Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

A Framework for Operator Assist Apps of Automated Systems

Jan Christoph Wehrstedt*, Annelie Sohr*, Tim Schenk*, Roland Rosen*, Yayun Zhou

**Siemens AG, Corporate Technology, Research in Digitalization and Automation, Munich 81739 Germany (e-mail: janchristoph.wehrstedt@siemens.com, annelie.sohr@siemens.com, tim.schenk@siemens.com, roland.rosen@siemens.com)*

Abstract: The role of assist functions for complex production systems is increasing. This paper introduces a framework for the developing simulation-based assist functions by using reusable microservices, a common data format and a low coding HMI. This approach is illustrated for an online bottleneck detection for a production system.

Keywords: Operator Assistance, Bottleneck Identification, Assist System Framework, Plant Simulation, OPC UA

1. INTRODUCTION

Efficient operation of automated systems like production systems in process and manufacturing industry is a challenge. In addition to the consideration of quality, time and cost targets, more and more flexibility is required. The lot sizes become smaller while at the same time the variety of product types increases, and order changes or rather additional orders are also to be realized at shorter notice. In addition to classic automation the development of specific software applications, which provides data- and simulation- based decision approaches promises a solution to handle the task (Kuehn, 2018). Such software will include beyond the current state of the automated system further information from different sources, like logistics, maintenance and production planning. Furthermore, the technological vision of Digital Twin can and will support this direction of development by providing and connecting various information and models, e.g. about the product types to be produced and the production system with all its hierarchical levels itself (Kuehn, 2018; Rosen et al., 2019)

First elements of such solutions are already realized, for example on a component level an assist system for the operation of point machines for train switches (Boschert et al., 2018) and on an infrastructure level an operational assist system for water supply systems (Rosen et al., 2018). Simulation as an integral part of the plant or system life cycle will be a core technology of these application (Rosen et al., 2019). The simultaneous consideration of the different information extends the function of simulation-based assist systems compared to high-level automation functions and APC solutions too. Increasing production flexibility and efficiency is the economic objective and the reason, why such functions are required in today's tenders for automation solutions.

To realize a cost and time efficient realization of assistance systems it is necessary to develop a systematic approach which based on reusable components. In this paper we introduce in chapter 2 a framework for simulation-based assist systems and will illustrate it on an use case of bottleneck identification. This use case is introduced in chapter 3. In chapter 4 the instantiation of the framework for this use case will be described in detail for a demo plant. The paper close with a short summary and outlook.

2. ASSIST SYSTEM FRAMEWORK

2.1 Assist systems in the CPS Environment

In spite of its other advancements, the Industrie 4.0 concept does not state an innovative way to realize applications in that environment that even more efficiently integrate model-based evaluations like simulations and optimizations. Up to now the implementation of such applications is still a project- or customer-specific solution business requiring a team of automation, software, simulation and domain experts for each solution. Raising the efficiency in development and deployment entails a much better realization and dissemination of cyber-physical system applications – especially for small and medium enterprises.

Considering the requirements for such assist system functions as the mentioned monitoring, deviation detection, bottleneck analysis and optimization, the executed simulations or optimizations must be integrated in a runtime environment deployed close to but as an addition to the classical automation. These functions do not rely on measurements and observations provided by field devices, but also require information from higher levels of the original automation pyramid, like ERP and MES software. Furthermore, it must be possible to integrate services available in the internet or offered by customers, suppliers or other company-internal

Copyright lies with the authors

departments via cloud or service infrastructures. This required openness and utilization of various distributed information has been stated by the Industrie 4.0 initiative in the RAMI 4.0 model for upcoming cyber-physical systems and applications (Monostori, 2018; Sbaglia et al., 2019).

To support the transformation from a concept towards actual implementation in production, this paper introduces a framework approach, called assist system framework (ASF), and a corresponding process. The main paradigms the framework is based on are a model-based information flow and a service-oriented architecture. Thereby, the framework considers the application of these two main aspects for all architectural layers of such assist system functions – including the UI, the assist functions themselves, individual calculations (model-based or not) and the sources and stores of information.

