Lyu, Tuojian; Atmojo, Udayanto Dwi; Vyatkin, Valeriy

On enhancing reconfigurability of I/O connection and access in IEC 61499

Published in:
2022 IEEE 31st International Symposium on Industrial Electronics, ISIE 2022

DOI:
10.1109/ISIE51582.2022.9831506

Published: 01/01/2022

Document Version
Peer-reviewed accepted author manuscript, also known as Final accepted manuscript or Post-print

Please cite the original version:
Lyu, T., Atmojo, U. D., & Vyatkin, V. (2022). On enhancing reconfigurability of I/O connection and access in IEC 61499. In *2022 IEEE 31st International Symposium on Industrial Electronics, ISIE 2022* (pp. 818-823). (Proceedings of the IEEE International Symposium on Industrial Electronics; Vol. 2022-June). IEEE. https://doi.org/10.1109/ISIE51582.2022.9831506

On enhancing reconfigurability of I/O connection and access in IEC 61499

Tuojian Lyu*, Udayanto Dwi Atmojo*, Valeriy Vyatkin*†
*Department of Electrical Engineering and Automation, Aalto University, Espoo, Finland
†Department of Computer Science, Electrical and Space Engineering, Lulea Tekniska Universitet, Sweden

Email: {tuojian.lyu, udayanto.atmojo}@aalto.fi, vyatkin@ieee.org

***Abstract*—The efficiency and flexibility of data connection establishment in the automation system development architecture of IEC 61499 is important to support portability and flexibility. To improve this aspect, this paper proposes a method integrating OPC UA functionalities and I/O-accessing capabilities, introducing two function blocks, SmartIX and SmartQX. The solution is illustrated with one use case and comparison with the existing solution. The comparison shows that our proposed solution can improve efficiency, flexibility and portability for IEC 61499 by making fewer changes to existing systems from debugging mode to production mode.**

***Index Terms*—IEC 61499, OPC UA, 4diac, digital twin, smart factory, virtual commissioning**

I. INTRODUCTION

An intelligent, automated, efficient and flexible smart production is a vision highlighted in Industry 4.0. IEC 61499 is a distributed automation standard that extends the currently dominating IEC 61131-3 standard by introducing event-driven execution and distribution, i.e., control applications can be distributed to multiple PLCs. With the aforementioned advantages and features provided by IEC 61499, this standard is an important enabling technology of Industry 4.0.

On the other hand, two widely used protocols for data connections in industrial production are OPC UA and MQTT. Thanks to the benefits that OPC UA offers, such as data security and reliability, and the standardization effort, OPC UA is now very widely used in automation systems to transfer production data. However, we noticed some challenges and difficulties that prevent OPC UA from being further applied in control application development. Those challenges include:

- A lot of manual work and modifications to the existing automation systems are required by engineers.
- It is often time-consuming and error-prone to engineers to deal with Input/Output (I/O) issues, particularly on mapping configuration into I/O and data connections establishment.

The way how to utilize OPC UA in IEC 61499 has not yet been standardized, and this paper intends to propose a pattern on how OPC UA can be used to build data connections to processes and their virtual models in automation systems based on the IEC 61499. Specifically, this is illustrated using the 4diac IDE implementation of IEC 61499.

This paper attempts to address this issue by integrating functionalities of OPC UA and local I/O reading/writing in the corresponding function block, prototyped as a composite FB (CFB). Two common use-case scenarios are considered:

- Virtual Commissioning: Creating data connections between virtual sensors/actuators and IEC 61499-based control applications, i.e., bringing data from virtual sensors to control applications for debugging.
- Digital Twin creation: Bring data read from physical sensors and data waiting to be written to actuators to OPC UA server where those data could be further utilized or visualized by external cloud-native services, e.g., big data analysis, failure prediction and business optimization.

The outline of this paper is as follows. First, section II presents the literature review about I/O-accessing and OPC UA. Section III demonstrates the proposed design of SmartIO as our proposed method. Next, section IV gives one use case and the comparison between our proposed solution and the existing solution. Finally, section V summarizes the performance and contribution of our proposed solution and discusses the future plan.

