Towards a real-time capable plug & produce environment for adaptable factories

Christian Eymüller, Julian Hanke, Alwin Hoffmann, Alexander Poeppel, Constantin Wanninger, Wolfgang Reif

Angaben zur Veröffentlichung / Publication details:

Eymüller, Christian, Julian Hanke, Alwin Hoffmann, Alexander Poeppel, Constantin Wanninger, and Wolfgang Reif. 2021. "Towards a real-time capable plug & produce environment for adaptable factories." In *2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Vasteras, Sweden, 7-10 September 2021*, edited by Moris Behnam, Lucia Lo Bello, Thomas Nolte, and Marco Porta, 1–4. Piscataway, NJ: IEEE. https://doi.org/10.1109/etfa45728.2021.9613729.

Towards a Real-Time Capable Plug & Produce Environment for Adaptable Factories

Christian Eymüller*, Julian Hanke*, Alwin Hoffmann*, Alexander Poeppel*, Constantin Wanninger*, and Wolfgang Reif*
Institute for Software and Systems Engineering
University of Augsburg
Augsburg, Germany
*{eymueller, hanke, hoffmann, poeppel, wanninger, reif}@isse.de

***Abstract*—Industrial manufacturing is currently undergoing a transformation from mass production with inflexible production systems to individual production with adaptable cells. In order to ensure this adaptability of these systems, technologies such as plug & produce are needed, to integrate, modify and remove devices at runtime. Therefor an exact description of the system, the products and the capabilities / skills of the devices is essential as well as a network for communication between the devices. Deterministic data transmission is particularly important for distributed control systems. We propose an architecture for plug & produce mechanisms with hard real-time capable communication paths between the cyber-physical components using OPC UA PubSub over TSN and the ability to load and execute real-time critical tasks at runtime.**

***Index Terms*—Skills, Plug-&-Produce, Real-Time, OPC UA, TSN**

I. Introduction

Industry 4.0 induces disruptive changes to manufacturing [1]: Development and innovation cycles are becoming increasingly shorter, while the products need to be individualised to the customers' requests. This requires manufacturing processes to be flexible and customizable to different types of products. Hence, these processes are required to be set up and adapted quickly. Moreover, less standardisation of products and their manufacturing poses new challenges to quality assurance. In particular, this leads to a flexible online monitoring of a manufacturing process and the possibility to adapt the process in real-time in order to avoid reject parts.

This results in large research efforts in intelligent manufacturing [2] in order to cope with these challenges. Cyber-physical production systems [3], digital twins [4] of the product and its manufacturing process as well as multi-functional robot-based automation systems for flexible and adaptable assembly [5] are promising research areas. However, most of these approaches require an environment where automation devices can simply be plugged in and communicate with other devices, exchange product descriptions or process models in order to start producing. Similar to Plug & Play in the computer domain, this idea is called *Plug & Produce* [6].

While most of the current plug & produce systems neglect the requirement for real-time communication between automation components, we think it is essential to cope with the aforementioned challenges, in particular for online monitoring and real-time adaption of manufacturing processes. The contribution of this paper is to introduce an architecture for a hard real-time capable plug & produce environment based on OPC UA and Time-Sensitive Networking (TSN).

The structure of this paper is as follows: Related work is presented in Sect. II. The objectives of our real-time capable plug & produce environment and an overview of its structure is given in Sect. II. A case study to show the advantages of the approach is introduced in Sect. IV. Sect. V concludes the paper and gives an outlook.

The following terms will be used in the further course of this paper: A *resource* is a device (e. g. a gripper) with a uniform communication interface and a description of the device and its capabilities. A *skill* (e. g. `GRIP`) is a executable capability of one or more resources.

II. Related Work

Fundamental for the realization of a plug & produce environment is a uniform, vendor-independent communication standard and a unified description of products, processes, and resources. OPC UA [7] covers several of these requirements through its various communication channels and an integrated information model [8]. Since many industrial applications have certain real-time requirements [9], a time-deterministic communication is also necessary [10]. Through the integration of Time-sensitive Networking (TSN) into OPC UA PubSub Communication also data transmissions with timing guarantees are possible as well [11].

