Reconfigurable Conveyor Transfer System using IEC-61499 Function Blocks

Leonardo Mendonça, Gustavo Funchal, Frederico Fagundes and Paulo Leitao

Abstract In the 4th Industrial Revolution era, automation processes operate in a more modular, flexible, and reconfigurable manner, leveraging decentralized decision-making and distributed control. The IEC-61499 Function Block (FB) technology offers a suitable approach to implementing distributed automation control systems. This paper describes the development of a dynamic and on-the-fly reconfiguration control system based on the IEC-61499 standard, applied in a conveyor transfer system case study to automatically adapt its operation to face changes in the number and position of the conveyor modules. The developed control system considers a Python-based FB network using the DINASORE framework. The experimental tests showed promising results regarding the ability of the system to adapt to condition changes automatically and on-the-fly, as well as scalability and robustness.

1 Introduction

The 4th Industrial Revolution, represented by Industry 4.0, introduces a new era in the manufacturing sector, driven by the integration of digital technologies, intelligent automation, and interconnected systems. The implementation of Industry 4.0 requires decentralized decision-making and distributed control across the shop floor, enabling a more cooperative and flexible production environment [1]. Therefore, industrial systems can be successfully implemented by connecting the industrial as-

---

Leonardo Mendonça, Gustavo Funchal e Paulo Leitao
Research Centre in Digitalization and Intelligent Robotics (CeDRI), Laboratório para a Sustentabilidade e Tecnologia em Regiões de Montanha (SusTEC), Instituto Politécnico de Bragança Campus de Santa Apolónia, 5300-253 Bragança, Portugal e-mail: `{leonardomendonca, gustavofunchal,pleitao}@ipb.pt`

Frederico Fagundes
Federal Center for Technological Education of Minas Gerais (CEFET-MG), Araxá, Minas Gerais, Brazil e-mail: `fredericofagundes@cefetmg.br`

sets, including networked smart objects, cyber-physical assets, associated generic information technologies, and cloud or edge computing platforms [2]. As a result, the large amount of collected data can feed analytical solutions, enabling the real-time and intelligent data analysis, improving the collaboration between components belonging to the industrial process, leading to optimal industrial operations [2, 3].

At the core of this new perspective lies the Cyber-physical Systems (CPS), which is a new structure that emphasizes the strong interlink and interdependence of the digital/cyber and physical components, decentralizing and distributing the computation entities among a network of nodes and subsystems [4, 5]. These systems can interact with humans and be engaged in a machine-to-machine communication, enabling them to make decisions and optimize the production system [6].

To ensure the optimal functionality, the control logic behind the automation process needs to be standardized, which was initially possible by following the IEC-61131 standard [7] that provides the guidelines and directives of how industrial control applications must be implemented, including the variables definition, data types, and programming languages. The IEC-61499 standard [8] provides a highly accessible framework for implementing distributed automation control systems, allowing the seamless integration of diverse controllers from different manufacturers into a single program and improving the aspects of portability, configurability, and interoperability. The central element of the IEC-61499 standard is the use of Function Block (FB), known for its graphical representation and characterized by a block structure with inputs and outputs specifically designed for events and data handling, and an embedded program within the block that orchestrates its functionality [9]. Therefore, the wide dissemination in the industrial environment is due to the easy and modular implementation of such systems, allowing the creation of blocks that meet the system's specific needs.

Having this in mind, this paper presents the use of an FB approach to develop a dynamic reconfigurable transfer conveyor system based on Fischetechniks conveyor modules. For this purpose, the developed solution uses the Dynamic Intelligent Architecture for Software and Modular Reconfiguration (DINASORE) platform that considers Python to develop FB-based systems [10]. This implementation forms a distributed control network within the conveyor system, allowing the development of a self-organized mechanism that adapts dynamically to changes in the conveyor's module sequence. The experimental results showed that the system could self-reconfigure on-the-fly to face the detected changes, i.e., without the need to stop, reprogram, and start the system again. Robustness and scalability are characteristics also exhibited by the developed system.

The remaining paper is organized as follows: Section 2 presents the related work in this field, mainly focusing on the dynamic reconfiguration using the IEC-61499 standard. Section 3 describes the experimental case study related to the cyber-physical conveyor system and overviews the DINASORE framework to develop FB-based systems. Section 4 presents the developed reconfigurable FB-based system, and Section 5 discusses the achieved experimental results. Finally, Section 6 rounds up the paper with the conclusions and points out some future work.

