Hierarchization and Integration of IEC 61131-3 and IEC 61499 for Enhanced Reusability

M. Melik Merkumians, P. Gsellmann, and G. Schitter
Automation and Control Institute
TU Wien
Vienna, Austria
melik-merkumians@acin.tuwien.ac.at

***Abstract*—Today's markets are short-lived and volatile. The producing industry needs to be able to react fast and cost-effective to product and process changes. Following the trend of Industry 4.0, current automation systems are designed as Cyber-Physical Production System (CPPS), which main properties are the distribution of capabilities and strict modularization to enhance reuse of off-the-shelf automation components. The currently dominant Programmable Logic Controller programming standard IEC 61131 was conceived with mainly centralistic systems in mind, which limits its use for nowadays CPPS scenarios. Focusing on the software aspect, the contribution of this work is an enhanced hierarchic development approach for automation systems, which aims to decouple all hardware from the control flow logic of an automation system for combined IEC 61131 and IEC 61499 systems. Therefore, a migration path for IEC 61131 Function Blocks into the development model of the event-driven IEC 61499 is presented, and issues are discussed. Both approaches pay special attention to the reusability of software.**

***Index Terms*—CPPS, PLC, Modularization, IEC 61131, IEC 61499**

I. Introduction

Plant engineering has become increasingly complex in recent years. The reasons for this are the increasingly demanding production processes that combine a variety of different technologies, but also the changed market conditions, with significantly higher diversity of variants, and shorter product- and process life cycles [1]. Industry and academia focused on modularization concepts to cope with these new demands. Following the trend of Industry 4.0, current automation systems are designed as Cyber-Physical Production System (CPPS), which main properties are the distribution of capabilities and strict modularization to enhance reuse of off-the-shelf automation components. Both properties can only be achieved if hardware (HW) and software (SW) are designed with this in mind. The currently dominant Programmable Logic Controller (PLC) programming standard, the IEC 61131 – Programmable controllers [2], unfortunately did not consider such highly distributed systems as CPPS, when it was first developed, as centralistic systems were sufficient back then. This limits the use of the IEC 61131 system and programming models for distributed systems, which were tried to remedy with additional external models or additions to the standard [3], [4]. The main reason for this limitation is the cyclic processing model of IEC 61131 [5], [6]. The IEC 61131 uses a cyclic processing model, which guarantees a unique system status during program processing. This regular "freezing" of the real world facilitates programming and is easy to understand for the user. At the same time, however, it is an obstacle for the modularization of the SW and the distribution of the program over several PLCs, since a uniform system state cannot be guaranteed. Even if the same cycle time is chosen for all PLCs, the individual PLC-cycles do not necessarily have to run isochronously. The PLCs can therefore have different snapshots of the system to be controlled, which can lead to errors in the overall application [7].

The IEC 61499 was developed with distribution aspects of HW and SW in mind, which also forces the modularization of SW [8]. IEC 61499 uses the Function Block (FB) as a basic programming element, which in principle similar to the IEC 61131 FB, but replaces the data-driven activation of FBs with event control. This means that IEC 61499 FBs only execute if they have received an event.

Still, as the IEC 61131 is the dominant standard for PLC development, a migration path for integrating existing PLC programs into IEC 61499 – Function blocks [9] systems has to be provided [10].

Focusing on the SW aspect, the contribution of this work are an enhanced hierarchic development approach for automation systems, which aims to decouple all HW from the control flow logic of an automation system, and a migration path for IEC 61131-based systems to be integrated into IEC 61499 systems.

II. Hierarchic Development

Modularization and hierarchization allow automation components to be easily reused in various plant sections and applications. To be composable, each automation component must be self-sufficient and complete [11], [12]. It is important that the components do not pass or expect any requirements of a specific HW-implementation via their interfaces [12], [13]. Unfortunately, automation programs are often tightly coupled to the used HW, due to the fact how the HW is supposed to be controlled [12]. The resistor sorting system in Fig. 1 serves as an example. Resistors are filled into the vibrating conveyor as bulk good. The resistors are separated, the resistance value is measured, and based on the measurement result put in one of the two storage pallets. Following the state-of-the-art methods for hierarchizing automation systems [14]–[16], the system can be divided into three subcomponents: the measuring station,

Post-print version of the article: M. Melik Merkumians, P. Gsellmann, and G. Schitter, "Hierarchization and Integration of IEC 61131-3 and IEC 61499 for Enhanced Reusability," *2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2019. DOI: 10.1109/ETFA45728.2021.9613226

