responsibility to make sure the information gets transferred, thus reducing the load of an individual producer. This helps in situations where multiple systems request data from e.g. a resource-constrained device. The broker model is also expected to open up new possibilities to ease utilization of OPC UA data as no OPC UA specific technology is required for the integrations. For a subscriber, this seems like regular MQTT or AMQP interaction although the payload has its typical OPC UA PubSub characteristic. It is worth noting that the PubSub specification excludes message routing which means it only standardises the data link.

TABLE I
THE FEATURES OF MESSAGING PROTOCOLS COMPARED FROM THE VIEWPOINT OF PLANT-WIDE COMMUNICATIONS.

	AMQP 0-9-1 [25]	AMQP 1.0 [27]	MQTT 3.1.1 [28]	MQTT 5.0 [29]	UDP [30]
Release time	2008	2012	2015	2019	1980
Mapping from OPC UA PubSub	-	✓	✓	-	✓
Topic-based routing	✓	-	✓	✓	-
Multicast routing	✓	-	-	-	✓
Brokerless communication	-	✓	-	-	✓
Explicit request-response support	✓	?	-	✓	-
Connection-oriented	✓	✓	✓	✓	-

Security in OPC UA PubSub concerns the transport security as well as integrity and confidentiality of the messages transferred. In the broker model, the transport can be ensured by securing transport between publishers and the broker as well as between consumers and the broker but this is transport protocol specific.

For message security, there are three levels: 1) no security; 2) signing but no encryption; 3) signing and encryption. Message security is end-to-end and implemented using SecurityGroups to manage cryptographic keys on both sides. A standard framework for this and the Security Key Services are defined in the OPC UA PubSub specification. [8]

IV. Protocol comparison

This section reviews the protocols included in OPC UA PubSub as well as related versions of the same protocols. First, an overview is provided about each, followed by a comparison.

A. Overview

The origin of AMQP 0-9-1 [25] is in the financial industry that used to deploy proprietary middleware for a high message volume, reaching hundreds of thousands of events per second [26]. AMQP would both be effective and enable interoperability for message routing.

AMQP 1.0 [27] is a later implementation that specifies a subset of the 0-9-1, namely a 'peer-to-peer' protocol and a message format. There is no compatibility with 0-9-1.

MQTT 3.1.1 [28] originates from Internet of Things (IoT) environments with a limited computational capacity compared to conventional servers. The specification was originally proprietary but later opened.

MQTT 5.0 [29] was released in 2019, adding new features to MQTT 3.1.1. The new features include, e.g., a built-in header called 'response topic' for request-response communication, payload format indicator, and more flexibility regarding authorization and authentication schemes.

Compared to others, UDP is a simplistic alternative, as it omits the message broker [30]. UDP is a basic Internet protocol and widely supported in networking libraries. UDP either sends messages directly or delivers these as multicast to the entire physical sub-network.

B. Comparison

Table I compares the protocols for the following aspects.

a) Release time: At the time of writing, only MQTT has recently evolved, the newest version being from 2019. UDP is a basic Internet protocol and therefore unlikely to develop considerably. The AMQP family received its latest update 10 years from the time of writing.

b) Mapping from OPC UA PubSub: Only AMQP 1.0, MQTT 3.1.1, and UDP have a mapping from PubSub.

c) Topic-based routing: AMQP 0-9-1 and MQTT support topic-based message routing. This is a requirement in networks that span over geographically large areas or the cloud, exploiting Internet communication. In UDP, this is out of scope, whereas AMQP 1.0 leaves it as a future item.

d) Multicast routing: AMQP 0-9-1 and UDP support multicast routing. This enables a type of loose coupling, because the data source directly points to no recipient. The feature is missing from MQTT as well as AMQP 1.0.

e) Brokerless communication: Routerless communication reduces overhead, as the nodes can create a direct communication link. This simplicity is advantageous when computational resources are limited. Still, the lack of broker would cause needless data traffic in cases with a need for selective message routing, hindering scalability especially when the number of nodes grows. AMQP 1.0 and UDP enable brokerless communication, whereas the others require a broker.

f) Explicit request-response support: AMQP 0-9-1 specifies a header to indicate which queue to reply to [25, p. 17]. However, AMQP 1.0 specification is vague, as there is a 'reply-to' field but no indication about the scope. Respectively, MQTT 5.0 specifies a response topic field. In contrast, MQTT 3.1.1 and UDP lack any explicit request-response reference.

g) Connection-oriented: All but UDP are based on Transmission Control Protocol (TCP) and therefore connection oriented, assuming a retry if data is lost. This enables reliability features but causes overhead.

