DOI: https://doi.org/10.46354/i3m.2019.emss.021

CLOSED-LOOP-ENGINEERING – ENABLER FOR SWIFT RECONFIGURATION IN PLANT ENGINEERING

Janek Bender[(a)], Jana Deckers[(b)], Simon Fritz[(c)], Jivka Ovtcharova[(d)]

[(a),(b),(c) ,(d)] FZI Research Center for Information Technology, Haid-und-Neu-Strasse 10-14, 76135 Karlsruhe, Germany

[(a)]bender@fzi.de, [(b)]deckers@fzi.de, [(c)]fritz@fzi.de, [(d)]ovtcha@fzi.de

ABSTRACT

Manufacturing industries face increasingly complex customer demands in terms of product customisation and delivery times. At the same time, batch sizes decline and production needs to adapt to flexible demands and highly configurable products. These circumstances pose significant challenges for both plant engineers and process planners. In this work, we explore a novel approach in order to establish a Closed Loop Engineering process chain for the Digital Factory by combining state of the art technologies from the fields of Production and Process Automation, Virtual Reality (VR), and Internet of Things (IoT). As a proof of concept, we developed a VR-demonstrator using the CONTACT Elements for IoT platform in conjunction with a web-based production process planner and Manufacturing Execution System (MES) written by the FZI in Java as well as the Unity game engine for real-time simulation in VR.

Keywords: Digital Twin, Plug&Produce, AutomationML, Virtual Reality

1. INTRODUCTION

Manufacturing industries face increasingly complex customer demands in terms of product customisation and delivery times. At the same time, batch sizes decline and the production needs to adapt to flexible demands and highly configurable products. These circumstances are also known in the literature as Mass Customisation (Tseng et al. 2017). Aside from the implications for product developers, they also pose significant challenges for plant engineers as well as process planners.

Industry, public, and science attempt to meet these challenges by working towards a large-scale transformation of the manufacturing industries, in Germany referred to as Industrie 4.0 (Industry 4.0). Other economic heavyweights employ similar strategies such as Made in China 2025, the Industrial Internet Consortium in the United States of America, or Society 5.0 in Japan.

At the core of the German efforts lies the Smart Factory, for which the Digital Factory forms a key requirement. In this work, we explore a novel approach in order to establish a closed-loop-engineering process chain for said Digital Factory by combining state of the art technologies from the fields of Production and Process Automation, Virtual Reality (VR), and Internet of Things (IoT).

In the following, the state of the art in the key technologies for this paper is reviewed. Then, the concept is described from a theoretical viewpoint leading to a section about the prototypic implementation in form of a research demonstrator. Finally, the results are evaluated and the paper is concluded.

2. STATE OF THE ART

This section gives a brief overview about the concepts and technologies required to describe and implement our approach. Namely we touch on Virtual Reality, (Industrial) Internet of Things, the Digital Twin, Manufacturing Execution Systems, and the Digital Factory as the element that ties everything together.

2.1. Virtual Reality

The idea to leverage Virtual Reality for factory planning and virtual commissioning is not entirely new. (Menck et al. 2012) There have been numerous academic and commercial solutions emerging in recent years like taraVRbuilder (tarakos 2019), IPO.Log (IPO.Plan 2019), R3DT (R3DT 2019), Boxplan (SALT AND PEPPER 2019) or game4automation (in2sight 2019) focussing on different aspects of plant engineering and virtual commissioning in VR, or simple 3D respectively. Choi et al. (2015) present a comprehensive list of VR applications in the manufacturing industries. Our approach however is tightly integrated into existing real-world monitoring and control software and thus provides a lightweight tool for plant engineers to swiftly simulate the impact of layout changes on a process level. Hence we required our own engine capable of VR, for which we employed the commercial game engine Unity (Unity Technologies 2019). It is affordable, accessible, surrounded by a large community, and built executables run on various operating systems including Windows and Linux distributions.

2.2. Industrial Internet of Things

(Boyes et al. 2018) define the Industrial Internet of Things (IIoT) as "*a system comprising networked smart objects, cyber-physical assets, associated generic information technologies and optional cloud or edge computing platforms, which enable real-time, intelligent,* *and autonomous access, collection, analysis, communications, and exchange of process, product and/or service information, within the industrial environment […]*". Note that we use the terms IoT and IIoT synonymously in this paper.

