Design Approaches for IEC 61499 Control Applications

Sebastian Preuße, Christian Gerber and Hans-Michael Hanisch

University of Halle-Wittenberg
Institute of Computer Science, Chair of Automation Technology
Kurt-Mothes-Straße 1, 06120 Halle (Saale), Germany
{sebastian.preusse, christian.gerber, hans-michael.hanisch}@informatik.uni-halle.de

Abstract: Current challenges require automation industry to be more flexible and tailored to a wide range of production scenarios. The trend shows up that the quantity of products is alternating, while the amount of possible product configurations increases. For this, manufacturing systems as well as control software applications have to be reconfigurable and reusable to meet these demands. The international standard IEC 61499 offers a lot of advantages to face these challenges. This contribution therefore proposes different design approaches for IEC 61499 control applications that are tailored to the demands of reconfiguration and reusability.

1 Introduction

The international standard IEC 61499 [IEC05] provides advanced possibilities and ways to design control applications. Crucial features are, amongst others, real object-orientation, event-driven execution behavior and distributed controller design. Furthermore, it claims vendor-independency, so that once designed control applications can be executed on any engineering environment.
Current demands, as for example system's reconfiguration and reusability, can only partially be fulfilled by the application of well-established methods, which are used for control software development following the international standard IEC 61131 [IEC03]. Usually, the control code consists of one main program that calls function blocks or functions and is processed cyclically or time-triggered. This design approach requests a complete re-engineering of the control program, when rebuilding or migrating the attached plant. However, plant operators demand a high grade of flexibility in combination with minimal downtimes. The IEC 61499 provides promising advances for tackling these problems. Anyway, application engineers have to be aware of the new functionality and have to reconsider the way of control software design.
This contribution therefore proposes IEC 61499 controller design approaches that have been developed during research work of the authors' workgroup. A short introduction on IEC 61499 is given in [PMG$^+$10]. The examples for the design approaches, described in Section 2, are taken from a manufacturing prototype, namely the EnAS testbed [PGH10]. Section 3 finally draws a conclusion.

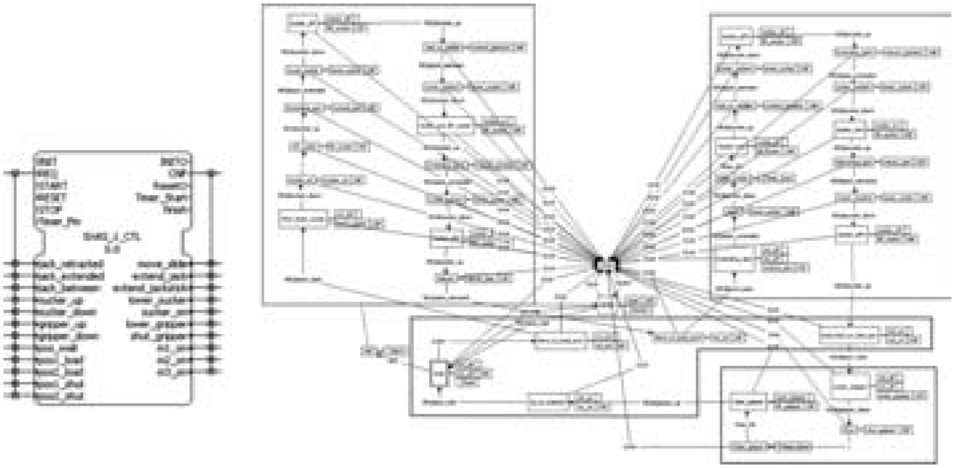

Figure 1: Central Controller of the testbed

2 Controller Design Approaches

This section introduces controller design approaches that have been developed during research work of the authors' workgroup. The order is chronological, since the approaches have been advanced due to solving new problems that appeared, when implementing control software for different testbeds. Therefore, the approaches are experimentally approved to be applicable for manufacturing systems and are suitable for large scale plants as well.