II. LITERATURE REVIEW

A. I/O-accessing in different IEC 61499 implementations

In the current (IEC 61131-3 compliant) PLCs, the developers can either use physical PLC I/O tags in the programs or use global variables with symbolic names and associate them with the I/O tags at the later stage of deployment. However, this approach is not applicable in IEC 61499 since there are no global variables and deployment can be to a network of controllers, and the I/O sampling is event-driven, which requires an explicit description of the source of events for each input variable.

IEC 61499 does not define a standard way to access input-output modules of PLC, but provides a mechanism of service interface function blocks for performing this task.

One way of providing access to IOs is defining a library of Service-Interface FBs (SIFB) corresponding to the available hardware resources of the PLC. Such I/O-accessing libraries vary from device to device. Fig. 1 presents an example of how NxtStudio uses the A6650 SIFB to interact with I/Os of the Adam-6650 PLC of Advantech. A similar approach has been used in such tools as FBDK, 4diac, and BlockIDE. On the contrary, ISaGRAF uses the classic ”mapping table” approach, as it is done in IEC 61131-3 PLCs.

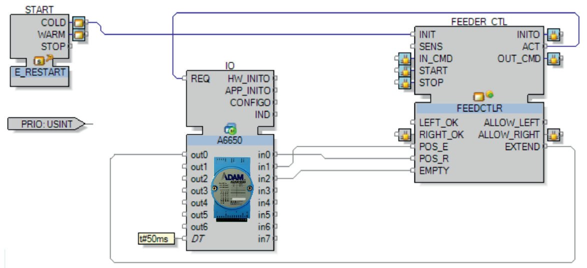

Fig. 1. I/O access implemented in NxtStudio for a PLC of Advantech.

However, this approach to I/O access introduces coupling between the control application and the underlying communication interface, making the control applications less portable and reusable. After mapping to the particular PLC, the control components, such as FEEDCTLR in the example, need to be manually connected to the SIFBs, representing PLC I/Os.

NxtControl has proposed a ”symbolic links” mechanism to rectify this problem. As illustrated in Fig. 2, the control component can be implemented as a composite FB, in which special symbolic links FBs represent I/O modules of the PLC with symbolic names, which can be built using the nested names of the FB instances within a hierarchically organized application. These names are independent of the particular type of PLC where the application will be deployed.

The lower part of the figure contains function blocks representing I/O modules of a particular PLC. These are contained in the corresponding device type and instantiated once a particular I/O configuration of the device is known. The symbolic names of the HW-independent application are assigned to the inputs of those I/O-representing FBs after the application is mapped to devices. In NxtStudio, a special kind of function blocks, called CAT, is used to implement the I/O FBs.

A similar approach was introduced in 4diac as described in [2]. I/O-accessing in application logic remains the same across various platforms while only device configuration function block network should be specified and modified. Based on their work, we tried their proposed mechanism on the Kunbus RevolutionPi PLC. Fig. 3 gives the control application implemented via the new I/O-accessing mechanism.

Here, for example, the instance LED_GREEN of FB QX implements the output of a Boolean value to a PLC output, whose exact pin number is determined by assigning the symbolic string LED_GREEN to the input Digital_Output2 of the FB IORevPiDIO.

B. Addressing OPC UA variables in IEC 61499

Typically, each PLC hosts an OPC UA server. Therefore, in IEC 61131-3 compliant PLCs, the same global variable table can be used to declare the OPC UA tags attached to the global variables.

In 4diac, access to the OPC UA server is organized using CLIENT and SUBSCRIBE communication FBs. This can also be seen in Fig.3. For example, the pair of SUBSCRIBE_1 and CLIENT_1_0 FBs implements writing and reading of the OPC UA copy of the LED_GREEN output signal.

In the NxtControl-developed implementation of IEC 61499, OPC UA tags can be assigned to inputs and outputs of the CAT FBs. Typically, I/O modules are represented by CATs. Therefore the OPC UA tags can be assigned to their inputs and outputs.

C. Reflecting I/O dynamics in cloud services

In order to achieve intelligent manufacturing, smart factory should be implemented such that intelligence can be present on each layer, i.e., from hardware layer with PLCs to cloud layer with machine learning services and big data analysis [3].