The capabilities of such systems provide great opportunities for plant engineers in terms of monitoring and analysis of the production line as well as automated workflows for the staff surrounding it. As such, IoT already needs to be considered during the design phase of a production line. In recent years, numerous commercial and open source IoT platforms have been emerging. For our work, we have been provided with the commercial platform Elements for IoT by CONTACT Software (CONTACT 2019). A key technology for IoT are message exchange protocols such as the publish-subscribe-based Message Queuing Telemetry Transport protocol (MQTT, OASIS 2019) which is used in this work. Originally created for smart home applications, MQTT is easy to integrate into a production context as it is lightweight in comparison to more complex industry standards such as the OPC Unified Architecture (OPC UA, OPC Foundation 2019).

2.3. Digital Twin

A Digital Twin is a digital representation of a real world object. (GI 2017) In the context of production, this could be anything ranging from a sensor, over a machine or a product, up to an entire production line or factory. The Digital Twin is not even tied to the existence of its real world counterpart. In the case of a product or a machine in development, the Digital Twin may be established long before the production of the real world object. It can be used over the entire product lifecycle to track and exchange information. In terms of granularity this information may encompass CAD data, parameters and configurations, meta data, behavioural models, and state information like machine health. The Digital Twin enables users in various roles to view the representation from different perspectives and interact with it without affecting the real world.

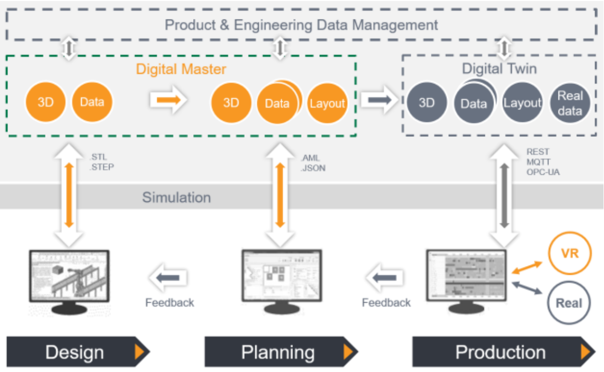

Figure 1: Digital Twin in Product Engineering. Source: Own Illustration

Figure 1 shows how we see the Digital Twin in plant engineering. Already during the early design stage of a machine, a Digital Master is created holding predominantly CAD data, meta data, and parameters. During the planning phase, this is embedded in a layout with other machines leading to the Digital Master of a production line or production centre. When entering production phase, the actual Digital Twin is derived from the Digital Master by deploying the real world objects. The Digital Twin then inherits its Master's information and completes it with information received at runtime, like sensor data or data about the health of the production line. There can be a multitude of Digital Twins derived from a Digital Master. Furthermore, the Digital Twins can be simulated entirely or feed on real world data.

2.4. Manufacturing Execution System

Manufacturing Execution Systems (MES) fill in the gap between Enterprise Resource Planning (ERP) and the shop floor by providing real time planning, monitoring, and control of production processes. According to the VDI (2016), MES provide ten core functions:

1. Detailed Planning and Control
2. Resource Management
3. Material Management
4. Personnel Management
5. Data Collection
6. Performance Analysis
7. Quality Management
8. Information Management
9. Energy Management
10. Order Management

As one can see, there is a slight overlap between MES core functions and the monitoring and analysis aspects of IoT platforms. However, the MES' sophisticated planning and control mechanisms play a key role in being able to realise the Digital Factory. Limits and potentials of these systems must be considered carefully during the process of plant engineering.

2.5. Digital Factory

Originally conceived as aggregation of various planning and simulation approaches for plant engineering (Kuhn 2006), the Digital Factory can be extended by the real time and real world data processing capabilities of the Digital Twin. In our understanding, simulation and planning are only part of the loop that connects engineering and production. The other part is the real world information and knowledge gathered in production being reflected back into the engineering in an iterative development process. Thus, the Digital Factory forms the cornerstone of our efforts.