2.1 Central-Controller Approach

The Central-Controller approach is quite common since the controller functions are implemented within one Function Block (FB), what makes it possible to quickly comprehend the sequence of the manufacturing steps. As the authors' testbeds have first been controlled by IEC 61131 conform controllers that usually consist of one main program and several function calls, the idea seems obvious to just map the functionality of the classic controllers to the IEC 61499 conform ones.
The FB presented in the left of Figure 1 is the central controller of the left plant part of the EnAS testbed. It controls all 3 conveyors and all processing stations. Therefore, the ECC shown in the right of Figure 1 gets huge and hard to maintain. Even though this FB is portable between several engineering environments conform to IEC 61499, it will not be reusable if the production scenario changes, and it will completely have to be re-engineered.
Regarding the ECC, one can see four different controller tasks, which are marked in Figure 1. The upper right ECC part controls the opening and unloading of a tin, whereas the upper left part controls the loading and closing of a tin. The down center part controls the

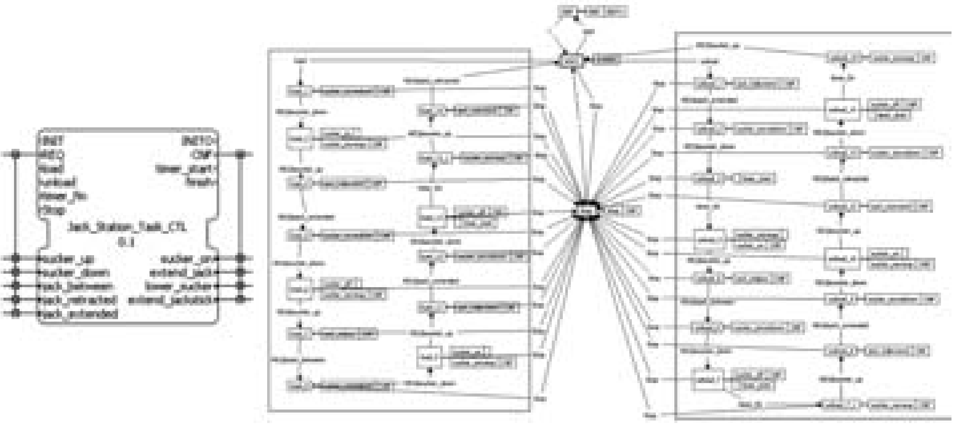

Figure 2: Task-Controller of the Jack Station

conveyors of the plant, and the down right part controls the movement and the closing of the gripper.
Using this information, several actions can be identified for every used mechanical component and distributed to separate FBs as described in [VHH06, HGVH07] for another plant. This leads to the Master-Task-Controller approach, presented in the following.

2.2 Master-Task-Controller Approach

The Task-Controller is the reusable part of the control application and can be stored in a library. It is once developed and verified for every mechanical component and controls the specific functions of it. If a component is removed from or is added to the plant, this Task-Controller FB will just have to be deleted from or inserted into the control application. Although the components are rearranged, the FB has only to be reconnected.
Figure 2 shows the FB and the ECC of the Task-Controller of the Jack Station. According to the ECC parts in Figure 1, the right rectangle borders all ECStates controlling the opening and unloading of a tin, while the left rectangle controls the loading and closing of a tin.
The control application is composed of all Task-Controllers, which handle the functionality. The program operation is coordinated by a Master-Controller. This FB triggers the execution of every task by events. As the Master-Controller is specific, it - but only it - will have to be re-engineered if the production scenario changes. But in contrast to the Central-Controller approach, the FBs controlling the elementary tasks are reusable. This makes the Master-Task-Controller approach more flexible and allows quick reconfiguration. Anyway, every new or changed FB has to be uploaded to the control device. Since not every runtime environment supports this dynamic reconfiguration as reported in [OWRSB05, VHH06], the application has possibly to be stopped and restarted again with the new configuration. The Parameterized Master-Task-Controller approach presents

