

Wagner, Sarah et al.

Article — Published Version

From framework to industrial implementation: the digital twin in process planning

Journal of Intelligent Manufacturing

Suggested Citation: Wagner, Sarah et al. (2023) : From framework to industrial implementation: the digital twin in process planning, Journal of Intelligent Manufacturing, ISSN 1572-8145, Springer US, New York, Vol. 35, Iss. 8, pp. 3793-3813,
<https://doi.org/10.1007/s10845-023-02268-0>

This Version is available at:

<https://hdl.handle.net/10419/317827>

Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

Terms of use:

Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.



<http://creativecommons.org/licenses/by/4.0/>



From framework to industrial implementation: the digital twin in process planning

Sarah Wagner¹ · Clemens Gonnermann¹ · Marc Wegmann¹ · Franz Listl² · Gunther Reinhart¹ · Michael Weyrich²

Received: 25 March 2023 / Accepted: 4 November 2023 / Published online: 28 December 2023
© The Author(s) 2023

Abstract

In today's fast-paced market, companies are challenged to meet increasing customer demands and shorter product life cycles. To successfully respond to these demands, companies must produce a wide variety of different products. This requires the determination of necessary processes and resources for each product, which can be difficult for process engineers due to the high manual effort and expertise involved. The current state of research has not yet provided explicit definitions of the necessary knowledge and has not fully achieved complete process planning automation. To address this challenge, a digital twin is a valuable tool for automating and understanding process planning. This paper presents a digital twin concept for process planning. It automatically analyzes the product, determines production processes, and selects appropriate resources by linking information about products, resources, and processes. The effectiveness of the digital twin concept is demonstrated through verified and validated use cases, including the production of a compressor element.

Keywords Assembly · Manufacturing · Simulation · Ontology · Requirement · Skill

Introduction

In today's fast-paced market, manufacturing companies face an increasingly dynamic environment, including shorter product life cycles, growing market saturation, and the spread of new technologies due to Industry 4.0 (Abele & Rein-

hart, 2011). This leads to increasing individualization, rising customer requirements, and a high diversity of variants (Michniewicz, 2019). Recent challenges, such as supply disruptions due to the Ukraine crisis, add to these fundamental challenges. For each product variant, process planning must determine the required production processes and adequate resources (ElMaraghy & Nassehi, 2019). Numerous challenges impede process planning: heterogeneous data sources, implicit expert knowledge, and a lack of integrated and automated decision support, for example, for planning the production processes of products (Wagner et al., 2021b). As a result, process engineers require support in planning production and logistics processes.

In this environment, the capabilities of a digital twin (DT) can address process planning challenges. A DT represents the physical world in the digital world through granular and realistic data and knowledge (Negri et al., 2017; Talkhestani et al., 2019). Through simulation, it predicts the impact of decision alternatives. Combined with simulation, automated planning capabilities and a task-oriented architecture, a DT appears suitable to support decisions in process planning.

Consequently, this paper presents a methodology to develop a digital twin in process planning (DTPP) supported by current technology trends. It automatically analyzes the

✉ Sarah Wagner
sarah.wagner@iwb.tum.de
Clemens Gonnermann
clemens.gonnermann@iwb.tum.de
Marc Wegmann
marc.wegmann@iwb.tum.de
Franz Listl
franz.listl@siemens.com
Gunther Reinhart
gunther.reinhart@iwb.tum.de
Michael Weyrich
michael.weyrich@ias.uni-stuttgart.de

¹ Institute for Machine Tools and Industrial Management, Technical University of Munich, Boltzmannstraße 15, Garching 85748, Bavaria, Germany

² Institute of Industrial Automation and Software Engineering, University of Stuttgart, Pfaffenwaldring 47, Stuttgart 70550, Baden-Württemberg, Germany

product, determines production processes, and selects appropriate resources by linking information about products, resources, and processes. The DTPP represents the first part of the DT in technical order processing (Wagner et al., 2021b). The paper introduces the basics of process planning and DTs as well as the state of research in Sect. “Literature review”. Sections “Digital twin design procedure” and “Digital twin concept” explain the DTPP design procedure and the resulting DTPP concept. Finally, the paper evaluates the concept and provides a summary and outlook (see Sects. “Discussion” and “Conclusion and outlook”).

