This is an electronic reprint of the original article.
This reprint may differ from the original in pagination and typographic detail.

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

Towards cloud-based virtual commissioning of distributed automation applications with IEC 61499 and containerization technology

Published in:
Proceedings of 47th Annual Conference of the IEEE Industrial Electronics Society, IECON 2021

DOI:
[10.1109/IECON48115.2021.9589945](https://doi.org/10.1109/IECON48115.2021.9589945)

Published: 13/11/2021

Document Version
Peer reviewed version

Please cite the original version:
Lyu, T., Dwi Atmojo, U., & Vyatkin, V. (2021). Towards cloud-based virtual commissioning of distributed automation applications with IEC 61499 and containerization technology. In *Proceedings of 47th Annual Conference of the IEEE Industrial Electronics Society, IECON 2021* IEEE.
<https://doi.org/10.1109/IECON48115.2021.9589945>

© 2021 IEEE. This is the author's version of an article that has been published by IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Towards cloud-based VC of distributed automation applications with IEC 61499 and containerization technology

Tuojian Lyu*, Udayanto Atmojo*, Valeriy Vyatkin*[†]

*Department of Electrical Engineering and Automation, Aalto University, Espoo, Finland

[†]Department of Computer Science, Computer and Space Engineering, Lulea Tekniska Universitet, Sweden

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

Abstract—Smart Factory is currently becoming a significant research object for the realization of Industry 4.0. Implementing a smart factory requires realizing some advanced features, comparing to traditional factories, such as high flexibility, stability, digital twin and low failure recovery cost. Compared to IEC 61131 standard used in traditional industrial production, IEC 61499 standard can bring more advantages to meet the realization of Smart Factory, such as flexibility, distribution and event-driven.

In this paper, we propose a new Virtual Commissioning (VC) architecture for IEC 61499-based control applications. We build an online collaborative VC platform with the help of cloud computing and containerization technologies. Furthermore, we discuss and propose the utilization of soft-PLC containers from different vendors to meet users' commissioning requirements for different hardware specifics.

Compared with the traditional offline single-person debugging method, our proposed architecture can improve the flexibility, collaboration and efficiency of control software debugging in a more interactive and collaborative style via the online platform we implemented.

Index Terms—IEC 61499, containerization, web, cloud computing, distributed control application

I. INTRODUCTION

Smart Factory as the core content of Industry 4.0, its significance is also gradually increasing. The smart factory gives several clear requirements for industrial automation control applications, such as flexibility, agility and efficiency [1]. In order to achieve shorter innovation and technology life cycle and fulfill the increasing demand for customized products, traditional factories are required to optimize their manufacturing process to be more intelligent and dynamic [2]. How to quickly and efficiently develop, maintain, or modify control applications for a complex and changing production environment is critical for realizing Smart Factory [3]. In a production environment full of dynamic changes, developers in traditional factories have to adjust the production processes and control applications according to devices from different vendors with various hardware facilities. Such agile and flexible changes to existing control software usually cost a lot of time and effort in traditional factories. The high cost of improvement and optimization is a significant reason preventing those traditional factories from being more intelligent. Therefore, an efficient

and effective control application commissioning approach can facilitate the optimization of traditional factories to be more intelligent and competitive in their marketplaces under the guidance of Industry 4.0 [4].

A standard that supports distributed, flexible and event-based application development is another major enabler, besides efficient commissioning, for achieving Smart Factory where the control application intelligence is decentralized across distributed devices [5]. In addition to the need for control applications to be virtually commissioned before deployment, control applications need to be flexible, highly reusable, and vendor-independent [6]. However, a significant issue now is that the IEC 61131 standard [7], as the major control application development language in industrial automation, shows many drawbacks gradually, such as lack of flexibility and portability [8]. Control applications developed based on the IEC 61131 standard are holistic and highly coupled to specific devices. In addition to control software, hardware devices may also need to be frequently reconfigured and upgraded along with environment dynamics and increasing demands of product customization from marketplaces. Even though the IEC 61131 is still the dominant control application development language, its drawbacks prevent this standard from being used in Industry 4.0, requiring a higher level of flexibility and portability.

