Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 83 (2019) 292–297

www.elsevier.com/locate/procedia

11th CIRP Conference on Industrial Product-Service Systems

Model for Web-Application based Configuration of Modular Production Plants with automated PLC Line Control Code Generation

Markus Schäfer[a,]*, Patrick Moll[a], Lukas Brocke[a], Sven Coutandin[a], Jürgen Fleischer[a]

[a]*wbk Institute of Production Science, KIT Karlsruhe Institute of Technology, Kaiserstr. 12, 76131 Karlsruhe, Germany*

* Corresponding author. Tel.: +49-721-608-47357; fax: +49-721-608-45005. *E-mail address:* markus.schaefer@kit.edu

Abstract

The international competition leads manufacturers in high-wage countries to focus more on high-value products, which often come at the disadvantage of small batch sizes. To remain competitive, the plant engineering for should be time and cost effective. One approach to achieve this are modular production lines. In the presented contribution, a product orientated web- service for the configuration of a modular production plant investigated. The resulting model then is interpreted by a code generator to generate a PLC line control. The approach is validated with a plant of metal hybrid carbon fiber seat rests.

Peer-review under responsibility of the scientific committee of the 11th CIRP Conference on Industrial Product-Service Systems

Keywords: Modular production plant; web-based configuration service; control system code generator

1. Introduction

The increasing trend towards shorter product life cycles and an increasing number of individual customer requirements has led to smaller batches in manufacturing. In order to meet such volatile market requirements, production plants must offer transformability at factory level as well as fast configuration and reconfiguration at manufacturing system level [1, 2]. In the context of the "Industry 4.0" also the number of interacting machines and complexity in hard- and software are rising. Hence, effort for interface engineering and integration are increasing rapidly [3].

Modularization offers the advantages to cover these requirements of flexibility and fast integration as well as scalability and adaptability. This applies not only to manufacturing machines but also to the software control architecture.

However, currently used control systems are not well suited to be quickly reconfigured [4]. The change of a process or product in production usually requires a time consuming reprogramming of the PLC.

This approach presents an engineering method to (re-)configure a modular production line via a web-based application to provide a process-model for a PLC runtime code generator to shorten the engineering process in order to reduce costs.

The contribution is structured as follows: first, the considered use case is explained. Followed by presenting the general architecture of the modular plant and the configuration tool in order to under-stand the configuration procedure in chapter five. Finally, a short summary of the code generation is given before the conclusions of this contribution are discussed.

Especially in the field of carbon fiber reinforced plastics (CFRP), which have relatively small batch sizes, a fast and cost-effective reconfiguration of the system is necessary to be economical [5]. There-fore the use case of this contribution is the configuration and set up of a modular plant for the production of metal hybrid thermoplastic carbon fiber composite parts. The demonstrator part is a hybrid seat-backrest of which the process chain includes 15 steps of manufacturing shown in Fig.1. A detailed description of the process is presented in [6].

2212-8271 © 2019 The Authors. Published by Elsevier B.V.
Peer-review under responsibility of the scientific committee of the 11th CIRP Conference on Industrial Product-Service Systems.
10.1016/j.procir.2019.03.126