3. CONCEPT

This section provides a brief introduction into the general ideas behind the Closed Loop Engineering approach by employing a Digital Twin. Figure 2 gives an overview. The illustration is divided into two parts, the production side, i.e. the production line on the shop floor producing goods, and the (re-)engineering side, i.e. the development of new production liens and changes on existing ones.

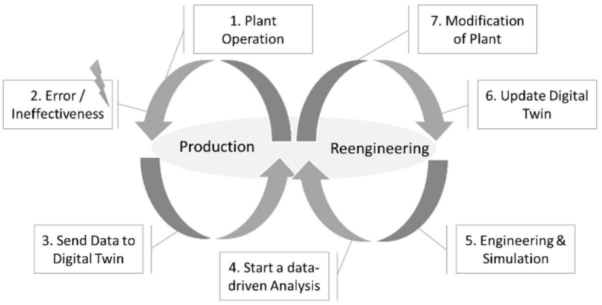

Figure 2: Closed-Loop-Engineering. Source: Own Illustration

Starting on the left side, the Closed Loop process typically assumes there is an existing factory or production line which is already operating (1). Somewhere down the line, an error or inefficiency occurs (2). This could be due to changes in the production process, wear and tear on the machines, or an overlooked mistake during the planning of the production. As data from the real world production line is constantly pushed to its Digital Twin (3), in order to identify the root cause of the problem, analysis can be conducted immediately once an error or inefficiency occurs (4) leading to the right part of the illustration. The results of the analysis are then used to perform re-engineering and simulate the changes, i.e. with methods of virtual commissioning (5). Once a solution has been found and validated through simulation, it is deployed in the Digital Twin (6) and the real world production line (7). In order to realise the concept in form of a demonstrator, we rely on two key approaches, the Product-Process-Resource-Skill model and AutomationML.

3.1. Model of Product-Process-Resource-Skill

To accurately describe our factory model, we employ a Product-Process-Resource-Skill (PPRS) approach. (Pfrommer et al. 2013, Aleksandrov et al. 2014) This allows us to define abstract high-level skills for machines without knowing the implementation details of any specific machine type. For instance, a generic robotic arm could provide the skills transport and assemble. Of course, at some point these high-level skills have to be translated into low-level machine instructions. However, the abstract representation is sufficient to outline machine capabilities and to model the products. A product is understood as defining its own production process in form of a sequence of skills needed to be applied in a certain order on the initial workpiece. At runtime, these required skills (product skills) are mapped onto the skills provided by the machines (resource skills) in the actual layout represented in a graph structure. This allows us to calculate all possible paths for the material flow and derive concrete production sequences in a simple and efficient fashion.

3.2. Exchange Format AutomationML

Both already existing as well as factory layouts still in development are thought to be exchangeable independently of deployed CAD and PLM systems as they come the format of AutomationML (AML, AutomationML 2019). Broadly speaking, AML is an XML-based exchange format for engineering information. (AutomationML 2018) It may contain meta-data and machine parameters but also links to actual CAD files. For simplification purposes, in this paper, AML contains everything required to plan and simulate a simple production process. The four main AML element types are used in our work as follows (note that we do not utilise the AttributeTypeLib yet which was introduced in AutomationML 2.10):

- InterfaceClassLib
  This element stores generic interface types for inter-machine connections.
- RoleClassLib
  This element contains semantic information about the different domain model classes factory, resource (machine), tool, sensor, product, and skill. It also contains descriptions of the various skill types.
- SystemUnitClassLib
  This element defines the generic resource types, i.e. machine, robot, and conveyor. It also defines the product including its skill-based production process using a series of nested nodes.
- InstanceHierarchy
  Finally, this element groups instances of the generic resource types from the SystemUnitClassLib in a defined layout in 3D space. This represents the actual production line on the shop floor.

The AML provides an almost self-contained semantic description of a factory including all its products, processes, and resources. However, we did not include CAD data at this stage of the development. 3D models of the resources and products are directly integrated in Unity.