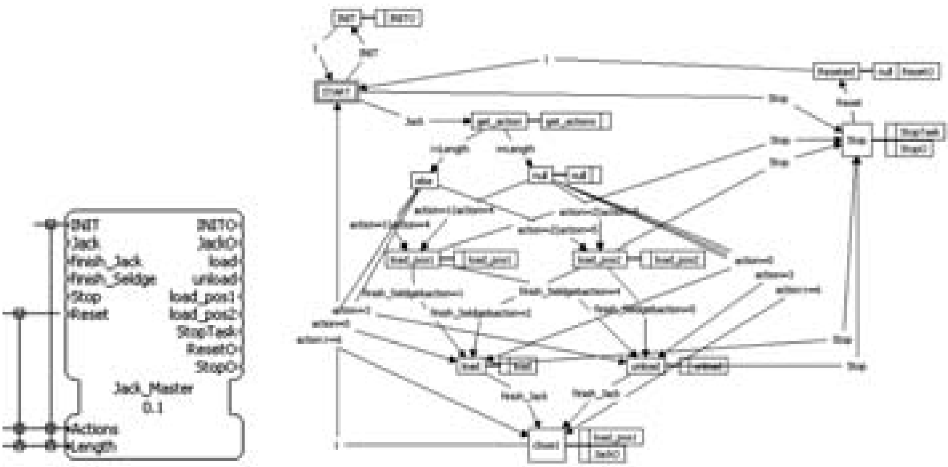

Figure 3: Parameterized Master-Controller of the Jack Station

a solution for this problem and is presented in the following.

2.3 Parametrized Master-Task-Controller Approach

The approach of the Parameterized Master-Task-Controller uses Master and Task-Controllers, too. The Master-Controller incorporates every possible production scenario and coordinates it with the other Master-Controllers of the plant. The control application can be reconfigured by implementing all possible production scenarios and switching between them through changed parameters.

Figure 3 shows the Parameterized Master-Task-Controller and its corresponding ECC of the Jack Station. Through the data input *Actions*, an array of sequentially performed actions can be parameterized. Whenever the FB receives the input event *Jack* and is in EC-State *START*, the next action is read out from the array, and the connected Task-Controllers are accordingly triggered. The number of actions is counted by an internal counter *i*. If this number equals to the value of the data input *Length*, the counter is reset, and the planned production scenario starts again from the beginning. As described in [GHH09], each production scenario is specified by an activity diagram of the Systems Modeling Language (SysML). The specific scenarios are transferred to the controller via the human machine interface, so that a high degree of flexibility is reached because the process can be adapted to new demands without having to restart or even to change the code of the control device.