A close integration between products, processes, and resources is necessary for the successful planning and implementation of production plants [12]. As mentioned by Schleipen et al. [13], the linking of information to processes, products, and resources is especially important. It must be clarified in which form and where the information of the individual elements is stored and how they can be efficiently combined. Here, it is possible to combine the advantages of the OPC UA information model with the advantages of semantic web technologies [14]. There have already been several

This work partly presents results of the project *WiR Augsburg* which is funded by the German Federal Ministry of Education and Research (BMBF) and the Bavarian Government.

approaches to implement plug & produce approaches [6], [15], [16]. Several of these systems also rely on OPC UA as communication middleware, however, OPC UA client-server communication is used. A real-time capable plug & produce environment can only be developed if the exchange of information about distributed resources is also real-time capable. Due to the communication overhead caused by the query-based and connection-oriented communication of OPC UA client-server communication, deterministic data exchange is only possible to a certain extent. The broker-less communication of OPC UA PubSub [17] is better suited for real-time transmission, but only with the addition of TSN it becomes real-time capable. In the approach of Zimmermann et al. [18], a plug & produce implementation with OPC UA PubSub communication has been presented. It is also mentioned that a simple switch to OPC UA PubSub over TSN should be possible. Since OPC UA PubSub over TSN is based on TSN Scheduled Traffic [19], the switch from OPC UA PubSub to OPC UA PubSub over TSN is associated with several challenges and is not trivial. Especially the configuration of the network is more complex, since, for example, time slots must be reserved for communication. How an automatic configuration of the TSN communication in the context of plug & produce can be implemented is presented in [20].

III. Objectives and System Overview

The aim of the approach presented in this paper is the design and implementation of a plug and produce environment with intelligent cyber-physical systems for an adaptive industrial production plant. For this purpose, the following objectives must be met:

1) Distributed and real-time capable execution of skills
2) Process modelling and process implementation with skills
3) Automatic monitoring of flexible processes
4) Linking the information on processes, products, and resources
5) Automatic online feedback of the quality data to adapt the production process

Figure 1 shows the basic architecture of the plug & produce environment. OPC UA is used as communication middleware. In this case, a distinction is made between two information channels. Client-Server OPC UA communication is used for the configuration of assets, the exchange of descriptions and the exchange of non-real-time critical data, while the OPC UA over TSN communication channel is used for applications in which hard real-time constraints must be adhered.

Each *resource* in the system has an asset administration shell [21] which consists of three essential parts. The first part are the control services, which in turn consist of the hardware driver for the resource (this represents the interface between the actual hardware and the basic skills, for example, a gripper only can communicate via a specific fieldbus), basic skills in form of executable services (for example, the skill `Grip` of a gripper or the skill `MoveToPosition` of the robot) and additional software components which can be loaded during runtime (for example, additional or composed skills). Preliminary work for loading software components during runtime

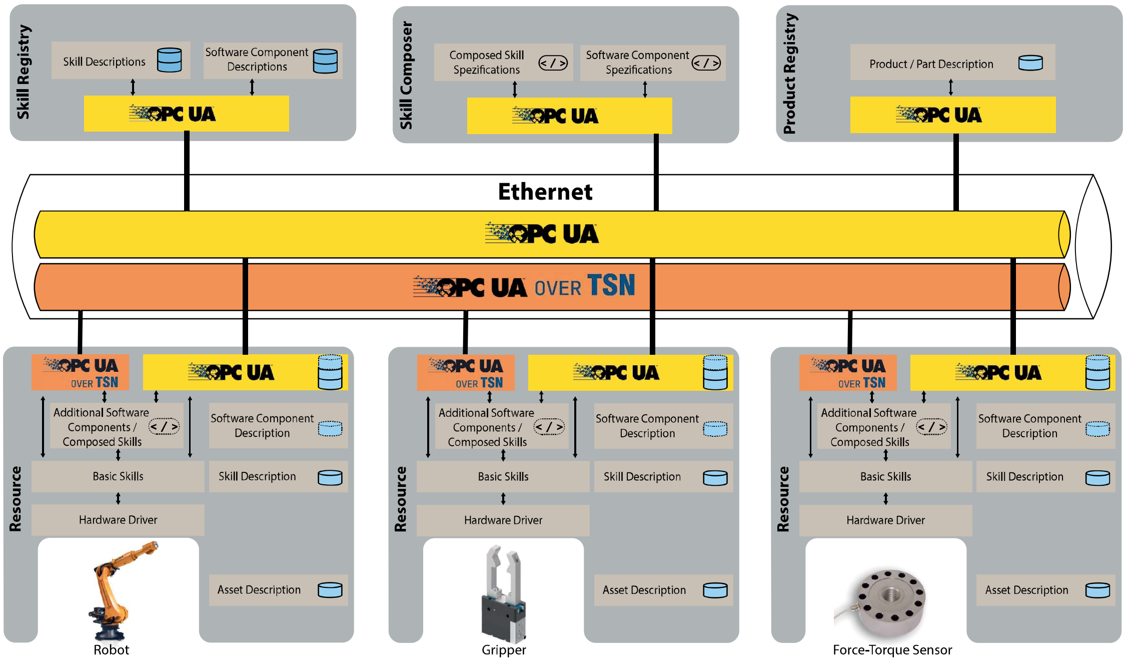

Figure 1: System architecure for a real-time capable plug & produce environment using OPC UA and OPC UA over TSN as middleware.

was done in [22]. The second part is responsible for describing the resource and its skills. If additional software components are loaded the description of these software components is added as well. The last and third part is the communication component. It creates a network between multiple devices and allows other elements in the system to access descriptions and control services of other resources.