4. DEMONSTRATOR

As a proof of concept, we developed a VR-demonstrator using the CONTACT Elements for IoT platform (CONTACT 2019) in conjunction with a web-based production process planner and Manufacturing Execution System (MES) written by the FZI Research Center for Information Technology in Java as well as the Unity game engine (Unity Technologies 2019) for real time simulation in VR. This demonstrator is subject of ongoing development and mainly serves as our technological and algorithmic testbed. It also provides an accessible research showcase for both industry and public.

4.1. Use Case

The showcase revolves around a bottling plant in a small enterprise where bottles are filled by a filling machine and then sealed, engraved, and packed for shipping by robot arms with multiple attachable tools at their disposal. As Figure 3 shows, the production line is assembled from one filling machine, two conveyors, up to two robotic arms, one cap stock providing bottle caps, and one box for storing the finished bottles. Parts of this setup were available to us as real world machinery.

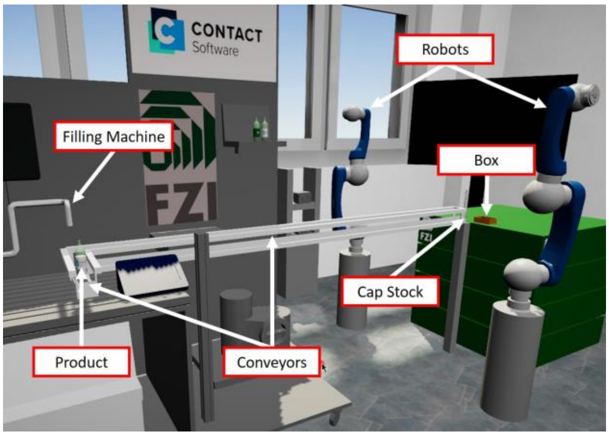

Figure 3: Machines of the Bottling Plant in Unity. Source: Own Illustration

Customer orders are highly customisable and define variable batch sizes, different bottle designs, a variety of fluids, and custom engravings. Thus, production planning and execution complexity rises requiring the MES to adapt to a variety of different circumstances. The subset of the production process illustrated in the demonstrator is thought to run in a fully automated environment. Human interaction in the actual production process is not required in this instance.

4.2. Functions

In order to simulate the use case according to the constraints defined by the concept, i.e. simulating an existing production line encoded in AML, we defined a set of functions the demonstrator has to support:

1. Import Factory Layout
   An existing factory layout encoded in AML is imported and transformed into the system's domain model.
2. Change Factory Layout
   The factory layout is altered by adding, removing, or re-configuring machines.
3. Export Factory Layout
   The factory layout is transformed and exported back into AML.
4. Display Factory Layout
   The factory layout is displayed in 3D and VR. The user can move freely within the scene.
5. Place Order
   An order can be placed containing quantity, desired fluid, and custom engraving.
6. Simulate Production Process
   Based on a placed order, a batch equating the order quantity is scheduled for production. A production plan outlining the specific sequence of machine operations including machine setup is computed and then executed on screen.
7. Simulate Sensors
   Using Unity's physics capabilities, sensor data like weight on the conveyor belts, pull on the robot arms, robot angles, and throughput of the filling machine can be calculated. This data is enriched by information about the internal state of the machines, such as which tools are equipped, at what angles the robot joints are, and what tasks are currently executed.
8. Export Sensor Data
   The simulated sensor data is serialised and pushed to the IoT platform via MQTT in steady intervals.
9. Reset Simulation
   The simulation can be reset and restarted at any given point.

4.3. System Architecture

In order to realise the concept, achieve the necessary functionality, and integrate the IoT platform, an architecture tying together software components from different domains, technologies, and programming languages was required.