2 Related Work

In industrial automation systems, the ability to dynamically adjust the system configuration and behavior in real-time is a pivotal requirement. Approaches like Multi-Agent Systems (MAS), Distributed Control Systems (DCS), and IEC-61499 emerge as focal points in addressing this demand. Therefore, IEC-61499 holds a unique position due to its ease of graphical programming and its modular design, which simplifies the system development and maintenance, in addition to its compatibility with Programmable Logic Controllers (PLCs), making it especially applicable and practical for industrial settings.

In this context, the use of supervisory control theory to reconfigure systems is discussed in [11], involving the temporary replacement or modification of FBs to ensure the system safety for device management and reconfiguration of control applications. Furthermore, [12] developed a standard-compliant reconfiguration methodology, mapping the existing system and adding/removing FBs to achieve the desired functionality. This approach uses management blocks within devices for issuing commands, ensuring the efficient system reconfiguration. A similar approach was introduced by [13], although using the OPC-UA, showing the potential to offer an open configuration interface for these devices. Furthermore, an alternative strategy is presented by [14] that relies on predefined functions to build FB networks, enhancing the precision and service continuity, while [15] further refines this approach with the Priority Ceiling Protocol (PCP) for addressing scheduling intricacies. Additionally, [16] focuses on real-time fault detection and program logic adaptation by developing a Reconfiguration Architecture for Fault Handling (RAFAH), although with constraints on fault types and FB libraries. Meanwhile, the concept of dynamic adapter connections discussed by [17] contributes to enable the run-time reconfiguration of component connections, addressing the limitations of static and design-time connections.

Nevertheless, [18] introduces the concept of reconfigurable FBs as an extension to the IEC-61499 standard, emphasizing the significance of dynamic reconfiguration within FBs and presenting a tool chain for modeling, formal verification and quantitative analysis of reconfigurable distributed control systems. The proposal includes a dynamic Master-Slave Execution Control Chart (MSECC) to separate the reconfiguration and control models. Nevertheless, developing the FBs using the Execution Control Charts (ECC) can introduce complexities in large projects, potentially impacting system resources and verification processes.

The implementation of a dedicated reconfiguration block capable to dynamically identify condition changes and adapt the system operation in real-time is a promising perspective. In this context, this work proposes an innovative approach utilizing a Python-based FB reconfiguration approach, seeking to implement a dynamic reconfigurable conveyor transfer system.

3 Experimental case study

This section describes the experimental case study and the DINASORE framework for developing the FB-based system.

3.1 Description of the Case Study

The case study is related to a conveyor transfer system based on modular Fischertechnik conveyors, as shown in Fig. 1. Each conveyor consists of two main components: a cyber part running on the Raspberry Pi 3 Model B+ computational platform (providing 28 GPIOs, from which four inputs and one output were used), and a physical part encompassing light sensors, indicating the part's arrival and departure from the conveyor, alongside with a motor powering the belt, both operating at 24V. The primary objective of this system is to transfer a part from its starting point to the target endpoint.

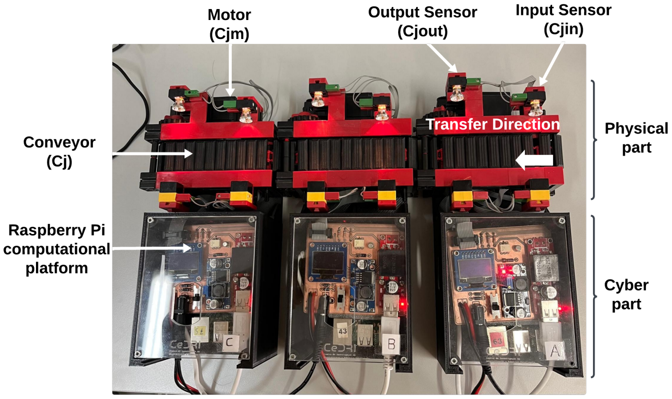

Fig. 1 Modular conveyor transfer system.

