Abstract. In a high-mix low volume production environment, time to market is a key factor. However, one bottleneck lies in the often times manual parametrization of machines for new or modified designs. A truly flexible manufacturing environment therefore requires a continuous data flow from the design stages to the shop floor. This paper presents a concept for the automated parameterization of machines at a large automotive plant. Therefore, this paper initially discusses the results of a stakeholder analysis. The stakeholders comprise of different departments related to the product design and manufacturing processes. The requirements resulting from the interviews conducted with the stakeholders are grouped and ordered by priority. Secondly a general architecture for the control interface is presented. It includes multiple submodules, which model the continuous data flow between the departments and the production systems. The first main step of the data flow is the transformation of the information which is presented in various styles depending on the source departments to structured and standardized data. Thereafter machine parameters are generated automatically by a submodule using the structured input data and inference rules. Finally, the architecture supports the automatic transfer of the machine readable output data to the assembly line. To test the architecture a prototype comprising of more than 100 robots in a live production environment is implemented. It allows for a continuous data flow from the design and productions planning department to the robots. This enables a flexible process control, which up to today has supported the fast roll out of more than 100 new product variants. In contrast to the conventional manual setup of the machines for operation, the prototype was able to show that the monetary and time expenditure could be reduced by 95 percent.
Therefore, to enable a continuous data flow between all involved parties a common, machine readable information model is needed. However, there currently is no such standard.

2. Contribution
This paper presents a conceptual architecture for a continuous data flow between all stakeholders involved in the design and production of mechanical components. The model is based on a stakeholder analysis conducted at an automotive plant. Based on the stakeholder analysis an information model is derived that maps the information requirements of the manufacturing equipment onto the information sources in the upstream design processes. It considers three different priorities for the requirements which deal with 1) safety of the operator and equipment, 2) data processing and security and 3) ease of use. The model is then used for the design of a continuous data flow architecture that connects the involved departments. Based on the initial results with a prototype implementation, this architecture allows to reduce the cost and time needed for the addition of a new product variant by 95%.

3. State of the Art
Looking at the state of the art, there are different developments that deal with the networking of different subsystems across the different company levels. The piCASSO project [4], [5] deals with a cloud-based, modularized control architecture for dynamic robot programming at the control level. It provides an architecture that considers all machines as cyber physical devices which are addressed through standardized interfaces. This allows to interchange hardware with equal capabilities while providing a common interface for purely virtual control services which provide for the actual process control. This service based architecture leads to an increase in flexibility of the production system. While the piCASSO project focused on the benefits of a distributed control architecture it relied on standardized machine to machine communication frameworks. The most common framework is OPC-Unified Architecture which provides software interfaces for communication as well as a means for data modelling on a process and enterprise level [6]. The BaSys framework [7] builds upon OPC UA and provides a reference architecture for production systems, which allows a service-based networking of different technologies in the production environment. AutomationML is a highly standardized file format based on XML, which is intended to enable the standardized exchange of machine, production and product data in order to ensure the consistency of the information available [8]. Based on the ideas of these technologies the concept of the data-driven parameterization of production machines has been developed.

4. Concept
This section first presents the requirements engineering process. Thereupon the information model is derived and used to design a system architecture for the continuous data flow.

4.1. Requirements
To identify the requirements of the information model a stakeholder analysis with all parties involved in the design and production of the parts has been conducted. In particular the analysis includes internal production-related departments (PRD) and production employees (PD), as well as departments for standardization (DS), process optimization (PO), improvement management (IM) and information technology (IT). Furthermore, external service providers (ESP) are included in the basis for decision-making.

In an initial step, the involvement of the different parties and their influence on the process has been rated as shown in Figure 1. It shows that the design related departments and the production workers are the most important stakeholders, as they directly interact with the system. The production departments also have detailed information about their equipment and its data requirements. In addition, the production operators have an in-depth knowledge of the machines and their programming. DS engineers are trained to support projects of this kind and to provide standardized solutions or specifications for potential problems. Therefore, their influence on the project is very high and the interest is high. The department of process optimization has a high influence and interest, because important information
from already implemented improvements could be contributed and the implementation represents a considerable optimization in the production process. The IT department has a high influence because it ultimately provides the resources for implementation. The low influence of the IM results from the fact that this department only deals with already existing processes. On the other hand, the high level of interest is due to the fact that after the first prototype has been started, improvement measures can be implemented in other departments. External service providers would have little influence and little interest in the project at this point, since basic requirements must first be established in order to formulate concrete questions or demands on service providers. In the following, the requirements are analyzed and specified based on workshops and interviews conducted with the stakeholders, available documentation as well as information from existing and related systems.

Figure 1: Interest and Influence of the possible stakeholders

During the analysis, the unstructured information obtained from the expert interviews were classified, assigned to one of the subsystems (shown in Figure 2) and prioritized. There are requirements for the following subsystems:

- the department(s) (Input Requirements - INR),
- the interface (Interface Requirement - IR) and
- the production machine(s) (Output Requirement - OUTR).

Aside this allocation, the requirements are further divided into the following three priority classes:

- Priority 1: Requirements for operational and functional safety as well as core functionality,
- Priority 2: Requirements which are decisive for the functionality of the automated parametrization and as well as the verifiability of production quality and finally
- Priority 3: Other requirements that are not necessary for the core functionality but contribute to the ease of use.