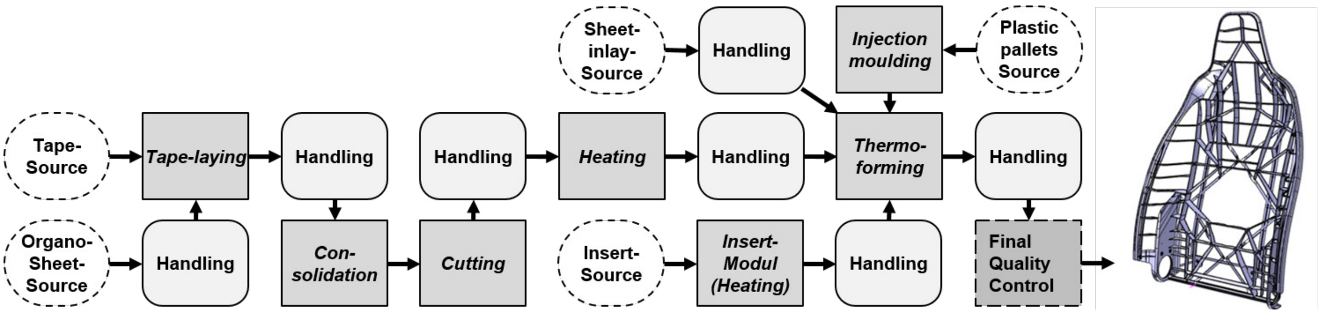

Fig. 1. The process for a hybrid seatrest made of CFRP and metal inlays.

2. Modular Plant Architecture

2.1. General Approach

To meet the stated requirements for the complex process chain of hybrid fiber reinforced parts, a distributed control system was installed in order to create a modular architecture and configuration method.

As the widely adopted IEC 61131 standard does not conceptually enable reconfigurability [4] the programming of the PLC is transformed into a user guided process. (See Fig. 2) This is done in form of a web-application, which provides more possibilities due to the use of high level programming language and interactive user interfaces. As Francalanza et al. have shown a guided engineering process is beneficial to support users [7]. Furthermore, predefined libraries and knowledge sources can be integrated to efficiently provide and process the needed information.

Subsequently, the configured plant model is exported into a single file model. This ‘production recipe’ is converted into the required runtime code components of the line control with the aid of a code generator. In order to realize this concept, a Service-oriented-Architecture (SoA) is developed to gain the aimed modularity.

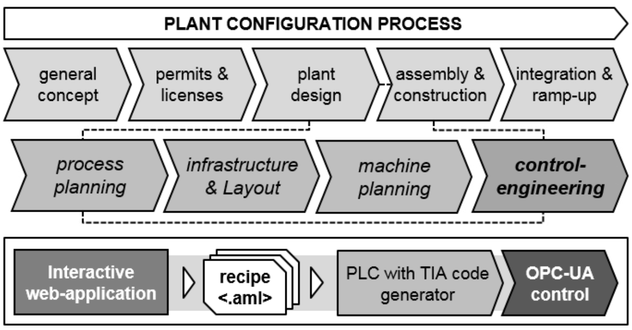

Fig. 2. Classification of the configuration workflow and new approach.

2.2. Service-oriented-Architecture (SoA)

To manage and fracture the complexity of the engineering workflow, the encapsulation of machine functions into services is beneficial [8]. This concept offers the advantage that the complexity for the plant operator is reduced as only the key parameters have to be known. Also the know-how of the module vendor is secured as the source code of the function remains hidden. Hence, the process chain itself is a sequenced compilation of the offered services being executed cyclic or event driven.

Creating such a SoA, the production modules offer their service capabilities to the line control via a defined OPC UA [9] interface. Using OPC UA not just decreases the effort for integration through the server-client principle, it also enables transferring the machine parameters to higher level analysis in the sense of Industry 4.0. The services are defined by the module vendor and can therefore vary in granularity as well as capability.

2.2.1. Service-Functions

A Service is a series of parameterized predefined functions based on the IEC61131 standard and executed by the module specific control. To call these services the needed parameters have to be provided with the respective call.

An important constraint for many plant operators is the use of existing machinery. The simple call of a predefined machine program stored locally on the module enables to integrate existing machines. In this case, the only transferred parameter is a string-type name of the program to be executed.

The third option for an offered service is the autonomous generation of the needed machine program. Hereby a file is transferred to the module and a preprocessing creates the local control program. By this way also more complex data can be submitted, e.g. the result file of a simulation.

2.2.2. Line-Control

For sequentially processing the respective services on the various production modules state manager systems are useful.[8] For this purpose, such a state model shown based on IEC 61512 [10] was implemented on each module and the control system.