The system's functionality is modeled using the Petri nets formalism, illustrated in Fig. 2, where all the conveyors modules present the same functionality. Briefly, when a part reaches the first conveyor, the input sensor is activated (transition *t1*), starting the conveyor motor and transferring the part, while the trigger of the output sensor transmits a signal to activate the next conveyor (transition *t10*). When the next conveyor receives this signal (identified by transition *t3*), it starts its motor, and when the part reaches its input sensor, a signal is sent to the previous conveyor (transition *t6*), informing that its motor can be turned off. This process is repeated until the part reaches the end of the last conveyor, with transition *t8* triggering the motor's deactivation.

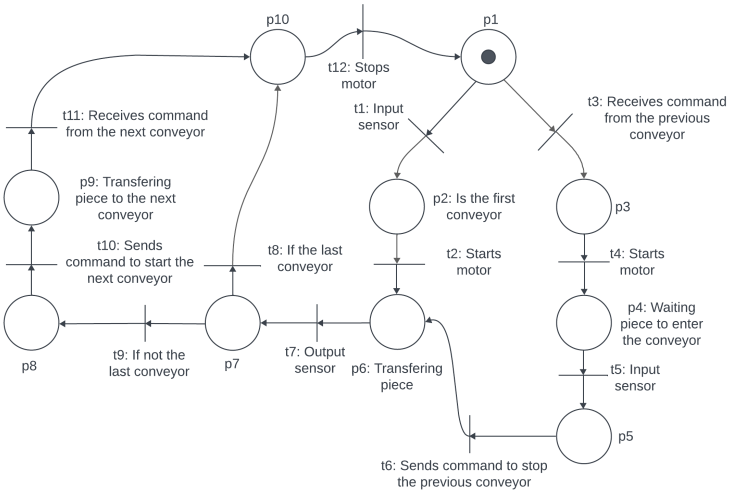

Fig. 2 Behavior model of the conveyor transfer system (adapted from [19]).

Moreover, the fundamental objective remains to empower conveyor systems to adapt seamlessly to changes on-the-fly, facing the addition of new modules, removal of existing modules, or reconfiguration of the order of the modules.

3.2 DINASORE Framework

DINASORE is a framework that facilitates the design, implementation, and management of FB-based control systems in a distributed manner, enabling the integration of Python into FBs. By utilizing a structured configuration and Python programming, it facilitates the real-time adaptation and efficient control systems [20], thus allowing an on-the-fly reconfiguration.

Moreover, it allows the orchestration of a pipeline of FBs across an array of devices, establishing a distributed and harmonized control paradigm [10]. Crafting FBs within the DINASORE framework entails the creation of two essential files: the *.xml* file that establishes the structural foundation of these FBs and the Python file that contains the coded implementation defining the functionalities of the FBs. The creation of a network of FBs requires the use of a *.txt* file called *data_model* that is specifically crafted for each module containing commands to create the blocks, make the connections between them, and set up the necessary parameters. The configuration defined by using the DINASORE platform establishes a network of FBs across the nodes, simplifying the integration of new devices and ensuring seamless communication.

4 Implementation of the Reconfigurable Control System

This section describes the practical implementation of the FBs-based control system for the conveyor transfer system case study. Fig. 3 illustrates the control system's structure, presenting the physical layer that supports the transfer of parts, the cyber part that uses FBs to implement the control logic for each conveyor transfer module, and the communication layer that ensures the interconnection between the individual control modules.

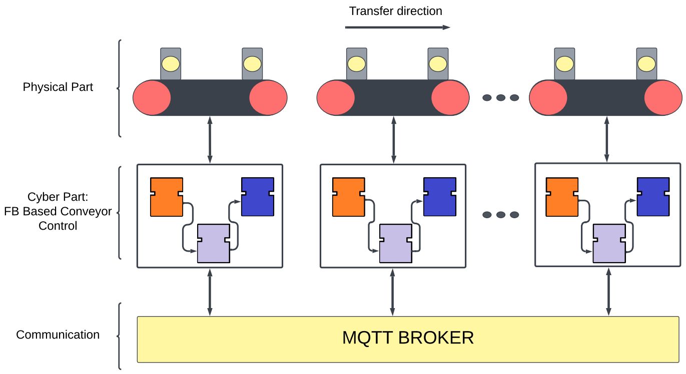

Fig. 3 Structure of the FBs based control system.

4.1 *Control of Individual Conveyor Modules using FBs*

The control of each conveyor module is managed by a network of FBs designed to perform specific functions. These FBs handle diverse tasks, including monitoring the sensor data to track the movement of parts, controlling the conveyor motor operation and the communication between modules. Additionally, the FBs ensure that the module adapts to operational changes and maintains the overall efficiency.

