



Multi-factory production planning using edge computing and IIoT platforms

Deepak Dhungana^{a,*}, Alois Haselböck^b, Sebastian Meixner^b, Daniel Schall^b,
Johannes Schmid^c, Stefan Trabesinger^c, Stefan Wallner^b

^a IMC University of Applied Sciences, Krems, Austria

^b Siemens AG Österreich, Vienna, Austria

^c Technische Universität Graz, Graz, Austria

ARTICLE INFO

Article history:

Received 10 January 2021

Received in revised form 6 July 2021

Accepted 2 September 2021

Available online 16 September 2021

Keywords:

Factory as a service

Industrie 4.0

Manufacturing ecosystems

Cloud manufacturing

Monitoring production

Edge analytics

ABSTRACT

An important prerequisite for determining whether a certain product is producible in any given production facility is an accurate assessment of which production lines and/or the machines are able to execute the necessary production steps. Not only the static information about the capabilities of the machines, but also the conditions of machines and tools are significant for this analysis. Because of the deviation of machine capabilities with increasing deterioration and wear of the equipment, it is also necessary to continuously monitor the status of the machine and analyze the machine conditions. In this paper, we present an approach for generating production plans across multiple factories, considering both static information and dynamic data analysis. Edge devices constantly monitor high frequency machine data and report condensed machine states to an Industrial IoT platform (IIoT). A marketplace within the cloud-application MindSphere enables us to integrate the requirements of the products and the capabilities of the production sites. Customers are able to evaluate these production plans based on duration, energy consumption, CO₂ footprint etc.

© 2021 The Authors. Published by Elsevier Inc. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

1. Introduction and motivation

Research on individualized production processes has skyrocketed in recent years with the increasing demand for individually tailored products. This has led to novel methods that can deal with the flexibility of a production plant and associated intelligent processes. Many initiatives in this area aim to revolutionize the future of industrial production, e.g., the Smart Manufacturing Leadership Coalition (SMLC),¹ the Industrial Internet Consortium,² Industrie 4.0,³ etc.

Traditionally, many product sellers have relied on manufacturing firms (MF) to produce components and products. The product designer typically approaches the MF with a design, and the MF offers to manufacture the part(s) based on internal processes, labor, tooling, and material cost etc. Before a contract is signed, the

factory must perform the so-called *Producibility Check* (Dhungana et al., 2017, 2018) (also called *Manufacturability Check* (de Silva et al., 2016, 2017)), which determines whether a product defined in the form of a bill of material (BOM) and a bill of process (BOP) can be produced in a given factory. If the production is technically feasible, the *Producibility Check* (cf. Producibility Checker in Fig. 1) returns one or more production plans, i.e. workflows of how to execute each process (i.e., production step) on a suitable machine. If there is more than one such production plan, optimization can find the quickest and/or the cheapest one.

Dealing with these issues requires a shift from a traditional “machine park operation” business to a “production as a service” model, also called “manufacturing as a service” in Lu et al. (2014). In this model, the production process, and even the companies involved in the production process, are transparent to the product seller. Because of the required flexibility, the infrastructure of one company alone often is no longer sufficient to fulfill customers’ needs. The goal is to provide an innovative customer experience, where – from the viewpoint of the customer – the production facility is always-on, always-available, and limitless in terms of production capacity.

To achieve this shift, innovation in both product and production design is needed. As described in one of our earlier papers, an alternative approach to design considers products not as sets

* Corresponding author.

E-mail addresses: deepak.dhungana@fh-krems.ac.at (D. Dhungana), alois.haselboeck@siemens.com (A. Haselböck), sebastian.a.meixner@siemens.com (S. Meixner), daniel.schall@siemens.com (D. Schall), johannes.schmid@tugraz.at (J. Schmid), stefan.trabesinger@tugraz.at (S. Trabesinger), stefan.wallner@siemens.com (S. Wallner).

¹ <https://smartmanufacturingleadershipcoalition.org/>.

² <https://www.iiconsortium.org/>.

³ <https://www.plattform-i40.de/>.

of features, but rather as sets of production operations and production constraints in a structured and formal model (Dhungana et al., 2017). Formally describing such a “digital twin” of the product enables algorithmic evaluation of whether a product can be manufactured at a given site.

These ideas have been discussed in our recent publications (Dhungana et al., 2017, 2020, 2015). This paper extends these ideas further and considers not only the static information about the capabilities of the factories but also the real-time data from the machines in the factory. The main contribution of this paper is the integration of operational data in the generation of production schedules and determining the producibility based on runtime-data of the involved machines and factories (cf. Production Scheduler in Fig. 1).

In particular, this paper addresses the following issues, which are eminent in individualized production processes:

- *Constrained production requirements:* Product designers and production planners need to consider the production requirements, including the tolerances and margins of each step before a production line and a schedule can be setup. Only after a detailed evaluation of the production constraints and the capabilities of the machines, it is possible to initiate the production steps. However the tolerances and error margins of the machines can change with time because of changing machine conditions.
- *Continuous monitoring infrastructure:* Production processes need to be continuously monitored, in order to ensure the product quality requirements. This demands an infrastructure for continuously collecting the data about the state of the machines, product parts, and the outputs of the involved manufacturing activities. Typically, data must be collected across multiple machines and factories to detect quality issues, which requires a networked data collection infrastructure. The approach shown here is suitable for both high and low precision manufacturing. For both types of accuracy, it helps to know what accuracies are available on the machines of the respective manufacturer. Through different selection options and thus degrees of accuracy of manufacturing machines, the respective required accuracy can be obtained at optimal cost.
- *Deviation of equipment properties:* Unavailability (e.g., because of deterioration) of equipment, tools, and other resources may also occur temporarily, because of short-term breakdowns, overload, or other operational issues. Manufacturing difficulties may mean that alternative production schedules (possibly even with external partners) need to be coordinated. The customer may have to bear long waiting times due to the unavailability of the equipment, tools, or resources (e.g., specialists), as small production orders place higher demands on the equipment coordination.

The rest of the paper is structured as follows: Section 2 provides an overview of the approach, giving the readers some background information about the tools and infrastructural elements used in the approach. In Section 3 presents a running example in the form of a use case to demonstrate our approach. The solution architecture for our approach is presented in 4. With details about the proposed dynamic production planning in Section 5, we describe our pilot system implementation in Section 6. Related work is presented in Section 7, before we summarize the paper in Section 8.

2. Approach overview

Our approach is based on a common platform that acts as a “virtual meeting room”, enabling interactions between stakeholders (factory equipment vendors, factory operators, product

sellers, and end-customers) in a production ecosystem (Dhungana et al., 2015). The platform allows users to publish and share artifacts such as an ontology describing the capabilities of factories and models of products and their manufacturing processes. After the customers create new production orders, factories can evaluate whether they have equipment and technical capability to manufacture the required variant of the product.

As part of our ongoing research in this area, we have developed such an online marketplace (Dhungana et al., 2018) and are using it for experimenting with different ways of modeling products and factories, and improving core algorithms for producibility checks, scheduling and re-scheduling production orders. The marketplace operator defines the vocabulary to be used by product and equipment sellers as a common language for factory and product specification. An ontology based on standards such as the ANSI/ISA-95 (Standard ANSI/ISA 95.03-2005, 2007) and IEC 62264 standards for enterprise-control system integration represents the common vocabulary and facilitates a broad acceptance of such marketplaces. Fig. 1 illustrates this approach and is explained in the following three sections.

2.1. Knowledge infrastructure

Knowledge about products, production equipment, and processes is shared in the form of models — which act as the knowledge backbone of our approach (Dhungana et al., 2018). In particular, the *Skill Ontology* contains declarations of all potential capabilities of the production equipment. The *Equipment Models* refer to the formal descriptions of the factory components that can provide production capabilities in a factory. Each *Equipment* provides a set of production and/or transportation skills. The equipment model is therefore another model layer sub-classing the skill ontology.

In our pilot system implementations (see Section 6), we evaluated both a conventional, relational database and an RDF/OWL graph database (a “knowledge graph”) to store factory and product models. Both approaches have proven their worth. Relational databases are a high-performance, well-established technology with strict table schemata on which the reasoning processes can rely. Graph databases are a bit more flexible in adding all kinds of supplementary data. We used JSON file formats for data disclosure and exchange. The JSON-LD variant of a JSON file format combines particularly well with a graph database.

2.2. Monitoring infrastructure

The marketplace is deployed in an Industrial IoT platform, which enables connectivity between the factories and the cloud platform for managing application-specific data. The Edge Devices deployed in the factories receive high frequency data from the machines, and a particular app pre-processes it locally, by, for example, computing certain machine parameters from the raw data. Once the pre-processing step is done and higher-level information is derived from the raw data, the information can be made available to other applications running on the Edge. Alternatively, the computed data can be transferred to the IIoT platform in the cloud where it is archived and the users have the possibility to view and use this information at a later point in time.