C. Comparisons in bibliography

Earlier studies have compared AMQP and MQTT as well. [31] concluded that AMQP excels at security, whereas MQTT is more efficient. Respectively, [32] state that AMQP is better in security and features but MQTT requires less of computational power. There are performance studies as well, but these are problematic due to the influence of the protocol implementations. Besides, many studies lack an indication of the AMQP version being evaluated.

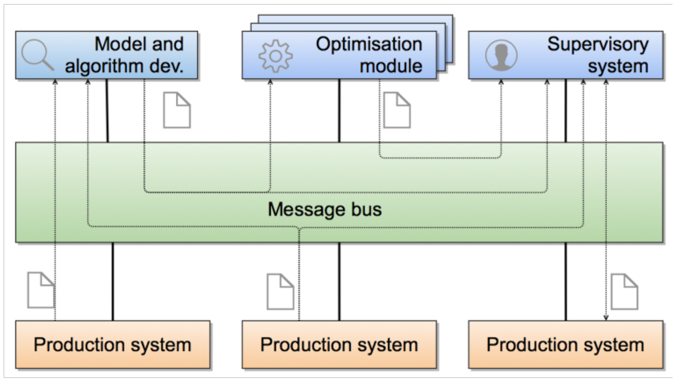

Fig. 1. In the COCOP plant-wide process monitoring and control architecture, systems for monitoring, optimisation and control are connected using a message bus architecture. The messaging layer acts as a unifying integration level between heterogeneous control applications and legacy control systems.

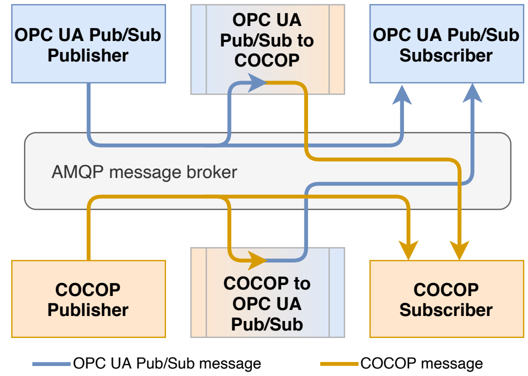

Fig. 2. Interoperability between OPC UA PubSub and the existing COCOP messaging architecture is implemented using translation components following the loosely coupled broker based implementation philosophy. In the model no changes are required to the system components being integrated.

D. Summary

The initial motivation of each protocol can be summarized as high volume for AMQP, IoT for MQTT and simplicity for UDP. Still, especially AMQP 0-9-1 and MQTT are competitors to each other with similar features, and the final performance likely depends on the implementation. AMQP 1.0 lacks routing features, whereas UDP has its use case in low overhead and situations that require no message routing. Among the brokered protocols, MQTT seems more actively developed and therefore tempting for long-term system designs.

For plant-wide communication, the most promising selection is MQTT 3.1.1 with its message routing support and mapping from OPC UA PubSub. For the future, there are hopes of including MQTT 5.0 in the official PubSub protocols.

V. Case Study Example

This section presents an example where OPC UA PubSub is integrated with an existing message broker approach. Using the example, we analyze the compatibility of OPC UA PubSub with the existing approach and if OPC UA PubSub can be used as an extension of similar existing integration models.