There is no doubt that data will play a more important role in Industry 4.0 [4] where intelligence is required to optimize traditional factories. The work [5] emphasized that twin data is the critical enabler when developing digital twin (DT). By establishing flexible data connections among automation systems, digital twins, and edge/cloud computing services, some advanced features can be obtained, including dynamic reconfiguration, deployment optimization, and agile manufacturing [6], [7], [8].

Qi et al. [9] presents a 5-dimensional DT model [10]. In order to provide advanced capabilities of dynamic reconfiguration, optimization, and agile manufacturing, DT should be provided data in various formats from physical sources and cloud services, domain experts, and virtual models. Thus six types of data connections were mentioned: 1) physical entities - virtual models, 2) physical entities - database, 3) physical entities - services, 4) virtual models - database, 5) virtual models - services, and 6) services - database. Among these, in this paper, we focus on the data connections between physical entities and virtual models. In this scenario, the physical part is responsible for collecting data from the real-world while collected data will be processed and knowledge will be extracted in DT with higher degrees of meaning [11].

The work [12] created a DT for a physical bending machine via using OPC UA. Communication protocols, including MQTT, CoAP, MTConnect, and OPC UA, were used. MQTT and OPC UA were focused upon, as they are most widely used. Compared to MQTT, OPC UA is more suitable for industrial automation due to its data security, reliability, and standardized information model and semantics. Moreover, the OPC UA server could be easily created on different platforms, from Windows to Linux. This helps distributed applications running on distributed PLCs subscribe and publish data via its server address to the OPC UA server.

D. Works of efficient data connection establishment

In the work [13], the issue of lack of remote data processing and remote PLC-controlling was pointed out. It is usually tedious to debug control applications with physical devices since each modification to the control applications requires recompilation of programs and redeployment. Moreover, PLCs are useful and powerful for executing control programs but not for data backup and data analytics due to the limitations