Literature review

The following sections describe the basics of process planning and digital twins and explain the current state of research.

Process planning

Process planning is a phase within technical order processing (hereafter referred to as order processing). The latter consists of all value-adding activities necessary for fulfilling customer orders. This encompasses various phases, including research and development, production management, and production and logistics, as illustrated in Fig. 1 (Eversheim, 2002). These phases are not purely sequential; for example, production provides feedback to production management, which includes all activities necessary to prepare for manufacturing and assembly. Within production management, work system planning aims to design and arrange resources (Eversheim, 2002). In contrast, process planning determines processes and selects resources needed to produce a product (ElMaraghy & Nassehi, 2019). Both phases together are called work planning. Production planning and control uses the resulting process plans to plan and control production operationally, temporally, and quantitatively (Schuh & Gierth, 2012). All order processing phases are complemented by information and material flows as indicated by the arrows in Fig. 1. This paper focuses on process planning and considers work system planning.

Process planning is performed at different levels of abstraction: generic planning to select technologies, macro planning to identify process sequences and alternative resources, detailed planning to assign individual processes to tools and specific machines or operators, and micro planning to provide optimal conditions and machine instructions (ElMaraghy, 1993). Due to numerous challenges (e.g., resource allocation, operation sequence determination), this paper focuses on macro planning (Schuh et al., 2020), which consists of four major iterative phases (ElMaraghy & Nassehi, 2019; Eversheim, 2002):

1. **Specification and requirement analysis (P1)** involves identifying the product properties that are relevant to the production of the product.
2. **Process identification and sequencing (P2)** identifies processes and their sequence and assign process properties.
3. **Resource selection (P3)** chooses appropriate resources (i.e. equipment) for the identified process steps. First, the production network with its resources, links and skills must be specified (Gonnermann et al., 2020). Skills are then compared to the properties to select all possible process resource combinations (Michniewicz, 2019; Gonnermann et al., 2021).
4. **Performance parameter determination (P4)** identifies key performance parameters (KPIs) as a basis for selecting a process plan. Note that the goal of this paper is not to automatically determine parameters for programming CNC machines. For that, please refer to computer-aided manufacturing.

As process planning becomes more complex, computer-aided technologies are increasingly important to reduce the planning effort and handle custom products, enabling small batch sizes (ElMaraghy & Nassehi, 2019):

- Computer-aided design (CAD) has become a widely used technology to assist product designers by digitizing product geometries. CAD tools in mechanical engineering create spatial three-dimensional (3D) models of parts or production equipment (Bracht et al., 2011; Dankwort et al., 2004).
- Computer-aided manufacturing (CAM) tools control machines such as robots using programmable software (micro process planning).
- Computer-aided process planning (CAPP) tools link product design and production at the macro (i.e., process sequences and resource selection) and micro (i.e., definition of operating parameters and auxiliary resources) levels.
- Group Technology (GT) aims to combine CAD, CAM, and CAPP (Ajmal, 1992). It involves classifying similar parts into groups based on similarities in design, manufacturing processes, and functional characteristics. These groups allow companies to leverage commonalities and achieve efficiencies in areas such as tooling, production planning, and knowledge sharing.

Although computer-aided tools can reduce the need for manual planning and expert knowledge, knowledge capture and automation of process planning is still part of today’s research. A generic and cross-application knowledge representation of product, process, and resource information would integrate existing knowledge and thus improve the

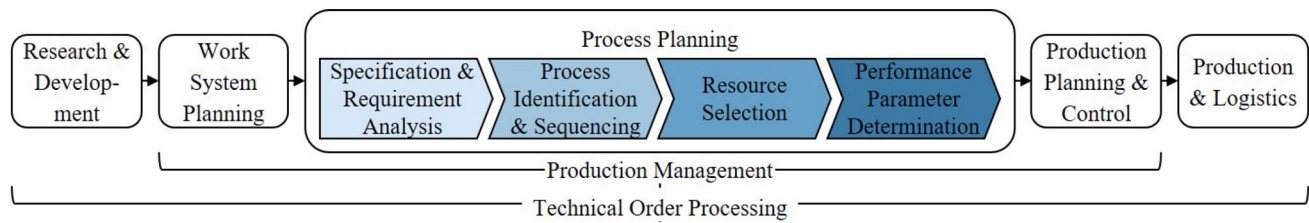


Fig. 1 Phases of process planning and their integration into order processing (ElMaraghy & Nassehi, 2019; Eversheim, 2002)

quality and speed of process planning (Liu et al., 2019). Simulation-based and semantic allocation of resources and processes could support automated process planning.