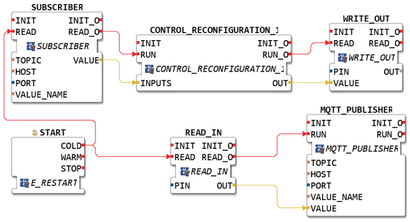

Fig. 4 FB network to control an individual transfer conveyor module (no reconfiguration considered).

4.2 Communication between Individual FB-based Control Transfer Systems

Since the conveyor transfer system is built up of modular conveyor modules, a direct connection between their FB-based control modules is not possible, requiring the use of a communication middleware that easily and efficiently manages and routes the exchanged messages. Even the OPC UA standing as the most widely used protocol in industry, MQTT also offers protocols that are well-suited for industrial environments due to its well-established reliability, efficiency, and scalability. MQTT is a lightweight publish-subscribe message pattern platform, being in this work used Mosquitto as the MQTT broker that provides robust message routing and management. The exchanged data were formatted according to JSON schema, which can be interpreted by various technologies that can be used in self-organization applications, facilitating the system interoperability.

To implement the communication between FB modules, the development of two FBs was required: SUBSCRIBER, used to subscribe to MQTT topics for receiving sensor and connection information, and MQTT_PUBLISHER, used to publish sensor values and connection information on MQTT topics. Moreover, each SUBSCRIBER FB handles one topic of the MQTT broker, increasing the message processing efficiency. Therefore, each conveyor module sends input information to the MQTT broker by publishing on the proper topic and subscribes to the topics to receive information related to the other conveyor modules, as shown in Fig. 5.

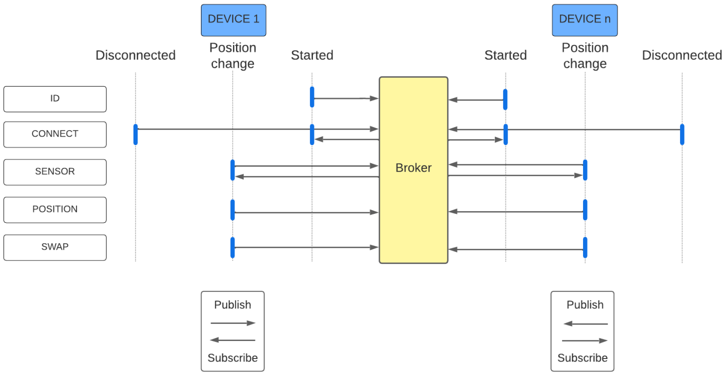

Fig. 5 Message routing scenarios using the MQTT broker.

Basically, the conveyor modules communicate during three situations: i) in the system initiation, sharing the device's ID (ID) and the information about its connectivity (CONNECT), ii) in the part transfer, exchanging the sensor input and output data (SENSOR) to learn the sequence order and to indicate the transportation of the part, and sharing information about its position (POSITION) and the position's change (SWAP), and iii) in the disconnection of the device.

4.3 Reconfiguration Mechanism

Upon receiving the information through the MQTT broker, the FB-based control module checks for possible changes and updates the output automatically to adapt and transfer the part. Furthermore, the development of the FBs depicted in Fig. 6 allows the system to implement the self-organization capability.

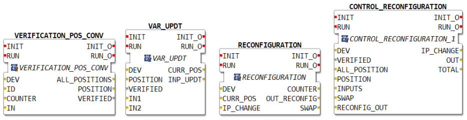

Fig. 6 Structure of the reconfiguration FBs.

The process begins with the VERIFICATION POS CONV FB, which scans the positions of connected conveyors and assigns them in a vector that reflects their physical sequence. This scanning process involves monitoring the order in which the conveyors' input sensors are triggered during a part test. Specifically, the FB receives sensor values and dynamically assigns an ID to each connected conveyor within the vector based on the sequence of the sensor's activation. This vector is then shared with all connected devices, allowing each conveyor to know its position in the sequence, as well as the position of the other conveyors. If any changes are detected, this FB reassigns the conveyors' positions to ensure that the system's representation matches the physical layout. In addition, the system can handle more than one part through a part-counting mechanism. Following this, the VAR_UPDT FB updates the system variables using the verified sequence information, adjusting sensor values to match the new configuration.