2.3. Marketplace services

Based on the knowledge shared through the platform, stakeholders are provided with various services (Dhungana et al., 2018, 2015):

Producibility Queries are services provided by the marketplace to factory operators and product sellers, enabling them to answer

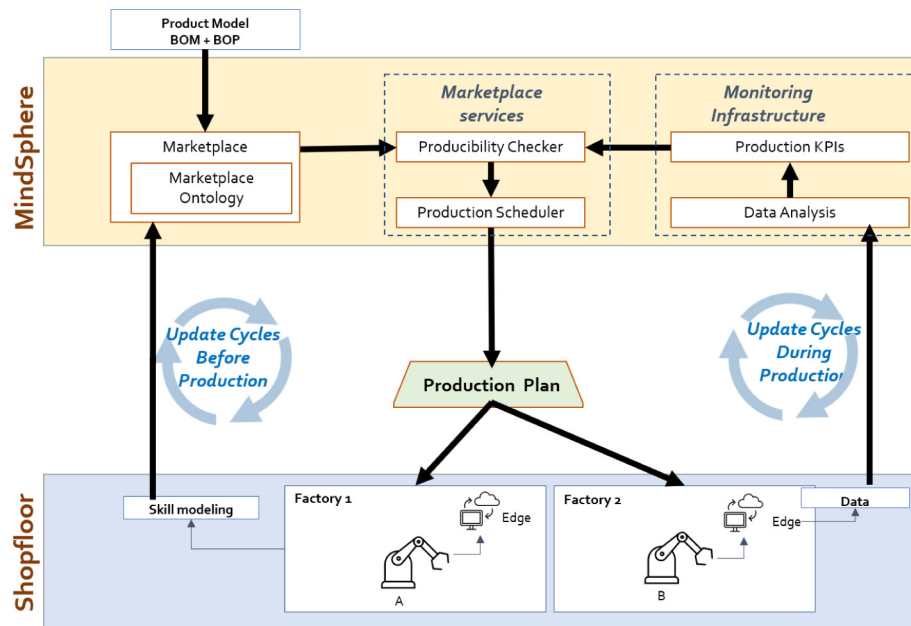


Fig. 1. Overview of the approach.

two basic queries: (i) Given a configured product (order model), find all the factories that can manufacture the product. (ii) Given a factory, find all products that can be manufactured in that factory. One of the key services required for producibility checks is the *Skill Matching Service* (also known as Capability Matching (Järvenpää et al., 2017; Bildstein et al., 2018)), which can be used to semantically identify matches and gaps between services provided by the factories/equipment and capabilities required for manufacturing a product.

Reconfiguration Recommendations can be used both by product designers and by factory operators to learn about potential changes to improve their products and services by obtaining answers to two basic queries: (i) Given a product model represented as BOM and BOP, and a factory where it should be produced: If the factory is currently not able to produce the product, what are the *recommended changes to the product design* to enable the factory to produce it? (ii) Given a factory model and a product model which it should produce: If the factory is currently not able to produce the product, what are the *recommended changes to the factory* to enable it to produce the product?

3. Use case

Smart automation aims for digitalization of all aspects of products, production, and fast order – and manufacturing cycles especially for goods with small lot sizes. An important prerequisite for this is the accurate assessment, which production lines and the related machines are able to execute the necessary production steps with the current equipment. Edge devices constantly monitor high frequency machine data and report condensed machine states to an Industrial IoT platform (IIoT). The implementation of this novel architecture ensures an accurate and timely production planning. The following use case describes the demonstration of this scenario using PROFACTO⁴ as marketplace within the cloud-application MindSphere.

In our scenario, we use the configuration and production of a wave gear product as sketched in Fig. 8. After finishing the

product development, the features that characterize a wave gear product are transferred to PROFACTO. Thereafter, the customer has the chance to select the product according to his/her requirements. These requirements of the wave gear are the required load, the resolution speed, the gear ratio, and the connecting dimensions. Furthermore, BOM and BOP structures are transferred in advance from the product life cycle management system (PLM) to PROFACTO. The BOM contains all components required for the production of the wave gear, whereas the BOP contains a detailed description of all production steps necessary to manufacture the new product. At the end of the product configuration process, PROFACTO knows the BOM and the BOP of the new product.

The most important part of a BOP operation's specification is the description of the required machine skills necessary to execute the corresponding production process. The evaluation whether a machine can execute a given BOP operation is called *skill matching*. Skill matching consists of two parts: (i) matching of skill type and (ii) evaluation of skill constraints. Both tasks rely on a common vocabulary used for describing required production operation skills and skills offered by a machine. For example, let us focus on the reaming process for a wave generator as shown in Fig. 8. Only machines that offer a *reaming* skill pass the first part of the skill matching test. In the second part, all required skill constraints must be satisfied for a given machine. These are, for instance: the machine must be able to work on the material type of the work-piece; the machine must be able to ream holes of a given width and depth; the machine must ream the hole within a given precision tolerance. Only when all these constraints are satisfied, the machine is said to be matching.

Skill parameters of machines are typically key performance indicators (KPI) and they are defined in the manufacturer specification of the machine. However, the actual state of the machine may deviate from that manufacturer specification. The achievable precision depends on several conditions, like how long the machine is running without maintenance or the workload of the machine over time. It could be the case that the current precision may be degraded below the required tolerance value, due to the previously mentioned factors. A machine assignment in a production plan, that does not take the current machine state into account, risks the usage of an unsuitable machine or the necessity of a re-planning of the production right before its start.

⁴ PROFACTO is a marketplace for products and production services. It is currently in an experimental state and not available outside of Siemens.

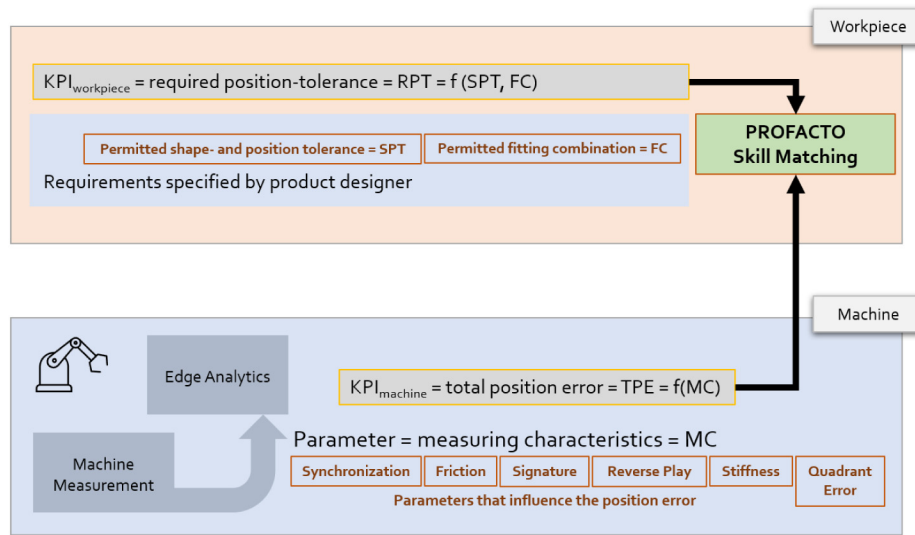


Fig. 2. Skill matching example.

Wrong tolerance values lead to an inadequate quality of the final work-piece and therefore a late and expensive intervention in the production process is required.

For this reason, we should aim for evaluating skill constraints against KPI values that are as up-to-date as possible (e.g., updated on a daily basis). In our example of the reaming process to drill a precision bore into the wave generator work-piece, one of the skill constraints claims that the precision of a reaming process has to be in permitted shape and position tolerances (SPT) and fitting combinations (FC) of the work-piece. As depicted in Fig. 2, the work-piece-parameters SPT and FC are combined to a KPI called *required position tolerance* (RPT). Based on the KPI RPT, a machine is required that can guarantee to ream a hole with equal or higher precision than required by the given RPT value.

On the milling machine side, several basic machine parameters determine the KPI *total position error* (TPE). These parameters can be obtained by the measuring characteristics (MC) Synchronization, Friction, Signature, Reverse Play, Stiffness and Quadrant Error. Within this use case, a machine has to be assigned to the required position tolerances of the precision bore. The assigned machine must guarantee that it can manufacture the precision bore of the component within its total position error.

To do so, the precision of a machine is frequently assessed and made available as KPI TPE to the skill matching routine. On a daily basis, a test routine transfers sensor data from a machine to a connected edge device. These time series are processed, condensed, and compared to previous results. The KPI TPE is computed and reported to the IIoT platform MindSphere, where the corresponding digital asset is updated.

Fig. 3 shows the skill matching part of the scenario: the skill constraint "machine.TPE $\leq 15 \mu\text{m}$ " checks the tolerance value of each machine and rules out those machines that cannot guarantee the required precision. The PROFACTO marketplace has access to the specifications of all registered plants run by different, often competitive, owners. Each production site has its own tenant in the IIoT platform and provides the necessary KPI values of its machines without disclosing any detailed or sensitive production or machine data.