Digital twin

Many publications characterize a DT as a digital world that represents the physical world (AboElHassan & Yacout, 2022). Since there is no uniform DT definition, Wagner et al. (2021b) derived six crucial DT requirements (R):

1. A DT must consider different levels of abstraction (varying granularity).
2. DTs have to provide a realistic representation of the reference object.
3. A DT must predict and simulate the behavior of the reference object.
4. The virtual reference object should follow the physical object's life cycle.
5. DTs require a task-oriented architecture that integrates all elements.
6. Quality requirements include high automation and real-time.

State of research

To identify relevant publications on decision support systems in process planning, Scopus was searched using the keywords *digital twin*, *process planning*, *assembly design*, and *process design*. The search was filtered to manufacturing, engineering, and production. Relevant articles were selected primarily based on the title and abstract. After a thorough

reading of the articles, they were classified and analyzed or discarded as irrelevant. The analysis considers two evaluation dimensions in line with the initial goal: the coverage of the planning steps (ElMaraghy & Nassehi, 2019; Eversheim, 2002) and the DT requirements (Wagner et al., 2021b) (see Sects. “Digital twin” and “Process Planning”). Harvey balls were used as a fulfillment indicator. Thus, a criterion is rated as not, partially, or fully met. Table 1 explains the different evaluation criteria.

Table 2 evaluates the twelve publications. Several publications describe DT frameworks and implementations. None of them provides detailed information on implementation and validation beyond the conceptual framework. Current research often ignores varying granularity (R1) and only a few publications focus on the application area of process planning (see fulfillment of P1 to P4). Several approaches target operational production management, such as scheduling orders. Other publications only describe the existing requirements and processes within a structured database without identifying them. They do not determine product requirements (P1) and necessary production processes (P2). Most of the literature analyzed describes match-making (P3) conceptually without implementing it. Only two publications implement P2 and P3 without aiming for a DT. The approach of Buechler et al. (2022) is strongly focused on manufacturing, while Michniewicz (2019) focuses on assembly. Neither sufficiently incorporates DT requirements. The rare publications that include process planning and fulfill at least some DT requirements focus on the evaluation and determination of various process parameters (P4). For example, Vishnu et al. (2021) analyze the spindle speed and feed rate in a DT to optimize the surface roughness of the product (Vishnu

Table 1 Assessment criteria for evaluating the state of research

Criteria	Not met ○	Partially met ●	Fully met ●
P1-P4	Not covered	Concept	Detailed implementation
R1	Single abstraction levels	Few abstraction levels	Varying abstraction levels
R2	No representation	Representation of parts	Realistic representation
R3	No prediction	Conceptual prediction	Detailed prediction
R4	No change adoption	Product or production	Product and production
R5	No modules or interfaces	Modules or interfaces	Modules and interfaces
R6	Not automated	Semi-automated	Fully automated

Table 2 Analysis of the state of research

Publication	P1	P2	P3	P4	R1	R2	R3	R4	R5	R6
Liu et al. (2019)	●	○	●	○	●	●	○	●	●	●
Tao et al. (2018)	●	○	●	○	○	●	●	●	●	○
Buechler et al. (2022)	○	●	●	○	○	●	○	●	●	●
Biesinger et al. (2019)	○	○	●	●	○	●	○	●	●	●
Chen et al. (2020)	○	○	●	●	○	●	●	●	●	●
Zhang et al. (2017)	○	○	●	●	○	●	●	●	●	●
Zhang et al. (2018)	○	○	●	○	●	●	●	○	○	○
Ma et al. (2020)	○	○	○	●	●	●	●	●	●	●
Shakirov et al. (2021)	○	○	○	●	○	●	●	●	●	●
Vishnu et al. (2021)	○	○	○	●	○	●	●	●	●	●
Zhao et al. (2020)	○	○	○	●	○	●	●	●	●	●
Michniewicz (2019)	●	●	●	○	○	●	○	●	○	●

et al., 2021). Overall, the current state of research lacks a DT that fully automates all relevant steps of process planning. None of the papers fulfills all planning phases and DT requirements. The paper addresses this research gap.

