Model-driven engineering of gateways for industrial automation.

P.Denzler [1], **D.Ramsauer** [1] **and W.Kastner** [1]
[1] TU Wien, Automation Systems Group, Treitlstr. 1-3/4., 1040 Vienna, Austria

Tel.: +43 1 58801-18311, fax: +43 1 58801-18391
E-mail: patrick.denzler@tuwien.ac.at

Abstract: Middleware gateways are a feasible solution for improving interoperability between legacy connectivity technologies in industrial automation and enterprise information technology. However, developing, configuring, and deploying gateways on different devices for connecting middlewares are not well examined. A proposed gateway generation process that builds upon a generic gateway model in the Architecture Analysis & Design Language (AADL) and allows platform-specific code generation addresses those challenges. Additional, configuration files define all necessary bindings. The paper concludes by outlining subsequent steps by pointing out research challenges in the presented process.

Keywords: middleware, gateways, industrial automation, Industrial Internet of Things (IIoT), Architecture Analysis & Design Language (AADL)

1. Introduction

Accessing information in industrial automation has always been a critical element for a system's optimal functioning, but often hindered by the used architecture [1]. From a historical perspective, typical automation systems follow a hierarchical architecture often visualised as an automation pyramid (Fig. 1) [2]. The first two levels represent sensors and actuators, followed by controllers (e.g., programmable logic controllers (PLCs)) which are responsible for the interaction with the underlying technical process and the processing of automation functions (e.g., open/closed-loop control), respectively. With the third SCADA systems level, the lower levels represent the operational technology (OT) part of automation [3].

Compared to the two top levels, summarised as information technology (IT), OT has more stringent requirements (e.g., on real-time and safety) achieved by automation and industrial communication technology. On the other hand, IT uses off-the-shelf components and communication systems ready for use with the Internet Protocol (IP) suite part of any enterprise / IT environment. Typical applications in the upper levels are Manufacturing Execution Systems (MES) and Enterprise-Resource-Planning (ERP) tool suites that process and abstract the data from the lower levels. Accessing OT information is complicated and is often referred to as the OT/IT gap [2].

Given more recent initiatives, e.g., the Industrial Internet of Things (IIoT) or Industry 4.0, access to information is becoming even more critical. IIoT aims to close the OT/IT gap by introducing the same IP technology at all levels, thus creating a more flat and transparent architecture. Such a system design would make it possible to address a single sensor on the shop floor directly from a top-level application (e.g., located in the cloud).

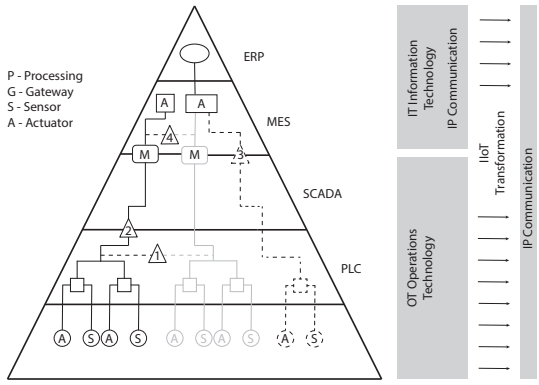

Fig. 1. *OT / IT gap and gateway positions in the Automation Pyramid*

New technological developments such as fog and cloud computing or time-sensitive networking (TSN) further promise to close the OT/IT gap [4].

An often forgotten problem in the long history of industrial communication is that the domain looks back on several decades of industrial communication technology. Due to long life cycles, it is not uncommon to find relatively old, still correctly working systems [5]. Therefore, a sudden change towards an all-new IP architecture is not very realistic. There is, therefore, a need for interim solutions, for example, gateways to connect older OT protocols with newer technology and Internet protocols of the IT levels [6].

In this context, this article describes the possible positions of gateways within the automation pyramid. It presents arguments why middleware gateways are a feasible solution to close the OT/IT gap. It also describes the related challenges, followed by a process that addresses them. The results form the basis for further research about middleware gateways for industrial automation.

Section 2 presents possible positions of gateways in the automation pyramid and the historical background. Section 3 introduces the middleware gateway concept and lays the ground for the related challenges in Section 4. Within Section 5, the gateway generation process is introduced and discussed in Section 6. Section 7 concludes the paper and provides an outlook on future work.

2. Gateways in the automation pyramid

Gateways have always been part of industrial automation. They found their application at different automation pyramid levels by connecting protocols horizontally or vertically with IT applications [1].