Each network requires a *skill registry*. The task of the skill registry is to collect all available skills in the system and provides the skill descriptions and the description of additional loaded software components to other system members. The skill registry also serves as an endpoint for querying existing skills. Another system member is the *skill composer*. The skill composer ensures that several skills can interact with each other and can be combined to more complex skills. The skill composer provides additional software component specifications for resources that can be loaded into the control services of a resource. For example, a software component specification is available that implements the composed skill `Pick&Place`. This composed skill has dependencies to the basic skills `MoveToPosition` and `Grip`. First the skill registry is queried whether all dependencies can be met and which resources fulfil the dependencies. Further, it must be checked whether the resources can be connected to each other. For example the gripper must be connected to the robot to be able to execute the `Pick&Place` skill. Subsequently, the software component of the composed skill `Pick&Place` is loaded to one of the resources holding one of the needed basic skills. For instance, the composed skill is loaded to the robot that has the basic skill `MoveToPosition`. The interface of the existing basic skill can be accessed directly on the resource the additional software component is loaded to. If additional skills are required, a TSN connection is established automatically to the resource which has the required skill. In the example, a TSN connection to the gripper would be established in order to access the basic skill `Grip`. Preparatory work has already been done in the fields of automatic configuration of TSN connections [20] and synchronisation of skills via TSN [23]. Only a small amount of data is required to trigger the skill of another resource. In contrast to other approaches that implement composed skills as pure service calls in the OPC UA network, this approach conserves bandwidth and enables the execution of real-time capable composed skills, since only the network resources actually required are used. Due to the low bandwidth requirements, many resources can be operated in parallel and the system can be scaled up easily. With this architecture, however, it is also possible to compose skills using simple service calls. Although, in this configuration only applications with soft real-time requirements are possible.

Apart from skills for controlling resources, also skills for monitoring the resource are offered. Analogous to the control skills, additional software components can be used here. For example, a force-torque sensor has the skill `GetForces`, which returns the forces in all three degrees of freedom. An additional software component can be added to the sensor to calculate the total force acting on the sensor. It is even possible to monitor multiple resources with composed monitoring skills. Further, additional skills and software components can be added to implement an automatic online feedback on the quality of the executed process and it may also be possible to adapt the process directly depending on the quality data.

Every system has a *Product Registry* to bring product and part descriptions into the system. Information about the products and parts is stored here, so that this data can be used to adapt skills. As a result, limitations of skills can be , such as the fact that heavy products require a different gripper than light and fragile products.

IV. Case Study

The concept is evaluated using a case study within the innovation laboratory of the project *WiR Augsburg*[1]. The aim is to produce variable products with an adaptable system while guaranteeing hard real-time communication and deterministic execution of skills in this changing system. A 6-axis robot on a linear axis equipped with an automatic tool change system serves as the basis for an adaptable system. The robot is equipped with a force-torque sensor in order to be able to implement precise force-controlled applications. The automatic tool change system enables a large number of different tools (e. g. different grippers, screwdrivers, cameras, sensors, ...) and thus a multitude of different skills (e. g. `Grip`, `Tighten`, `Inspect`, `Measure` ...). Figure 2 illustrates the experimental setup. In the robot cell, there are several tool trays to equip the robot with different skills. Furthermore, several processing stations are installed for the assembly and handling of products. Due to the large working area of the robot with its linear axis the different processing stations and tool trays can be reached without problems.

Each system component (i.e. robots, sensors, tools) is equipped with a TSN capable industrial computer (IPC) which serves as the hardware component of the presented administration shell of a resource. It contains the self-description of the resource (e. g. parallel gripper with a holding force of 200 N, a full stroke of 85 mm, ...) as well as the executable skills (i. e. `Grip`) and can be supplemented with additional software components (e. g. `Pick&Place`).