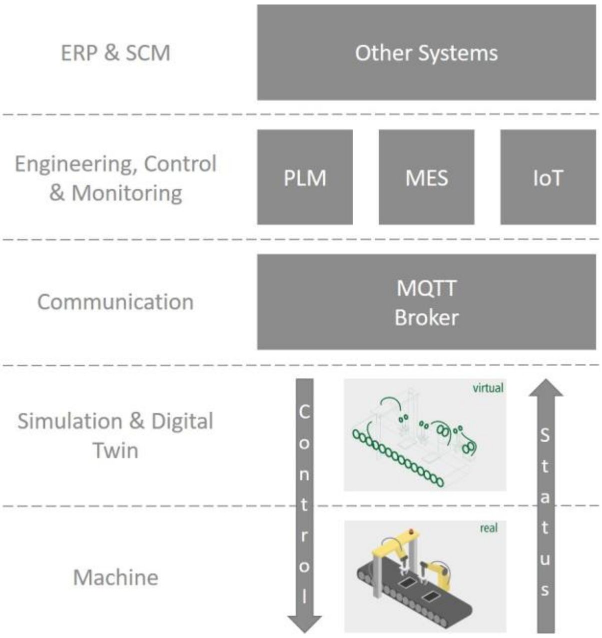

Figure 4: System Architecture & Flow of Information. Source: Own Illustration

As Figure 4 shows, on the very top lie the upstream systems, like Enterprise Resource Planning (ERP) and Supply Chain Management (SCM). These components are not directly integrated in the demonstrator. Instead, data exchange with these systems is mocked using AML files for machine data and JSON strings for order data.

Right below the top level, the actual development and control systems are situated. This encompasses the Product Lifecycle Management (PLM) software for component development, the MES for control of the shop floor, and the IoT for surveillance of the deployed machines. While the PLM system is again not integrated but only mocked through AML files, the other two systems are hooked up to the shop floor via an MQTT Broker. The MQTT Broker serves as central communication hub for all data exchange in the demonstrator across individual system borders. The message format is implemented as JSON strings containing serialised objects.

Downwards of the MQTT Broker lies the shop floor, both virtual and real. The MES is set up to communicate with the Digital Twin which runs in the Unity simulation. This is done by sending instructions from the MES to the Digital Twin which only responds with state reports. Instructions contain the high-level skill to execute, a set of customisable parameters, and a reference to the product it should execute the skill on. State reports contain information about the Digital Twin, whether it is idling, busy, or in error state, which skill it currently executes, its position in 3D space, and a reference to the currently selected tool (if available). These state reports are sent by the Digital Twin when asked for by the MES, an instruction is received or finished, or an error occurs.

The Digital Twin translates the high-level skill references from the instruction into concrete machine code and passes it down to its real world counterpart for execution. In return, the real world machine pushes all its sensor data to the Digital Twin. Hence, the real world machine is capsuled entirely through its Digital Twin. The sensor data as well as data about the internal state of the machine is aggregated by the Digital Twin and sent to the IoT instance periodically. The Digital Twin provides low-level error detection, i.e. when it cannot locate the designated product or execute a given skill, and will report errors to the MES through the state report mechanism. The MES then decides whether it can recover from the error or has to halt production entirely.

4.4. Interaction in 3D and Virtual Reality

The demonstrator can be interacted with through both regular 3D on a monitor plus mouse and keyboard setup as well as in VR using a headset and controller. Users can move freely within the scene and examine all machines in detail. Furthermore, they are able to place orders in-game to start the simulation and stop or reset it at will. Finally, users can deploy new machines from a catalogue and thus try out different factory configurations. At this stage, machines can only be placed at pre-defined locations, however a dynamic placement system is subject to ongoing development.

4.5. An Exemplary Rundown

At this point, an ideal step-by-step rundown of the use-case in the demonstrator is given:

1. In a first step, the PLM system transforms the real world factory and product contained in AML into our domain model.
2. The PLM system then sends the factory over MQTT to our MES and the Unity simulation.
3. The MES calculates a production process graph based on layout information and the product description.
4. Unity renders the 3D scene based on the factory information it received as seen in Figure 5. The Digital Twins connect via MQTT to the MES and to their real world counterparts.