Three central research questions (RQ) are formulated below to address the identified research gap in detail. First, the paper will answer how the DT creates granular (R1) and realistic (R2) transparency about process planning using appropriate data and knowledge (RQ1). In addition, the paper investigates how the DT can automate process planning (P1–P4) with simulation (R3) support to consider its execution during production and logistics (RQ2). Finally, it answers the question of how to create a lifecycle-oriented (R4) and enterprise-integrated DTPP architecture for decision support (R5) taking into account qualitative (R6) requirements (RQ3).

Digital twin design procedure

The development of the DTPP followed the DT design procedure presented in Wagner et al. (2021a). It uses software design phases (see Fig. 2) and concretizes each phase towards DT design. In addition to the concept model (see Sect. “Digital twin concept”), this section describes the main artifacts, including the system context, use cases, and specified requirements. Section “Discussion” finally verifies the requirements and validates the concept.

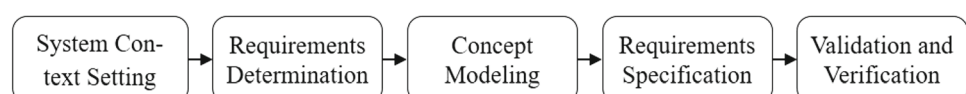
System context

The procedure starts with the separation of the system context from the system to be developed (system boundary) and

the irrelevant environment (context boundary). The DTPP system context (see Fig. 3) has been developed through interviews with industry and research partners and literature analysis.

Within the system boundary, the DTPP targets process planning by considering its phases (see Fig. 1) and touches on work system planning by including reconfiguration. Several authors model process planning on the basis of the threefold concept of product, process, and resource (PPR) (Michniewicz, 2019; Pfrommer et al., 2013; Schleipen & Drath, 2009). It includes products produced or consumed, resources available in the production network, and processes performed by resources to produce products. The DTPP plans manufacturing, assembly, and logistics processes performed by manufacturing, assembly, supply and transportation systems. The resulting process plan produces fabricates or assemblies with different performance parameters, such as cost, time, and quality. They are the basis for selecting the optimal one from alternative process plans. To provide full DT functionality, the DTPP implements all the requirements described in Sect. “Digital twin” (Wagner et al., 2021b).

Considering the context boundary, several processes influence the DTPP with their events and documents. Research and development transfers three-dimensional product designs as input. The production network resulting from work system planning includes all resources with their skills. Production planning and control uses the resulting process plans to generate production plans and control strategies. Finally, production executes the plans and strategies, supported by logistics. Although the DTPP focuses on process planning, it considers the impact of all process planning decisions on production and logistics through its simulation

Fig. 2 Digital twin design procedure adapted from Wagner et al. (2021a)