The assembly of aluminium groove profiles is examined as a case study for a flexible production. The flexibility of the products results from the different lengths of profiles, different profile connectors (fixed and adjustable, cf. Figure 3) and different mounting options of the connectors (plug, screw, pin, ...). Even through the use of only one connector type, countless product variants can be produced [24]. This means that the assembly process as well as the handling process of the individual parts and the finished product is highly dependent on the product specification and must therefor be adaptable.

Due to the wide variety of the products and parts, simple descriptions of skills (e. g. `Grip`) are not sufficient. For example in the case of gripping, a precise description is

[1] https://www.uni-augsburg.de/de/forschung/einrichtungen/institute/amu/wir-augsburg/innovationslabor/

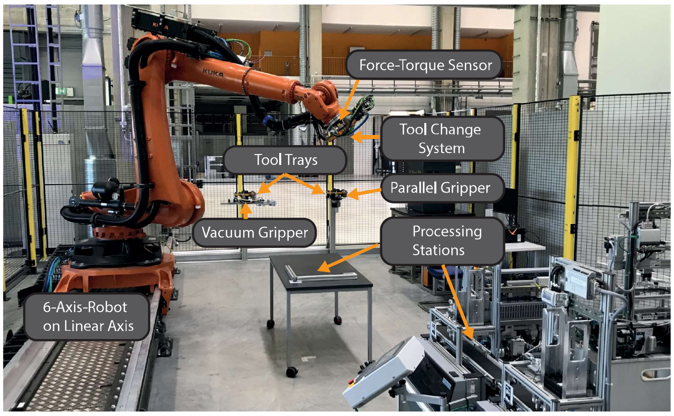

Figure 2: Experimental setup for the plug & produce environment

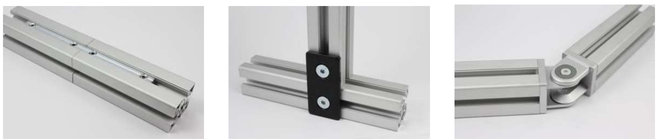

Figure 3: Aluminium groove profiles with different types of connectors (straight, 90-degree, adjustable)

needed up to which size and weight a part can be grasped and information is required whether a part is gripped positively or non-positively. A close connection between the product description and the skill description is therefor required.

V. Conclusion

In this paper, an architecture is presented for the realization of plug & produce mechanism which can handle hard real-time requirements. A combination of OPC UA and OPC UA PubSub over TSN is used as real-time capable communication middleware. Also, a concept for loading real-time skills into production-resources at runtime is described. In the next step, the missing components of the architecture have to be implemented, for example, the online adaption of production processes. Existing ones have to be developed further (i. e. adding and removing devices and communication paths in OPC UA over TSN) and subsequently the robustness of the architecture must be evaluated using the presented case study.

References

[1] H. Lasi, P. Fettke, H.-G. Kemper, T. Feld, and M. Hoffmann, "Industry 4.0," *Business & information systems engineering*, vol. 6, no. 4, pp. 239–242, 2014.

[2] R. Y. Zhong, X. Xu, E. Klotz, and S. T. Newman, "Intelligent manufacturing in the context of industry 4.0: a review," *Engineering*, vol. 3, no. 5, pp. 616–630, 2017.

[3] L. Monostori, "Cyber-physical production systems: Roots, expectations and R&D challenges," *Procedia Cirp*, vol. 17, pp. 9–13, 2014.

[4] T. H.-J. Uhlemann, C. Lehmann, and R. Steinhilper, "The digital twin: Realizing the cyber-physical production system for industry 4.0," *Procedia Cirp*, vol. 61, pp. 335–340, 2017.

[5] L. Nägele, A. Schierl, A. Hoffmann, and W. Reif, "Multi-robot cooperation for assembly: Automated planning and optimization," in *Informatics in Control, Automation and Robotics, 2019 Revised Selected Papers*, ser. LNEE. Springer, 2021.

[6] S. Profanter, A. Perzylo, M. Rickert, and A. Knoll, "A Generic Plug & Produce System Composed of Semantic OPC UA Skills," *IEEE Open Journal of the Industrial Electronics Society*, vol. 2, pp. 128–141, 2021.

[7] T. Burke, "OPC Unified Architecture: Interoperability for Industrie 4.0 and the Internet of Things," *Opc Foundation*, vol. 1, pp. 01–44, 2017.

[8] "OPC Unified Architecture Part 1: Overview and Concepts (OPC 10000-1)," OPC Foundation, v 1.04, Nov. 2017, https://opcfoundation.org/developer-tools/specifications-unified-architecture/part-1-overview-and-concepts/.