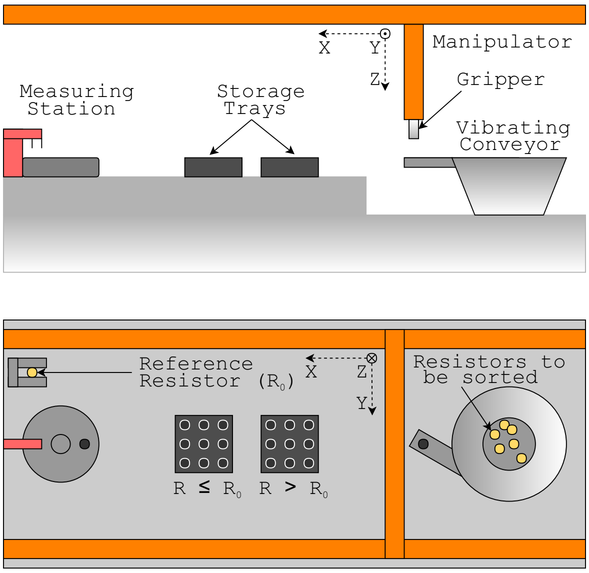

Figure 1: The considered system, which sorts resistors, provided as bulk material, according to its value in relation to a reference resistor. The resistors are separated and transported to the outlet of the vibrating conveyor, then moved to the measuring station, where it is compared to a reference resistor. Based on the comparison, the resistor is then sorted into one of the two storage trays.

the vibrating conveyor and separator, and the gantry robot with a vacuum gripper. The individual subcomponents can be further divided. For the portal robot-component this would be the three axes or axis drives for x, y and z, and the gripping tool. On the HW side, these three components are completely decoupled, as there are no dependencies between the individual components. However, inspecting the SW-components, it is evident that the components are coupled to each other, due to expectations on the representation of process parameters. The axis drive control of the individual axes of the portal robot serves as an example (see Fig. 2). The servomotor-axis drive expects the position representation in Cartesian coordinates (x, y or z for each individual axis) as input, visible as the output Position of the FB's on the right in Fig. 2, which seems to be the obvious choice, due to the gantry robot structure. This assumption is passed on to the next higher hierarchical level via the Controller_Interface interface, which expects a specific position value (Final _Position).

Thus, the next higher component is forced to implement this interface and is therefore strongly linked to subcomponents that expect coordinates as input. However, if the servomotor is replaced by, e.g., a pneumatic cylinder, the representation of position changed drastically, as a pneumatic cylinder is usually moved into its end positions, without intermediate positioning possible. This is also represented by its control FB in Fig. 3, where the movement is controlled by the Extend and Retract events to perform the respective movements. The proposed component-based design approach tackles this issue, by further decoupling the control logic from the HW implementation abstracting HW-dependent parameters into abstract symbols.

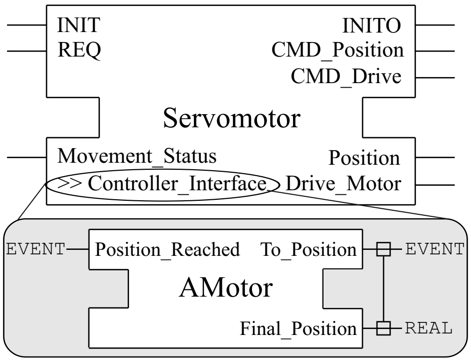

Figure 2: An IEC 61499 servomotor controller FB according to the state-of-the-art approach [14]. The interface consists of the hardware-dependent interface Movement_Status, Position, Drive_Motor, and all input and output events. The interface to the next hierarchy level is the Adapter Interface (AI) socket Controller_Interface, which is a logical interface consisting of the desired position Final_Position, and the events To_Position to trigger the movement and Position_Reached indicating that the desired position has been reached.

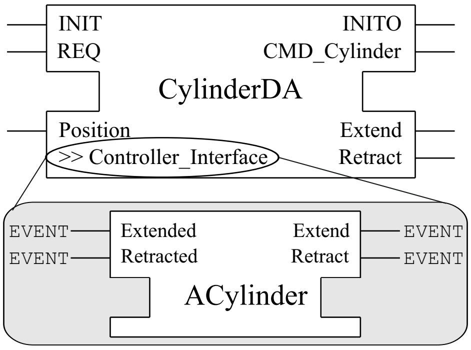

Figure 3: A 5/3-Directional Control Valve (DCV) controller FB for a double-acting cylinder in IEC 61499 according to the state-of-the-art approach [14]. The interface consists of the hardware-dependent interface Position, Extend, and Retract, and all input and output events. The interface to the hierarchy's lower level is the AI socket Controller_Interface, implementing an event driven interface to actuate a pneumatic cylinder via the Extend and Retract events, and the command completion events Extended and Retracted events.