A. Case Requirements and the COCOP integration model

The example case is an information exchange and communication architecture for the plant-wide monitoring and control of industrial processes, i.e. coordinating control and optimisation together with the local control systems of subprocesses [33]. The original approach follows a data-driven model with loose coupling using a message broker [34]. Fig. 1 illustrates the coupling of different systems. To facilitate interoperability, various information models, e.g. Observations and Measurements (O&M) and TimeseriesML from Open Geospatial Consortium (OGC), have been selected for conveying various data between decentralized operations [35].

In the COCOP model, the message broker and agreed information models act as the unifying interoperability layer between local control systems and any other applications requiring data. The broker and the information model each form a sub-layer independent of the other. Therefore, one sub-layer can be extended or even replaced without side effects.

B. Data model translations between OPC UA PubSub and existing COCOP message broker system (AMQP)

The most convenient way to implement COCOP-compliant messaging is using an SDK called COCOP Toolkit[1]. To connect with the broker in the protocol level, the software clients would use any generic client libraries available for the message bus. For the information model, there are COCOP-specific software libraries to facilitate development. These have been published as a part of COCOP Toolkit, implemented in C#.NET and Java [36].

Still, there is no necessity to utilize COCOP Toolkit as seen in this example case. Any software platform is possible as long as the required protocol is applied and the information models match the standards in COCOP. This example case builds upon the JavaScript implementation NodeJs, which has no support in COCOP Toolkit but has a client library for AMQP.

The example case demonstrates a situation where COCOP communicates with OPC UA PubSub nodes (see Fig. 2). The case necessitates translators to enable cross-technology communication from PubSub publisher to COCOP subscriber or COCOP publisher to PubSub subscriber. It is notable that as long as the message bus remains the same for COCOP and PubSub, this protocol layer requires no translation. COCOP uses AMQP 0-9-1, which has no mapping from PubSub, but this protocol is conceptually similar to MQTT and AMQP 1.0. Furthermore, some message bus products, at least RabbitMQ, support multiple protocols and also enable protocol translation.

Fig. 3 visualizes the mapping from a COCOP-compliant XML data record to the OPC UA information model (JSON). The example shows the composition of a matte sample taken from the flash smelting furnace (FSF) of a copper smelter. The percentage of each substance (copper, iron, nickel, and sulphur; the rest omitted), has an entry in the payload. Still, some metadata lacks a direct mapping as PubSub delivers

[1]https://kannisto.github.io/Cocop-Toolkit/

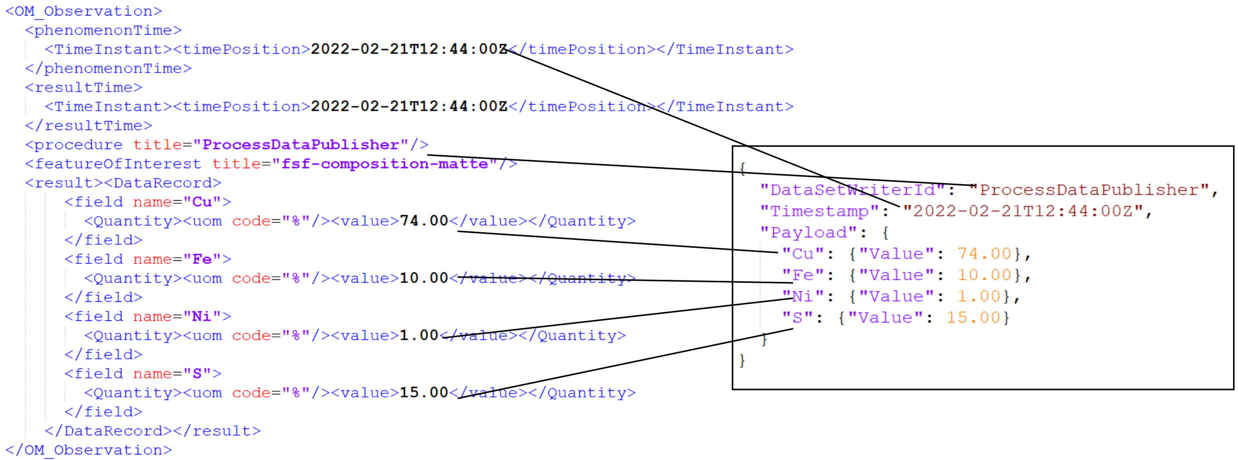

Fig. 3. Visualized mapping of the XML-based OGC O&M data model (simplified) and a JSON model developed for OPC UA PubSub for process data mesurements. The PubSub message omits some of the metadata, as this would be delivered in separate messages in a complete PubSub setup. Furthermore, the PubSub data model used here is exemplary and not fully compliant.

