centurio.work - Modular Secure Manufacturing Orchestration

Florian Pauker, Juergen Mangler, Stefanie Rinderle-Ma, Christoph Pollak

Austrian Comptetence Center for Digital Production, Vienna, Austria,
{juergen.mangler, florian.pauker, christoph.pollak}@acdp.at

University of Vienna, Faculty of Computer Science,
Research Group Workflow Systems and Technology, Vienna, Austria,
stefanie.rinderle-ma@univie.ac.at

Abstract. Organisational and technical processes of manufacturing companies are becoming more and more complex while the pressure for flexibility increases. The biggest challenge here is still the integration of different systems in a coherent transparent way. In order to further vertical integration in the automation pyramid, this paper presents a solution based on use BPM technology, more specifically process modelling and execution based on BPMN to integrate shopfloor with top level planning. Our full-stack solution is currently in active development with our customers. One main advantage is that through coherent usage of BPMN processes from top to bottom, all monitored data during execution can always be stored in the right context, which saves a lot of effort compared to the state of the art in manufacturing systems.

1 Introduction

In today's global competition, manufacturing companies must adapt their technical processes and resources to constantly changing conditions. Additionally the demand to smaller lot size is increasing and so higher flexibility is needed. In this high mix low volume environment, the use of digital tools is becoming increasingly important. Data management in the factory relies on information flows that access different systems. The problem in many manufacturing companies is that most software tools are hard-coded stand-alone solutions with no or proprietary software interfaces. This poses a huge interoperability challenge. The lack of software interface compatibility makes data exchange between individual applications difficult and increases the effort of integrating new systems. Tailor-made software solutions are often used for coupling the individual software solutions. This kind of hard-coded integration can usually only be maintained, changed and expanded by software developers. This leads to high maintenance costs due to the complexity. What instead is required is an infrastructure that enables efficient adaptation of system orchestration.

Current initiatives like Industry 4.0 (I4.0), Industrial Internet of Things (IIoT) demand a fully connected environment consisting of so called Cyber-physical Systems (CPS). CPS are objects consisting of a physical part with an

F. Casati et al. (Eds.): Proceedings of the Dissertation Award and Demonstration, Industrial Track at BPM 2018, CEUR-WS.org, 2018.

virtual representation which can communicate with other CPS over a partially global network [4]. This technology approach leads to new infrastructure in companies. The well known automation pyramid [10], decomposes to loosely coupled services that interact with each other and the hardware and sensors on the field level (see Fig. 1).

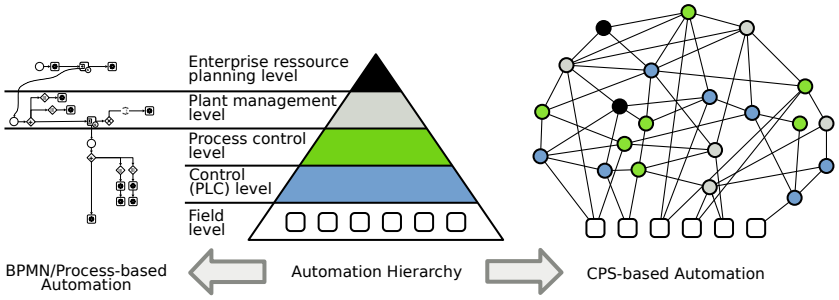


Fig. 1: Decomposition of the Automation Pyramid [10]

The contribution of this paper is the centurio.work approach for simplifying the integration of heterogeneous systems throughout the whole automation pyramid, by utilizing a true service-oriented oriented approach and advances like the BPMN [7] modelling language, and flexible execution [1,5]. As seen in Fig. 1

In Sect. 2 we start out by describing currently existing standards and requirements on the shopfloor (lower levels of the automation pyramid). In Sect. 3 we then describe the overall architecture of the system in relation to the shopfloor requirements. In Sect. 4 when then describe how the architecture is adapted by our customers.