[9] A. Hoffmann, A. Angerer, F. Ortmeier, M. Vistein, and W. Reif, "Hiding real-time: A new approach for the software development of industrial robots," in *Proc. 2009 IEEE/RSJ Intl. Conf. on Intell. Robots and Systems, St. Louis, Missouri, USA*. IEEE, Oct. 2009, pp. 2108–2113.

[10] J. Otto, S. Henning, and O. Niggemann, "Why cyber-physical production systems need a descriptive engineering approach – a case study in plug & produce," *Procedia Technology*, vol. 15, pp. 295–302, 2014.

[11] J. Pfrommer, A. Ebner, S. Ravikumar, and B. Karunakaran, "Open source OPC UA PubSub over TSN for realtime industrial communication," in *23rd IEEE Intl. Conf. on Emerging Technologies and Factory Automation (ETFA)*, Sep. 2018, pp. 1087–1090.

[12] A.-F. Cutting-Decelle, R. I. Young, J.-J. Michel, R. Grangel, J. Le Cardinal, and J. P. Bourey, "ISO 15531 MANDATE: a product-process-resource based approach for managing modularity in production management," *Concurrent Engineering*, vol. 15, no. 2, pp. 217–235, 2007.

[13] M. Schleipen and R. Drath, "Three-view-concept for modeling process or manufacturing plants with AutomationML," in *2009 IEEE Conference on Emerging Technologies & Factory Automation*. IEEE, 2009, pp. 1–4.

[14] J. Bakakeu, M. Brossog, J. Zeitler, J. Franke, S. Tolksdorf, H. Klos, and J. Peschke, "Automated reasoning and knowledge inference on OPC UA information models," in *2019 IEEE International Conference on Industrial Cyber Physical Systems (ICPS)*. IEEE, 2019, pp. 53–60.

[15] M. Onori, N. Lohse, J. Barata, and C. Hanisch, "The IDEAS project: plug & produce at shop-floor level," *Assembly automation*, 2012.

[16] J. Pfrommer, D. Stogl, K. Aleksandrov, S. E. Navarro, B. Hein, and J. Beyerer, "Plug & produce by modelling skills and service-oriented orchestration of reconfigurable manufacturing systems," *at-Automatisierungstechnik*, vol. 63, no. 10, pp. 790–800, 2015.

[17] "OPC Unified Architecture Part 14: PubSub (OPC 10000-14)," OPC Foundation, v 1.04, Feb. 2018, https://opcfoundation.org/developer-tools/specifications-unified-architecture/part-14-pubsub.

[18] P. Zimmermann, E. Axmann, B. Brandenbourger, K. Dorofeev, A. Mankowski, and P. Zanini, "Skill-based engineering and control on field-device-level with OPC UA," in *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*. IEEE, 2019, pp. 1101–1108.

[19] IEEE Standards Association *et al.*, "IEEE Standard for Local and Metropolitan Area Networks—Bridges and Bridged Networks—Amendment 25: Enhancements for Scheduled Traffic," *Amendment to IEEE Std*, vol. 802, pp. 1–57, 2016.

[20] C. Eymüller, J. Hanke, A. Hoffmann, M. Kugelmann, F. Grätz, and W. Reif, "RealCaPP: Real-time capable Plug & Produce communication platform with OPC UA over TSN for distributed industrial robot control," in *17th IEEE Intl. Conf. on Automation Science and Engineering (CASE)*. IEEE, 2021.

[21] T. Bangemann, C. Bauer, H. Bedenbender, M. Diesner, U. Epple, F. Elmas, J. Friedrich, T. Goldschmidt, F. Göbe, and S. Grüner, "Industrie 4.0-Technical Assets: Basic terminology concepts life cycles and administration models," *VDI/VDE and ZVEI*, 2016.

[22] C. Eymüller, C. Wanninger, A. Hoffmann, and W. Reif, "Semantic Plug and Play—Self-Descriptive Modular Hardware for Robotic Applications," *International Journal of Semantic Computing*, vol. 12, no. 04, pp. 559–577, 2018.

[23] C. Eymüller, J. Hanke, A. Hoffmann, M. Kugelmann, and W. Reif, "Real-time capable OPC-UA Programs over TSN for distributed industrial control," in *25th IEEE Intl. Conf. on Emerging Technologies and Factory Automation (ETFA)*, vol. 1. IEEE, 2020, pp. 278–285.

[24] K. Nottensteiner, T. Bodenmueller, M. Kassecker, M. A. Roa, A. Stemmer, T. Stouraitis, D. Seidel, and U. Thomas, "A complete automated chain for flexible assembly using recognition, planning and sensor-based execution," in *Proceedings of ISR 2016: 47st International Symposium on Robotics*. VDE, 2016, pp. 1–8.