separate *DataSetMetadata* messages that are omitted in this demo. It is notable that the JSON format is an exemplary subset of PubSub and therefore not fully compliant.

The mapping appeared straightforward due to the modern data formats (XML and JSON) and their support in standard libraries of modern software tools. This suggests that OPC UA PubSub can wrap any legacy or otherwise incompatible systems and therefore reach interoperability with OPC UA.

C. Information security considerations

The COCOP model relies on securing the communication of publishers and consumers with the AMQP broker, including the access control regarding publishing and consuming the data. RabbitMQ, used in the COCOP implementation, supports authentication using username/password and certificates, and enables setting authorizations for the exchanges and queues.

The OPC PubSub model supports different levels of information security, as explained in section III. To enable the integration in the example, the adapters needed a permission to the message broker and the exchanges. From the outlined differences, it can be seen that the translator component becomes the critical part in the integration and must be part of the OPC UA PubSub Security Group if more advanced PubSub message security means are used. It is notable that the two security models do not match and, therefore, more of integration effort is necessary, including encryption/decryption.

VI. Results and Discussion

Our simple example shows that although OPC UA PubSub and existing AMQP (0.9.1) communications seem similar they are not interoperable as such. Using an adapter approach, we were successful in integrating data payloads from proprietary data models to those used by OPC UA PubSub.

The adapter pattern can also be used for information security, which is essential in Internet-wide communications. Our existing AMQP system uses encryption and security in the broker implementation, which means that the adapter component needs to transform the security model as well. OPC UA PubSub message security is specified outside of the actual communication protocol, which makes this a necessity for any existing broker-based integration.

More generally, the use of message-oriented integration approaches, such as message brokers, implies a shift in how application systems are developed compared to the typical request-response pattern usually involving client and server applications. In a publish-subscribe model, the system components react upon new data instead of being invoked. This can be seen to increase scalability and prospects of using data in new, previously unknown applications.

OPC UA PubSub extends the widely used OPC UA communication model towards this direction. However, it is notable that not all traditional OPC UA features are available as such and there are separate mechanisms for metadata and discovery.

UA PubSub can improve integration prospects in industrial systems. Unlike traditional OPC UA, PubSub does not require the systems to be UA servers or clients in the traditional sense. Thus, PubSub can unify practices, especially how information security is managed across various communication media.

Regarding future directions, PubSub is essential as OPC UA enters Operational Technology (or OT) through the initiative Field Level Communications (FLC) [37]. Conventionally, this area has been dominated by heterogeneous fieldbus technologies, but FLC aims to enable flexible communications with interoperability, security, and determinism.

VII. Conclusion

In this paper, we compared OPC UA PubSub with other message-broker-based integration models. We also studied, using an example for plant-wide process monitoring and control, how OPC UA PubSub can be integrated with an existing AMQP message broker integration approach that uses other than OPC UA based information models. We showed that an adapter approach is required for translating the data payloads as well as transforming the information security practices.

PubSub defines many features outside of the transport protocol, which means that they need to be implemented separately for the different transport protocol mappings supported. This should unify how PubSub systems are implemented and thus increase interoperability that is currently not available between, e.g., existing AMQP or MQTT solutions.

The transition from traditional OPC UA to UA PubSub implies a shift in the operation model and system design philosophy but allows an easier integration compared to traditional OPC UA, e.g. to other data applications, as the systems are more decoupled by nature. In our case example we also showed that integration is straightforward and the support for OPC PubSub structures can improve the interoperability of existing message-broker-based approaches with minor changes.