Using this SoA, an engineering method to configure a production line for the use case of CFRPs can be developed, following three-view-concept [11] of products, processes and resources. To implement this workflow, a web application and its structure are presented.

3. Web Application / Toolbox

The control programming process in classic PLC configuration is often cluttered and time consuming due to the adaptation of interfaces and variables. The recent advancements in web interface design, on the other hand, allow very variable and intuitive workflows. To use these advantages, a configuration method has been developed and implemented in a web-application named “modular toolbox” (Fig. 3).

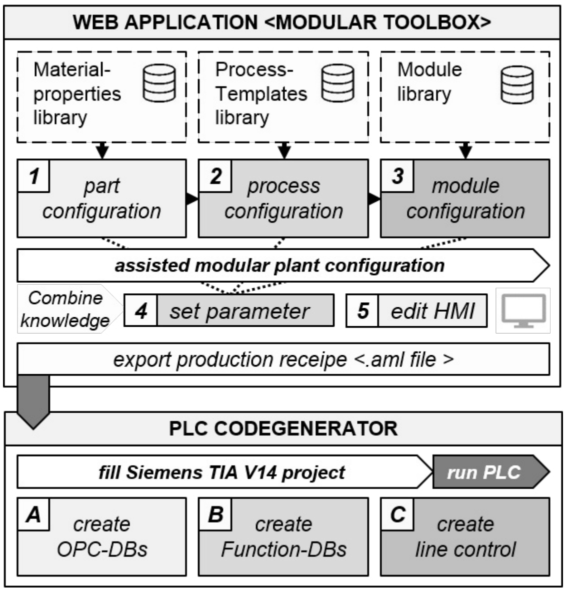

Fig. 3. Structure of web application and toolbox.

The method is clustered into five steps: part, process and module configuration, parameter setting and the setup of the human machine interface (HMI).

The result is a plant model which contains all necessary data for the line control to run the plant later on. For this, the XML based AutomationML (AML) format [12] has been used, which allows interoperability storing and exchanging engineering data in plant planning as it fulfills the requirements of CAEX [13] standard. The web application has been designed to create a tree based topology suited for AML. Schleipen et al. [11] have shown that the object orientated approach of AML is suitable to mod-el complex engineering data such as plants. The engineering procedure is supported by the use of several databases, which are briefly introduced here.

3.1.1. Material Library

The material library contains a variant of datasets of commercially available semi-finished products and materials. Later on the process parameters e.g. the melting temperature of a thermoplastic are linked to the module’s attributes according to these properties.

3.1.2. Hybrid-Lightweight Process library

The process library provides templates of commonly known process chains to provide an easy start for the user. For this project variants of thermoforming [5] have been implemented.

Every process is modeled using an inverse tree structure. Each node represents a process step containing information about possible preceding process steps (child nodes), as well as the succeeding process steps (parent node). This structure uniquely identifies a specific process chain. All nodes are stored as tuples in a relational database table with a unique identifier added to every node.

Additionally, to provide a grid layout in the process configuration step, horizontal and vertical coordinates are associated with each process step.

3.2. Module-Profile Library

The importance of standardized interfaces for multi-vendor modular production plants as a key to Industry 4.0 is often stated [14]. Yet it is complicated to unify all interfaces as many machine manufacturers have their own requirements, which often prevent adaptation. Therefore, the concept of Module Type Packages [15], a semantic information model of each module, is used.

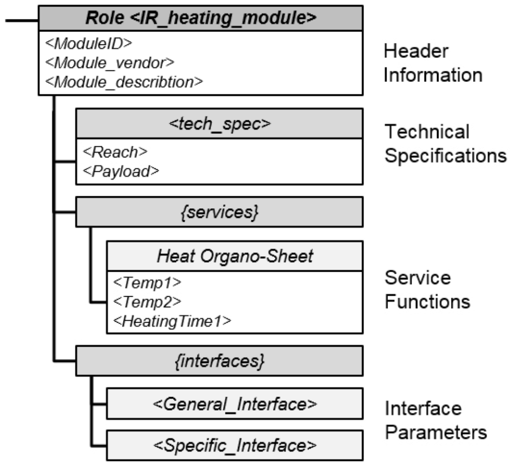

Fig. 4. Instance of a Module Profile.