2 Modelling of Shopfloor Processes

For orchestration of manufacturing different solutions are available. These solutions typically depend on the hierarchy level in the automation pyramid [10]. For orchestration of processes on Control level graphical languages are used as a basis. Graphchart [11] is one example for a graphical language for sequential supervisory control of systems. It is based on Sequential Function Charts (SFC), one of the programming languages for PLC described in IEC 61131-3 [2]. On higher organization levels other languages like BPMN or BPEL are used. For lower levels traditional programming is prevalent.

One requirement of Cyber-physical Systems (CPS) based automation is the need of orchestrating different services, which is implemented e.g. by the Manufacturing Service Bus (MSB) [6], an specialization of the Enterprise Service Bus. The main integration services of an ESB are: (i) transformation services that bridge differences in data formats and data models; (ii) a routing service

that handles messages and sends them to designated recipients; and (iii) an orchestration service which uses predefined process models and controls the flow of messages between service consumers and service providers. For different artefacts, such as orchestrations and integration services, different languages are used e.g BPEL and XSLT, which leads to rather unnecessary complexity. In other words: the MSB is struggling to gain traction with companies because it is a medley of technologies and components with high configuration and maintenance complexity. To the best of our knowledge only one implementation [9] exists. The motivation for the centurio.work framework is to bring down the complexity of introducing an MSB (or ESB) by configuring the bag of adapter and translation, and message exchange components through BPMN processes, thus using a single concept of configuring, using and maintaining infrastructure.

3 Modular Secure Manufacturing Orchestration

centurio.work was designed as an interoperability and integration framework for manufacturing companies. The motivation is bringing the flexibility of Business Process Management Systems from the management level to the shopfloor. Fig. 2 shows the architecture of the framework.

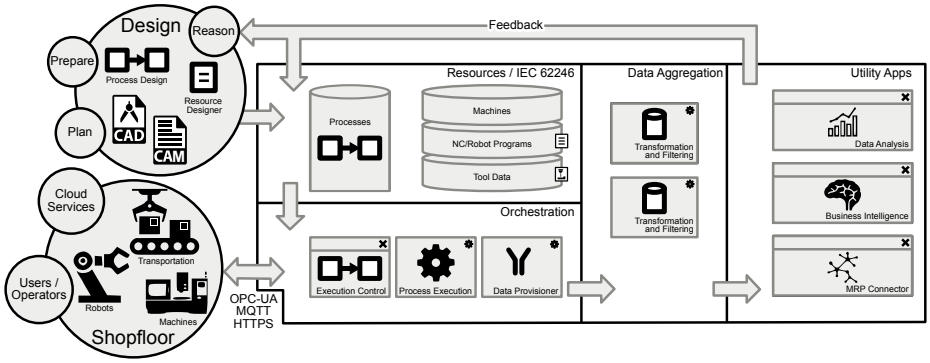


Fig. 2: centurio.work Architecture

The centurio.work framework was designed from the ground up with security in mind. It groups applications into four contexts that are strictly separated and are allowed to either push or read data. The arrows represent the data flow between these contexts. While every component inside a context is intended to be replaceable, the arrows represent strictly defined protocols, data formats and security mechanisms. Contexts which are not connected by arrows do not interact. Orchestration components are allowed to access Resources and can push information to Data Aggregation components. Data Aggregation components provide Data to Utility Apps but data change is not allowed.

The centurio.work framework stores static (design-time) information in the resource context. This information stems from various company systems and is fingerprinted upon insertion to be able to detect corruption and tampering.

For interacting with the shopfloor (run-time) centurio.work supports different interfaces. For orchestration of shop floor equipment OPC-UA is preferred solution. OPC-UA has strong modelling capabilities and is an agreed-upon standard for European companies. So the shopfloor equipment can offer their abilities as OPC-UA methods and static information as variables and properties. Therefore, we can utilize the client server paradigm and the built in security mechanisms of OPC-UA e.g. user authentication and certificates.