By enabling a flexibility in the assembly of the assist systems functions and considering also the UI aspects in the framework, the target group of such applications is no longer restricted to the plant operators. Further stakeholders of the production system, such as management, service teams, engineers and even customers can now be provided custom-made UIs with underlying assist functions that are tailored to their specific needs.

The goal of the framework is to enable easy configuration, assembly and deployment of simulation- and optimization-based individual operational support functionality for all domains, stakeholders and plant sites.

2.2 Architecture

The model- and service-based approach of the framework results in the following four main implications and potentials. Encapsulating each assist system sub-function in a service enables extensive modularity in assembling complex applications out of single-purpose, often small-sized services. The definition of a common meta-model, by implication, facilitates firstly a much easier integration of simulations and optimizations into complex assist functions as they already utilize a model-based approach. Secondly this enables a seamless data exchange between all services (model-based or not) just by connecting them to a workflow sequence and relieves the developer of managing proprietary variable mappings. By having defined and selected the relevant services and having engineered a project-specific model of the plant, it is possible to use low-coding frameworks to assemble individual UIs, as these also base on parameterizable toolboxes and domain models for easy configuration. The process and concept of the assist system that enables all these potentials has been outlined in (Zhou et al., 2019).

As in the next chapter the application of the framework on a concrete assist system solution for manufacturing plants is explained in detail, the general concept behind the framework is explained in the following. The architecture depicted in Fig. 1 shows that the core framework consists of four building blocks.

There is a meta-model structure for the definition of domain-specific types of entities. These can be physical entities as well as information entities of every kind. The structure also capable of representing an instance-oriented plant/system model engineered by utilizing and parameterizing the domain-model types. An interface specification prescribes how each service declares its data contract with both the domain and the plant meta-model and specifies the required functionalities each service must implement. The core server performs the execution of an assist system function by both triggering the interface functions of the connected services in the correct order of sequence and interchanging current runtime model data with the services according to their data contracts. And finally, a backend provides generic endpoints for the access to all model values as well as for status and control functions for arbitrary assist system functions.

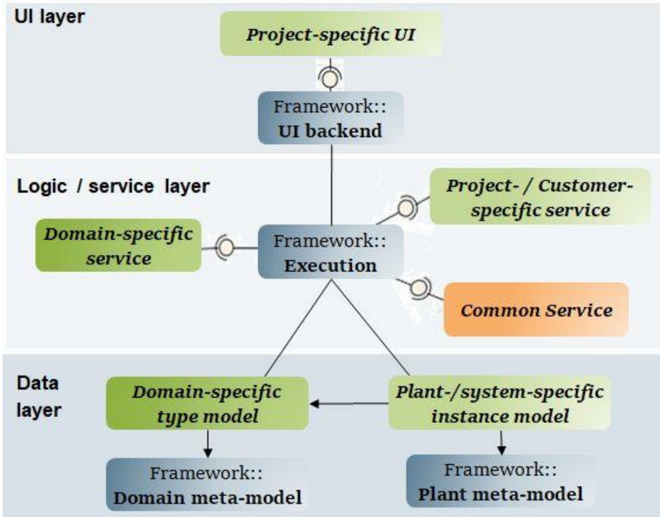

Fig. 1: General architecture of the assist system framework containing the four building blocks UI Backend, Execution, Domain meta-model and Plant meta-model

There is a meta-model structure for the definition of domain-specific types of entities. These can be physical entities as well as information entities of every kind. Then there is a meta-model structure capable of representing the plant/system specific instance-oriented model engineered by utilizing and parameterizing the domain-model types. An interface specification prescribes how each service declares its data contract with both the domain and the plant meta-model and specifies the required functionalities each service must implement. The core server performs the execution of an assist system function by both triggering the interface functions of the connected services in the correct order of sequence and interchanging current runtime model data with the services according to their data contracts. And finally, a backend provides generic endpoints for the access to all model values as well as for status and control functions for arbitrary assist system functions.