This model of the machine interfaces in this approach is named Module-Profile. It is a class consisting of four major parts (Fig. 4): the header info to identify the module, metadata about technical specifications to enable comparability, the provided service functions for the SoA and the interface description itself.

All of the information contained in these module profiles and the other libraries are used during the configuration process.

4. Configuration Process

The configuration process follows the principle of Product-Process-Resources (PPR) described in [11]. The main purpose of the configuration is to link all information with their respective partners together.

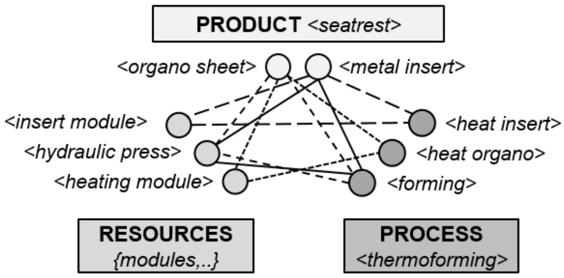

Fig. 5. The configuration links the attributes of product, process and the resources to each other.

4.1. Part Configuration

First the product “seat rest” is modeled by its consisting educts via drop down menus. For the given use case of thermoplastic hybrid CFRPs three main roles are given: organo sheet, tape laying structure and sheet metal. Furthermore the actual material data sheets either supplied by the vendor or in a materials database can be selected which results in adding the materials properties to the respective educt node. For the example of the organo sheet an actual semi-finished part with polyamide 6 (PA6) matrix is chosen. The value of “melting point temperature” is then set to 220°C. The result of this step is a collection of educt nodes with the corresponding material property attributes. Hybridization components like thread metal insert, sheet metal insert and more can be added.

4.2. Process Definition

Second, the process is modeled as a tree that can have multiple sources and one final resulting part. For this purpose, the chosen educts of the part configuration and the (end-) product were defined as start and end points with a single branch each.

When a template is selected in the “Module Toolbox”, the previously described tree structure is fetched from the database and converted into the grid structure using the horizontal and vertical coordinates. After that, each node is connected to its preceding and succeeding process steps. The user can now further individualize the process by adding, moving or deleting steps from the grid via drag and drop. As responsive design is used it can also be done on touch devices. If there was a part configuration beforehand, the start nodes already contain information.

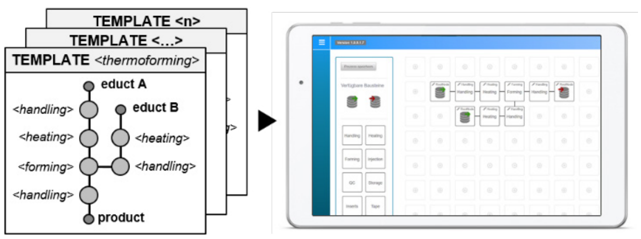

Fig. 6. The templates are loaded into an Inter-face where the process can be edited e.g. by drag & drop gestures.

A configured process is saved in the described inverse tree structure with a user-defined name is used to identify the process. This name is used by the next step, the machine selection, to load an existing process including all process steps and assign actual machines.

While the coordinates corresponding to the process definition grid aren’t used by the module selection, they are still saved in case the user wants to open a previously configured process again. The process can then be reconfigured on the grid.

Each process step node supplements a role class related to typical operations in CFRP manufacturing. According to every class there is a mapping to the class of machines offering the corresponding service. Figure 6 shows the example of the handling class offering a mapping to robots or a manual worker. Both can carry out the task of handling a part from one machine to another so the instance can now be created.