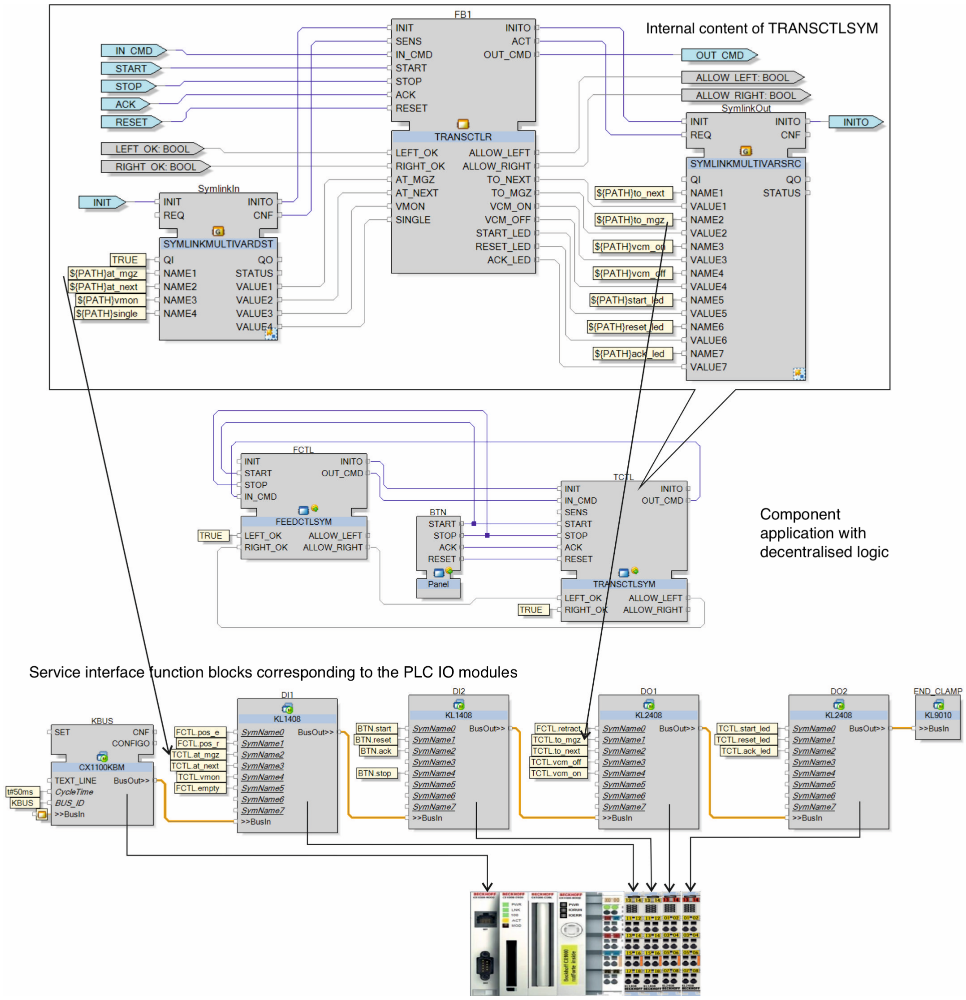

Fig. 2. I/O access implemented in NxtStudio using symbolic links [1].

of PLCs. A solution was proposed to solve those problems mentioned above by utilizing the Node-Red and cloud computing platform. Manufacturing data can be transferred from PLCs to the Node-Red gateway and be further transferred to a cloud platform. However, one issue that was not discussed in this paper is how to collect data inside PLCs efficiently. This is the topic that we focused in this paper.

The work [14] proposed a method for inter-PLC communication generation. The implementation of communication is always error-prone and tedious for automation system engineers. A plugin to 4diac IDE was implemented to automatically process system files and generate data connections in PLCs. Processing XML-format system files of 4diac projects would be one possible to meet the motivation of auto-generating communications.

Romanato et al. [15] presented a mechanism to bridge automation systems and Arrowhead service platform [16]. They developed composite FBs to read from and/or write to Arrowhead Datamanager via the HTTP communication protocol. Users can input different parameters to achieve different functionalities. We borrow this idea in this paper, i.e., implementing composite FBs to communicate with the OPC UA server in the cloud side. Furthermore, I/O-accessing capabilities are included in OPC UA functionalities, which further enables the flexibility of data connection creation.

E. Conclusions from the literature review

Many works justify the need for data connection among automation systems, virtual models and service platforms, while in the current IEC 61499 implementations, access to PLC I/Os and OPC UA servers is not standardized and is often tedious. This paper attempts to propose a design pattern for data connections in IEC 61499-based automation systems, in order to improve the overall portability of the IEC 61499 control application. The proposed solution is based on I/O accessing function blocks combining the existing I/O-accessing capabilities (which are provided by IX, QX, IW and QW FBs, e.g., in 4DIAC) with OPC UA functionalities, forming composite FBs referred in this work as SmartIX and SmartQX, both referred as SmartIO.

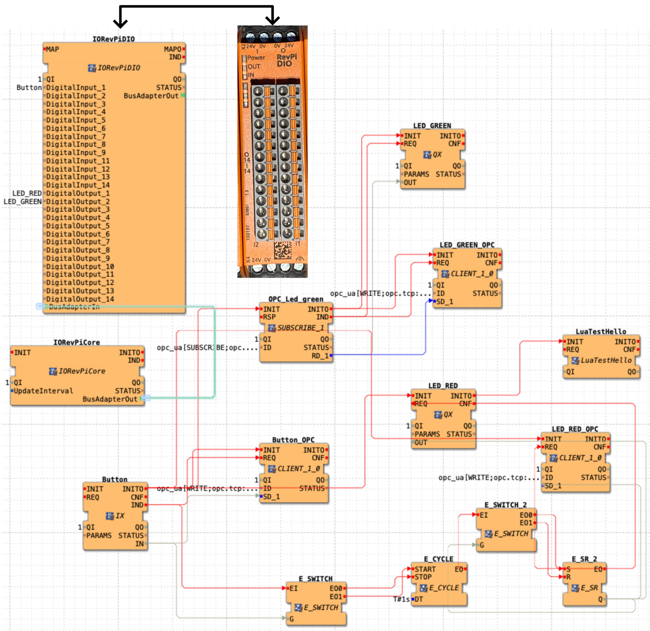

Fig. 3. I/O access implemented in 4diac for a PLC of Kunbus.