Another reason that hinders the development of smart factories is that many factories still retain a large number of legacy devices [9]. In order to enhance the advanced flexibility of control systems in smart factories, programmable logic controllers (PLCs) should be upgraded from legacy devices to more powerful and intelligent ones with more computing resources. Nevertheless, an important issue that plant owners do not want to face is how to upgrade the equipment along with legacy control applications developed by IEC 61131 [10]. Different devices often have different hardware specifics, such as inputs and outputs (I/Os), and different software, such as operating system and firmware. Directly redeploying control applications from legacy devices to new devices can damage the hardware or even lead to unrecoverable disasters due to improperly defined interfaces. Before redeploying existing

control applications or deploying newly developed applications, those applications should be fully verified and validated via VC (VC). In this paper, we propose a new architecture to address the inadequate VC problem control software and code utilizing cloud computing, web and containerization technologies. Because of the distributed, event-based, and highly flexible nature of the IEC 61499 standard [11], IEC 61499 will be used as the system-level modeling language in our experiment. It should be noted that IEC 61499, as an enabler for Industry 4.0, is gaining more and more attention and utilization due to some advantages IEC 61131 could not offer. Another enabler for achieving Smart Factory is cloud computing. The computing and storage resources provided by cloud computing platforms can further accelerate the development of Smart Factory [1]. Moreover, containerization virtualization technologies, such as Docker and Kubernetes, are gradually becoming an efficient and flexible development tool in the IT field due to advantages, including lightweight and portability. Although containerization technology is not yet widely used in industrial control applications, the potential benefits that containerization may bring to the Smart Factory should be discussed and noted from many aspects. These potential advantages include faster control application development and deployment, various runtime environments deployment, load balancing and legacy devices upgradation [10]. Considering the potential advantages and features brought by containerization, we utilize a cloud computing platform and containerization technology, Docker in our experiment, to deploy IEC 61499 runtime environments and IEC 61499-based control applications. Finally, a web platform was built to integrate the interfaces connecting developers, various runtime environments and virtual entities.

The outline of this paper is as follows. First, section II gives the literature review about VC. Then, section III presents the design and implementation of our proposed online collaborative VC platform. Next, section IV gives the steps and performance viewed via an experiment with a virtual LED and a virtual button use case. Finally, section V summarizes the performance and contribution of our proposed web-based VC platform and gives our future work.

II. LITERATURE REVIEW OF VC

Achieving agile and flexible manufacturing in Smart Factory should reduce collisions and frictional losses during the commissioning of control software and code from multiple disciplines (i.e., mechanical, electrical and software engineering) [12]. These collisions and frictions during commissioning will hinder the development and upgrading of control applications. If manufacturers want to keep their competitiveness in the ever-changing marketplace, they have to find a better approach to improve their agile production capability with mass customized production [4]. A more advanced approach to commissioning and upgrading control applications in a real production environment is VC. VC refers to the virtualized commissioning of production systems and software in a simulated environment without impacting or damaging the actual

hardware/devices. By modeling the production environment and behavior, VC can lead to shorter product development life cycles, more agile manufacturing and lower development cost. In general, VC can be categorized in three ways: (1) Hardware-in-the-loop (HIL), (2) Reality-in-the-loop (RIL), and (3) Software-in-the-loop (SIL) [12]. Different VC methods aim to solve different problems in various scenarios.