This architectural setup serves as the basis for all three steps of the assist system process, namely the (project-unspecific) preparation by software, simulation and domain experts, the project-specific configuration and engineering by an assist system engineer and the site-deployment by an assist system service technician. The generated artefacts build upon each other as displayed in Fig. 2.

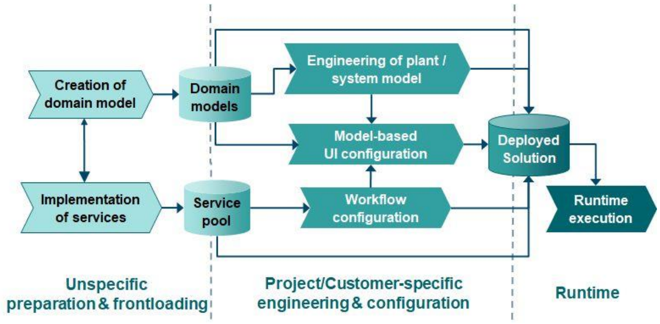

Fig. 2: Three steps (Preparation & Frontloading, Engineering & Configuration and Runtime) and artefacts of the ASF process

2.3 Model-based flow

One specific aspect of the framework approach is the separation of the sequential workflow of services and the flow of data during a runtime execution. In the IoT domain there are also visual programming approaches that configure sequential workflows out of small existing function blocks (e.g. "Node-RED," n.d.). But in that approaches the process flow is always equivalent to the data flow. This might be reasonable, if the exchanged data between individual function blocks is both restricted to simple, non-structured data – as the functions themselves are very simple – and still diverse enough so that it is not possible to specify a domain model on which all functions can operate on.

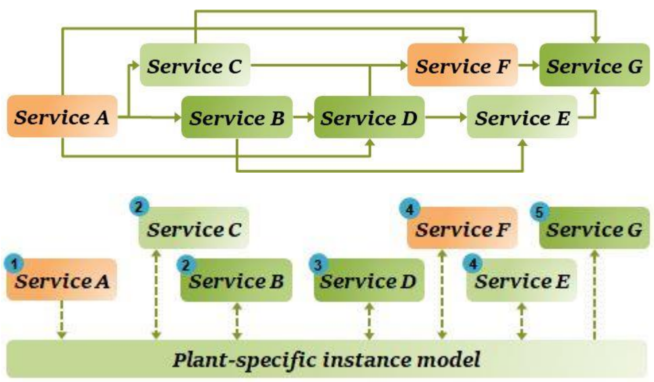

Fig. 3: Combined (top) and separated (bottom) process and data flow

The type-oriented ASF approach enables that services exchange their input and output data during runtime only with the current runtime model and not in between the services. Therefore, the connections only specify, which service is executed in the sequence flow before, after or in parallel to other services. Which service calculated a specific variable value is unknown to a service that requires that variable as input. This ensures consistency of data over the whole execution, simplifies the configuration of a complex workflow sequence as the order of sequence can be even identified automatically (see Fig. 3) and enables flexibility as services do not have to stick to a certain input / output specification to be used in combination with other services.

3. USE CASE: BOTTLENECK IDENTIFICATION

In this section we are explaining a typical situation where assist functions can play a role for production systems. In a manufacturing line, a bottleneck is a work stage that gets more work requests than its maximal capacity. A bottleneck can be identified as a root cause, which interrupts the material flow, so the process takes longer time than during normal operation. If one can identify bottlenecks in a manufacturing line and remove the causes, the manufacturing can be speeded up and the productivity can be increased. Usually the root cause and the position of the actual bottleneck varies so that the identification is not trivial.