In the early days of automation when fieldbus systems replaced parallel and discrete cabling in the lower levels of the automation pyramid, early suggestions appeared to improve interoperability with gateways [7]. The focus of these dedicated automation networks (e.g. Controller Area Network (CAN), PROFIBUS or INTERBUS) however was to overcome the restrictions of parallel cabling and not on interoperability [7]. Therefore, low-level horizontal gateways, as depicted in Figure 1, remained an exception.

With the advent of the Internet, the relevant technology also made its way into the automation domains. Due to the uniform Ethernet standard, gateways seem to be a relic of the past. However, Ethernet lacked actual real-time capabilities, which led to the development of several new dedicated protocols and reintroduced the need for gateways.

Protocols such as PROFINET or EtherCAT either use parts of the OSI layers or adapt them to achieve low latency communication. In contrast to Internet technology, such protocols are OT specific and not intended for communication with the IT domain [8]. The specific real-time requirements make creating horizontal gateways a problematic endeavour. Research has shown that directly connecting such protocols can create serious security risks [1]. Nowadays, the new initiative creating time-sensitive networks (TSN) opens up new possibilities [4].

Vertical gateways are another solution to overcome OT and IT's separation as they transfer automation data to the IP-based networks. The gateways abstract a certain amount of data and functions from OT to the enterprise environment [9]. However, creating vertical gateways requires extensive knowledge of the protocol and the connected application (MES, ERP system). The uniqueness of each gateway makes it challenging to maintain and reuse or adjust it for other applications. One possible solution is to connect vertical gateways with middleware that is better suited to the IT environment.

With middleware, IT applications can access the OT world's data points with the necessary abstraction and additional information. There are different ways to access legacy protocols. Either directly with specific APIs and drivers or as mentioned via vertical gateways. Since middlewares offer standardised access to all IT applications, their use would reduce the number of required gateways [6]. However, the number of middleware and protocol combinations is still considerably large and would hinder interoperability.

At this point, middleware gateways could improve the situation by building bridges between middlewares. The complexity of such gateways is high because middlewares sometimes use different communication patterns, e.g., OPC Unified Architecture (OPC UA) (Server-Client), MQTT (Broker) or Data Distribution Service (DDS) (publish-subscribe) and means of transport (TCP/IP, UDP, HTTP or CoAP) [9]. Nonetheless, the IT compatibility allows the flexible deployment, e.g., on fog nodes or in the cloud and are therefore a suitable solution to play an essential role in closing the OT/IT gap.

3. Middleware gateway concept

Middleware gateways could reduce the total amount of required gateways in IIoT. The goal is to connect the existing middlewares, which are either already attached to legacy protocols or provide drivers or APIs. Such a proposal is not new. The Industrial Internet Consortium (IIC) published a related study to identify the leading middleware technologies for their suitability for IIoT applications [10]. After an in-depth evaluation OPC UA, DDS, oneM2M and Web Services turned out to be best candidates for all types of industrial sectors. Connecting those "main" middlewares with gateways would significantly improve interoperability. Secondary gateways would connect other middlewares or legacy protocols to one of the main middlewares (cf. Figure 2).

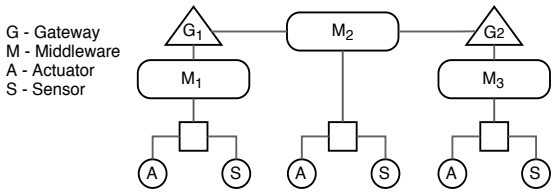

Fig. 2. *Main middleware gateways concept*

As part of the IIC, the OMG Group published an OPC UA / DDS gateway specification [11] to follow up on the concept. Since OPC UA is a server-client-based middleware and DDS follows the publish-subscribe paradigm, the specification addresses these differences by utilising configurable bi-directional bridges. Since its publication in 2019, the gateway specification received some attention in the industry and led to a proprietary implementation. In academic research, the focus remains mainly on the gateway's information exchange and functionality aspects by implementing reduced prototypes as in [12]. However, none of the studies focused on the challenges associated with implementing, configuring, and deploying such gateways.

4. Challenges related to middleware gateways

A recent use-case study about fog computing in automation systems to consolidate SCADA functionality, and using an OPC UA / DDS gateway prototype revealed several challenges [13]. In this use case, DDS acted as a bridge between two OPC UA networks and provided remote maintenance functionality. The trials revealed mostly problems related to the implementation of the gateway prototype itself.