An extension to single-factory production planning was described in Dhungana et al. (2020). Whenever no single factory can be found to manufacture a given product, a minimal set of factories is computed that cooperatively can do the job. Access to up-to-date machine KPI values complements this scenario by identifying which machines are currently ready to produce the

desired products in the claimed quality. To ensure that the total position error of the milling machine required at the time of planning still has the necessary precision, the KPI TPE is queried again before the order is placed.

4. Architecture

To allow the assessment of the dynamically changing quality parameters a machine is capable of achieving and to provide PROFACTO's production planner with the possibility to make use of these KPIs, we need to employ a combination of cloud and edge computing, exploiting their respective strengths.

Generally speaking, the process of transforming raw production data into usable quality related KPIs can be split up into several distinct steps. First, the MC collection, during which high frequency machine data is obtained directly from the machines' Numerical Control Unit (NCU), pre-processed locally, and then transferred into the IIoT platform in the cloud. There, the computed MC are stored and used during the production planning to derive certain quality related KPIs for the given production steps. By aggregating the data locally and pre-computing certain MCs on the edge, the load on the network is reduced and the required bandwidth is minimized (Shi et al., 2016). Furthermore, a reduction of the storage costs in the cloud can be achieved in this way. Fig. 4 gives an overview of the entire system with all its components.

Shopfloor. On this layer of the system reside the physical machines that are located in the machine operator's plant. These can, for example, be milling or turning machines, which can be used to execute certain production steps. Which concrete production steps they are able to execute are described by the machine operator using the skill ontology introduced in Section 2. Automated manufacturing equipment is generally equipped with an NCU which allows the executing of NC programs and which controls the tools that are chucked into the tool holder. These NCUs have access to a multitude of control data, such as the spindle's position, the torque that is exerted by the individual axes, or the current that is drawn by them.

Connectivity. To receive the aforementioned data, such as position, torque, or current, there are industry standards, which provide uniform access to this data. One such widely used standard

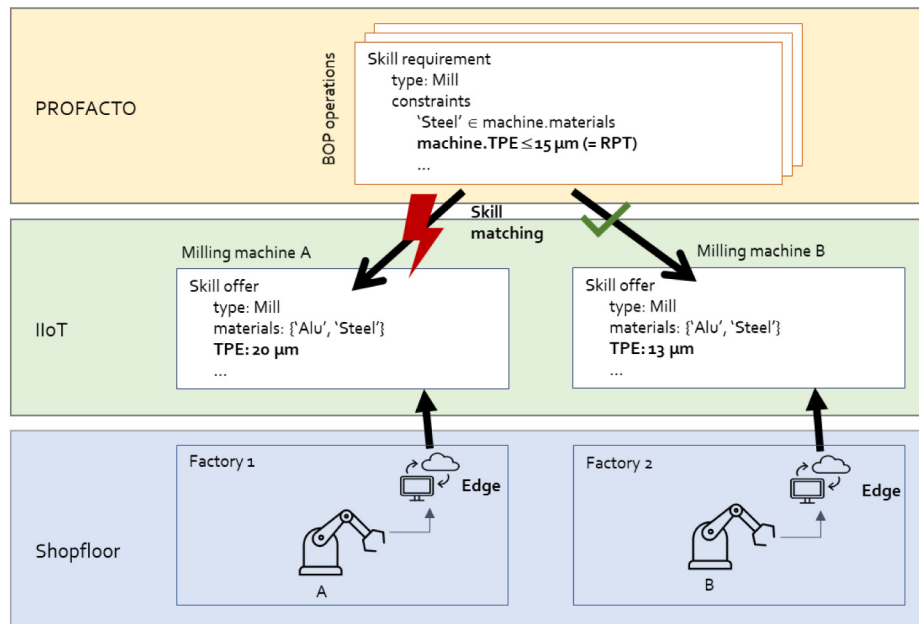


Fig. 3. Skill matching example.

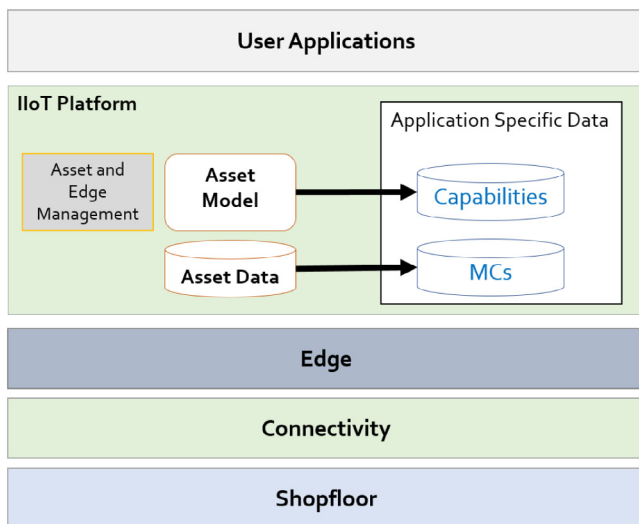


Fig. 4. System overview.

is OPC UA,⁵ which offers a generic and datapoint-centric interface to access machine data. Although widely used and standardized, the OPC UA protocol has a severe limitation in the form of its maximum update frequency, which does not offer high enough sampling rates, while reaching the needed bandwidth (Cavalieri and Chiacchio, 2013). There are many applications for which such an update frequency is adequate, e.g., receiving status updates about the production process. Nevertheless, if one wants to gain deep insights into the production process for more involved tasks such as predictive maintenance, or the quality assessment of a machine, this update frequency no longer suffices. For example, the stiffness of an axis cannot be easily determined directly without additional tools. However, small deterioration in the stiffness can already lead to lower quality pieces. Thus, to track even comparatively tiny changes requires high frequency data.

Therefore, a custom protocol is needed that allows much smaller update intervals (Trabesinger et al., 2020). However, since the NCU's primary task is to control the machine, it can be used to collect and transfer this data, but not to process and analyze it.

Edge. The first station of the raw data once it is transferred from the NCU is an Edge Device. These devices are relatively low-powered, when compared to a full-blown server infrastructure, but are still high-powered appliances in the context of a shopfloor device (Trabesinger et al., 2020). They usually have several gigabytes of RAM, a multi-core processor, and at least two network adaptors to properly keep the operational technology (OT) and the IT network separated. The management of the Edge Devices is usually done via the IIoT platform, which enables users to easily manage the devices remotely by installing or updating applications, as well as upgrading the device's firmware. A prerequisite for such a management is an Edge Computing platform connecting the device to the IIoT platform, offering the appropriate communication mechanism to interact with the Edge Device remotely, as well as providing an application runtime to enable users to develop their own edge applications. As indicated before, the Edge Device receives the high frequency data from the NCU, and a particular app pre-processes it locally, by, for example, computing certain MCs from the raw data. Once the pre-processing step is done and higher-level information is derived from the raw data, the information can be made available to other applications running on the Edge. Alternatively, the computed data can be transferred to the IIoT platform in the cloud where it is archived and the users have the possibility to view and use this information at a later point in time.

User applications and IIoT platform. There are two main stakeholders in the proposed system. First, a factory operator, who manages her machines and their skills. Second, the product designer, who wants his product manufactured in an optimized fashion. The operator interacts with *User Applications*, which expose the platform's basic functionality. Most importantly, this includes the management of edge devices, the creation of an asset model, and the archiving of asset data. The edge management allows users to administrate their edge devices by, for example, checking their status, remotely installing and configuring applications, or updating a device's firmware. It also enables them

⁵ <https://opcfoundation.org/about/opc-technologies/opc-ua/>.

to link data that are received from the edge devices to the respective assets, that are managed within the IIoT platform. Apart from the platform's basic capabilities, on which user applications can be based, the IIoT platform also offers the possibility to manage application-specific data, whereby developers are able to extend the platform's functionality by building upon the basic capabilities, the generic, asset-related data, as well as the application-specific data. This data can be in the form of factory capabilities which are used by the factory operator to describe what her machines can do. Information about the skills of the machines are then used by the *User Applications*, with which the product designer interacts to create a production plan based on the designer's BOM and BOP.

4.1. Machine-to-cloud data transfer

There are two distinct challenges for the transfer of the pre-processed high-frequency data to the cloud. First, a proper connectivity needs to be established that separates IT and OT networks. Secondly, the collected high-frequency data needs to be handled at the edge and only higher-level MCs should be transferred to the cloud.