III. THE DESIGN OF SMARTIO

SmartIO includes two CFBs, SmartIX and SmartQX. Inside the 4diac development tool, IX is the SIFB to read data from IOs of hardware, while QX is the SIFB to write data to IOs to control actuators. To utilize OPC UA functionalities, e.g., subscribe and publish, customized parameters must be manually defined and set for corresponding FBs. The fact is that I/O-accessing and OPC UA functionalities are separated inside 4diac. However, it would be more natural that I/O-accessing FBs can also subscribe or publish data externally. For example, once the IX reads the data from sensors, the data could also be published spontaneously to the OPC UA server. Currently, in order to support the scenarios mentioned above, additional manual work to control applications is required for engineers. Moreover, those manually modified control applications are not portable in some cases where OPC UA is not supported.

Fig. 4 presents the implementation of SmartIX and SmartQX that are CFBs with customized parameters, including Mode, OPC_Variable, OPC_Address, and I/O. Specifically, users are able to select the mode, i.e., Mode 0 indicating I/O-accessing while Mode 1 indicating I/O-accessing with OPC UA SUBSCRIBE and/or WRITE. OPC_Variable and OPC_Address are two parameters used to generate the OPC UA string. By switching the parameter Mode between 0 and 1, users can flexibly and easily control the data flow from IOs whether to be sent to the cloud platform via OPC UA or not. In addition to improved flexibility, engineers would not need to replace or modify existing control applications in different scenarios. After correctly configuring and mapping IOs and OPC UA nodes, the only thing to do is switching between Mode 0 and Mode 1.

TABLE I
THE MAPPING TABLE OF IOS, OPC UA VARIABLES, AND OPC UA SERVER ADDRESSES.

I/O Dev.	I/O Pin	OPC_Variable	OPC_Address
Sensor1	1	Sensor1_OPC	IP:Port
Sensor2	2	Sensor2_OPC	IP:Port
Sensor3	3	Sensor3_OPC	IP:Port
Actuator1	4	Actuator1_OPC	IP:Port
Actuator2	5	Actuator2_OPC	IP:Port

Table I demonstrates the mapping mechanism among different parameters. For example, if there is one I/O device "Sensor1" attached to I/O pin "1", the data read from this sensor could be automatically transferred to OPC UA server "IP:Port" as a variable "Sensor1_OPC". All the processes are automatic once those parameters are prepared for SmartIX and SmartQX. Those OPC UA variables could also be synchronized automatically at the OPC UA server-side, i.e., cloud-native services could process those data in various scenarios, such as digital twin generation and optimization.

IV. USE CASE

In this section, one use case of the proposed SmartIO will be presented while comparing with the existing solution. This use case requires manufacturing data to be collected from PLCs and sent to the edge/cloud platform for further processing after debugging with virtual models. There are many types of data during manufacturing, such as processing data, sensor data, actuator data, and environmental data. Among others, I/O data, sensor data and actuator data are the two most used data inside control applications, and the I/O data directly reflect the dynamics of whole manufacturing systems.

Therefore, besides processing those data inside control systems, modern industry trends to process those data in edge/-cloud servers with virtual models or digital counterparts. Using the proposed solution, physical devices, OPC UA variable, OPC UA server address, and digital counterparts can be easily customized and grouped together directly in the IEC 61499-based control applications.

A. Virtual Commissioning

Generally, there are three types of VC: Hardware-in-the-Loop (HiL), Reality-in-the-Loop (RiL), and Software-in-the-Loop (SiL). HiL means physical PLCs are used with virtual models, e.g., virtual sensors and virtual actuators, to debug control applications. RiL indicates that soft-PLCs are used with physical devices. Lastly, SiL means that soft-PLCs and virtual devices are used to test developed automation systems. Those three types of VC cover most cases when debugging control applications. One scenario is that physical sensors might be missing when engineers are trying to test their developed automation systems with PLCs. For this case, HiL might be the solution where virtual sensors would be connected to physical PLCs via OPC UA data connections. Many software tools, e.g., FactoryIO and Visual Components, could be used to create virtual plants with virtual sensors. Fig. 5 depicts the architecture of data flow in VC.