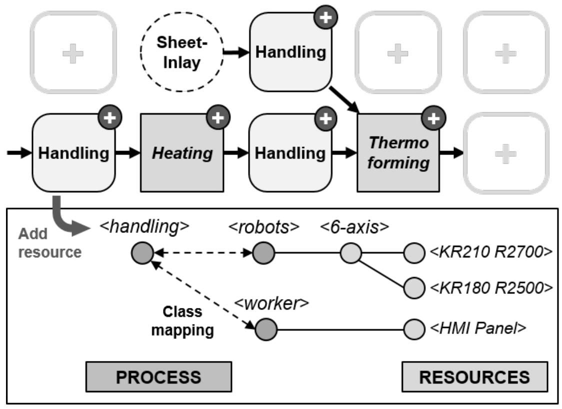

Fig. 7. Every process node gets assigned to a module instance

4.3. Module Configuration

The offered modules for the machine configuration can be many depending on the class of the process step. To limit the amount of possible selections additional methods are investigated and implemented.

4.3.1. Machine Selection Assistant

The selection can be supported in various ways. For the next paragraphs, the heating-module will be taken as a simple example. The task of this module is to heat up an organo-sheet blank to a specific temperature, hold this temperature for a defined time and then transfer it to the next step.

To select the best fitting module the user can fill in filter limits adjusted to the class and thus reduce the options. It is also possible to call up an assistance widget to be guided through the selection process. The widget shall be designed differently for each class to benefit the user guidance. For this contribution the rather simple example of heating up an organo sheet is presented.

One assistance feature is the estimation of relevant selection properties which are not directly apparent from the technical specifications. The necessary selection criteria are calculated using formulas and assumptions from literature. E.g. the time to heat up the organo sheet can be estimated using the power of the module, the thickness of the semi-finished product and the material properties. This allows the user to be offered important cornerstones for the decision making directly in the tool. However, it should be noted that these values are estimates for user support only. As there are many different and more advanced ways to design the support with data driven methods this will also be a part of future work.

4.4. Setting Service Parameters

By selecting the resource to carry out the process step, also the offered services are confined. A ser-vice itself consists of parameterized functions or a combination of further services. Each service is fully described by a set of parameters to be defined then.

For the example of the heating module, the parameters include the target temperature, the allowed tolerances and the various heating areas. Linking the properties of the semi-finished part to the service object, the parameters can be predefined by the toolbox. In the presented use case, the organo sheet to be heated up has a PA6 matrix. In the knowledge database, the materials are linked to the according temperatures, which then are set as service parameters (Fig.7). When the service is fully parameterized, the data object is created and stored to the database. All parameters are put into a separate database table including the attributes name, value, default value, description and unit.

These knowledge-based transformations can be spread to a wide field of applications. In the further investigations learning based models can be used to predefine parameters more precise.

To create the HMI visualization of the production line the roster of the first step is used and can be extended by the selection of several parameters to be periodically transferred to quality control analysis.

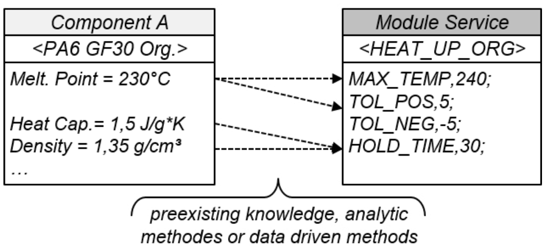

Fig. 9. Example of how the properties of the component e.g. a organo sheet are matched to the module parameters of a service e.g. heating up.

4.5. Export to AutomationML Recipe

When every service is parameterized, the acquired plant model is exported into an AutomationML file. Based on the SoA, a topology to resemble the gained configuration was developed.