References

[1] A. W. Colombo, S. Karnouskos, O. Kaynak, Y. Shi, and S. Yin, "Industrial cyberphysical systems: A backbone of the fourth industrial revolution," *IEEE Industrial Electronics Magazine*, vol. 11, no. 1, pp. 6–16, March 2017.

[2] R. F. Babiceanu and R. Seker, "Big data and virtualization for manufacturing cyber-physical systems: A survey of the current status and future outlook," *Computers in Industry*, vol. 81, pp. 128–137, 2016, emerging ICT concepts for smart, safe and sustainable industrial systems. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0166361516300471

[3] P. Leitão, F. Pires, S. Karnouskos, and A. W. Colombo, "Quo vadis Industry 4.0? Position, trends, and challenges," *IEEE Open J. Ind. Electron. Soc.*, vol. 1, pp. 298–310, 2020.

[4] V. Jirkovský, M. Obitko, P. Kadera, and V. Mařík, "Toward plug play cyber-physical system components," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 6, pp. 2803–2811, 2018.

[5] F. B. Vernadat, "Technical, semantic and organizational issues of enterprise interoperability and networking," *Annu. Rev. Control*, vol. 34, no. 1, pp. 139–144, 2010.

[6] "New European interoperability framework," European Commission, 2017, URL http://doi.org/10.2799/78681 [Retrieved 29 Oct 2021].

[7] "OPC unified architecture specification part 1, overview and concepts, release 1.04," OPC Foundation, 2017.

[8] "OPC unified architecture specification part 14, PubSub, release 1.04," OPC Foundation, 2018.

[9] A. A. Nazarenko, J. Sarraipa, L. M. Camarinha-Matos, C. Grunewald, M. Dorchain, and R. Jardim-Goncalves, "Analysis of relevant standards for industrial systems to support zero defects manufacturing process," *J. Ind. Inf. Integr.*, p. 100214, 2021, in press.

[10] F. Tao, Q. Qi, A. Liu, and A. Kusiak, "Data-driven smart manufacturing," *J. Manuf. Syst.*, vol. 48, pp. 157–169, 2018.

[11] N. Benias and A. P. Markopoulos, "A review on the readiness level and cyber-security challenges in industry 4.0," in *2017 South Eastern European Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM)*, 2017, pp. 1–5.

[12] Z. Liu and P. Bellot, "OPC UA PubSub implementation and configuration," in *2019 6th International Conference on Systems and Informatics (ICSAI)*, 2019, pp. 1063–1068.

[13] ——, "A configuration tool for MQTT based OPC UA PubSub," in *2020 RIVF International Conference on Computing and Communication Technologies (RIVF)*, 2020, pp. 1–6.

[14] C. Lee, N. Kim, and S. Hong, "Toward industrial IoT: Integrated architecture of an OPC UA synergy platform," *IEEE Access*, vol. 9, pp. 164 720–164 731, 2021.

[15] A. Eckhardt, S. Müller, and L. Leurs, "An evaluation of the applicability of OPC UA publish subscribe on factory automation use cases," in *2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)*, vol. 1, 2018, pp. 1071–1074.

[16] J. Pfrommer, A. Ebner, S. Ravikumar, and B. Karunakaran, "Open source OPC UA PubSub over TSN for realtime industrial communication," in *2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)*, vol. 1, 2018, pp. 1087–1090.

[17] S. K. Panda, M. Majumder, L. Wisniewski, and J. Jasperneite, "Real-time industrial communication by using OPC UA field level communication," in *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, vol. 1, 2020, pp. 1143–1146.

[18] P. Zimmermann, E. Axmann, B. Brandenbourger, K. Dorofeev, A. Mankowski, and P. Zanini, "Skill-based engineering and control on field-device-level with OPC UA," in *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2019, pp. 1101–1108.

[19] J. M. Gutierrez-Guerrero and J. A. Holgado-Terriza, "Automatic configuration of OPC UA for industrial internet of things environments," *Electronics*, vol. 8, no. 6, 2019. [Online]. Available: https://www.mdpi.com/2079-9292/8/6/600

[20] F. Gosewehr, J. Wermann, W. Borsych, and A. W. Colombo, "Specification and design of an industrial manufacturing middleware," in *2017 IEEE 15th International Conference on Industrial Informatics (INDIN)*, 2017, pp. 1160–1166.