Additionally MQTT support is built in because of its light-weight publish/-subscribe mechanisms and the possibility to run on embedded systems. This is especially useful for high-performance data gathering.

The centurio.work Execution Engine itself is configured to push data for certain topics to Data Aggregation components. In other words: it communicates with Data Aggregation Apps through a secure publish/subscribe mechanism.

3.1 Resources

There exist some basic types of resources, which may be utilized by the Orchestration components. The Processes which provide a logic for coordinating the interaction between different machines. They define sequences of interactions with decisions and parallel execution. Processes typically refer to other information Resources, such as specific Machines, NC/Robot Programs, and Tool-Data. Process of course may also access this information a run-time, e.g. to decide which machine to utilize for a certain job.

The structure of the Resources container is based on the IEC 62264 [3]. The Resources context may accessed from the outside to add or modify processes and other static data.

3.2 Orchestration Components

This context is the sole context that is allowed to interact with the shopfloor. It does so by utilizing processes, which in turn contain all information how to interact with machines for the purpose of production and for the purpose of monitoring. Currently three types of apps can exist in this context: (i) An Execution Control App which allows to start and modify processes at runtime in the case of errors (ii), an Execution Engine which enacts the processes and (iii) a Data Provisioner which acts as an efficient gateway for machines that have the capability to push sensor data to subscribers.

The Execution Engine in turn is the sole source of data that is available to Data Aggregation. It pushes all aspects of interaction with machines to these Data Aggregation components, which may (or may not) filter, transform or otherwise process and store this data. Data may be duplicated, as each Data Aggregation App serves one specific purpose, and is to structure data for optimum performance and usability.

3.3 Data Aggregation Apps

Data Aggregation Apps, may (or may not) filter, transform or otherwise process and store data. Each app is intended to implement its storage and an interface how to access this storage. Data Aggregation Apps may: (i) log the execution (ii) log / monitor the execution for possible security violations (Secure Logging), and (iii) calculate Key Performance Indicators (like Overall Equipment Efficiency) of the equipment or the process. They may also (iv) aggregate data so that Utility Apps can efficiently perform their tasks. We use standards such as IEEE 1849-2016 XES [1] and tools such as InfluxDB to allow for optimized access by Utility Apps implementing dash-boarding and data- and process analytics.

3.4 Utility Apps

All other apps/functions are intended to run in this context. The basic functions of Utility Apps are to read data stored in Data Aggregation context. The Apps also can interact with each other and can additionally implement their own storage. A containerization salutation e.g. Docker is used to fulfil these requirements and to increase the usability for customers.

These Apps are also the basis for improvement recommendation, which in some cases (e.g. optimizations for the invocation of machines) automatically flow back into to the Resources context, or be passed on as documents to humans which may decide how the process structure or its parameters can be changed.

4 Example Scenarios From Our Customers

centurio.work was created as the result of feedback from our customers. In order to evaluate its applicability in an iterative manner, it is deployed in an increasing number of small and medium Austrian enterprises. Some use cases which have been developed and tested are described in the following.

4.1 Orchestration of a Manufacturing Process

As proposed in [8] for orchestration of a flexible manufacturing cell, on a hardware level, we need some atomic operations. Current restrictions for these are the real time capabilities of interfaces and network infrastructure. Some tasks have to be performed on the real time controller of the machines.

Typically we have different granularity on different levels in a manufacturing company (cmp. Fig. 3). This is represented in the automation pyramid as different levels which have different requirements on information, communication and data granularity. Approaches like the MSB use different modelling languages for the different levels. centurio.work uses BPMN for all levels, so we have a seamless modelling and process execution from business to shopfloor.

Fig. 3 shows a common process in a company with different information granularity in each level. On the top level (ERP) the process owner only sees a process consisting of three tasks, order, produce and deliver.