In HIL, actual PLCs will be used to perform automation applications. Virtual plants or plant emulators will be connected to actual PLCs for testing control software results from PLCs [13]. Control applications deployed to the PLCs will be compiled and executed, and then the results will be output to corresponding I/Os. All the I/Os operations can be mapped in the virtual plant by simulating the hardware via sensors and actuators. Because HIL does not need to be connected to actual manufacturing hardware, this HIL VC method is a good way to verify and test the actual behavior of control applications in actual PLCs and monitor how it interacts with the simulated plant devices. Another advantage is that HIL does not lead to unrecoverable or costly damage to actual manufacturing devices. On the other hand, as pointed in the work [14], in the field of electric energy systems, utilizing HIL is a good way to VC but requires a high level of stability and accuracy from the simulation results. Usually, RIL or SIL can improve efficiency and reduce the cost of debugging control applications to a greater extent than HIL. Moreover, another disadvantage of HIL is the higher commissioning cost, design and programming conflicts, and even the possibility of permanent damage to the PLCs [15]. For example, in HIL, it is costly and difficult for developers to commission their control applications in various PLCs since they usually have different hardware from different vendors. Under the context where existing control software should be ready for migration due to dynamic manufacturing environments, HIL is not the VC method used in our online collaborative VC platform.

The RIL approach uses actual plant devices and virtual PLCs to commission control applications. Because actual production hardware is used, RIL has apparent advantages and disadvantages. Compared to HIL, RIL has advantages, such as being a decision support tool for diagnosing and debugging devices, maintaining existing control software, and exploring alternative system-control strategies based on existing plant devices [16]. However, a disadvantage is that the method runs the risk of costly repairs and even causing permanent damage to manufacturing devices.

Unlike HIL and RIL, SIL refers to commission control applications with virtual PLCs and simulated plant hardware. Since SIL uses completely virtualized PLCs and devices, the method will provide developers with great flexibility and efficiency via reducing development cost and life cycles [17]. Specifically, developers can validate and upgrade control software with SIL by interacting directly with the digital factory model via a local development PC. SIL has two significant advantages over HIL and RIL. The first advantage is the versatility of PLCs. Because of the virtual PLCs used in SIL, it is possible to commission applications against

Fig. 1. The framework of the web-based collaborative VC platform.

different brands of PLCs with different hardware specifics. The second advantage is the high efficiency. Because there is no real hardware or equipment involved, SIL will improve the efficiency but will not result in costly recovery and longer downtime for the actual plant. Based on the SIL used in our implementation, we aim to utilize other technologies, such as cloud computing, containerization and web, to further boost its flexibility and efficiency for control software VC in Smart Factory.

III. IMPLEMENTATION

This paper aims to build an online collaborative VC platform for IEC 61499 standard-based automation applications using containerization, web, and cloud computing technologies to boost VC flexibility and efficiency. This VC platform can provide users with the virtual simulation environment and virtual PLCs for commissioning of control software or code. A more advanced feature is that this VC platform supports the customized deployment of soft-PLC containers from different vendors and various IEC 61499 runtime environments. To achieve this goal, we designed three tasks as follows:

- 1) Build IEC 61499 runtime images.
- 2) Create an online, soft real-time collaborative web server.
- 3) Connect virtual devices and soft-PLCs to the web.

Fig. 1 presents the framework of integrating these three tasks for creating this VC platform in a big picture. The left side of Fig. 1 is the local development workstation for developing control applications using 4diac IDE, the middle is the online collaborative web that provides the VC platform, and the right side represents the computing cluster including computing nodes with soft-PLC containers.

A. IEC 61499 Runtime Images

Because of multiple benefits of the IEC 61499 standard, such as distribution and high flexibility, control applications based on the IEC 61499 standard will be used as the system

level modeling language. We use the open-source software 4diac as the IDE and Forte as the runtime for developing IEC 61499 control applications. It should be noted that IEC 61499 runtime environments are usually different vendors to vendors. Furthermore, for a same runtime, e.g., FOrte, different parameters should be set during runtime compilation according to hardware specifics and operating systems. As a result, automation application developers often need to compile and manage multiple runtimes to virtually commission control software. This is the primary reason that hinders the further development of the IEC 61499 standard. To solve the difficulties caused by hardware gap, we use containerization technology to create images of different IEC 61499 runtimes compiled for different PLC vendors. The runtime images for different PLCs are tagged accordingly and stored in the cloud for future use. These runtime container images can be easily used in containerized software, such as Docker and K8s. The use of containerization technology provides many benefits, including rapid remote creation, deletion, modification and scaling.