Simulation is often used to help identifying bottlenecks before planning a new production system. Simulation allows a user to execute virtual experiment by running "what-if-scenarios". These virtual experiments can be conducted at different levels of production facilities. It is an efficient way to optimize material flow, resource utilization and logistic (Bangsow, 2008). This optimization is usually based on selected scenarios and not a global optimization.

There are many tools available on the market, which can run material flow simulation and identify bottlenecks in a manufacturing system. It allows users to model, simulate, visualize and optimize production systems and process. However, between engineering and operation certain factors may change, after the simulation model is prepared. In those cases, the simulation result does not predict the manufacturing system behavior accurately.

This paper suggests an online assist system which integrates simulation into the operation phase in order to give additional assistance to the operators. In order to avoid the previous mentioned problem, a manufacturing system is often monitored. The processing time of individual sub steps is measured and compared with the prepared simulation model respectively previous runs. Once detected deviations exceed specified tolerances, the simulation model will be updated and together with the current production orders consequences of this change will be identified based on the current system state. Possible upcoming bottlenecks can be detected before they occur, and countermeasures can be taken.

4. ASSIST FUNCTIONS DESCRIPTION

With the help of the ASF, assist functions can be developed for the given use case. In order to verify the assist functions in our lab, we adopt a FischerTechnik model as a mockup of a manufacturing system. In this session, the construction of this verification system is introduced, then the assist functions required for the use case are described. Afterwards, the functions are realized by implementing reusable services with the ASF. Lastly, we show the orchestration of the services for every function and describe the corresponding User interface.

4.1 Assist Functions for the Use Case

We construct a manufacturing system with FischerTechnik components combined with real automation soft- and hardware to verify our use case in our lab. The mini manufacturing system consists of five stations and several conveyors. The five stations include a warehouse and its corresponding gripper robot, a multi-process station, a sorting line with a color sensor and three buffers, a pick-and-place robot, and a second multi-processing station with a drilling machine as well as a milling machine.

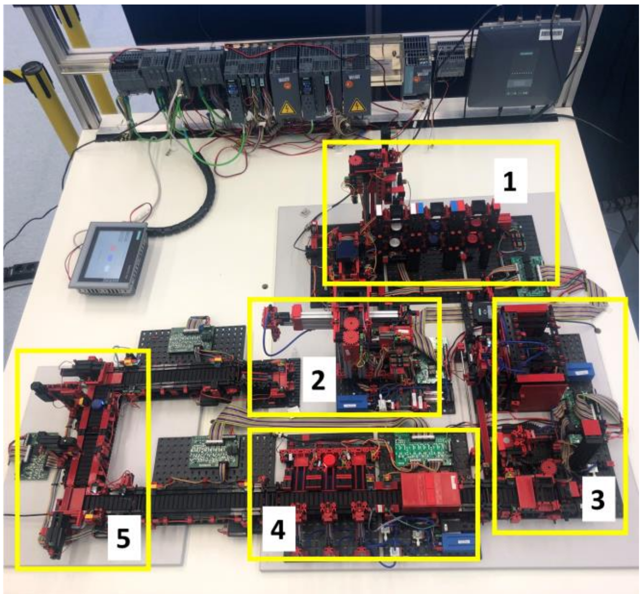

Fig. 4: FischerTechnik factory model

Fig. 4 shows a manufacturing system. On the top of the picture is the automated warehouse (1), where all the workpieces are stored. They are distributed by a gripper robot (2) and placed to a conveyor system near to the warehouse. In the center, a pick-and-place robot picks a workpiece from conveyor and place it to corresponding stations. The multi-process station (3) on the right-hand side contains a heating system, a turn table and a milling machine. A sorting line (4) on the bottom is connected to the multi-process station via conveyor, which sorts the workpieces by their colors. Certain workpieces are pushed into the buffer and will be collected by the pick-and-place robot. If a workpiece is not pushed to the buffer, it travels further via conveyor and passes two further stations (5) on the left-hand side. After the two manufacturing steps milling and drilling, the workpiece is stored in the buffer.