These symbols only state the intent of a control-logic action,

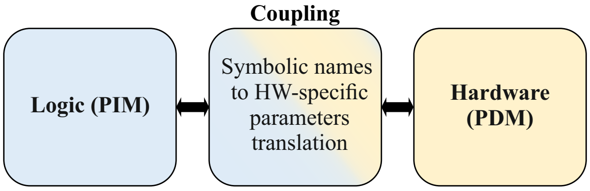

Figure 4: The proposed Model Driven Architecture (MDA)-like design approach separates the logical control application (Platform Independent Model (PIM)) from the hardware-specific layer (Platform Description Model (PDM)). The two layers are connected to an intermediate layer, responsible for translating the logical identifiers to the corresponding hardware-specific parameters.

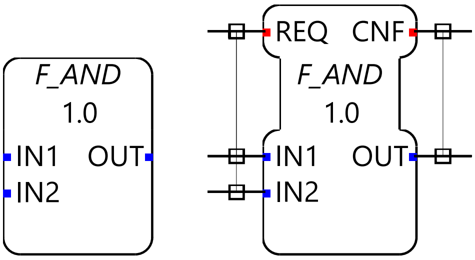

Figure 5: The Simple Function Block (SFB) is used in a dual way to represent IEC 61499 Simple Function Blocks (SFBs) and IEC 61131 FBs simultaneously. This reduces development effort and IEC 61131 FBs can easily be transferred to the IEC 61499 [24].

shielding the application from any HW-dependent values until the concrete HW implementation is chosen. These symbols get translated at the deployment stage of the automation project to the necessary hardware-specific representations, similar to the well known MDA development approach (see Fig. 4).

III. Migrating of IEC 61131 Applications to IEC 61499

As mentioned in Section I, a migration path from classic IEC 61131 based systems to IEC 61499 systems has to be provided, as it is not economical to completely revamp existing automation systems to be able to use new technologies and methods. Still, a method is needed to integrate those legacy systems into newer methodologies, such as the IEC 61499 development model, to reap the benefits of its distributed execution model, which usually builds the base of CPPS [17]–[19]. Several approaches to either combine IEC 61131 and IEC 61499 environments have been proposed [5], [20] or the automatic transformation of IEC 61131 applications to IEC 61499 models [10], [21], [22] to provide. Aside from the migration path aspect, certain classes of automation applications are easier implemented in either IEC 61131 or IEC 61499, so the ability to mix and match is highly desired to reduce the development effort [23].

The integration approach presented by Gsellmann et al. [24] not only combines IEC 61131 and IEC 61499 execution environments, but also development models. IEC 61131 `TASK`s are represented as specialized IEC 61499 `RESOURCE`s, which implement the cyclic execution model of the IEC 61499, running independent of other IEC 61499 `RESOURCE`s on the same `DEVICE`, as the IEC 61499 demands.

Interaction between the IEC 61131 and IEC 61499 parts is enabled via communication FBs, which respect the IEC 61131 cyclic execution model for the IEC 61131 parts and utilizing the event model for IEC 61499 `RESOURCE`s. A very interesting property of this approach is the dual use of SFBs as non-Object-Oriented (OO) IEC 61131 FBs (see Fig. 5), which enables an easy automatic transformation from legacy IEC 61131 FBs into IEC 61499 FBs, by simply adding an `REQ` input event and an `CNF` output event and `WITH`-ing all data in- and outputs with the respective event.

The OO extensions for IEC 61131, added in the 3rd edition of the standard, poses a challenge, as such FBs can implement `METHOD`s beside the regular code body of a regular IEC 61131 FB. Currently, IEC 61499 FBs can implement multiple `ALGORITHM`s, but these `ALGORITHM`s neither provide input nor output parameters in contrast to `METHOD`s. Therefore, such FBs cannot be translated to IEC 61499 FBs. Another current issue is the lack of IEC 61131 `FUNCTION`s in the model of the IEC 61499. Typical IEC 61499 environments provide a basic set of `FUNCTION`s, but do not provide custom `FUNCTION`s, as this is not covered by the IEC 61499. To fully integrate IEC 61131 into the models of IEC 61499 these issues have to be solved. The integration of `FUNCTION`s seems trivial at first glance, but there are several contexts where direct usage of `FUNCTION`s could lead to unintended side effects, such as the usage in Execution Control Chart (ECC) transition conditions and by that manipulating the encapsulating FBs state. This is even more a threat if IEC 61131 `METHOD`s would be directly transferred to the IEC 61499 development model. As a consequence, for a final harmonization these issues have to be solved.