However, it is crucial to develop a dedicated FB capable of managing the behaviors of the conveyor modules and identifying changes on-the-fly. For this purpose, the CONTROL_RECONFIGURATION FB, is responsible for managing the behavior of the conveyor modules according to the sequence position. This FB implements two strategies to verify the reconfigurations: i) the flag strategy involves raising a flag when a part moves from the first to the last conveyor. The lack of a raised flag, coupled with the arrival of a part on a different conveyor, signals a configuration change in the first position, and ii) the timeout strategy involves the activation of a timer when a conveyor's output sensor is triggered and resetting it when the next conveyor's input sensor is activated. Any configuration change during this time frame invalidates the condition. The reconfiguration information is sent to the RECONFIGURATION FB, which is responsible for updating the outputs according to the changes, activating the verification process, and sending the reconfiguration information to the broker, maintaining the system's functionality.

Interconnecting the previously discussed FBs forms a comprehensive control system tailored for one conveyor module, illustrated in Fig. 7, that can be easily deployed in the other conveyors. The FB network forms a system that supports the changes without the need to stop, reprogram, or restart, implementing the functionality of on-the-fly reconfigurability, scalability, and interoperability.

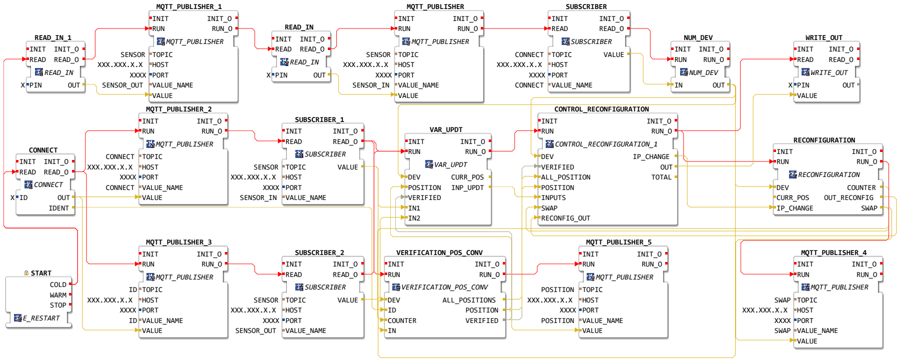

Fig. 7 FB control network to one conveyor module, applying self-organization in the conveyor transfer system.

5 Experimental Results

This section provides comprehensive insights into the achieved experimental results concerning reconfigurability and scalability, detailing the various scenarios tested using the conveyor transfer system.

5.1 Reconfigurability

To verify the reconfigurability of the conveyor system, an evaluation test was conducted with three independent conveyors, assessing its ability to adapt dynamically to changes in the conveyor configuration. A video showing the system operation is located at https://youtu.be/2N5E6h_n16Q. Here, the system was initiated with a single conveyor and was progressively integrated with additional conveyors, changing their positions and even removing conveyors. The experimental tests clearly show the system's adaptability and the capability to self-reconfigure on-the-fly, i.e., without stopping, reprogramming, and restarting the control system.

5.2 Scalability

The system's capability to handle increased message demands and maintain an optimal performance was evaluated through stress and performance tests, utilizing the Apache JMeter to simulate high message volumes within the MQTT broker topics, in which the MQTT broker was deployed on a Raspberry Pi 3 model B+.

5.2.1 Stress Experimental Test

The stress test evaluated the performance of the SUBSCRIBER FB in receiving and processing messages under extreme workloads to determine its limits. The test was conducted by starting with ten messages and doubling this value until a limit of processed messages was reached, repeating each test 5 times to take the average. The achieved results are illustrated in Fig. 8-a).

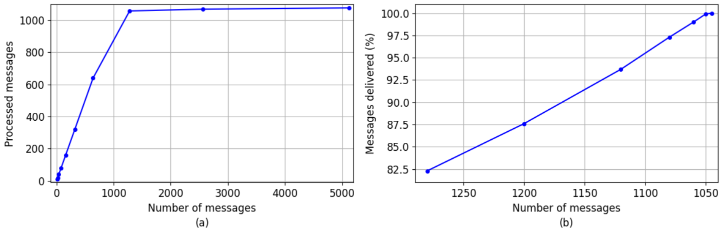

Fig. 8 Stress test results: a) Stress test curve, b) Loss mitigation test.