Table 1 presents the resulting catalog of priority 1 requirements. An example for a priority 2 requirement is the provision of metadata and process results, which provides necessary information for the verifiability of production quality, but does not contribute to safety or core functionality. An example for priority 3 is the overall usability, which neither contributes to safety or the core functionality nor provides functionality for the automated parametrization or data flow and yet is of importance for the acceptance and deployment of the system.
Table 1: Priority 1 requirements as found in the stakeholder analysis

| ID   | Keyword                  | Explanation                                                                 |
|------|--------------------------|-----------------------------------------------------------------------------|
| INR01 | Input data format        | Minimum semistructured, database convertible, human readable and modifiable |
| INR02 | Standardization          | Individual attributes defined by a uniform nomenclature                     |
| INR03 | Data provision           | Provision of input data in a defined structure                              |
| IR01  | Interoperability         | Bidirectional communication via standardized communication protocol         |
| IR04  | Security                 | (traceable) data access management                                          |
| IR05  | Safety                   | Safety checks prior to execution of a process with new parameters           |
| IR20  | Standardization          | Uniform data types and structures, uniform naming schemas                   |
| IR21  | Process data format      | Minimum semistructured, machine-readable and human-readable                 |
| IR22  | Data provision           | (Automatic) provision of process data in a defined structure                |
| OUTR01| Integration              | Data processing and update of machine parameters between production cycles   |
| OUTR02| Data Consumption         | No computation within the machine program, Templated programs               |
| OUTR03| Safety                   | Update notification, enforce testing of new motion sequences                 |
| OUTR04| Communication            | Common protocol that is widely available on machines                        |

4.2. Architecture and data flow

Finally, based on the requirements gathered through the stakeholder analysis an architecture for the data flow is designed as presented in Figure 2. The individual sub-systems, departments (grey), interface (green) and production (orange) are shown. The input data is generated and stored within a system of the corresponding department. Examples for input data are CAD data (i.e. geometry or material information), machine data, tool data, and quality data or logistics information. Afterwards the input data is automatically transferred to the interface, where different services realize the data processing to generate machine-readable process data. The process data is then distributed to the respective production machines in a way that ensures data synchronization over the entire production line. The interface is divided into three sublayers: presentation, application and data layer. The presentation layer (blue)
provides the contents (such as current input and process data, archived data, change and error logs) of the service for different user groups.

The application layer (red) is the layer in which different algorithms for data processing and orchestration are implemented. In the data layer (yellow), the input and process data are stored. Depending on the data structure, persistent data storage in an SQL or NoSQL database is conceivable. Furthermore, a file-based storage for the process data is necessary. These must be machine-readable, at least semi-structured and stored in a defined file format (e.g. json, csv or xml) at a previously defined storage location. The data transfer to the production machines (orange) is carried out via a standardized communication protocol. The process data is then processed and persistently stored locally on the machine so that the machine program can access it. This means that in the event of a network failure, synchronized production data is stored on the machine. If the process data has an impact on the machine behaviour, a manual function check by a machine operator is necessary prior to clearance of the automatic program execution. If this check passes, series production is continued with updated process parameters otherwise the old parameters are kept in use until a new update is provided.

5. Conclusion and Outlook

The concept outlined in the paper provides standardized specifications to establish a continuous data flow between departments and shop floor via a central interface, thus enabling flexible production.

To test the architecture and the general data flow, a prototype has been implemented for an assembly line consisting of more than 100 6-axis articulated robots. The input data for more than 100 variants is automatically read from the construction and production planning departments and processed via a central interface. Robot program parameters are then automatically generated by inference rules and forwarded into program templates for the different machining processes. Initial results of the prototype implementation are very promising. As no external robot programmers were required to manually program all robots for all product variants, the costs and time taken for robot programming has been reduced by 95%. With the proposed automatic parameterization, the robots are ready for production in just 15 minutes. This includes the required manual safety check of the newly parametrized robot program. The previous process requires about eight hours of machine downtime. The more transparent change management further allows to identify error causes faster and has helped to reduce the number of rejected or faulty products.

In future work we like to expand the current prototype to a wider spectrum of machines (e.g. machine tools, measuring machines) and address limitations in the current implementation.

6. References

[1] Peschke and Eckardt. Flexible Produktion durch Digitalisierung. Carl Hanser Verlag; 2019.
[2] T. Strasser, A. Zoitl and G. Ebenhofer. 4diac – “Ein Open Source Framework für verteilte industrielle Automatisierungs- und Steuerungssysteme” in INFORMATIK 2010. Service Science – Neue Perspektiven für die Informatik. Band 1, K.-P. Fähnrich and B. Franczyk. Eds. Bonn: Gesellschaft für Informatik e.V.; 2010, pp. 435–440.
[3] V. M. Banholzer. Gestaltungsdiskurs Industrie 4.0: Akzeptanzaspekte, Frames. Institutionalisierung. Wiesbaden: Springer; 2018. p. 221–239.
[4] A. Vick, V. Vonásek, R. Pěnička and J. Krüger. Robot control as a service — Towards cloud-based motion planning and control for industrial robots. 2015 10th International Workshop on Robot Motion and Control (RoMoCo). Poznan; 2015, pp. 33-39, doi: 10.1109/RoMoCo.2015.7219710.
[5] A. Vick, C. Horn, M. Rudorfer and J. Krüger. Control of robots and machine tools with an extended factory cloud. 2015 IEEE World Conference on Factory Communication Systems (WFCS). Palma de Mallorca; 2015, pp. 1-4, doi: 10.1109/WFCS.2015.7160575.
[6] M., Schleipen, S.-S., Gilani, T., Bischoff, J. Pfrommer. OPC UA & Industrie 4.0 - Enabling Technology with High Diversity and Variability. Procedia CIRP, 2016;57:315–320.
[7] https://www.basys40.de/ (18.11.2020)
[8] AutomationML e.V. AutomationML and eCl@ss integration: Whitepaper. AutomationML e.V, 2015.