4.1.1. Connectivity overview

Fig. 5 illustrates the connectivity which is needed for the presented edge computing use case. Similar applications are realized using this connectivity approach, as presented in [Trabesinger et al. \(2020\)](#). A *Network Switch* inside the machine's cabinet is used to separate the *Machine LAN* and the *Factory LAN* (the OT and the IT network, respectively). The *NCU* is connected to the *Network Switch* and can communicate with the *Edge Device* over the *Machine LAN*. This communication channel allows the transmission of the high-frequency data to the *Edge Device*, as well as for the *Edge Device* to set user-defined variables on the *NCU*. As a safety precaution, individual apps have to be granted read and write access to these variables explicitly. Also, the user-defined variables that do not directly influence the machine, therefore these variables have to be handled in the NC program or on the *NCU* directly. An additional connection to the *NCU* is established from the *HMI Panel*, via the *Network Switch*, which allows machine operators to interact with the *NCU*. Thereby, they can, for example, start and stop NC programs. Furthermore, since the *HMI Panel* and the *Edge Device* are both located in the *Machine LAN*, it is possible to provide access from the *HMI Panel* to the *Edge Device*. Therefore, each *Edge Application* can optionally provide a web-based user interface for the operator to interact with the application, e.g., to configure the *Edge Application* or to trigger some action. Both the *HMI Panel* and the *Edge Device* are also connected to the *Factory LAN*, which enables them to reach other infrastructure inside the factory over the network, as well as establishing a connection to the public internet. The latter point is crucial for the *Edge Device* to allow the users to manage their devices, comfortably, from a cloud-based user interface. However, the connection to the *Factory LAN* is secured by the *Factory LAN Firewall* which needs to be strictly configured to protect the machine and its components from unauthorized access.

4.1.2. Edge system architecture

Fig. 6 depicts the edge architecture of the overall system. High frequency machine data can be obtained by installing an appropriate *Compile Cycle* onto the *NCU*. This *Compile Cycle* offers the possibility to extract relevant control data such as the current and voltage drawn, or the torque created by a specific machine axis. An *Edge Device* is connected to the *NCU* via the standard network connection, and the data is received by an appropriate

edge application (i.e., the *Sinumerik Adapter*). The data is obtained in 2 millisecond intervals by the *Compile Cycle*, and forwarded to the *Edge Device* in bulk, in 100 millisecond intervals via a proprietary protocol, which allows the transmission of these large amounts of data in an efficient manner. From the *Sinumerik Adapter*, whose only task it is to receive the high frequency data and make it available to other applications, the data is forwarded to the *Analyze MyMachine/Condition (AMM/C)* edge app. There, the low-level and high frequency data is transformed into machine-specific MCs, which can be used to represent the status of the machine itself. However, they do not allow to directly draw conclusions about the machine's quality-related capabilities.

Examples for such MCs are the stiffness or the friction distribution of an axis, or the quadrant error of the machine. Once the desired MCs are computed, they are transferred to the IIoT platform in the cloud (i.e., *MindSphere*) for further use via a dedicated *Cloud Connector*.

As Fig. 6 illustrates, the communication within the *Edge Device* is realized via a data-bus onto which applications can publish data. Other interested applications can then subscribe to this data. Thereby, the individual edge applications can be easily decoupled from each other, since they do not care which applications are subscribed to the data they publish or which application publishes the data they are interested in.

To allow the computation of the above-mentioned MCs (such as an axis' stiffness), for each of them a specific NC program is automatically created by the *AMM/C* application. When the program is started it sets a specific user-defined machine-variable which acts as a trigger for the *AMM/C* application, instructing it to start recording the high-frequency data. The rest of the program is specifically designed to generate data that allows to compute the above-mentioned MCs. Before the NC program ends, it clears the variable it set before, thereby signaling the *AMM/C* application that it can stop recording and compute the MCs from the recorded data.

4.2. Cloud application

Fig. 7 depicts the cloud architecture of the overall system. It shows the individual parts of the *PROFACTO* application, which is hosted in the cloud. The application allows product designers to submit the BOM and BOP of their respective product in order to obtain an optimal production plan, and which can be broken down into five components.

Via the web-based *User Interface* product designers can configure their products and upload a description of the production process as a BOM and BOP. This description is passed to the *Production Planner* which has the task of actually planning the production process and optimize it with regards to a certain target value (e.g., CO₂ produced during the process). The *Production Planner* uses the *Factories Capabilities*, which are stored in a dedicated database and provided by the factory operator, in conjunction with certain quality metrics, to determine the optimal production process, which is described in more detail in Section 5. Based on a certain machine's MCs its quality related KPIs can be derived. Since this process might involve complex computations and will likely be refined over time, the actual computation of the quality metrics is done by the *Quality Service*, which offers the appropriate APIs and accesses the database containing the *Machine MCs* as they are received from the edge devices. The *Quality Service* might, for example, use the stiffness of an axis in combination with the quadrant error of the machine to determine the maximum amount of precision that a certain machine can possibly achieve. Based on the outcome of this computation a machine can be included or excluded from the set of possible machines for a certain production step specified in the BOP.

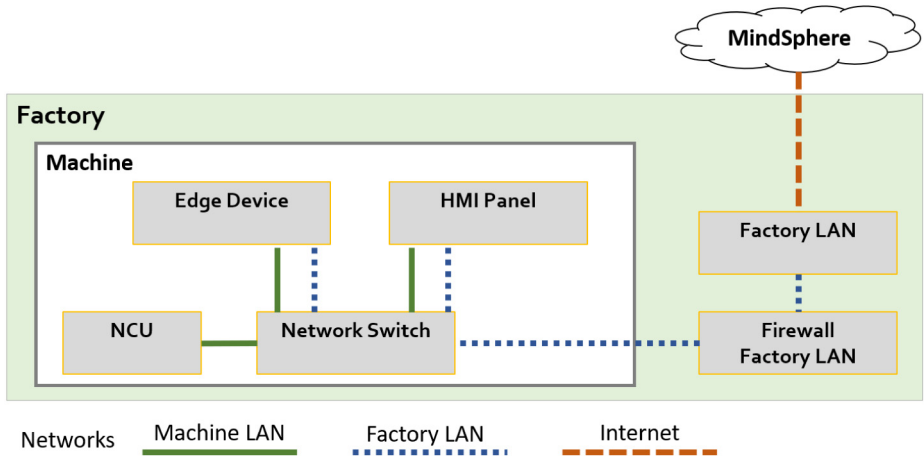


Fig. 5. Edge networks.
Source: Adapted from [Trabesinger et al. \(2020\)](#).

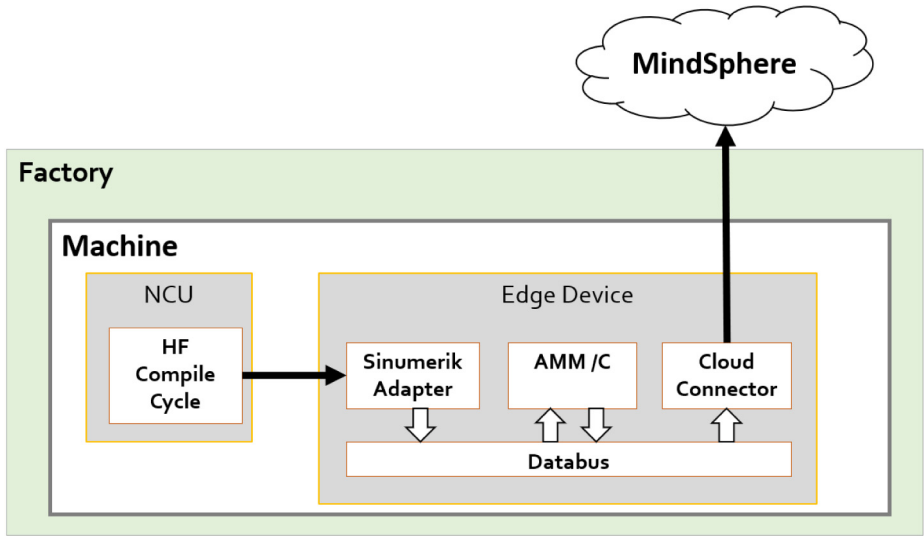


Fig. 6. Edge architecture.

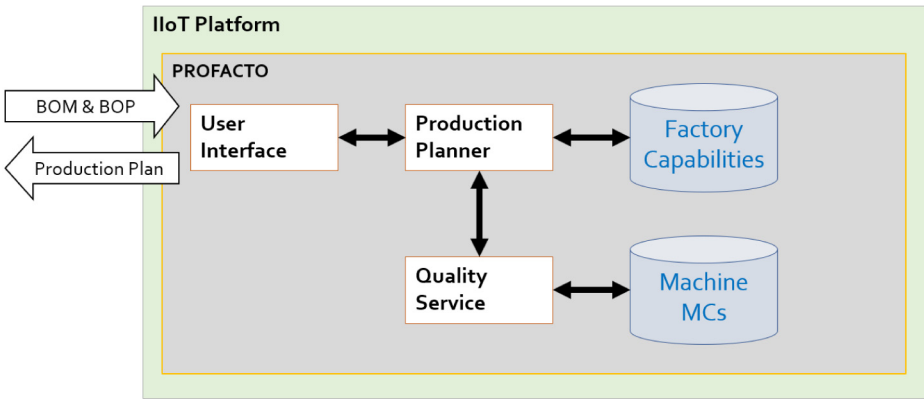


Fig. 7. Cloud architecture.