The limit of processed messages was around 1050, which required a loss mitigation test to find an optimum value at which all the messages sent could be delivered and processed by the FB. In this way, the test reduced the number of messages sent as the percentage of messages delivered increased, as shown in Fig. 8-b). The achieved results show that to guarantee 100% of messages delivered, up to 1045 messages can be sent simultaneously. Therefore, as each SUBSCRIBER FB handles one message topic, the system can support up to 1045 connected conveyors, which is a very high number for this modular conveyor system.

5.2.2 Performance Experimental Test

The performance test aimed to measure the system's processing time for the varying message quantities, ranging from 1 to the limit of 1045. Across multiple repetitions, the results, illustrated in Fig. 9-a), show that as the batch of messages increases, there is a correspondent rise in the average latency time to process each message, indicating that larger batches introduce complexity that impacts the system's ability to process messages.

However, the overall processing time for the given amount of messages remains satisfactory, underscoring the system's reliability even when subjected to different message loads. Moreover, the use of WiFi in communication may introduce reliability challenges like interference or bandwidth limitations, affecting the performance. Additionally, the latencies observed could be due to DINASORE's Python implementation, which may introduce overhead.

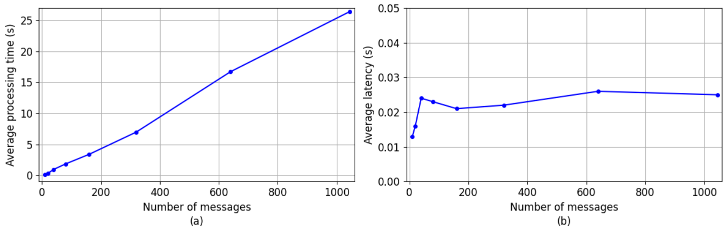

Fig. 9 Performance test results: a) Performance test, b) Latency test.

6 Conclusion

This paper details the development of IEC-61499 Python-based FBs within the DINASORE framework for a modular conveyor transfer system. The approach enables the on-the-fly reconfiguration in response to condition changes while ensuring scalability. The developed control system introduces a reconfiguration FB to dynamically identify and adapt to changes, modifying the system's behavior in real-time. However, the learning curve associated with DINASORE was notice, with challenges in mastering its configuration and integration processes. Despite these difficulties, the implemented FB-based control system demonstrated significant adaptability, seamlessly adjusting to changes in the conveyor configuration and exhibiting robust scalability. Notably, its processing efficiency allowed the system to accommodate approximately 1045 conveyor modules without losing information.

When compared with the MAS approach (see the work described in [21]), both technologies exhibit good reconfigurability, scalability, and adaptability, making them suitable for dynamic applications. However, MAS provides more robust and stable communication patterns than the ones used by FBs, taking advantage of the established FIPA protocols. Also, MAS offers high flexibility and adaptability due to its distributed and autonomous nature, but currently, the implementation of FB-based control systems in industrial environments is more well-adopted. Future work is devoted to expand the current linear conveyor case study to more complex flow topologies, including integrating the approach with systems like AGVs and robotic arms. Additionally, it is crucial to improve communication capabilities, particularly addressing issues related to WiFi reliability and latency.

Acknowledgments

This work has been supported by national funds through FCT/MCTES (PIDDAC): CeDRI, UIDB/05757/2020 (DOI: 10.54499/UIDB/05757/2020) and UIDP/05757/2020 (DOI: 10.54499/UIDP/05757/2020); and SusTEC, LA/P/0007/2020 (DOI: 10.54499/LA/P/0007/2020). The author Gustavo Funchal thanks the FCT Portugal for the PhD Grant 2022.13712.BD.