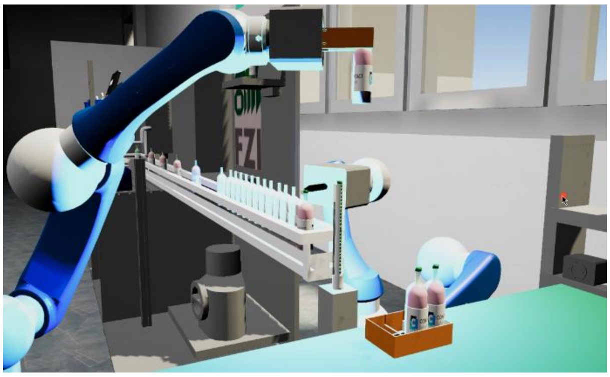

Figure 5: VR-Simulation of the Production Process in Unity. Source: Own Illustration

5. The user places an order either from within Unity or via a webpage provided by the MES.
6. The MES calculates a production plan and starts executing that plan by issuing instructions over MQTT to the Digital Twins.
7. The Digital Twins execute the instructions in the simulation while at the same time translating them to machine code which they send down to their real world counterparts.
8. The real world counterparts execute the orders received from their Digital Twins and send back sensor and machine data in return.
9. The Digital Twins aggregate the incoming sensor and machine data and enrich it with additional information about their own state. This data is periodically pushed to the IoT instance.
10. An error occurs either in the real world or the simulation and is communicated to the MES by the Digital Twins. At the same time, the IoT instance detects inefficiencies based on the incoming sensor data and kicks off a workflow.
11. The MES halts production for the user to review the problem both in the simulation and in the IoT instance as illustrated in Figure 6.

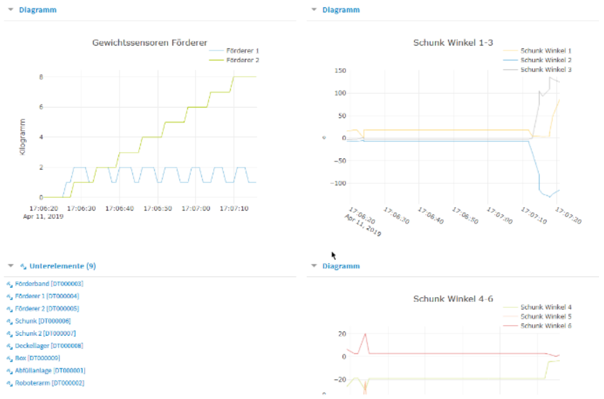

Figure 6: Presentation of Sensor Data in the CONTACT Elements for IoT Platform. Source: Own Illustration

12. The user decides to reconfigure the plant in VR, i.e. by adding an additional robot arm.
13. The user restarts the simulation, this time without connecting to the real world machines and validates the changes made to the layout.
14. The user deploys the additional robot arm in the real world plant and the production starts again.
15. The layout changes are exported back into AML.

The power of the system lies in its flexibility. It can be run in simulation only, or be connected to either the complete or just a subset of the real world machinery. Users can interact with the system via VR, regular 3D, or through a web interface.

5. EVALUATION

Both the demonstrator and the concept work it is based on are still subject to further research and development, hence a final evaluation cannot be given at this point. However, the demonstrator has been exposed to figures from industry and public on several occasions, like visits, trade fairs, and public events. The receptions were positive overall for its ability to demonstrate complex production processes in an immersive format. However, fair criticism was levelled at its physical inaccuracy in contrast to other professional simulation tools which are more focused on accurate physical representations over the process simulation we are interested in. On a positive side, a lightweight version of the demonstrator is used by our partners at CONTACT for training and evaluation purposes.

6. CONCLUSION

In this paper, we presented a novel approach on how to establish a Closed Loop Engineering process chain for the Digital Factory. We did this by combining technologies from the fields of Production and Process Automation, Virtual Reality, and Internet of Things. The approach connects the engineering and simulation aspects of the Digital Factory with the shop floor by exploiting the concept of the Digital Twin. As a proof of concept, we implemented a research demonstrator comprised of multiple software systems integrated through the MQTT protocol. The demonstrator is still subject to ongoing developments and evaluation efforts. However, we are confident our work provides a first step towards the implementation of a widely applicable Closed Loop Engineering approach for plant engineering in the context of the Digital Factory.

7. FUTURE WORK

For the future, we plan to expand the demonstrator hardware by the implementation of a more sophisticated real world miniature production line with support for a wide array of potential production scenarios. This is to be tightly coupled with the virtual environment and serves as a testbed for plug & produce approaches in particular. On the software side, we are mainly interested in driving our model of the Digital Factory more towards the Smart Factory, and use the demonstrator as algorithmic testbed for AI-supported production planning and machine parameter optimisation. Finally, we consider integrating more industry standards, like OPC UA for example, into the showcase.