[21] G. Angione, J. Barbosa, F. Gosewehr, P. Leitão, D. Massa, J. Matos, R. S. Peres, A. D. Rocha, and J. Wermann, "Integration and deployment of a distributed and pluggable industrial architecture for the PERFoRM project," *Procedia Manuf*, vol. 11, pp. 896–904, 2017.

[22] P. Varga, F. Blomstedt, L. L. Ferreira, J. Eliasson, M. Johansson, J. Delsing, and I. M. de Soria, "Making system of systems interoperable – the core components of the Arrowhead framework," *J. Netw. Comput. Appl.*, vol. 81, pp. 85–95, 2017.

[23] "OPC UA part 100: Devices, release 1.02," OPC Foundation, 2019.

[24] "OPC UA for asset administration shell (AAS), release 1.00," OPC Foundation, 2021.

[25] "AMQP: Advanced message queueing protocol version 0-9-1," AMQP Working Group 0-9-1, 2008, URL http://www.amqp.org/specification/0-9-1/amqp-org-download [Retrieved 27 May 2020].

[26] J. O'Hara, "Toward a commodity enterprise middleware," *Queue*, vol. 5, no. 4, p. 48–55, May 2007.

[27] "OASIS advanced message queuing protocol (AMQP) version 1.0," OASIS, 2012, URL http://docs.oasis-open.org/amqp/core/v1.0/os/amqp-core-complete-v1.0-os.pdf [Retrieved 27 May 2020].

[28] "MQTT version 3.1.1," OASIS, 2015, URL http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/errata01/os/mqtt-v3.1.1-errata01-os-complete.html [Retrieved 8 Jul 2020].

[29] "MQTT version 5.0," OASIS, 2019, URL https://docs.oasis-open.org/mqtt/mqtt/v5.0/os/mqtt-v5.0-os.html [Retrieved 27 May 2020].

[30] "RFC 768: User datagram protocol," ISI, 1980, URL https://tools.ietf.org/html/rfc768 [Retrieved 18 May 2020].

[31] N. Naik, "Choice of effective messaging protocols for iot systems: MQTT, CoAP, AMQP and HTTP," in *2017 IEEE International Systems Engineering Symposium (ISSE)*, 2017, pp. 1–7.

[32] N. Q. Uy and V. H. Nam, "A comparison of AMQP and MQTT protocols for internet of things," in *2019 6th NAFOSTED Conference on Information and Computer Science (NICS)*, 2019, pp. 292–297.

[33] P. Kannisto, D. Hästbacka, T. Gutiérrez, O. Suominen, M. Vilkko, and P. Craamer, "Plant-wide interoperability and decoupled, data-driven process control with message bus communication," *Journal of Industrial Information Integration*, vol. 26, p. 100253, 2022.

[34] D. Hästbacka, P. Kannisto, and M. Vilkko, "Data-driven and event-driven integration architecture for plant-wide industrial process monitoring and control," in *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*, Oct 2018, pp. 2979–2985.

[35] ——, "Information models and information exchange in plant-wide monitoring and control of industrial processes," in *Proc. of the 10th Int. Joint Conf. on Knowledge Discovery, Knowledge Engineering and Knowledge Management - Volume 3: KMIS*, 2018, pp. 216–222.

[36] P. Kannisto, A. Kätkytniemi, M. Vilkko, and D. Hästbacka, "Software toolkit for development of interoperable communications in data-driven systems," *IFAC-PapersOnLine*, vol. 54, no. 1, pp. 845–850, 2021.

[37] "OPC UA for field level communications – a theory of operation, version 1," OPC Foundation, 2020, URL https://opcfoundation.org/wp-content/uploads/2020/11/OPCF-FLC-Technical-Paper-C2C.pdf [Retrieved 7 Sep 2021].