4.5.1. Production recipe topology

The concept of AML files contains four sections: A library of predefined roles (Role Class Library) and a library of actual elements such as the modules and the services they offer (System Unit Class). Also a Library for AML-internal relation of the data to each other and linking of services is provided. To depict the process itself the Instance Hierarchy is used.

4.5.2. Instance Hierarchy

The production recipe conceptually follows the tree structure. The top node is the reference of the con-figured process and contains additional information e.g. date and batch size. Linked to this node is an instance for every production module used in the line. Linked to these are the configured services. Figure 8 shows a short section of the configured use case with connected services.

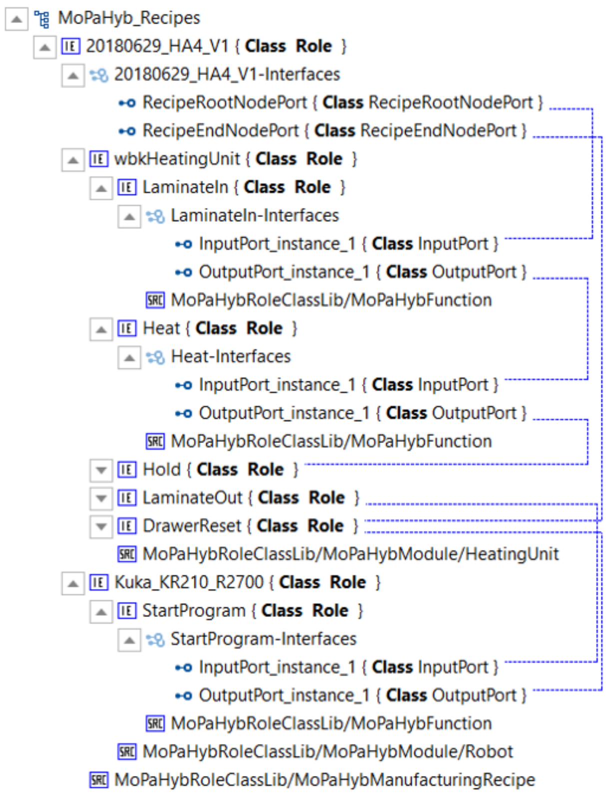

Fig. 8. Section of the Instance hierarchy of the modelled manufacturing process for the use case.

To preserve the integrity and order of the configured process, all function calls of the services are linked together using ports with unique IDs. While the recipe itself contains a start (root) and end port, input and output ports are added to each function found in the process. A link is unidirectional and connects an output port with an input port. Starting with the root port, a link is created to the input port of the starting services.

From there, the order de-fined in the process definition is used to create all links from functions output ports to the successors input port. The final functions output port is linked to the recipe end port. By this principle the whole process chain is described.

4.5.3. Export from SQL to AML

When the export of a process chain is started, an empty file in the Extensible Markup Language (XML) format will be created. Then, the module library and its roles will be reconstructed from the database into the system unit class.

The Instance Hierarchy has been partly created and configured by the steps made in the web application. Some information however hasn't been acquired in the configuration process and is therefore added in the export process. This mainly includes general, inconfigurable, module parameters, which are copied from the module library without modification. Modules and services in the Instance Hierarchy will also be assigned predefined roles from the Role Class Library which are required by AutomationML.

With the Instance Hierarchy being complete and valid, it will be added to the XML file as well. The file then is offered as a downloadable file so it can be transferred the code generator of the PLC.

5. PLC Code Generation

The central line control is based on the TIA Portal Version 14 from Siemens. Using the TIA ODK inter-face, a code generator can import and interpret the AML production recipe file. The code generator creates instances of generic data blocks (DB) of the control system and parameterizes them. For each module, Communication-DBs are generated for the basic structure of the OPC-UA connection containing the IP address and OPC-namespace. Together with the Function-DBs for calling the individual functions, a line control template is filled, which fulfills the sequential line control. This DB contains instructions for transitions (switching conditions), which vary depending on the state of the system.