5. Dynamic production planning

To digitize and automate the task of planning the production of a new product is crucial for smart automation systems. A product is typically specified by a BOM associated with a set of production processes, the BOP, that define the operations

and machine skills that are necessary to manufacture the product ([Dhungana et al., 2018](#)). [Fig. 8](#) shows the BOM/BOP of an example product, a simplified wave gear. This product consists of two parts: the input side (also called the drive side) with the wave generator and the motor flange. The ball bearings and the motor

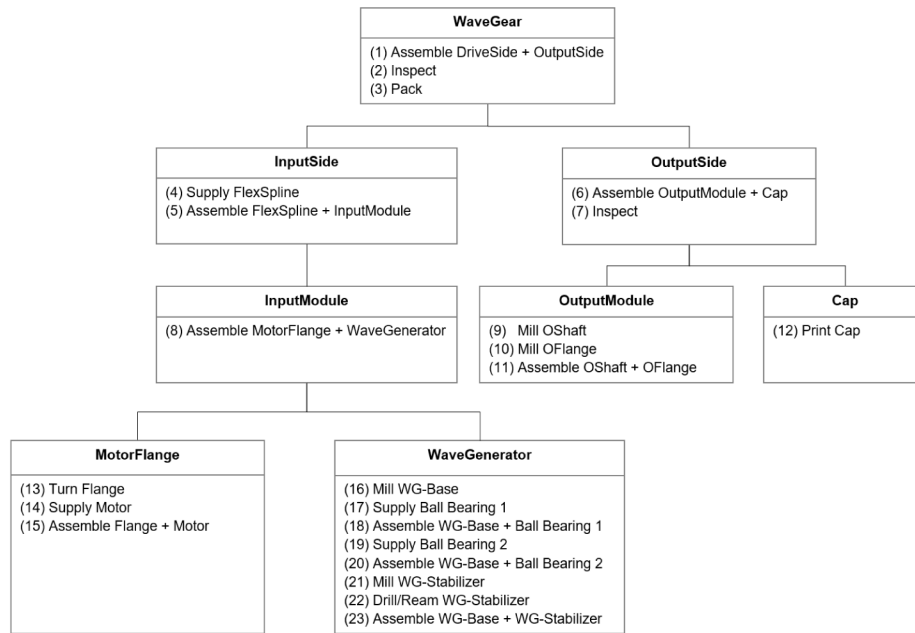


Fig. 8. BOM/BOP of a simplified wave gear product.

are supplied as 3rd-party products, the other input material parts are produced by milling processes.

Associated to each BOM material/part is a sequence of production processes that is necessary to manufacture the corresponding part. For instance, the wave generator stabilizer is produced by a milling operation and a subsequent drilling/reaming step. Assembly steps are special processes that have 2 input materials that are joined together.

The function for computing valid production plans, i.e., machine assignments and production step sequences for a given factory, is usually called *Producibility Check*. If no such production plan could be computed, an answer not producible is returned. Additionally, an explanation is provided to explain the reasons for non-producibility, like missing machines. In the case of producibility, the function computes some or all possible production plans. When an optimization criterion is given, like minimal makespan or minimal production costs, the producibility check finds a production plan that is optimal according to the given criterion.

The computation of production plans is a combinatorial problem. Constraint satisfaction problem (CSP) techniques (Rossi et al., 2006) are well-suited to represent and solve such problems. A CSP consists of variables, their domains, and a set of constraints. A solution is a variable assignment – each variable gets a value from its associated domain – that satisfies all constraints. The formulation of the producibility check as a CSP program is straight-forward. Fig. 9 shows a simple CSP encoding for computing valid production plans. We mainly use two types of variables: $v_m[i]$ is a variable whose value assignment represents the machine that should perform production process i . The variables $v_{start}[i]$ and $v_{end}[i]$ represent time slots when the execution should take place. $v_{end}[i]$ is determined by a simply being the start time of the process plus its expected duration. The input array $duration[]$ represents these execution durations. It should be noted that the time slots in a production plan mainly represent precedence of production processes. Responsible for the planning of the absolute time slots is a shop floor scheduler that dispatches the many tasks of an entire order stack.

Fig. 10 sketches a part of a production plan, i.e., a solution to the corresponding CSP. The nodes are the production processes.

Assigned to each process is a machine that shall execute this process. Directed edges between processes stand for material flow between the acting machines and therefore induce a temporal sequencing of the processes.

The case that a product, represented by a BOM/BOP tree, cannot be produced in a single factory could be mitigated by distributing production to multiple factories. Lately, a number of major, global events proved that flexible/local manufacturing can be essential to maintain a timely production of products. Examples of such events are the COVID-19 pandemic, where free movement of people and goods were again and again only possible to a limited extent, and the blocking of the Suez Canal by a cargo ship for about a week, delaying the delivery of commodities and goods to Europe.

We sketch a simple multi-factory producibility check approach that extends single-factory production plans to multiple factories. This and a second, more elaborate approach was introduced in Dhungana et al. (2020). Production plan generation is controlled by two types of cost factors: *production costs* and *transportation costs*. Production costs are a measure of how expensive it is for a certain machine to manufacture a product or a part of a product. The definition of an appropriate cost function (e.g., minimal monetary costs, minimal makespan, minimal energy consumption, or a combination of several objectives) is up to the application. We assume the availability of a function

$$prodCosts : P \times M \rightarrow \mathbb{R}_{\geq 0}$$

that provides the execution costs for a process p on a machine m . P is the set of all processes of the product's BOM/BOP tree, M is the set of machines of a given factory or a set of factories. The check whether a machine can execute a certain process or not is called *Skill Matching*. Skill matching provides the *prodCosts* values. If a machine cannot execute a certain process, ∞ (infinity) is used as production costs.

When two subsequent processes acting on the same work-piece are assigned two different machines, the work-piece must be routed from the first machine to the second one. This generates transportation costs:

$$transCosts : M \times M \rightarrow \mathbb{R}_{\geq 0}$$

Variables:

$v_m[i] \in machines(f)$
 // $v_m[i]$ represents the assignment of a machine to process i ; N is the number of
 // processes, $i \in \{0 \dots N-1\}$; $machines(f)$ is the set of all machines of factory f

$v_{start}[i] \in [0 \dots N-1]$
 $v_{end}[i] \in [0 \dots N-1]$
 // Variables Representing the start/end time of a process

Constraints:

$v_m[i] \in matching(i)$
 // Only matching machines are allowed values for variables v_m .
 // $matching(i) \subseteq machines(f)$ is a skill-matching function, i.e., it returns
 // all machines that can perform process i .

$v_{end}[i] = v_{start}[i] + duration[i]$
 // The duration of a process determines the end of the process execution.

$v_m[i] = v_m[j] \rightarrow v_{end}[i] < v_{start}[j] \vee v_{end}[j] < v_{start}[i]$
 // Serialize processes on same machine for all processes i and j , $i \neq j$.

$workOnSameMaterial(i, j) \rightarrow v_{end}[i] < v_{start}[j] \vee v_{end}[j] < v_{start}[i]$
 // Serialize processes that work on same materials for all processes i and j , $i \neq j$.

$workOnSameMaterial(i, j) \wedge v_{start}[i] < v_{start}[j] \rightarrow path(v_m[i], v_m[j])$ // There must be a mate-
 rial transportation path (function $path$) between two assigned
 // machines, if the two processes i and j , $i \neq j$, work on the same material.

... Additional, product-specific constraints ...

Fig. 9. CSP formulation of the production plan generation problem.

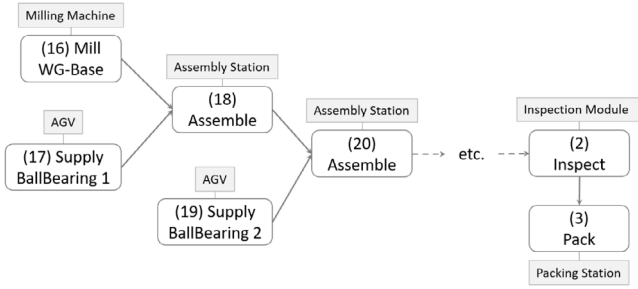


Fig. 10. Example of a production plan for the wave gear product in Fig. 8.

Compared to production costs, transportation costs *within* a factory are typically very small or even negligible. But in a multi-factory scenario, transport between factories – we use the term *shipping* here – is often a dominant cost factor. We again use an infinity value ∞ when transportation between two machines or shipping between two factories is not possible.