B. Online Collaborative VC Platform

The web server in Fig. 1 is an essential part of our online collaborative VC platform, providing an online workspace for multiple users to cooperate. In addition, users can customize their VC simulation environment via various soft-PLC containers in selected computing nodes. There are three critical components within this VC platform as follows:

- 1) Interactive Front Page: The web page, as presented in Fig. 2, behaves as the interface between users and the VC platform. Users can customize their VC models and configuration according to various requirements from their projects via the web page.
- 2) Web Server: This web server is used as a coordinator between the front page and Worker in remote VMs with functionalities, such as establish communication

Fig. 2. The front-end web page functionalities and layout.

between each user and remote computing nodes, and organize multiple remote Workers.

- 3) Worker Client: Each Worker is also a web client to cooperate with the web server to start/update/delete soft-PLC containers and provision cloud VMs via inputs from users. In addition, the Worker will monitor specified GPIO pin so that GPIO data can be read from or written to a specified GPIO pin via the file path (/sys/class/gpio/gpio*/value). The number of GPIO pin can be obtained from the manuscript of control applications.

The work flow is, first, the front-end page, as presented in Fig. 2, will be sent to the users' browser to provide an interface. The web page as the interface will provide information about the runtime containers, the node names and port numbers. In addition to the ability to remotely customize and create runtime containers, the front-end page can also provide monitoring information for runtime containers running on all computing nodes, including IP addresses, port numbers, images names and node names. The third feature provided by the front web page is an interactive virtual device panel. This part is the same as the definition of the virtual factory model in traditional VC, which means that users can monitor the performance of digital twins and support the online operations, clicking a virtual button to control a virtual LED in a soft real-time manner through the virtual device panel.

In addition to handling the front-end interaction page, our implemented web-based VC platform also includes a web server and Worker clients, as shown in Fig. 3. The web server can send the runtime creation and configuration requests from the front-end page to the Worker clients corresponding to a

specified computing node. In other words, if the user specifies the runtime image, port number and node name on the front-end page, the web server will receive these configuration requests and automatically generate the corresponding container creation commands and send them to the Worker clients. When the Worker clients have completed requests from web servers, they will return the monitored soft-PLC containers and computing nodes' information to the front-end page.

Such bi-directional communication allows developers to significantly improve the efficiency and flexibility of commissioning environment configuration before VC. Specifically, multiple users across different regions and smart factories can operate collaboratively through this online VC platform to commission the same project or control software. Moreover, according to different scenarios and needs, VC environments, including the simulation models and virtual PLCs, can be customized.

Fig. 3. The collaboration between the web server and distributed workers.

C. Virtual Device/Plant

This paper uses SIL, which means that virtual PLCs and virtual factories will be used instead of actual entities. The virtual PLCs will be created via corresponding soft-PLC Docker images. Fig. 4 presents the method how to configure devices inside 4diac IDE by using soft-PLC Docker containers running in the cloud with public VM IP addresses and port numbers.

Since the focus of this paper is to propose a new online commissioning solution based on web and container technology, not the creation of virtual factories, this paper will not specifically introduce the creation of virtual factories or virtual devices. Just as a demonstration, a simple virtual device including a virtual button and a virtual LED will be used as the test bed. The virtual device will be integrated into a web page to monitor the execution of the automation application code in the virtual PLCs. This virtual plant with both inputs and outputs can represent a wide range of automation applications and can be extended to a more complex control application in future.

Fig. 4. The devices configuration inside 4diac IDE via soft-PLCs in the cloud.

IV. EXPERIMENT

There are three most essential components in our proposed architecture, including IEC 61499 runtime images, web servers and virtual PLCs. We have first tested them separately for each of these three components and then tested the whole implementation in a completed VC loop. Fig. 5 shows the VC and data flow between the different parts via different colored lines. This experiment aims to control a virtual button to change a virtual LED on and off in a soft real-time, online collaborative manner on a web page. The virtual hardware, containing the button, the LED and the controller, can be used as a digital twin or a virtual factory as a simplest demonstration. The automation application that controls this virtual device (buttons and LEDs) was developed and deployed at 4diac. To achieve the VC via the web server, we conducted three experiments, 1) soft-PLC containers deployment, 2) enabling virtual GPIO of VMs, and 3) monitoring the virtual LED and button.