Three PLCs (two PLC 1500 & one PLC1200) are used to control the system. The PLCs are connected through PROFINET. The manufacturing system is also equipped with RFID antenna. The workpiece carries a RFID tag that is coded with the manufacturing information such as processing time of evolving stations.

The workpieces are processed by different stations and the process time is also different. The color of the workpiece stands for the product type. We can call the analysis function to identify possible bottlenecks for given production order (such as red, white, blue). In the optimization, alternative scenarios with different production orders are simulated. The scenarios are sorted with total production time.

To verify the assist function monitor, we change the heating system process time of the red workpiece. This change should be detected by the monitor assist function and raises an alarm. The operator can trigger the analysis function to identify possible bottlenecks in the system. If the change causes bottleneck and prolongs the production time. The optimization function can be called to select the best production order to avoid bottlenecks.

4.2 Development of assist functions

Coming back to the use case described above, five assist function are required in order to assist the operator

Calibration: After the simulation model is prepared for the manufacturing system, all parameters are calibrated based on field data. This step should be done once at the commissioning respectively after reconfigurations.

Monitor: The manufacturing system is monitored and processing time of each station for given product is measured. The field data is collected by the system monitor function and compared with predefined simulation models. Once a detected deviation exceeds tolerance, an alarm / warning is triggered, which allows the operator to start an additional investigation. Once a confirm signal is sent by the operator, the simulation model is updated. The field data can be provided via OPC UA interface provided by the PLC and the data is transferred periodic during operation.

Analysis: Based on the updated processing time, a simulation is started to analyze the production order. The simulation analyses the material flow in the manufacturing system and identifies possible bottlenecks in the system. Certain KPIs such as total manufacturing time and outputs are also calculated. The operator can make decisions based on the analysis results and KPI values.

Optimization: A series of alternative scenarios are created and simulated. Given KPIs are evaluated and the alternative scenarios are sorted according to the KPIs. This function is triggered by the operator and provides operation assistance. There are different strategies to eliminate bottlenecks. The easiest solution is to change the order of producing certain products. More complex solutions include changing production steps for certain selected product, changing manufacturing system configuration (add extra stations or re-configure station to perform different functions).

Adaption: An operator can accept and passes the change to the control system to continue production with optimized system configuration and control strategy.

In the next part, we will show the implementation in ASF to realize the assist functions.

4.3 Services to realize Assist Functions

Following the approach described in Fig. 3 the following required services are described:

OPC UA Service: This service is a module for the communication with the plant respectively its automation system. On one hand it is a classical OPC client connected to the automation system, on the other hand it is connecting to the framework to provide monitor variables and to trigger or parameterize functions in the automation system based on decision supported by the framework.

Data Preparation: As the data from/and to the automation system follows the purpose of classical network, the values read by the OPC Service must be prepared, e.g. correcting of units, ensuring the quality, identifying thresholds or new values.

Model Update: The service Model Update updates the common data household after the run of every service if necessary. This service should only be triggered if the quality is enough.

Anomaly Detection: This is a monitoring service which must be parameterized for every use. Its goal is to detect system changes and to inform the operator about this. Basically, it watches if certain system variables are leaving a certain range. Alarms and warnings will be displayed on the GUI

Order intake: As the OPC service connects the system to the automation system, the order intake service is giving the connection to the ERP system.

Production Plan: This is the interface to the MES system as a third source of input for the assist system.

Plant Simulation: This service updates the simulation model based on the changes in the data model and performs a material flow simulation. Selected results will be transferred to the service bottleneck detection

Bottleneck Detection: This service analysis the simulation results regarding possible bottlenecks. Its results will be displayed to the user to get informed about the current situation.

Scenario Generator and Evaluation: This is a corresponding service pair to find a better configuration. The generator creates different scenarios that are validated using simulation, until the evaluation model identifies the best configuration.

4.4 Developing an assist system using this framework.

In this section we show how the microservices described in 4.3 can be orchestrated to engineer the assist function for the Use case as described:

Calibration: The simulation model will be calibrated by measuring and processing time in all cells based on real values (see Fig. 5).