In the multi-factory version of the producibility check, only minimal changes to the original, single-factory version (see Fig. 9) are necessary. Let $machines(f)$ be the set of all machines at a factory f . We only have to extend the domains of the machine variables $v_m[i]$ to all machines of all available factories F :

$$v_m[i] \in \bigcup_{f \in F} machines(f)$$

The skill matching function (see function $matching(i)$ in Fig. 9) is extended to all those machines. A solution to this multi-factory producibility check problem must minimize the total costs c_{total} :

$$c_{total} = c_{prod} + c_{trans}$$

$$c_{prod} = \sum_{i \in \{0 \dots N-1\}} prodCosts(i, v_m[i])$$

$$c_{trans} = \sum_{(i,j) \in materialFlowEdges} transCosts(v_m[i], v_m[j])$$

The overall production costs are the sum of all individual production costs of all selected machines. The overall transportation costs are the sum of all material flow costs between machines. We omit implementation details here and use the function $materialFlowEdges$, that represent all pairs of machines where transportation of work-pieces is necessary. Transportation of work-pieces between machines of different factories represents work-piece shipping. Usually, shipping is very costly, therefore optimal solutions will use as few factories as possible.

A constraint solver can solve the above sketched problem formulation, minimizing the total costs c_{total} . The result is a production plan that assigns a machine and a time slot to each production step (variables v_m , v_{start} and v_{end}). If necessary, machines of more than one factory are involved. Fig. 11 indicates an example of an assignment of factories to production processes of the wave gear example product (see Fig. 8). In this case, two factories are necessary. Manufacturing of the drive side and the final assembling is done in factory A, while the output side is manufactured in factory B. A dashed arrow in the production plan indicates shipping of work-pieces between two factories.

In the context of this paper, we are particularly interested in the process of finding out whether a machine is able to execute a production operation or not. This task is called *skill matching*. It is realized in the function $matching(i)$, see Fig. 9. Skill matching is based on a model of skills *required* by production processes and skills *offered* by machines. Constraints formulate conditions that match or match not required and offered skill.

A typical example is that of a reaming process. Amongst other skill properties like position of the required hole, its depth and width, the required position tolerance (RPT) is a crucial parameter. It makes demands on the precision of a milling machine that shall execute the process. A KPI that represents this precision is the total position error (TPE) of the milling machine. The simple constraint

$$v_m[i].TPE \leq RPT$$

states that the TPE of the machine assigned to process i (which is meant to be the reaming process) is less or equal to a given

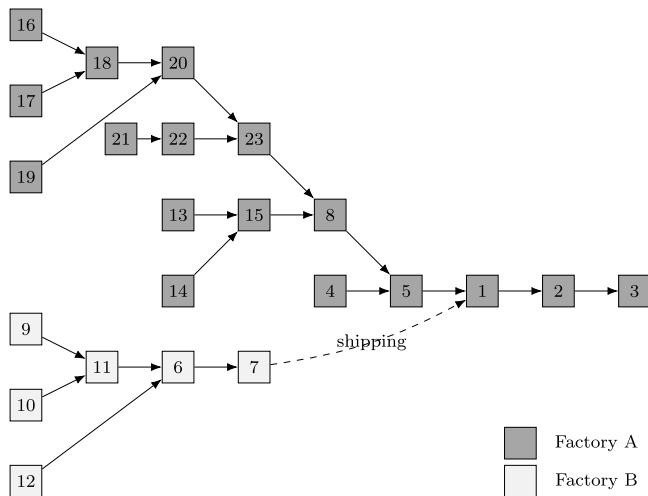


Fig. 11. Example of a production plan that involves two factories.

RPT. Only those machines that fulfill this constraint are considered to be candidates for assignment. It should be noted, that similar constraints are given for all other skill parameters (like: $v_m[i].MAX_DEPTH \geq DEPTH$), and a machine only matches, if none of these constraints is violated.

In a static setting, machine parameters like TPE are given by the vendor of a machine and they are updated never or very rarely. A regular actualization of such KPIs allows skill matching to return machine candidates that are accurate on a daily basis. Short after maintenance of a machine, the TPE may be much lower than given in the spec sheet, and could therefore pass skill matching. On the other hand, after many reaming cycles, the TPE value may fall below the required tolerance level RPT and cannot be assigned.

6. Pilot system implementation

Currently, our research is in an experimental phase. In this section, we describe our approach to self-assessment including stakeholder evaluation as part of our internal validation. To this end, we are piloting our ecosystem approach PROFACTO in several research production facilities in the Austrian cities of Vienna, Graz and Linz (Pichler et al., 2020; Trabesinger et al., 2019; Hennig et al., 2019). The goal thereby is to evaluate the overall system design and get additional feedback from domain experts to better understand the requirements which are significant for production planning.

6.1. Considering real-time machine KPIs for production plan generation

The pilot system is integrated as an application in the IIoT cloud platform MindSphere.⁶ This platform offers a data access layer where assets, meaning a digital representation of a machine, are modeled as aspects that group related data points based on their logical association. For example, the edge-device (see Fig. 4) of a particular machine represents an asset offering a KPI aspect providing one or more real-time KPI values (e.g., total positioning error). There also exists the concept of aspect types which are intended as a pre-configured template for an asset. This allows defining a generic interface for different possible KPI providers for our pilot. Moreover, required essential features for real business

case scenarios like data security, scalability and resiliency are ensured by the cloud platform.

As described in Dhungana et al. (2018), the application allows different stakeholders to interact through the production ecosystem. Product designers can offer their product lines, end-customers can configure and order lot-size-one products, and factory operators can offer their services to manufacture these customized products. Moreover, equipment vendors can offer their equipment to the factory operators as the building blocks of a factory. Hence, factories are modeled based on the configuration of available equipment and how these cyber-physical units are interconnected. This topology is a crucial aspect of a factory. The equipment itself is modeled as a set of capabilities which are required to be able to autonomously execute particular production operations. These capabilities or skills (e.g., supply, transport, drill, ream) form a common vocabulary defined by the ecosystem operator and are essential for collaboration in the ecosystem. Products are modeled in terms of BOMs, BOPs, and feature models, representing the product variability. The materials of the product's BOM use this information to steer their own production and their step-wise transformation towards concrete product instances or product batches (Dhungana et al., 2015). The whole ecosystem PROFACTO is a static view of reality. Hence, equipment used in particular factories has predefined skills as well as quality parameters like total position error (TPE) predefined by the equipment vendors. However, during production, these parameters or KPIs can change for concrete equipment and it is valuable to have a more accurate dynamic view on a factory and its KPIs. The ecosystem thereby allows factory operators to define for a concrete factory exactly which KPIs for which machine should be continuously published to the ecosystem and thus considered for the decision of how to manufacture a concrete product.

In the rest of this section, we continue the example of the production plan generation for a wave gear product (see Fig. 8) which we have modeled with PROFACTO (Dhungana et al., 2017, 2018, 2015) and describe the workflow scenario of an end-user utilizing our prototype app.

6.2. From configuration to the final product

After using the ecosystem's product configurator to customize a wave gear product variant, the end-user performs a producibility check. The results are several valid execution graphs which are presented to the user as production plans, with a ranking based on minimal total costs. During skill matching, the newest published KPI parameters are considered and have a direct influence on the generated production plans. For example, in one first pilot in the factory Graz the KPI total positioning error (TPE) of a SPINNER U630 machining center is monitored by an edge device and continuously published at the ecosystem. These KPIs need to fulfill a certain position tolerance to be able to execute a reaming process to produce a WaveGenerator sub-part of the wave gear.

In our first use case scenario, this tolerance for reaming is satisfied by the SPINNER U630 machining center in the factory Graz. Fig. 12, depicts the screenshot of the production plan showing the production of the InputModule sub-part of the wave gear product in factory Graz. The visualization uses the concept of swimlane diagrams showing which machine is involved in a production process and in which order particular production processes are assigned and executed. Horizontal lanes for each machine visualize the production steps which are executed. Directed arrows connect the production steps with each other and represent the production workflow. For part-joining operations (e.g., Assemble) the input parts are depicted as red labels. In the example, the factory mills a Wave Generator Base using a DMG MORI NLX 2500/700 and assembles Ball Bearing 1 and

⁶ <https://siemens.mindsphere.io/en/docs>.

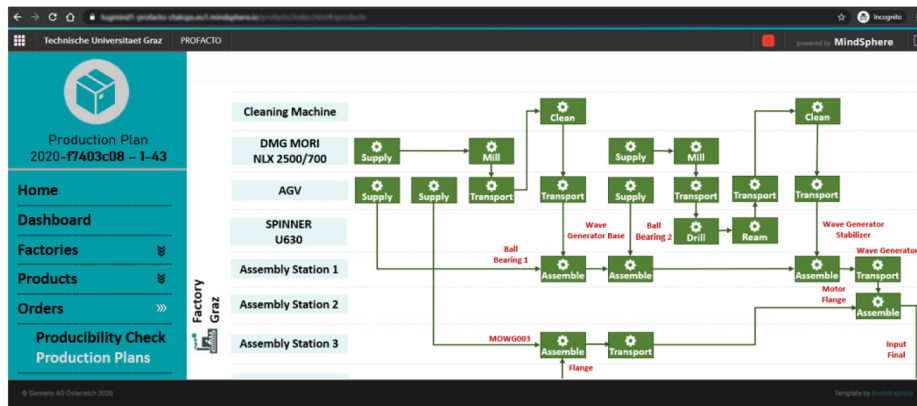


Fig. 12. Screenshot showing part of a production plan of a wave gear product in PROFACTO.