A. Soft-PLC Containers Deployment

In order to properly run distributed control programs based on the IEC 61499 standard, runtimes running IEC 61499 blocks need to be appropriately deployed to microcontrollers, PLCs or virtual machines. In the 4diac IDE, the device's IP and runtime port numbers need to be correctly configured. This step is called device configuration. One example of the device configuration in 4diac IDE is presented in Fig. 4. Each IP and port number can only correspond to a defined runtime. In our experiment, we deployed two cloud VMs with IP addresses 35.228.36.148 and 35.228.50.58. Since the control application in this experiment is simple, we run only one runtime container on each VM, and the port numbers of these two runtime containers are 61499 and 61500, respectively. After deploying runtime containers to the cloud with the correct configuration, it will be possible to configure two devices in 4diac IDE that support running IEC 61499-based distributed control applications via 35.228.36.148:61499 and 35.228.50.58:61500.

B. Enabling Virtual GPIO of VM

In the 4diac IDE, FBs such as QX and IX interact with the IO of hardware devices, respectively. Different microcontrollers and PLCs may have different hardware architectures. Different hardware and operating systems determine the way QX and IX interact with the underlying IO. For example, if a Raspberry Pi is used as the microcontroller, the QX or IX interacts with the Raspberry Pi's IO in a three-step process. The three steps are 1) identifying and activating the corresponding pin via PARAMS of QX or IX, 2) setting `/sysfs/class/gpio/gpio*/direction` to out (for writing data via QX) or in (for reading data via IX) to set the direction of the IO data, and 3) setting `/sysfs/class/gpio/gpio*/value` to read or write data. In this experiment, a Debian-based VM is used to simulate a microcontroller to provide computing resources. Because VMs are different from the actual microcontrollers, VMs do not provide the IO interaction capability. Therefore, a virtual GPIO interaction module (gpio-mockup) needs to be compiled and imported into the virtual machine. This mockup is the one that comes with the Linux kernel Linux-5.11.13 used in our experiment. By compiling and importing this gpio-mockup module into the VM kernel, QX and IX can read and write virtually by providing a virtual GPIO driver.

C. Monitoring the Virtual LED and Button

QX and IX can interact with the virtual driver of gpio-mockup on the cloud VM. Data written by QX or read by IX is passed through a virtual file path `/sys/class/gpio/gpio*/value`. Therefore, to transfer QX results and IX results to the web page in real-time, we implemented a Worker script via JavaScript. The role of this Worker script is to monitor all active GPIO pins in the distributed control application in soft real-time. In the IEC 61499 based control application, each QX and IX corresponds to the data of a defined GPIO pin via PARAMS. The Worker script transfers the GPIO data from the `soft-gpio/gpio*/value` to the front web page. Worker utilizing Socket.IO library establishes a connection to the web server and then displays the monitored GPIO data, LED's on or off. Similarly, for the interaction between the user and the front page button, the web server also uses Socket.IO to pass the button click command to the remote Worker script, and then the Worker writes the received data to the specified GPIO pin.

V. CONCLUSION AND FUTURE WORK

This paper proposed a web-based online collaborative VC platform for IEC 61499-based control applications utilizing web technologies and containerization. Specifically, with many advantages of containerization, such as high flexibility and portability, we use containerization technology to create IEC 61499 runtime images. The IEC 61499 runtime images can be deployed to cloud platforms or edge computing platforms. For 4diac IDE developing IEC 61499 control applications, these runtime containers deployed to cloud or edge computing platforms are indistinguishable from local microcontrollers or PLCs. In addition, we use the library provided by Socket.IO

Fig. 5. The workflow of controlling an LED via a button on a virtual device on the web page.

to implement the communication between the web server and Worker for transferring the monitored GPIO data (LED and Button). In the experiment, one virtual device with a LED and a button, as a simple case with input and output, is used to demonstrate the feasibility of the architecture we implemented. However, the control application of the LED in the experiment is very simple. We will optimize this online VC platform with more complex control applications in future work. Corresponding to the more complex control applications, we will try to integrate more advanced plant modelling software, e.g., FactoryIO, to further improve our online collaborative VC platform.