One other significant difficulty is the configuration of the gateway. Configuration requires in-depth knowledge about both networks; i.e., the node addresses, the data to be transferred or access rights. These hurdles directly impact the gateway's deployment and scalability because some information must be in place before the compilation. If such dependencies exist, the dynamic deployment of a gateway on different resources, for example on fog nodes, would require an automated compile process or a change in the gateway architecture. Other studies report similar problems with gateways, despite different protocols and architectures [14]. For example, Dionisio et al. [15] present an OPC UA / MQTT gateway based on Node-RED for "loom machines" where they use a static gateway configuration. Some authors [16] try to counter the configuration issue with a software-defined perspective or with a particular additional middleware [17].

Other problems encountered in the use case are related to security and latency issues. An additional firewall that protects the gateway from malicious access solved the problem. However, this solution required additional configuration effort. It would have been more user-friendly if the gateway had supported the security functions used by DDS and OPC UA. While latency and overhead were not a critical problem in the use case, some considerations had to be made. Other authors report similar problems and use HTTP to bridge protocols to reduce overhead and latency [18].

In summary, to enable middleware gateways to bridge the OT/IT gap and connect legacy protocols, those challenges need to be addressed first.

5. Gateway generation process

Gateway configuration, deployment and scalability are the most relevant challenges that a potential solution needs to address. The suggested process foresees using a modelling language to model the different aspects of a gateway and code generation to achieve higher flexibility. Potential candidates are Unified Modeling Language (UML), Systems Modeling Language (SysML) and the Architecture Analysis & Design Language (AADL). All languages support the modelling of complex applications and allow code generation. However, only SysML and AADL are capable of specifying hardware and communication. AADL originating out of avionics provides analysis possibilities for embedded, real-time systems, supporting its use in industrial automation [20].

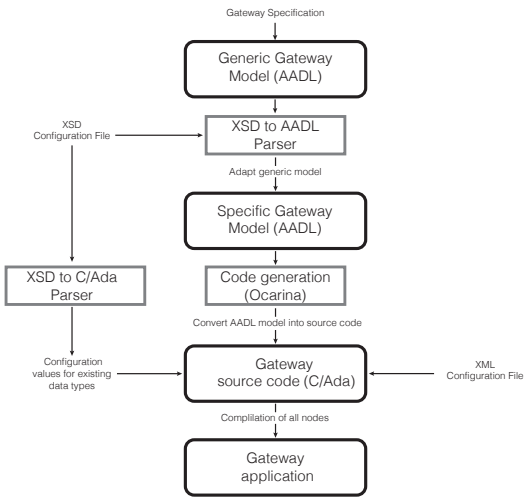

Fig. 3. *Gateway generation process*

The suggested process assumes the usage of AADL and related tools. However, it is also applicable to other modelling languages. The first step is the creation of a generic gateway model based on a specification. This unique specification depends on the middleware combination and defines necessary bindings and features. The created general model is the foundation for the further code generation and acts as an empty shell for specific gateway instances. For the actual code generation, each gateway instance requires further "hardcoded" information, e.g., data type mappings or node addresses. The proposed process foresees two stages for adding the required information. In the first stage, a configuration file provides the parser everything to create a specific gateway model. The following code generation then produces code. In AADL, the Ocarina toolchain creates either C or ADA code out of the model [20]. It might be worth to mention that the code generation is dependent on the chosen hardware environment, e.g., an embedded system. This feature is essential when considering that gateways might run on resource-restricted edge devices or fog nodes. In the last step before compiling the code, an XML configuration file provides the final mappings that end the gateway creation process.

6. Discussion

Although the proposed process is relatively simple, it addresses the most relevant challenges related to middleware development and use of gateways. Adopting a general gateway model allows the generation of specific and compiled gateway instances and reduces the configuration effort when adjustments are required. A weak point of the process is that the model entirely depends on a given gateway specification unique for each middleware combination; however, following the main gateway idea with only a few combinations reduces the effort.

As other studies indicated, the configuration of the gateway is a core challenge. Despite the proposed approach to integrating the configuration into the build-

ing process, the required configuration files are complex. This fact limits the flexibility as the configuration is static. A possible solution would be using an external tool that creates the files. In the best case, the tool obtains the needed (network) information from both middlewares automatically. OPC UA, for example, applies information models to structure all involved nodes. Another option is to change the architecture of the gateway, that would allow dynamical changes during runtime. Such functionality, however, might increase the complexity and is not supported in all programming languages.

If the configuration is static, the process will allow an automated generation of gateways. For instance, applied in fog computing, a deployment tool creates new instances when one gateway cannot handle all traffic. Moreover, though the process aims for middleware gateways, it is applicable for direct vertical gateways to connect legacy protocols with IT applications.