Ball Bearing 2 on the Wave Generator Base using Assembly Station 1. The Wave Generator Stabilizer is produced by processing a milling operation (DMG MORI NLX 2500/700) and by subsequent drilling and reaming operations processed by the SPINNER U630. Assembly stations 1, 2, and 3 take part in succeeding assembling the Input Module sub-part of the wave gear product.

In the second use case scenario, the TPE KPI of the SPINNER U630 has decreased over time and now it does not satisfy the quality constraint defined by the well gear product's BOP any more.

Fig. 13 shows a screenshot of the calculated production plan view for this scenario. Thereby compared to the first example, the reaming process and its subsequent cleaning process are planned to be executed by an EMCO MAXMILL 500 machine located in a different factory (in Vienna). Transport and shipping operations are adapted accordingly. This example shows that a slight decrease of the precision of a machine may completely change a production plan. In our case, production had to be split between two factories, because the original factory was no longer able to perform the reaming process with the required quality constraints.

The use case scenarios demonstrate how factory operators can adjust their production when certain machine quality KPIs change over time. Early rescheduling of production to another machine in the same factory or switching to another factory for a product part makes it possible to prevent low quality of production and contributes significantly to production's resilience. Producers can ensure the utilization of their factories even if they cannot manufacture a product completely. The continuous monitoring of the quality KPIs also allows to safely process orders close to the specified standard quality limits of the machines or even exceed certain of them through, e.g., improved maintenance measures. Further use case scenarios are going to be developed together with the domain experts of the pilot factories.

7. Related work

Monitoring of machine conditions is heavily important in the context of maintenance (i.e. condition-based maintenance) (Pranjapati et al., 2012). Moreover the conditions of machines are significant in the area of production planning and scheduling. However, traditionally *production planning and scheduling* and *condition monitoring* are considered as separate disciplines.

However, when combining both of these disciplines, the quality of decision making can be improved and, by utilizing the data-integration possibilities of cloud-enabled manufacturing services, the resilience of production capabilities can be increased.

For instance, Karner et al. (2019) describes an integrated machine condition-based production scheduling approach using vibration sensors and machine learning methods to classify machine conditions. Another work in the context of the semiconductor industry (Kao et al., 2018) considers so-called dynamic equipment health indicators (e.g., condition deteriorates over time) to schedule production jobs for wafer batching machines. Thereby the health indicator is integrated into production scheduling formulated as a mixed integer linear programming (MILP) model.

Another approach (Schmid et al., 2020, 2021) describes how validated machining data is obtained from the machine via OPC UA and then made available for future production planning. The values acquired using this method are intended to make future production planning more reliable and faster.

The method shown in Li et al. (2020) describes the matching of machine tools based on the Markov Decision Process (MDP) and Cross-Entropy (CE) and the creation of a universal framework. Requirements for machine tools and the machine tools themselves are described in detail. Then, machine tools are filtered to obtain machine candidates for requirements such as machine category, machine accuracy, or energy consumption. To describe the optimal fitting process and model, a set of finite discrete MDPs is applied. The solution is done with the CE algorithm to obtain an optimal machine tool selection that satisfies the total service cost or total service time requirements. Simulation experiments are used to validate the proposed method.

A unified simulation research platform for academic research and industry is created in Tao et al. (2017). A subsection shows the selection and design of matching and scheduling algorithms and their application in a matching simulator.

The article (Mourtzis et al., 2016) deals with the identification of feasible and adaptive process plans. A service-oriented, cloud-based software framework is used for this purpose. It is based on non-linear process plans on the one hand and on machine tool data provided by sensors, operator inputs, and machine plans on the other hand. The process planning service is supplied with data based on processing techniques using information fusion.

The paper (Yang et al., 2016) describes a manufacturing cloud concept that is intended to enable effective handling of uncertainties in the context of manufacturing. The work focuses on the development of a dynamic service selection across several manufacturing clouds. The use of IoT real-time sensors is made possible for the detection of faults such as machine failure.

With regard to technical possibilities, cloud manufacturing (Zhang et al., 2014; Gao et al., 2015) is gaining momentum in several industries. Thereby supporting technologies like Internet of Things (IoT), embedded systems technology, semantic web and open standards for machine-to-machine communications (e.g., OPC UA, MTConnect) are essential to integrate complex

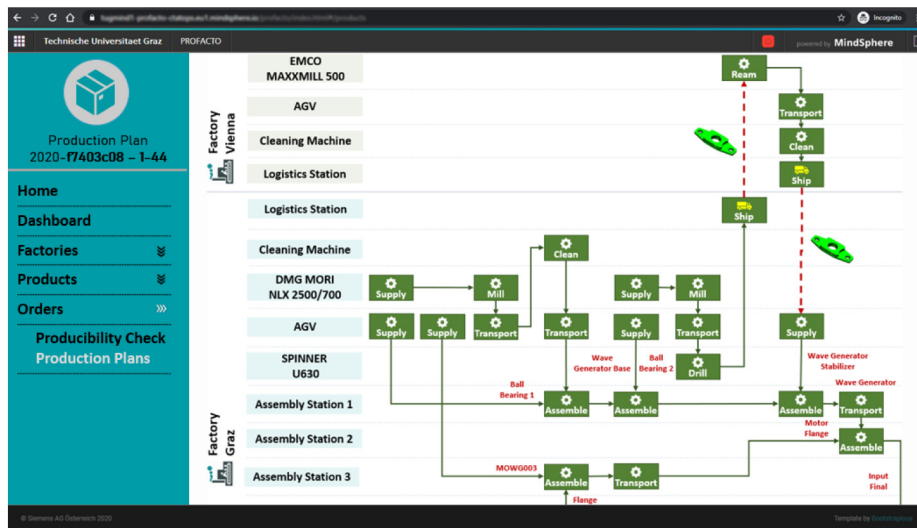


Fig. 13. Screenshot showing part of a multi-factory production plan of a wave gear product in PROFACTO.

distributed manufacturing scenarios. Our idea based on preliminary work (Dhungana et al., 2020) is using especially the possibilities of edge devices (Shi et al., 2016) to preprocess and consolidate real-time machine data for upstream cloud infrastructures and to use these parameter data to infer machine conditions and consider them during production planning.

Architectures similar to the integrated cloud manufacturing architectures described in the present paper are presented in Mourtzis and Vlachou (2019) and Yang et al. (2020). However, the approach described here considers also the concept of collaborative manufacturing, where a machine with deteriorated production skills can dynamically be replaced by another one located in a different factory.

8. Conclusion and future work

This paper described our ongoing efforts in establishing a model-driven collaborative approach for fulfilling production requirements of lot-size-one production orders through collaboration of multiple production facilities. In addition to the static information about the availability of machines and their capabilities in different production sites, our approach also considers production data to assess the condition of machines. This allows for a precise calculation of expected and offered tolerances and leeway in production. Our approach is based on a platform that brings together the involved stakeholders and offers various services for testing how a certain product can be produced through collaborative efforts of the involved factories.

By using the wave gear product as an example, we evaluated the generation of multi-factory production plans. A pilot system implementation based on the IIoT platform MindSphere and three production sites in three cities of Austria was used to demonstrate the approach. In addition to the marketplace for bridging the gap between product designers and production planners, this paper included novel mechanisms for monitoring the production processes using Edge Devices that continuously analyze machine data and ensure that the production constraints are fulfilled at all times.

So far, our approach looks promising, as we have demonstrated its feasibility in pilot settings. However, additional research and continuous refinement through feedback from industry and real world scenarios is still necessary to prove its adequacy for a publicly available, generic production ecosystem supporting multi-factory collaboration scenarios.

Apart from improving the tool support, adding new product models and factory models to the collaboration platform, we are considering data collection during the production process to ensure that the production plan is being executed as suggested by the platform, in order to detect any delays and trigger re-generation of an adapted production plan. This naturally leads to further research questions related to enforcing contractual agreements between the production facilities, e.g., through smart contracts or even building new production consortia on the fly by exchanging factories as required. Quite often there are scenarios where a cyber-physical unit needs a new tool which again can be seen as a product which needs to be produced. Another feature we plan to support is how to handle larger lot sizes, which usually influence transport costs as well as tool management. For instance, a injection moulding machine requires different types of moulds based on the lot size. These issues are being considered in our ongoing research.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work has been partially funded by the Austrian Research Promotion Agency (FFG) under the project grants 852105 (Pilot-fabrik Industrie 4.0 - Smart Production), 861798 (LIT Factory) and 861800 (smartfactory@tugraz).