REFERENCES

- [1] 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*, 6:6505–6519, 2017.
- [2] Zhan Shi, Yongping Xie, Wei Xue, Yong Chen, Liuliu Fu, and Xiaobo Xu. Smart factory in industry 4.0. *Systems Research and Behavioral Science*, 37(4):607–617, 2020.
- [3] Mehrshad Mehrpouya, Amir Dehghanghadikolaie, Behzad Fotovvati, Alireza Vosoughnia, Sattar S Emamian, and Annamaria Gisario. The potential of additive manufacturing in the smart factory industrial 4.0: A review. *Applied Sciences*, 9(18):3865, 2019.
- [4] Chi G Lee and Sang C Park. Survey on the virtual commissioning of manufacturing systems. *Journal of Computational Design and Engineering*, 1(3):213–222, 2014.
- [5] Valeriy Vyatkin. Iec 61499 as enabler of distributed and intelligent automation: State-of-the-art review. *IEEE transactions on Industrial Informatics*, 7(4):768–781, 2011.
- [6] Neil Higgins, Valeriy Vyatkin, Nirmal-Kumar C Nair, and Karlheinz Schwarz. Distributed power system automation with iec 61850, iec 61499, and intelligent control. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 41(1):81–92, 2010.
- [7] M. Tiegalkamp and K. John. *IEC 61131-3: Programming industrial automation systems*. Springer, 2010.
- [8] U. Atmojo and V. Vyatkin. Programming future industrial cyber physical systems: A review and perspective on the state of the art. 2018.
- [9] Prasanna Kumar Illa and Nikhil Padhi. Practical guide to smart factory transition using iot, big data and edge analytics. *Ieee Access*, 6:55162–55170, 2018.
- [10] Thomas Goldschmidt, Stefan Hauck-Stattelmann, Somayeh Malakuti, and Sten Grüner. Container-based architecture for flexible industrial control applications. *Journal of Systems Architecture*, 84:28–36, 2018.
- [11] V. Vyatkin. The iec 61499 standard and its semantics. *IEEE Industrial Electronics Magazine*, 3(4):40–48, 2009.
- [12] Tobias Lechler, Eva Fischer, Maximilian Metzner, Andreas Mayr, and Jörg Franke. Virtual commissioning—scientific review and exploratory use cases in advanced production systems. *Procedia CIRP*, 81:1125–1130, 2019.
- [13] H. Schludermann, T. Kirchmair, and M. Vorderwinkler. Soft-commissioning: hardware-in-the-loop-based verification of controller software. In *2000 Winter Simulation Conference Proceedings (Cat. No.00CH37165)*, volume 1, pages 893–899 vol.1, 2000.
- [14] Alexander Viehweider, Georg Lauss, and Lehfuss Felix. Stabilization of power hardware-in-the-loop simulations of electric energy systems. *Simulation Modelling Practice and Theory*, 19(7):1699–1708, 2011.
- [15] Jiewu Leng, Man Zhou, Yuxuan Xiao, Hu Zhang, Qiang Liu, Weiming Shen, Qianyi Su, and Longzhang Li. Digital twins-based remote semi-physical commissioning of flow-type smart manufacturing systems. *Journal of Cleaner Production*, 306:127278, 2021.
- [16] Konstantinos Mykoniatis and Gregory A Harris. A digital twin emulator of a modular production system using a data-driven hybrid modeling and simulation approach. *Journal of Intelligent Manufacturing*, pages 1–13, 2021.
- [17] M Ben Ayed, Lilia Zouari, and Mohamed Abid. Software in the loop simulation for robot manipulators. *Engineering, Technology & Applied Science Research*, 7(5), 2017.