References

1. G. Lyu and R. W. Brennan, "Towards IEC 61499 Based Distributed Intelligent Automation: Design and Computing Perspectives," in *Proc. IEEE 17th Int. Conf. Ind. Informatics (INDIN)*, vol. 1, pp. 160–163, IEEE, 2019.
2. H. Boyes, B. Hallaq, J. Cunningham, and T. Watson, "The industrial internet of things (IIoT): An analysis framework," *Comput. Ind.*, vol. 101, pp. 1–12, 2018.
3. E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, "Industrial Internet of Things: Challenges, Opportunities, and Directions," *IEEE Trans. Ind. Informatics*, vol. 14, no. 11, pp. 4724–4734, 2018.
4. D. Oliveira, J. Pinheiro, L. Neto, V. H. Pinto, and G. Gonçalves, "A Plug-and-Play Solution for Smart Transducers in Industrial Applications Based on IEEE 1451 and IEC 61499 Standards," *Sensors*, vol. 22, no. 19, p. 7694, 2022.
5. P. Leitão, A. W. Colombo, and S. Karnouskos, "Industrial automation based on cyber-physical systems technologies: Prototype implementations and challenges," *Comput. Ind.*, vol. 81, pp. 11–25, 2016.
6. J. P. F. de Moura Pinheiro, "Internet of things software modules marketplace," 2020.
7. R. Ramanathan, "The IEC 61131-3 programming languages features for industrial control systems," in *Proc. World Automation Congr. (WAC)*, IEEE, 2014.
8. N. Higgins, V. Vyatkin, N.-K. C. Nair, and K. Schwarz, "Distributed Power System Automation With IEC 61850, IEC 61499, and Intelligent Control," *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.*, vol. 41, no. 1, pp. 81–92, 2011.
9. V. Vyatkin, "The IEC 61499 standard and its semantics," *IEEE Ind. Electron. Mag.*, vol. 3, no. 4, pp. 40–48, 2009.
10. E. Pereira, J. Reis, and G. Gonçalves, "DINASORE: A Dynamic Intelligent Reconfiguration Tool for Cyber-Physical Production Systems," in *Proc. Eclipse Conf. Secur. Artif. Intell. Model. Next Gener. Internet Things (Eclipse SAM IoT)*, 2020.
11. L. I. Pinto, A. B. Leal, and R. S. U. Rosso, "Safe dynamic reconfiguration through supervisory control in IEC 61499 compliant systems," in *Proc. IEEE 15th Int. Conf. Ind. Informatics (INDIN)*, IEEE, 2017.
12. T. Strasser, M. Rooker, G. Ebenhofer, and A. Zoitl, "Standardized Dynamic Reconfiguration of Control Applications in Industrial Systems," *Int. J. Appl. Ind. Eng.*, vol. 2, no. 1, pp. 57–73, 2014.
13. M. Shakil and A. Zoitl, "OPC UA based IEC 61499 Device Configuration Interface," in *Proc. IEEE Conf. Ind. Cyberphysical Syst. (ICPS)*, IEEE, 2020.
14. L. Prenzel and S. Steinhorst, "Automated Dependency Resolution for Dynamic Reconfiguration of IEC 61499," in *Proc. 26th IEEE Int. Conf. Emerg. Technol. Factory Automat. (ETFA)*, IEEE, 2021.
15. L. Prenzel, S. Hofmann, and S. Steinhorst, "Real-time Dynamic Reconfiguration for IEC 61499," in *Proc. IEEE 5th Int. Conf. Ind. Cyber-Physical Syst. (ICPS)*, IEEE, 2022.
16. H. A. S. Leitao, R. S. U. Rosso, A. B. Leal, and A. Zoitl, "Fault Handling in Discrete Event Systems Applied to IEC 61499," in *Proc. 25th IEEE Int. Conf. Emerg. Technol. Factory Automat. (ETFA)*, IEEE, 2020.
17. P. Kajola, J. O. Blech, U. Dwi Atmojo, and V. Vyatkin, "Dynamic Adapter Connections for IEC 61499," in *Proc. 22nd IEEE Int. Conf. Ind. Technol. (ICIT)*, IEEE, 2021.
18. S. Guellouz, A. Benzina, M. Khalgui, G. Frey, Z. Li, and V. Vyatkin, "Designing Efficient Reconfigurable Control Systems Using IEC61499 and Symbolic Model Checking," *IEEE Trans. Autom. Sci. Eng.*, vol. 16, no. 3, pp. 1110–1124, 2019.
19. J. Barbosa, P. Leitão, and J. Teixeira, *Empowering a Cyber-Physical System for a Modular Conveyor System with Self-organization*, pp. 157–170. Springer Int. Publ., 2018.
20. G. Lyu and R. W. Brennan, "Towards iec 61499-based distributed intelligent automation: A literature review," *IEEE Transactions on Industrial Informatics*, vol. 17, pp. 2295–2306, Apr. 2021.
21. P. Leitão, J. Barbosa, G. Funchal, and V. Melo, "Self-organized cyber-physical conveyor system using multi-agent systems," *Int. J. Artif. Intell.*, pp. 171–185, 2020.