IV. Conclusion and Further Work

This contribution presents the basis for an extended hierarchization approach which takes HW-specifics into account and further separating flow control from HW-control. In addition, it presents a migration path for legacy PLC system to the new CPPS design architecture by integration into IEC 61499 systems. Here, a first analysis has shown that non-OO IEC 61131 FBs can be migrated straightforward to IEC 61499, but newer OO-based ones cannot currently be easily represented by IEC 61499 models and language elements.

To solve these issues, the `FUNCTION` concept will be transferred to the IEC 61499 as a first-class member, and new usages explored and analyzed on their advantages and disadvantages. Also, the issues with the OO extensions and the possibility of `METHOD`s inside FBs and how to transfer them to SFBs needs further analysis. Here, a first idea would be to transfer them, similar to the main code body of IEC 61131 FBs, as an additional `ALGORITHM` and to represent the `METHOD`s interface as an additional set of data in- and outputs and the respective events. This first idea seems unsatisfying, as the `METHOD` could only be called from the outside, but not from the inside of an IEC 61499 FB.

References

[1] Y. Koren, *The Global Manufacturing Revolution: Product-Process-Business Integration and Reconfigurable Systems*, ser. Wiley Series in Systems Engineering and Management. Wiley, 2010.

[2] IEC TC65/WG6, *IEC 61131: Standard - Programmable controllers – Parts 1 to 8*. International Electrical Commission, 2003.

[3] A. Zoitl, T. Strasser, C. Sünder, and T. Baier, "Is IEC 61499 in Harmony with IEC 61131-3?," *IEEE Industrial Electronics Magazine*, vol. 3, no. 4, pp. 49–55, 2009.

[4] K. Thramboulidis, "Development of distributed industrial control applications: the CORFU framework," in *4th IEEE International Workshop on Factory Communication Systems*. Vasteras, Sweden: IEEE, 2002, pp. 39–46. [Online]. Available: http://ieeexplore.ieee.org/document/1159698/

[5] C. Sünder, A. Zoitl, J. H. Christensen, H. Steininger, and J. Rritsche, "Considering IEC 61131-3 and IEC 61499 in the context of component frameworks," in *2008 6th IEEE International Conference on Industrial Informatics*. IEEE, 2008.

[6] L. A. C. Salazar and O. A. R. Alvarado, "The future of industrial automation and IEC 61499 standard," in *2014 III International Congress of Engineering Mechatronics and Automation (CIIMA)*, 2014, pp. 1–5.

[7] R. Schoop and A. Strelzoft, "Asynchronous and synchronous approaches for programming distributed control systems based on standards," *Control Engineering Practice*, vol. 4, no. 6, pp. 855–861, 1996, publisher: Elsevier BV.

[8] I. Hegny, T. Strasser, M. Melik-Merkumians, M. Wenger, and A. Zoitl, "Towards an Increased Reusability of Distributed Control Applications Modeled in IEC 61499," in *Proceedings of 2012 IEEE 17th International Conference on Emerging Technologies and Factory Automation (ETFA 2012)*, 2012.

[9] IEC TC65/WG6, *IEC 61499: Function blocks for industrial-process measurement and control systems – Parts 1 to 4*. Geneva: International Electrotechnical Commission (IEC), 2005.

[10] C. Sunder, M. Wenger, C. Hanni, I. Gosetti, H. Steininger, and J. Fritsche, "Transformation of existing IEC 61131-3 automation projects into control logic according to IEC 61499," in *2008 IEEE International Conference on Emerging Technologies and Factory Automation*, 2008, pp. 369–376, iSSN: 1946-0759.

[11] R. Hametner, A. Zoitl, and M. Semo, "Automation component architecture for the efficient development of industrial automation systems," in *2010 IEEE International Conference on Automation Science and Engineering*, 2010, pp. 156–161.

[12] M. Melik-Merkumians, M. Wenger, R. Hametner, and A. Zoitl, "Increasing Portability and Reuseability of Distributed Control Programs by I/O Access Abstraction," in *Proceedings IEEE Emerging Technologies and Factory Automation (ETFA 2010)*, 2010.