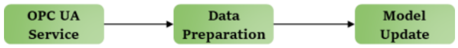

Fig. 5: Orchestration of microservices for calibration

Monitor: Runtime values coming via OPC will be permanently monitored, If anomalies are detected an alarm will be set and the model will be updated with the newly identified parameters (see. Fig. 6).

Fig. 6: Monitoring and anomaly detection

Analysis: The updated simulation model will now be updated with the parameters identified above as well as the current loads. A simulation will be then triggered to identify unwanted consequences Based on the simulation results will be analyzed regarding upcoming bottlenecks (see Fig. 7).

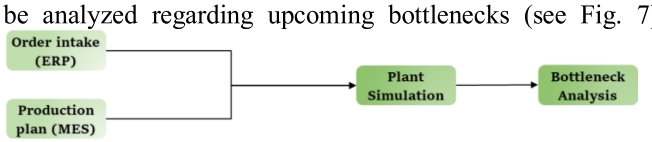

Fig. 7: Simulation for bottleneck identification

Optimization: A optimization kernel will now trigger a number of simulations in order to find the best configuration, e.g. production order, assignment from product towards production statement. This optimized order then will be suggested to the operator as a possible improvement of the production order (see Fig. 8).

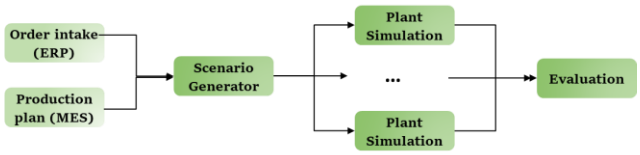

Fig. 8: Optimization of the production order

Adaption: Finally, after selecting the new operation strategy an update of the system via the OPC Client as well the internal data model initiate an update of the system (see Fig. 9).

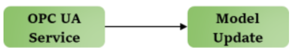

Fig. 9: Update of the model and the configuration of the production system

4.5 Developing a graphical user interface for the application.

The assist functions will be deployed together with a graphical user interface as shown in Fig. 10. As calibration usually is not a task/ interaction with the operator the GUI only shows the methods monitoring, analyze, optimize and adaption. In monitoring cyclic information either directly from the plant as well as the results of the monitoring are displayed (2). In case of a deviation the analysis method (3) can be triggered to analysis the effect of detected deviations. In case the analysis function shows any deviation the optimize function (4) can be used to find possible better configurations. The adaption method (5) triggers the reconfiguration with the changed production order.

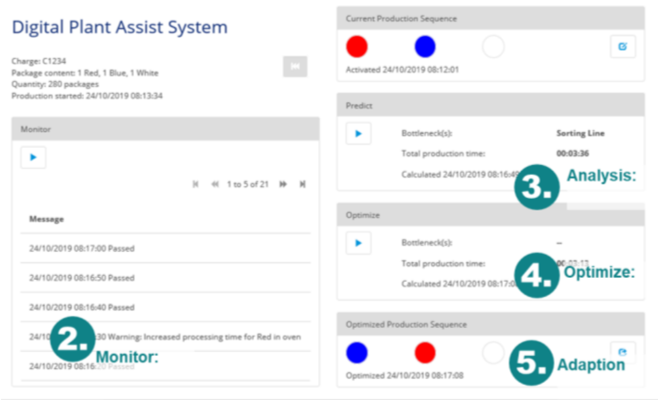

Fig. 10: Software GUI for assist the assist functions (2) Monitor, (3) Analysis, (4) Optimization and (5) Adaption

5. SUMMARY AND OUTLOOK

Assist systems with built-in data- and simulation- based decision approaches compromise more efficiency and flexibility for the operation of automated systems like process plants and manufacturing systems. They allow users to manage complex situations which arise, for example, through short-term order changes, new product types to be produced and additional operation goals and constraints like energy and resource efficiency. We presented a framework for the development of assist systems. Key features of this framework are the integration of data from various systems, the definition of workflows including simulation execution, and the preparation of configurable and extendible user interfaces. The assist system framework is illustrated on a use case of a bottleneck situation in a demo manufacturing plant. The presented assist system framework is not limited to production systems, it can be used for any kind of operator assistance with domain-specific libraries and further modules.