6. Conclusion

The paper presents a process to address challenges related to the development and deployment of middleware gateways which are an efficient way of connecting legacy OT protocols to the IT environment. A previously conducted use case study and related work indicate that the most relevant challenges are configuration, deployment, scalability and security. The model-based approach includes a two-step configuration, code generation for different platforms, and automation potential. Further studies will include the actual modelling of an OPC UA / DDS gateway with the proposed AADL language.

Acknowledgements

The research leading to these results has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 764785, FORA—Fog Computing for Robotics and Industrial Automation.

References

[1]. T. Sauter and M. Lobashov, "How to Access Factory Floor Information Using Internet Technologies and Gateways," in *IEEE Transactions on Industrial Informatics*, vol. 7, no. 4, pp. 699-712, Nov. 2011.

[2]. D.R. Harp, and G. Bengt, "IT/OT Convergence Bridging the Divide," *NexDefense*, p. 23, 2015.

[3]. M. Wollschlaeger, T. Sauter and J. Jasperneite, "The Future of Industrial Communication: Automation Networks in the Era of the Internet of Things and Industry 4.0," in *IEEE Industrial Electronics Magazine*, vol. 11, no. 1, pp. 17-27, March 2017.

[4]. M. Aazam, S. Zeadally and K. A. Harras, "Deploying Fog Computing in Industrial Internet of Things and Industry 4.0," in *IEEE Transactions on Industrial Informatics*, vol. 14, no. 10, pp. 4674-4682, Oct. 2018.

[5]. K. F. Tsang, M. Gidlund and J. Åkerberg, "Guest Editorial Industrial Wireless Networks: Applications, Challenges, and Future Directions," in *IEEE Transactions on Industrial Informatics*, vol. 12, no. 2, pp. 755-757, April 2016.

[7]. A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," in *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2347-2376, 2015.

[8]. G. Wood, "Survey of LANs and standards," *Computer Standards & Interfaces,* vol. 6, no. 1, pp. 27 – 36, 1987.

[9]. P. Danielis *et al*., "Survey on real-time communication via ethernet in industrial automation environments," *Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA)*, Barcelona, 2014, pp. 1-8.

[10]. T. Sauter, "The continuing evolution of integration in manufacturing automation," in *IEEE Industrial Electronics Magazine*, vol. 1, no. 1, pp. 10-19, 2007.

[11]. Industrial Internet Consortium, "Industrial Internet Reference Architecture (IIRA), 2015.

[12]. Object Management Group. OPC UA/DDS Gateway Version 1.0. (2020, October 15). [Online]. Available: https://www.omg.org/spec/ DDS-OPCUA/1.0

[13]. R. Endeley, T. Fleming, N. Jin, G. Fehringer and S. Cammish, "A Smart Gateway Enabling OPC UA and DDS Interoperability," *2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, 2019, pp. 88-93.

[14]. P. Denzler, J. Ruh, M. Kadar, C. Avasalcai and W. Kastner, "Towards Consolidating Industrial Use Cases on a Common Fog Computing Platform," 2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Vienna, Austria, 2020, pp. 172-179.

[15]. R. R. Peña, R. D. Fernández, M. Lorenc and A. Cadiboni, "Gateway OPC UA/Modbus applied to an energy recovery system identification," *2019 XVIII Workshop on Information Processing and Control (RPIC)*, Bahía Blanca, Argentina, 2019, pp. 235-240.

[16]. R. Dionisio, S. Malhao, and P. Torres, "Development of a Smart Gateway for a Label Loom Machine using Industrial IoT Technologies", *International Journal of Online and Biomedical Engineering (iJOE)*, vol. 16, no. 4, pp. 6–14, 2020.

[17]. Z. Jiang, Y. Chang, and X. Liu, "Design of software-defined gateway for industrial interconnection," *Journal of Industrial Information Integration*, vol. 18, 2020.

[18]. A. Ioana and A. Korodi, "VSOMEIP - OPC UA Gateway Solution for the Automotive Industry," *2019 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC)*, Valbonne Sophia-Antipolis, France, 2019, pp. 1-6.

[19]. MAA da Cruz, J.J.P.C.Rodrigues, P. Lorenz, P. Solic, J. Al-Muhtadi, and V. H. C. Albuquerque, "A proposal for bridging application layer protocols to HTTP on IoT solutions," *Future Generation Computer Systems*, vol. 97, pp. 145–152, 2019.

[20] P.H., Feiler, D.P. Gluch, J.J. Hudak, "The architecture analysis &design language (AADL): An introduction". *Technical Report*. Carnegie-Mellon Univ Pittsburgh PA Software Engineering Inst., 2006