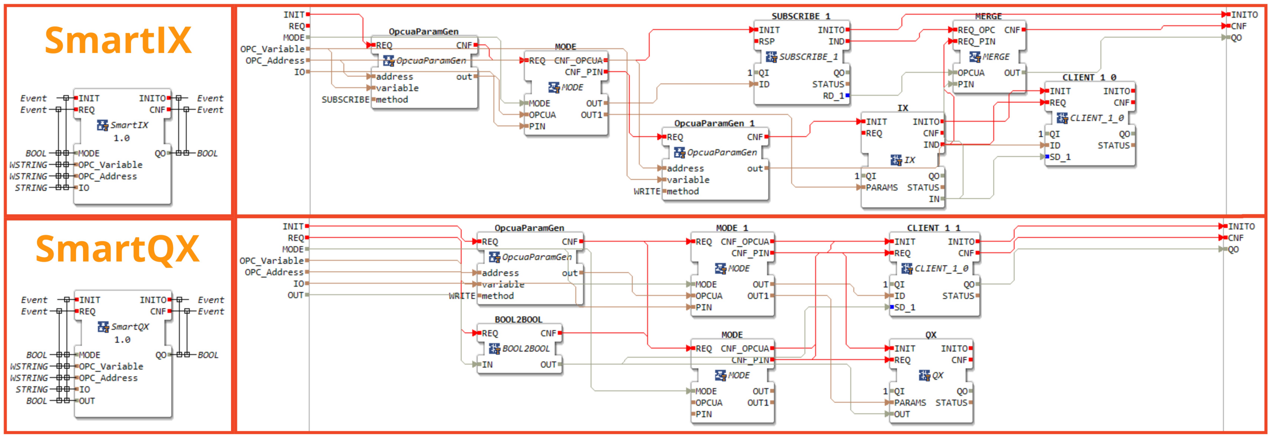

Fig. 4. The design of CFB SmartIX and SmartQX in 4diac IDE.

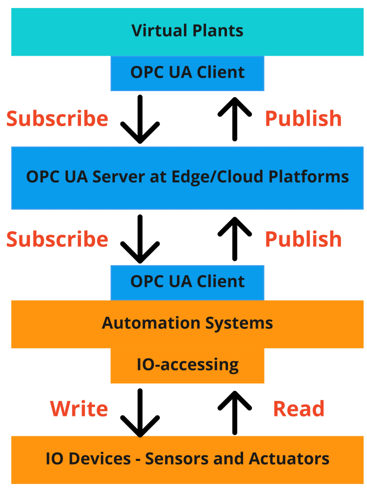

Fig. 5. The architecture of VC and Digital Twin creation.

Usually, hybrid VC can be seen in real developing cases, i.e., HiL, RiL, and SiL might be used along with the development phase. For example, SiL is used firstly to debug the control code and afterward, HiL and RiL are used to test the functionalities of devices with the existing control code. One major problem from this hybrid VC is that existing control applications have to be frequently modified according to the type of VC. These manual modifications introduce additional work and cost to companies. Therefore, the SmartIO is proposed as one of the solutions which would be helpful and flexible for various kinds of VC.

Fig. 6 presents the comparison of our proposed solution and the existing solution. Specifically, the existing solution includes two different scenarios. One scenario is the SiL, where all the input sensor data is from virtual models for debugging. Another scenario is that I/O data from physical devices will be reflected in digital counterparts to synchronize the performance of digital counterparts and real devices.

Usually, switching from debugging to production is error-prone and time-consuming since existing control applications need to be frequently modified. As the example of existing solution in Fig. 6, function blocks ”SUBSCRIBE_1” and ”CLIENT_1_0” are replaced by ”IX”, ”CLIENT_1_0”, and ”QX”. Considering this example is just one sensor and one actuator, if the control systems are complex on a large scale, developers will spend much time modifying existing control systems.

Due to the proposed solution, SmartIO is more flexible when switching from VC to production via the trigger ”MODE”. By switching the trigger ”MODE”, developers can easily switch their automation systems from debugging mode to production mode without changes to existing function block networks.

V. CONCLUSION AND FUTURE WORK

Digital transformation is critical to the success of Industry 4.0, which enables traditional or existing factories and manufacturing businesses to be more efficient, dynamic, and agile. In the context of digital transformation, Information Technologies (ITs) are suggested to integrate with Operational Technologies (OTs), i.e., bringing more advanced features to traditional manufacturing processes. Virtual commissioning and digital twin are two scenarios where ITs and OTs are integrated. For agile development in the industry, virtual commissioning helps to reduce debugging time and cost via soft-PLC, virtual models, and edge/cloud computing resources. For digital twin creation, bridging data of I/O devices to cloud-based virtual plants/sensors/actuators is one of the most significant steps.