[1] `http://xes-standard.org/` [Last accessed: 2018-04-12]

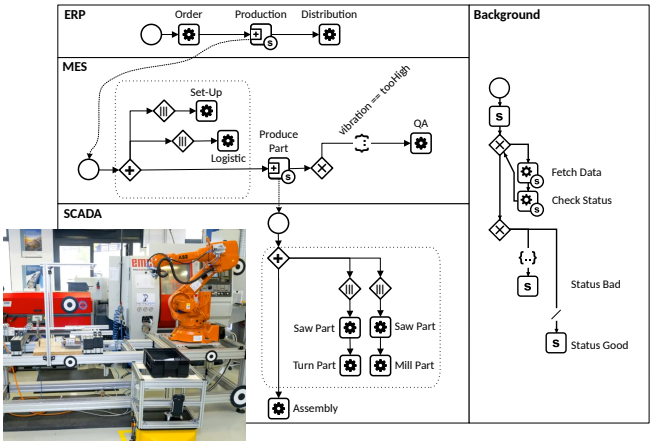


Fig. 3: Example Process with Hierarchy

The production planning needs more information of the Production process. Thus the sub process is more detailed with additional information on how to get the raw material and how to set up the machine. A worker on the shopfloor is involved in a subprocess with all tasks to produce individual parts. Additionally there are some background processes which for example gather sensor data from machines or calculate KPIs (cmp. Sect. 4.3).

4.2 Worker Machine Interaction

Currently on the shopfloor, human involvement is bound to hard-coded software on the machine. The big advantage of centurio.work is the independent coordination of human tasks with tasks from software systems or machines. A simplified process for setting up a machine (equip the machine with cutting tools) is displayed in Fig. 4. In the first task the process interacts using a UI with the worker. After finishing the task the next process step is executed. All information which are displayed on the UI are defined in the process task.

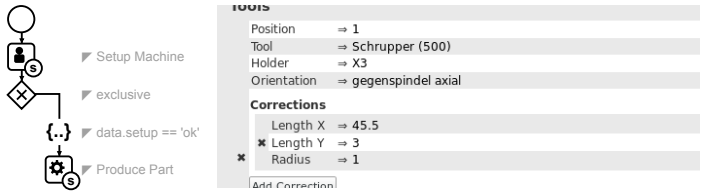
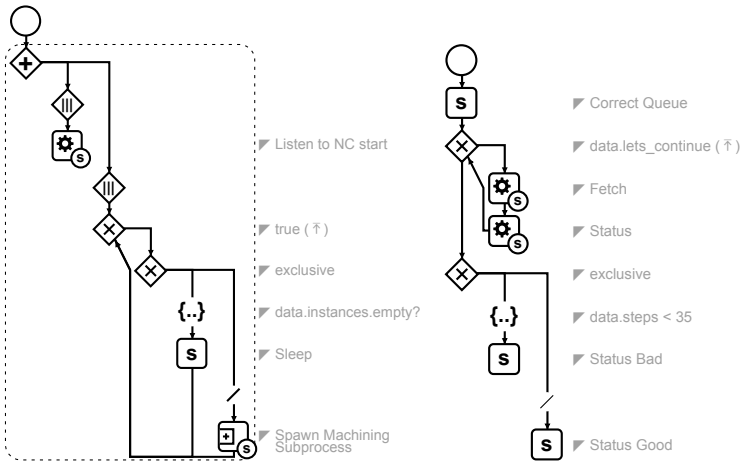


Fig. 4: Setup and Produce

4.3 Data Collection for Traceability

While traditionally data is collected individually for each machine, and is contextualized as part of a data analysis, in the centurio.work framework all data is collected from a single source, the Process Execution component. There is always an explicit connection between all monitored data artefacts.

This holds true even when only soft integration is required: i.e. when machines are not actively triggered, but just passively monitored without any changes to shopfloor at all. Fig. 5 shows a simplified version of the processes at our customers plant. One background process monitors the NC machines and if a NC Program is started it spawns a new instance of the machining process. The generated data is buffered in the Data Aggregation context.