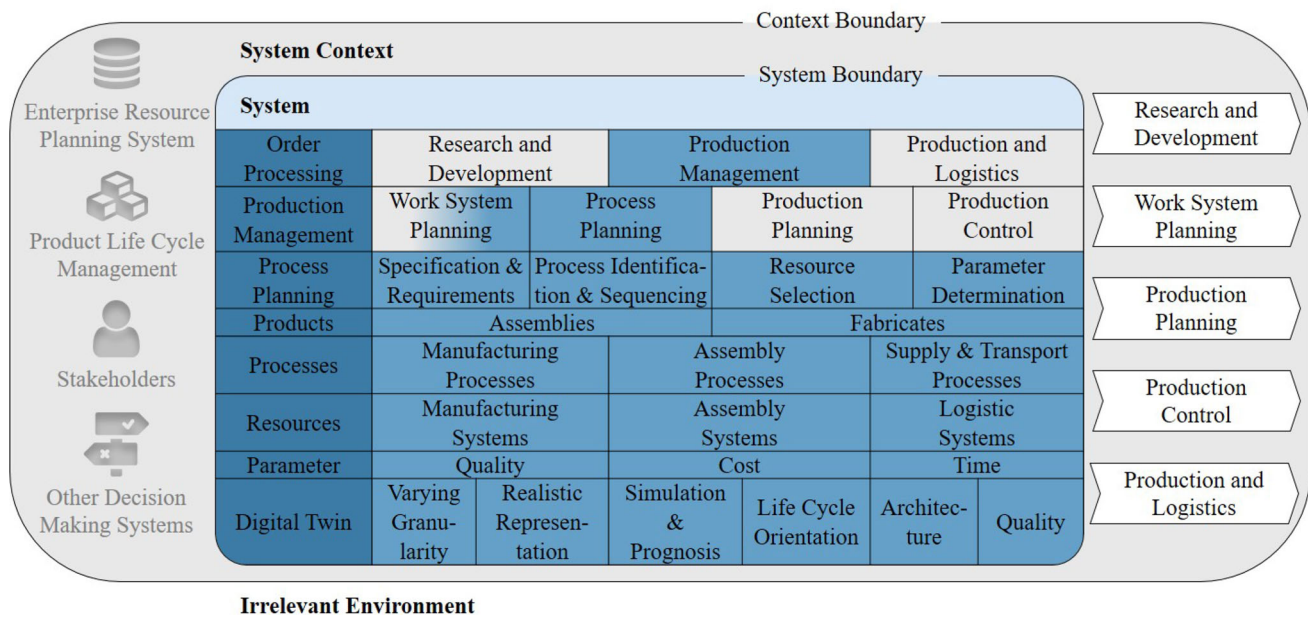


Fig. 3 System context of the digital twin in process planning

capability. Thus, only those parts of production and logistics that execute process planning decisions are within the system boundary. Information systems store the required DTPP input. Enterprise resource planning (ERP) systems manage resources, capabilities, and performance parameters. Product life cycle management (PLM) systems store three-dimensional products. Finally, decision making systems and stakeholder interests impact the system: Designers, process engineers, production planners, and operators have different goals and use cases (see Sect. “Use cases”).

Use cases

Process engineers, the phases of the process planning, and the system context helped define the DTPP use cases and users. Figure 4 illustrates both in a use case diagram modeled using the unified modeling language (UML). In addition to process engineers as primary DTPP users, upstream product and work system designers and downstream production planners and operators use the DTPP. However, they do not have the rights and responsibilities of process engineers. For process engineers, the core DTPP use case is to *manage process planning* (ElMaraghy, 1993). This includes *executing process planning* for all planning steps (see Sect. “Process planning”). In addition, process engineers must *modify process planning* as new knowledge arises, keeping decision making with the responsible person. Last, to *analyze process planning* captures the planning results. Product and work system designers do not require full execution or modification rights. They only analyze the resulting process plans and execute product requirement specification and resource allo-

cation to improve their designs. The resulting process plans are the input to production planners, who generate appropriate production plans and control schedules. Finally, operators in charge of production need access to the process plans for which they are responsible. To ensure privacy across operators, the use case *analyze anonymously* extends the analysis of process planning.

Requirements

The use cases and system context form the basis for determining DTPP requirements. Although preliminary requirements are developed as notes during determination, this section presents all requirements formulated within the specification. The six DT requirements of Sect. “Digital twin” guide the DTPP development (Wagner et al., 2021b). The following paragraphs explain all the so-called building and detailed requirements derived from these base requirements.

Existing abstraction level definitions help to specify varying granularity (R1) of all process planning inputs and outputs. Building requirements include varying granularity of the product (R1.1), resources (R1.2), resulting processes (R1.3), parameters (R1.4), and process plans (R1.5). As an example of a detailed requirement, the production network is specified at multiple levels, including network, site, area, and system.