All the objectives for achieving digital transformation require efficient, flexible, and secure data connections among automation systems, virtual models, and/or remote cloud/edge-based services. In this paper, we aim to provide a solution for automation system developers to quickly set up data connections between I/O devices and automation systems and between automation systems and remote virtual models. We focus on the engineers using IEC 61499 in 4diac IDE. Traditionally, those engineers should make a lot of manual modifications to existing control applications. We propose one

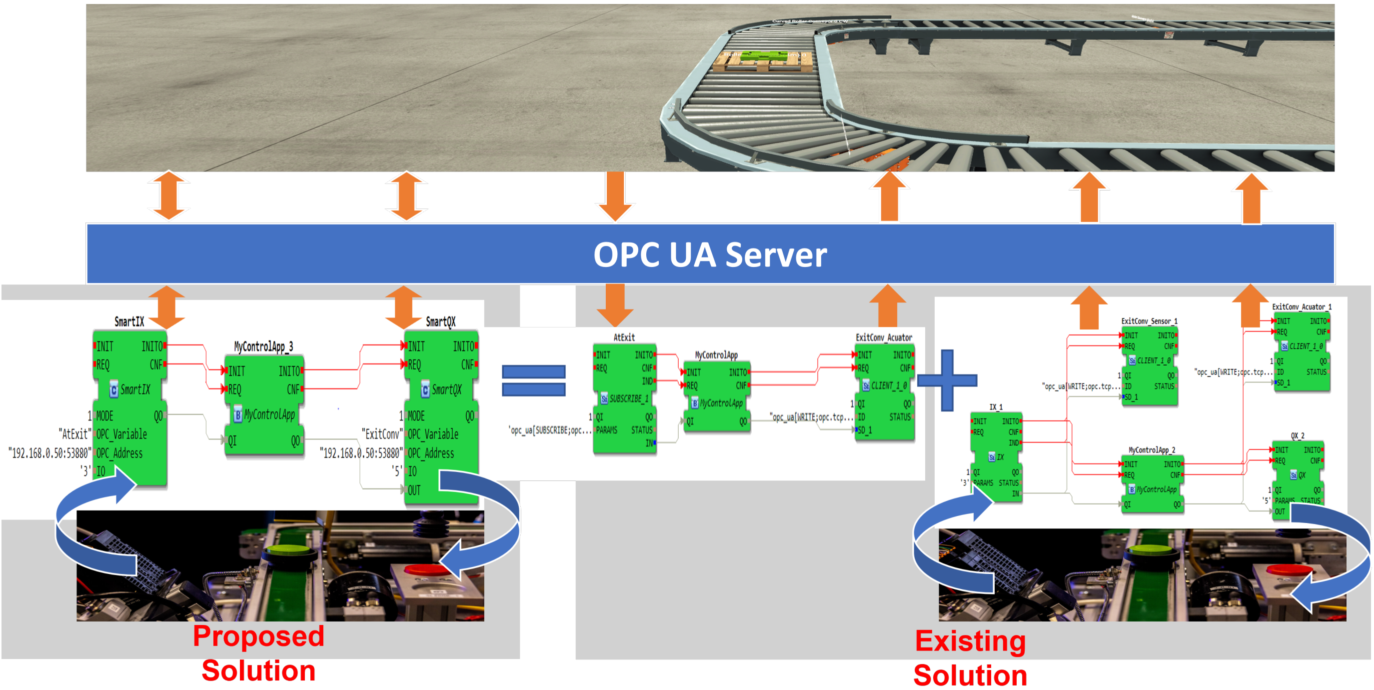

Fig. 6. The comparison between the proposed solution and existing solution.

solution to implement composite FBs where I/O-accessing capabilities are integrated with OPC UA operations, e.g., subscribe and publish. Through the experimental results, this solution proves to improve a lot of efficiency, flexibility, and portability of IEC 61499-based control applications.