ACKNOWLEDGMENTS

We would like to thank our partner CONTACT Software GmbH for providing us with both the Elements for IoT platform as well as their in-depth domain knowledge.

REFERENCES

Aleksandrov K., Schubert V., Ovtcharova J., 2014. Skill-Based Asset Management: A PLM-Approach for Reconfigurable Production Systems. In: Fukuda S., Bernard A., Gurumoorthy B., Bouras A., eds. Product Lifecycle Management for a Global Market. PLM 2014. IFIP Advances in Information and Communication Technology, vol 442. Berlin, Heidelberg: Springer.

AutomationML consortium, 2018. Whitepaper AutomationML Edition 2.1. Available from: www.automationml.org/o.red.c/dateien.html [accessed 09 July 2019]

AutomationML e.V., 2019. Homepage. Available from: www.automationml.org [accessed 08 July 2019]

Boyes H., Hallaq B., Cunningham J., Watson T., 2018. The industrial internet of things (IIoT): An analysis framework. Computers in Industry 101: 1-12.

Choi S., Jung K., Noh S.D., 2015. Virtual reality applications in manufacturing industries: Past research, present findings, and future directions. Concurrent Engineering 23(1): 40-63.

CONTACT Software GmbH, 2019. Homepage. Available from: www.contact-software.com/de/produkte/iot-plattform-fuer-digitale-geschaeftsmodelle/ [accessed 08 July 2019]

Gesellschaft für Informatik e.V. (GI), Kuhn T., 2017. Lexikon: Digitaler Zwilling. Available from: www.gi.de/informatiklexikon/digitaler-zwilling/ [accessed 08 July 2019]

Kuhn W., 2006. Digital Factory - Simulation Enhancing the Product and Production Engineering Process.

Proceedings of the 2006 Winter Simulation Conference, 1899-1906. 03.12.2006 - 06.12.2006, Monterey, CA, USA.

in2sight GmbH, 2019. Homepage. Available from: www.game4automation.com [accessed 08 July 2019]

IPO.Plan GmbH, 2019. Homepage. Available from: www.ipoplan.de/software-ipolog/ [accessed 08 July 2019]

Menck N., Yang X., Weidig C., Winkes P., Lauer C., Hagen H., Hamann B., Aurich J.C., 2012. Collaborative Factory Planning in Virtual Reality. Procedia CIRP, Volume 3, 317–322. Athens, Greece.

OPC Foundation, 2019. Homepage. Available from: www.opcfoundation.org [accessed 08 July 2019]

Open Source Robotics Foundation, Inc., 2019. Homepage. Available from: www.ros.org [accessed 08 July 2019]

Organization for the Advancement of Structured Information Standards (OASIS), 2019. MQTT Homepage. Available from: www.mqtt.org [accessed 08 July 2019]

Pfrommer J., Schleipen M., Beyerer J., 2013. PPRS: Production skills and their relation to product, process, and resource. IEEE 18th International Conference on Emerging Technologies and Factory Automation (ETFA), 1-4. 10.09.2013 - 13.09.2013, Cagliari, Italy.

R3DT GmbH, 2019. Homepage. Available from: www.r3dt.com [accessed 08 July 2019]

SALT AND PEPPER Holding GmbH & Co. KG, 2019. Homepage. Available from: www.salt-and-pepper.eu/en/virtualreality/ [accessed 08 July 2019]

tarakos GmbH, 2019. Homepage. Available from: www.tarakos.de/taravrbuilder [accessed 08 July 2019]

Tseng M. M., Wang Y., Jiao R. J., 2017. Mass Customization. In: Laperrière L., Reinhart G., eds. CIRP Encyclopedia of Production Engineering. Berlin, Heidelberg: Springer, 1-8.

Unity Technologies, 2019. Homepage. Available from: www.unity.com [accessed 08 July 2019]

VDI, 2016. Fertigungsmanagementsysteme (Manufacturing Execution Systems - MES). In: VDI 5600 Blatt 1:2016-10. VDI-Gesellschaft Produkt- und Prozessgestaltung.