References

- Bildstein, A., Feng, J., Bauernhansl, T., 2018. Information flow-based capability matching service for smart manufacturing. *Proc. CIRP* 72, 1015–1021.
- Cavalieri, S., Chiacchio, F., 2013. Analysis of OPC UA performances. *Comput. Stand. Interfaces* 36 (1), 165–177.
- Dhungana, D., Falkner, A.A., Haselböck, A., Schreiner, H., 2015. Smart factory product lines: a configuration perspective on smart production ecosystems. In: *SPLC 2015*, Nashville, TN, USA, pp. 201–210.
- Dhungana, D., Falkner, A.A., Haselböck, A., Taupe, R., 2017. Enabling integrated product and factory configuration in smart production ecosystems. In: *43rd Euromicro Conference on Software Engineering and Advanced Applications*, SEAA 2017, Vienna, Austria, August 30 - Sept. 1, 2017. IEEE Computer Society, pp. 266–273. <http://dx.doi.org/10.1109/SEAA.2017.26>.

- Dhungana, D., Haselböck, A., Taupe, R., 2018. A marketplace for smart production ecosystems. In: Hankammer, S., Nielsen, K., Pillar, F.T., Schuh, G., Wang, N. (Eds.), *Customization 4.0*. Springer International Publishing, Cham, pp. 103–123.
- Dhungana, D., Haselböck, A., Wallner, S., 2020. Generation of multi-factory production plans: Enabling collaborative lot-size-one production. In: 46th Euromicro Conference on Software Engineering and Advanced Applications, SEAA 2020, Portoroz, Slovenia, August 26–28, 2020. IEEE, pp. 529–536. <http://dx.doi.org/10.1109/SEAA51224.2020.00088>.
- Gao, R., Wang, L., Teti, R., Dornfeld, D., Kumara, S., Mori, M., Helu, M., 2015. Cloud-enabled prognosis for manufacturing. *CIRP Ann. - Manuf. Technol.* 64, <http://dx.doi.org/10.1016/j.cirp.2015.05.011>.
- Hennig, M., Reisinger, G., Trautner, T., Hold, P., Gerhard, D., Mazak, A., 2019. TU wien pilot factory industry 4.0. *Proc. Manuf.* 31, 200–205. <http://dx.doi.org/10.1016/j.promfg.2019.03.032>, URL <http://www.sciencedirect.com/science/article/pii/S2351978919303968>, Research. Experience. Education. 9th Conference on Learning Factories 2019 (CLF 2019), Braunschweig, Germany.
- Järvenpää, E., Siltala, N., Hylli, O., Lanz, M., 2017. Capability matchmaking procedure to support rapid configuration and re-configuration of production systems. *Proc. Manuf.* 11, 1053–1060.
- Kao, Y., Dauzère-Pérès, S., Blue, J., Chang, S., 2018. Impact of integrating equipment health in production scheduling for semiconductor fabrication. *Comput. Ind. Eng.* 120, 450–459. <http://dx.doi.org/10.1016/j.cie.2018.04.053>.
- Karner, M., Glawar, R., Sihni, W., Matyas, K., 2019. An industry-oriented approach for machine condition-based production scheduling. *Proc. CIRP* 81, 938–943. <http://dx.doi.org/10.1016/j.procir.2019.03.231>.
- Li, X., Fang, Z., Yin, C., 2020. A machine tool matching method in cloud manufacturing using Markov decision process and cross-entropy. *Robot. Comput.-Integr. Manuf.* 65, 101968. <http://dx.doi.org/10.1016/j.rcim.2020.101968>, URL <http://www.sciencedirect.com/science/article/pii/S0736584519300924>.
- Lu, Y., Xu, X., Xu, J., 2014. Development of a hybrid manufacturing cloud. *J. Manuf. Syst.* 33 (4), 551–566.
- Mourtzis, D., Vlachou, E., 2019. A cloud-based cyber-physical system for adaptive shop-floor scheduling and condition-based maintenance. *J. Manuf. Syst.* 47, 179–198. <http://dx.doi.org/10.1016/j.jmsy.2018.05.008>.
- Mourtzis, D., Vlachou, E., Xanthopoulos, N., Givehchi, M., Wang, L., 2016. Cloud-based adaptive process planning considering availability and capabilities of machine tools. *J. Manuf. Syst.* 39, 1–8. <http://dx.doi.org/10.1016/j.jmsy.2016.01.003>, URL <http://www.sciencedirect.com/science/article/pii/S0736584516000042>.
- Pichler, R., Gerhold, L., Pichler, M., 2020. Seamless data integration in a CPPS with highly heterogeneous facilities - architectures and use cases executed in a learning factory. In: Arseniev, D.G., Overmeyer, L., Kälviäinen, H., Katalinić, B. (Eds.), *Cyber-Physical Systems and Control*. Springer International Publishing, Cham, pp. 1–10.
- Prajapati, A., Bechtel, J., Ganesan, S., 2012. Condition based maintenance: A survey. *J. Qual. Maint. Eng.* 18, <http://dx.doi.org/10.1108/13552511211281552>.
- Rossi, F., van Beek, P., Walsh, T. (Eds.), 2006. *Handbook of Constraint Programming*, Vol. 2. Elsevier.
- Schmid, J., Trabesinger, S., Brillinger, M., Pichler, R., Wurzing, J., Ciumasu, R., 2020. Tacit knowledge based acquisition of verified machining data. pp. 117–121. <http://dx.doi.org/10.1109/ICITM48982.2020.9080346>.
- Schmid, J., Vallant, D., Butzerin, A., Brillinger, M., Suschnigg, J., Rudolf, P., Haas, F., 2021. Acquisition of machine tool data via the open source implementation open62541 for OPC-UA. Publication in process.
- Shi, W., Cao, J., Zhang, Q., Li, Y., Xu, L., 2016. Edge computing: Vision and challenges. *IEEE Internet Things J.* 3 (5), 637–646. <http://dx.doi.org/10.1109/JIOT.2016.2579198>.
- de Silva, L., Felli, P., Chaplin, J.C., Logan, B., Sanderson, D., Ratchev, S., 2016. Realisability of production recipes. In: *Proceedings of the Twenty-Second European Conference on Artificial Intelligence*. pp. 1449–1457.
- de Silva, L., Felli, P., Chaplin, J.C., Logan, B., Sanderson, D., Svetan, R., 2017. Synthesising industry-standard manufacturing process controllers. In: *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*. International Foundation for Autonomous Agents and Multiagent Systems.
- Standard ANSI/ISA 95.03-2005, 2007. Enterprise-control system integration, part 3: Activity models of manufacturing operations management. Also available as ISO/IEC 62264-3, IEC.
- Tao, F., Cheng, J., Cheng, Y., Gu, S., Zheng, T., Yang, H., 2017. Sdmsim: A manufacturing service supply?demand matching simulator under cloud environment. *Robot. Comput.-Integr. Manuf.* 45, 34–46. <http://dx.doi.org/10.1016/j.rcim.2016.07.001>, URL <http://www.sciencedirect.com/science/article/pii/S0736584516302605>, Special Issue on Ubiquitous Manufacturing (UbiM).
- Trabesinger, S., Butzerin, A., Schall, D., Pichler, R., 2020. Analysis of high frequency data of a machine tool via edge computing. *Proc. Manuf.* 45, 343–348. <http://dx.doi.org/10.1016/j.promfg.2020.04.028>, URL <http://www.sciencedirect.com/science/article/pii/S2351978920310660>.
- Trabesinger, S., Pichler, R., Schall, D., Gfrerer, R., 2019. Connectivity as a prior challenge in establishing CPPS on basis of heterogeneous IT-software environments. *Proc. Manuf.* 31, 370–376. <http://dx.doi.org/10.1016/j.promfg.2019.03.058>, Research. Experience. Education. 9th Conference on Learning Factories 2019 (CLF 2019), Braunschweig, Germany.
- Yang, C., Lan, S., Wang, L., Shen, W., Huang, G.G.Q., 2020. Big data driven edge-cloud collaboration architecture for cloud manufacturing: A software defined perspective. *IEEE Access* 8, 45938–45950. <http://dx.doi.org/10.1109/ACCESS.2020.2977846>.
- Yang, C., Shen, W., Lin, T., Wang, X., 2016. Iot-enabled dynamic service selection across multiple manufacturing clouds. *Manuf. Lett.* 7, 22–25. <http://dx.doi.org/10.1016/j.mfglet.2015.12.001>, URL <http://www.sciencedirect.com/science/article/pii/S2213846315000218>.
- Zhang, L., Luo, Y., Tao, F., Li, B.H., Ren, L., Zhang, X., Guo, H., Cheng, Y., Hu, A., Liu, Y., 2014. Cloud manufacturing: a new manufacturing paradigm. *Enterpr. IS* 8 (2), 167–187. <http://dx.doi.org/10.1080/17517575.2012.683812>.