Based on our solution, parameters from mapping tables should be input manually to SmartIX and SmartQX. This step could be automated further by auto-processing XML format system files via programs. Therefore, this might be our future work to implement the software tool to automatically process system files with information from I/O-OPC UA mapping tables. The presented prototypes are implemented as composite FBs, which can be later re-implemented as SIFBs for better efficiency.

REFERENCES

[1] V. Vyatkin, *IEC 61499 function blocks for embedded and distributed control systems design*. ISA-Instrumentation, Systems, and Automation Society and Oneida, 2015.

[2] W. Eisenmenger, J. Meßmer, M. Wenger, and A. Zoitl, "Increasing control application reusability through generic device configuration model," in *2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*. IEEE, 2017, pp. 1–8.

[3] B. Chen, J. Wan, L. Shu, P. Li, M. Mukherjee, and B. Yin, "Smart factory of industry 4.0: Key technologies, application case, and challenges," *Ieee Access*, vol. 6, pp. 6505–6519, 2017.

[4] M. V. Garcia, E. Irisarri, F. Pérez, E. Estévez, and M. Marcos, "Opc-ua communications integration using a cpps architecture," in *2016 IEEE Ecuador technical chapters meeting (ETCM)*. IEEE, 2016, pp. 1–6.

[5] Q. Qi, F. Tao, Y. Zuo, and D. Zhao, "Digital twin service towards smart manufacturing," *Procedia CIRP*, vol. 72, pp. 237–242, 2018, 51st CIRP Conference on Manufacturing Systems. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2212827118302580

[6] T. Terzimehic, S. Voss, and M. Wenger, "Using design space exploration to calculate deployment configurations of IEC 61499-based systems," in *2018 IEEE 14th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2018, pp. 881–886.

[7] M. Kumar, V. Shenbagaraman, R. N. Shaw, and A. Ghosh, "Predictive data analysis for energy management of a smart factory leading to sustainability," in *Innovations in electrical and electronic engineering*. Springer, 2021, pp. 765–773.

[8] A. Gunasekaran, Y. Y. Yusuf, E. O. Adeleye, and T. Papadopoulos, "Agile manufacturing practices: the role of big data and business analytics with multiple case studies," *International Journal of Production Research*, vol. 56, no. 1-2, pp. 385–397, 2018.

[9] Q. Qi, F. Tao, T. Hu, N. Anwer, A. Liu, Y. Wei, L. Wang, and A. Nee, "Enabling technologies and tools for digital twin," *Journal of Manufacturing Systems*, 2019.

[10] F. Tao, W. Liu, M. Zhang, T. Hu, Q. Qi, H. Zhang, F. Sui, T. Wang, H. Xu, Z. Huang *et al.*, "Five-dimension digital twin model and its ten applications," *Computer integrated manufacturing systems*, vol. 25, no. 1, pp. 1–18, 2019.

[11] C. Boje, A. Guerriero, S. Kubicki, and Y. Rezgui, "Towards a semantic construction digital twin: Directions for future research," *Automation in Construction*, vol. 114, p. 103179, 2020.

[12] E. Hinchy, N. P. O'Dowd, and C. T. McCarthy, "Using open-source microcontrollers to enable digital twin communication for smart manufacturing," *Procedia Manufacturing*, vol. 38, pp. 1213–1219, 2019.

[13] A. Gavlas, J. Zwierzyna, and J. Koziorek, "Possibilities of transfer process data from plc to cloud platforms based on iot," *IFAC-PapersOnLine*, vol. 51, no. 6, pp. 156–161, 2018.

[14] D. Sieffert and K. Dorofeev, "Generation of inter-plc communication in distributed control systems using IEC 61499," in *2021 IEEE 47th Annual Conference of the IEEE Industrial Electronics Society (IECON)*. IEEE, 2021.

[15] M. Romanato, D. Drozdov, S. Patil, J. Delsing, and V. Vyatkin, "Arrowhead datamanager integration with eclipse 4diac environment," in *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, vol. 1. IEEE, 2020, pp. 1377–1380.

[16] P. Varga, F. Blomstedt, L. L. Ferreira, J. Eliasson, M. Johansson, J. Delsing, and I. de Soria, "Making system of systems interoperable–the core components of the arrowhead framework," *Journal of Network and Computer Applications*, vol. 81, pp. 85–95, 2017.