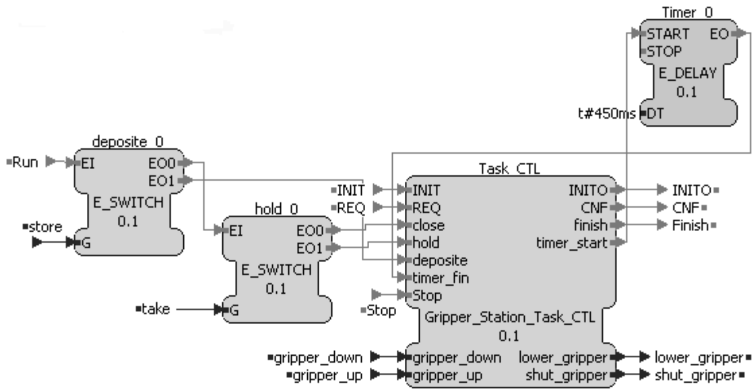

Figure 4: Function Block network to control the Gripper Station

2.4 Workpiece-Controller Approach

The complexity of the planned production scenario can increase because of several reasons. For example, pallets are added to or removed from the manufacturing system or there exist production alternatives, which influence each other for some reason. Consequently, the planning process will become the most time-consuming part of the reconfiguration.
The procedure can be simplified by developing the production scenario separately for every pallet. For this, each scenario is represented by one FB, namely the Workpiece-Controller, which controls the actions of the Task-Controllers. Each Workpiece-Controller allocates the Task-Controllers, which are necessary for the next production step, and releases them afterwards. If not all required Task-Controllers are available and a production alternative exists, this one will be chosen. Regarding the idea of prioritizing the ECTransition presented in [TD06], it is further possible to prioritize the production alternatives.
Figure 4 shows the FB Network, which controls the Gripper Station. It receives the *RUN* event, which is split and propagated according to the value of the event qualifiers *store* and *take* by the *E_Switch* FBs. Doing so, the actions *close, hold* and *deposite* are triggered. The event qualifier values are provided by the Workpiece-Controller FBs. For this, the production process is handled by the pallets that ”know” what actions have to be executed to process their workpieces.

3 Conclusion

Although the IEC 61499 standard supports issues as reconfiguration, reusability, and flexibility of controller design, it does not automatically guarantee that the designed controllers have these capabilities. An engineering methodology must be provided to guide the designer through the development process to really achieve that goal.
This contribution has shown some steps towards these goals that have been shown to be

useful. There are other approaches as well in this quickly emerging field of technology. What has not been mentioned in this short contribution is the fact that the described controller design methodologies are further enhanced by means of formal verification. This is a rather significant means to ensure correctness and reliability of controller design. This goes, however, far beyond the scope of this contribution.

Acknowledgment

This work was partially funded by the Federal Ministry of Economics and Technology (BMWi) under reference 16 IN 0651 and 01 MG 566 on account of a decision of the German Bundestag and by the German Research Foundation (DFG) under reference Ha 1886/16-1 and Ha 1886/16-2. The authors are responsible for the contents of this contribution.

References

[GHH09] C. Gerber, M. Hirsch, and H.-M. Hanisch. Automatisierung einer energieautarken Fertigungsanlage nach IEC 61499. *Automatisierungstechnische Praxis (atp)*, 51:44–52, 2009.

[HGVH07] M. Hirsch, C. Gerber, V. Vyatkin, and H.-M. Hanisch. Design and Implementation of Heterogeneous Distributed Controllers according to the IEC 61499 Standard - A Case Study. In *Proceedings of the Conference on Industrial Informatics (INDIN)*, pages 829–834, Vienna, Austria, July 2007. INDIN.

[IEC03] IEC 61131, Programmable Logic Controllers - Part 3: Programming languages, 2003.

[IEC05] IEC 61499, Function Blocks - Part 1: Architecture, 2005.

[OWRSB05] S. Olsen, J. Wang, A. Ramirez-Serrano, and R. W. Brennan. Contingencies-based reconfiguration of distributed factory automation. *Robotics and Computer-Integrated Manufacturing*, 21(2005):379–390, 2005.

[PGH10] S. Preuße, C. Gerber, and H.-M. Hanisch. Virtual Start-Up of Plants using Formal Methods. *Int. J. of Modelling, Identification and Control (IJMIC)*, 2010. Manuscript submitted for publication.

[PMG+10] S. Preuße, D. Missal, C. Gerber, M. Hirsch, and H.-M. Hanisch. On the Use of Model-Based IEC 61499 Controller Design. *Int. J. of Discrete Event Control Systems (IJDESC)*, 2010. Manuscript submitted for publication.

[TD06] K. Thramboulidis and G. Doukas. IEC61499 Execution Model Semantics. In *Proceedings of the Conference on Industrial Electronics, Technology & Automation (CISSE-IETA)*, 2006.

[VHH06] V. Vyatkin, M. Hirsch, and H.-M. Hanisch. Systematic Design and Implementation of Distributed Controllers in Industrial Automation. In H.-M. Hanisch V. Vyatkin, M. Hirsch, editor, *Proceedings of the Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 633–640. IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2006), September 2006.