(a) Spawn Machining Process (b) Machining Process

Fig. 5: Processes for data collection of a NC-machine

5 Conclusion and Outlook

On the shopfloor some of the challenges hindering tight IT integration are:

- Enterprise service bus (ESB) based systems provide the necessary ingredients, but require a lot of time and know-how to introduce, to configure and to maintain and are thus are typically employed only by big companies. Smaller companies typically have home-grown solutions that work.
- The shop floor is heterogeneous and machine manufacturers often still focus on building support for actively communicating with other machines into their proprietary software (resulting in a Mesh-Topology). Open, well documented interfaces for their own features are sadly still rare.

– Although OPC-UA gains traction, its focus is on standardizing the metamodel how to interact with machines. While taking away some communication issues, all the semantic issues remain.

centurio.work focuses on solving the above challenges by (1) providing a set of standardized components which can be deployed without any configuration. All configuration is handled through modelling BPMN processes for improved usability. (2) Experience values and improvements can easily flow back into the stored processes, instead of flowing back into configuration of individual components, or flowing back into static code. This again is a main usability improvement for our customers. (3) The bus topology with a centralized Orchestration Engine, allows for all the usual advantages like standardized monitoring and security auditing.

centurio.work focuses on a plug&play infrastructure, while moving all complexity into easy to interact with BPMN processes.

Acknowledgements: This work has been partially supported and funded by the Austrian Research Promotion Agency (FFG) via the "Austrian Competence Center for Digital Production" (CDP) under the contract number 854187.

References

1. Dadam, P., Reichert, M., Rinderle, S., Jurisch, M., Acker, H., Göser, K., Kreher, U., Lauer, M.: Towards Truly Flexible and Adaptive Process-Aware Information Systems. In: Int'l Conference UNISCON. pp. 72–83 (2008)
2. DIN Deutsches Institut für Normung e. V.: IEC 61131-3: Speicherprogrammierbare Steuerungen-Teil 3: Programmiersprachen. Berlin: Beuth (2003)
3. DIN Deutsches Institut für Normung e. V.: IEC 62264 - Enterprise control system integration (2016)
4. Geisberger, E., Broy, M.: agendaCPS: Integrierte Forschungsagenda Cyber-Physical Systems, vol. 1. Springer-Verlag (2012)
5. Mangler, J., Rinderle-Ma, S.: CPEE - Cloud Process Exection Engine. Eindhoven (Sep 2014), `https://eprints.cs.univie.ac.at/4136/`
6. Minguez, J., Lucke, D., Jakob, M., Constantinescu, C., Mitschang, B., Westkämper, E.: Introducing SOA into production environments–the manufacturing service bus. 43rd. CIRP International Conference on Manufacturing Systems, Vienna (2010)
7. OMG: Business Process Model and Notation (BPMN) Version 2.0. OMG Document formal/2011-01-03 (Jan 2011), `http://www.omg.org/spec/BPMN/2.0/`
8. Pauker, F., Ayatollahi, I., Kittl, B.: Service Orchestration for Flexible Manufacturing Systems using Sequential Functional Charts and OPC UA. IN-TECH - International Conference on Innovative Technologies 9, 11–09 (2015)
9. Schel, D., Henkel, C., Stock, D., Meyer, O., Rauhöft, G., Einberger, P., Stöhr, M., Daxer, M.A., Seidelmann, J.: Manufacturing service bus: an implementation. In: 11th CIRP Conf. Intell. Comput. Manuf. Eng. (2017)
10. VDI/VDE-Gesellschaft Mess-und Automatisierungstechnik, Kowalewski, S.: Cyber-physical systems: Chancen und Nutzen aus Sicht der Automation. VDI/VDE-Gesellschaft Mess-und Automatisierungstechnik (2013)
11. Årzén, K.E.: Grafchart: A graphical language for sequential supervisory control applications. IFAC Proceedings Volumes 29(1), 4831–4836 (1996)