[13] M. Wenger, M. Melik-Merkumians, I. Hegny, R. Hametner, and A. Zoitl, "Utilizing IEC 61499 in an MDA Control Application Development Approach," in *2011 IEEE Conference on Automation, Science and Engineering, August 24-27,2011, Trieste, Italy, Proceedings*, 2011, pp. 495–500.

[14] A. Zoitl and H. Prähofer, "Guidelines and Patterns for Building Hierarchical Automation Solutions in the IEC 61499 Modeling Language," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 4, pp. 2387–2396, 2013, publisher: Institute of Electrical and Electronics Engineers (IEEE).

[15] E. Faldella, A. Paoli, A. Tilli, M. Sartini, and D. Guidi, "Architectural design patterns for logic control of manufacturing systems: The generalized device," in *2009 XXII International Symposium on Information, Communication and Automation Technologies*. Sarajevo, Bosnia and Herzegovina: IEEE, Oct. 2009, pp. 1–7. [Online]. Available: http://ieeexplore.ieee.org/document/5348451/

[16] A. Tilli, A. Paoli, M. Sartini, C. Bonivento, and D. Guidi, "Hierarchical and cooperative approaches to logic control design in industrial automation," in *2009 IEEE Conference on Emerging Technologies & Factory Automation*. Palma de Mallorca, Spain: IEEE, Sep. 2009, pp. 1–8. [Online]. Available: http://ieeexplore.ieee.org/document/5347063/

[17] C. A. Garcia, E. X. Castellanos, C. Rosero, C. Sanchez, M. V. Garcia, F. Perez, and M. Marcos, "CPPS on low cost devices for batch process under IEC-61499 and ISA-88," in *2017 IEEE 15th International Conference on Industrial Informatics (INDIN)*. Emden: IEEE, Jul. 2017, pp. 855–860. [Online]. Available: http://ieeexplore.ieee.org/document/8104884/

[18] M. V. Garcia, E. Irisarri, F. Perez, M. Marcos, and E. Estevez, "From ISA 88/95 meta-models to an OPC UA-based development tool for CPPS under IEC 61499," in *2018 14th IEEE International Workshop on Factory Communication Systems (WFCS)*. Imperia: IEEE, Jun. 2018, pp. 1–9. [Online]. Available: https://ieeexplore.ieee.org/document/8402362/

[19] E. X. Castellanos, C. A. Garcia, C. Rosero, C. Sanchez, and M. V. Garcia, "Enabling an automation architecture of CPPs based on UML combined with IEC-61499," in *2017 17th International Conference on Control, Automation and Systems (ICCAS)*. Jeju: IEEE, Oct. 2017, pp. 471–476. [Online]. Available: http://ieeexplore.ieee.org/document/8204485/

[20] S. Campanelli, P. Foglia, and C. A. Prete, "Integration of existing IEC 61131-3 systems in an IEC 61499 distributed solution," in *Proceedings of 2012 IEEE 17th International Conference on Emerging Technologies & Factory Automation (ETFA 2012)*. IEEE, 2012.

[21] M. Wenger, A. Zoitl, C. Sünder, and H. Steininger, "Semantic correct transformation of IEC 61131-3 models into the IEC 61499 standard," in *2009 IEEE Conference on Emerging Technologies Factory Automation*, 2009, pp. 1–7, iSSN: 1946-0759.

[22] W. Dai and V. Vyatkin, "Redesign distributed IEC 61131-3 PLC system in IEC 61499 function blocks," in *2010 IEEE 15th Conference on Emerging Technologies Factory Automation (ETFA 2010)*, 2010, pp. 1–8, iSSN: 1946-0740.

[23] P. Gsellmann, M. Melik-Merkumians, and G. Schitter, "Comparison of Code Measures of IEC 61131-3 and 61499 standards for Typical Automation Applications," in *Proceedings of the 2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation*, 2018, pp. 1047–1050. [Online]. Available: https://publik.tuwien.ac.at/files/publik_271875.pdf

[24] P. Gsellmann, M. Melik-Merkumians, A. Zoitl, and G. Schitter, "A Novel Approach for Integrating IEC 61131-3 Engineering and Execution into IEC 61499," *IEEE Transactions on Industrial Informatics*, pp. 1–1, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9238463/

Post-print version of the article: M. Melik Merkumians, P. Gsellmann, and G. Schitter, "Hierarchization and Integration of IEC 61131-3 and IEC 61499 for Enhanced Reusability," *2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2019. DOI: 10.1109/ETFA45728.2021.9613226