In the next step, further assist functions like “Forecasts on production system level” or “Optimized production of new orders” could be implemented. For that scenario, simulation will be used to predict upcoming system behavior as well as to schedule and control production changes. The mid-term vision goes in the direction to support several stakeholders with different task, e.g. maintenance engineers, plant operators and managers, in a seamless way. In a long-term vision, assist systems are a path to autonomous systems (Rosen et al., 2015). The number of operator interventions is getting less and done on higher and more abstract objectives.

REFERENCES

Bangsow, S., 2008. Fertigungssimulationen mit Plant Simulation und SimTalk, in: Anwendung und Programmierung Mit Beispielen und Lösungen. Carl Hanser Verlag München Wien.

Boschert, S., Rosen, R., Heinrich, C., 2018. Next generation digital Twin, in: Proceedings of the 12th International Symposium on Tools and Methods of Competitive Engineering TMCE 2018. Presented at the Proceedings of the 12th International Symposium on Tools and Methods of Competitive Engineering TMCE 2018, Delft, p. S.209-218.

Kuehn, W., 2018. Digital twins for decision making in complex production and logistic enterprises. Int. J. Des. Nat. Ecodynamics 13, 260–271. https://doi.org/10.2495/DNE-V13-N3-260-271

Monostori, L., 2018. Cyber-Physical Systems, in: Chatti, S., Tolio, T. (Eds.), CIRP Encyclopedia of Production Engineering. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 1–8. https://doi.org/10.1007/978-3-642-35950-7_16790-1

Node-RED [WWW Document], n.d. URL https://nodered.org/ (accessed 29.10.19).

Rosen, R., Jaekel, J., Barth, M., Stern, O., Schmidt-Vollus, R., Heinzerling, T., Hoffmann, P., Richter, C., Puntel Schmidt, P., Scheifele, C., 2019. Simulation und Digitaler Zwilling im Engineering und Betrieb automatisierter Anlagen - Standpunkte und Thesen des GMA FA 6.11, in: Automatisierungskongress. Presented at the Automatisierungskongress, VDI, Baden-Baden.

Rosen, R., von Wichert, G., Lo, G., Bettenhausen, K.D., 2015. About the importance of autonomy and digital twins for the future of manufacturing. IFAC-Pap. 48, 567–572.

Rosen, R., Boschert, S., Sohr, A., 2018. Next Generation Digital Twin. in atp magazin. Atp-Mag. 60, 86–96.

Rosen, R., Fischer, J., Boschert, S., 2019. Next Generation Digital Twin: an Ecosystem for Mechatronic Systems?, in: 8th IFAC Symposium on Mechatronic Systems MECHATRONICS 2019. Presented at the 8th IFAC Symposium on Mechatronic Systems MECHATRONICS 2019, Vienna.

Sbaglia, L., Giberti, H., Silvestri, M., 2019. The Cyber-Physical Systems Within the industry 4.0 Framework, in: Carbone, G., Gasparetto, A. (Eds.), Advances in Italian Mechanism Science, Mechanisms and Machine Science. Springer International Publishing, Cham, pp. 415–423. https://doi.org/10.1007/978-3-030-03320-0_45

Zhou, Y., Schenk, T., Allmaras, M., Massalimova, A., Sohr, A., Wehrstedt, J.C., 2019. Flexible Architecture to integrate Simulation in Run-Time Environment. Presented at the Automationkongress, VDI - Verein Deutscher Ingenieure e.V., Baden-Baden.

The work was funded by the BMBF project CrESt – Collaborative embedded systems (01IS16043Q).