Finally, a complete TIA project is defined within a short period of generation time, which provides a significant reduction of engineering effort and time. Thus, a production line can be set up faster and a reconfiguration can take place in the web-application instead of the PLC code project.

To validate the presented concept, a production plant for the presented use case with nine multi-vendor modules has been installed e.g. a hydraulic press, an injection unit and an industrial robot. The plant model has been configured in the web-application and successfully exported. Based on the resulting AML file the line control has been generated. A total of 46 seat rests have been produced.

6. Conclusions

In this contribution the authors have presented a method to configure a modular plant based on a web-application with user assistance tools. It is shown that the necessary information of product, process and resources can be modeled within the process and exported into an AutomationML file. Subsequently this file can be interpreted by a PLC code generator to create the runtime elements for an OPC UA based control system within minutes. By this principle the plant control system can be quickly configured and reconfigured inside the high level web-application. In future work more advanced user assistants will be investigated learning from quality data recorded during the process.

Acknowledgements

This paper is based on work within the MoPaHyb project (02P14A178) supported by the Federal Ministry of Education and Research (BMBF) and super-vised by the project management agency PTKA. The authors especially thank all MoPaHyb partners.

References

[1] European Commission (2004) Manufuture a Vision for 2020: Report of the high-level group, ISBN 92-894-8322-9, Brussels

[2] Koren Y, Heisel U, Jovane F et al. (1999) Reconfigurable Manufacturing Systems. CIRP Annals 48(2): 527–540

[3] Aicher T, Schutz D, Vogel-Heuser B (2014 - 2014) Consistent engineering information model for mechatronic components in production automation engineering. In: Proceedings of IECON 2014 - Annual Conference of the IEEE Industrial Electronics Society. IEEE, pp 2532–2537

[4] Strasser T, Zoitl A, Christensen JH et al. (2011) Design and Execution Issues in IEC 61499 Distributed Automation and Control Systems. IEEE Transactions on Systems, Man, and Cybernetics C. 41(1): 41–51.

[5] Fleischer J, Teti R, Lanza G et al. (2018) Composite materials parts manufacturing. CIRP Annals 67(2): 603–626

[6] Schäfer M, Moll P, Fleischer J (2018) Modular production plants for hybrid lightweight components. 22nd Dresden International Lightweight Engineering Symposium

[7] Francalanza E, Borg J, Constantinescu C (2017) A knowledge-based tool for designing cyber physical production systems. Computers in Industry 84: 39–58

[8] Bloch H, Hensel S, Hoernicke M et al. (2018) State-based control of process services within modular process plants. Procedia CIRP 72: 1088–1093

[9] International Electrotechnical Commission (2016) OPC Unified Architecture - Part 1: Overview and Concepts (62541-1)

[10] International Electrotechnical Commission (1997) Batch control - Part 1: Models and terminology (IEC/EN 61512-1)

[11] Schleipen M, Drath R Three-View-Concept for modeling process or manufacturing plants with AutomationML. In: IEEE (ed) Proceedings of IEEE Conference on Emerging Technologies & Factory Automation, pp 22–25

[12] Drath R, Lüder A, Peschke J et al. (2008) AutomationML - the glue for seamless Automation Engineering. In: 2008 IEEE International Conference on Emerging Technologies and Factory Automation, pp 616–623

[13] International Electrotechnical Commission (2016) Representation of process control engineering - Requests in P&I diagrams and data exchange between P&ID tools and PCE-CAE tools (62424:2016)

[14] Stephan Weyer, Mathias Schmitt, Moritz Ohmer et al. Towards Industry 4.0 - Standardization as the crucial challenge for highly modular, multi-vendor production systems

[15] Wassilew S, Urbas L, Ladiges J et al. (2016) Transformation of the NAMUR MTP to OPC UA to allow plug and produce for modular process automation. In: 2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA): September 6-9, 2016 Berlin, Germany. IEEE, Piscataway, NJ, pp 1–9