Realistically mapping (R2) the reference object results in the consideration of the process planning phases with their inputs and outputs. As inputs (R2.1), the DTPP must realistically map the production network and the product. The

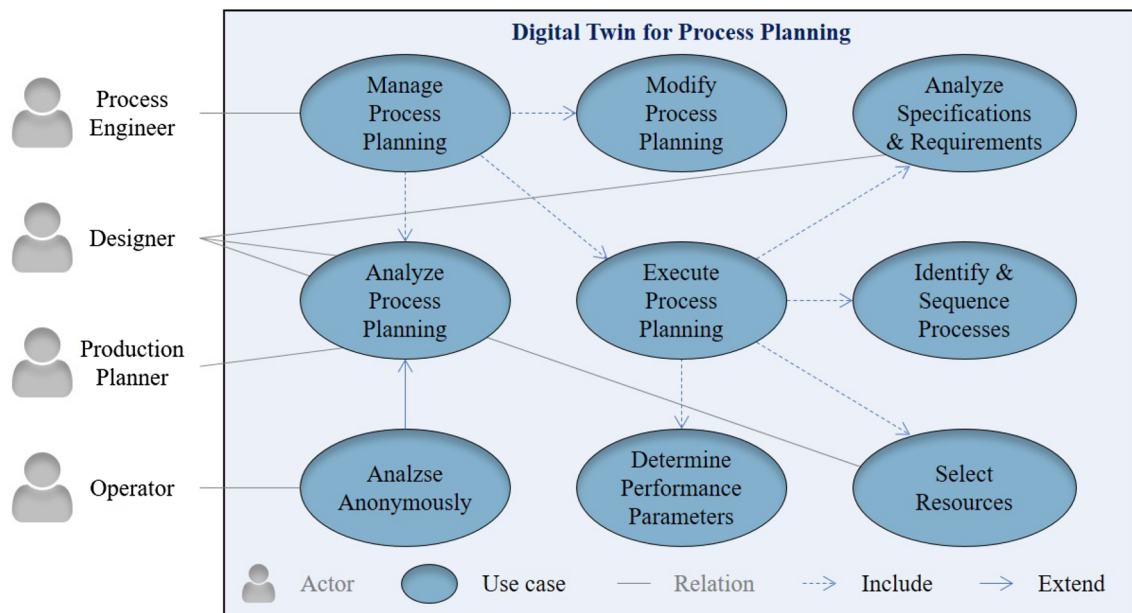


Fig. 4 Use case diagram for the digital twin in process planning

processes, the determined parameters, and the resulting process plan are part of the output (R2.2) to be represented.

The DTPP must simulate and forecast (R3) to generate process plans, validate their feasibility, and select the best performing ones. This includes analyzing product specifications and requirements (R3.1), identifying processes (R3.2) and resources (R3.3), and predicting performance parameters (R3.4). Within R3.2, the DTPP must predict all assembly, manufacturing, and logistics processes to produce a product. It must also forecast appropriate process resource assignments and simulate material flow between resources. Finally, the DTPP must predict the essential parameters of each process resource combination. The main objective of simulation is to consider the effects of process planning on production and logistics in the decision phase.

As more DTs focus on defined life phases, the life cycle (R4) view within the phase becomes important because it accounts for change. In the context of process planning, inputs must be considered throughout their life cycle. These are both production (R4.1) and the product itself (R4.2). Both refer to technical changes, which are defined as modifications of products or production (Koch et al., 2016). Product changes include changes in product material (R4.2.1), form (R4.2.2), or other requirement changes (R4.2.3), while reconfiguration and process changes must be considered as production changes.

A task-oriented architecture (R5) unifies all DTPP components. This requirement is concretized by the main goal of DT: data-based decision support. Numerous publications explain the data information knowledge pyramid to support efficient and effective decisions (Rowley, 2007; Cleveland,

1982). This pyramid describes how signs can be transformed into data, information, knowledge, and thus decisions. For the DTPP, a database (R5.1) must store and regularly update relevant process planning data. The DTPP must include algorithms to generate process plan information (R5.2). In addition, the DT must provide transparency to the complex process planning by adding semantics and logic to the data and information in the form of up-to-date domain knowledge (R5.3). Finally, the DTPP has to support decision making within the company (R5.4) through appropriate interfaces to its systems.

To ensure high quality requirements (R6), the DTPP must avoid redundancies, be automated and understandable, and have high data quality (Tao et al., 2018). The DTPP must be universally applicable, ensured by changeability, scalability, modularity, and economy (Wagner et al., 2021b).

Digital twin concept

The concept of the DTPP (considered as DT for production networks by Lu et al. (2020)) is to support the process planning of existing companies with their processes and systems, as shown in Fig. 5. As described in the context boundary, producing companies already store process planning data in information systems. Until now, decision making systems, such as process engineers, have made unsupported manual decisions by accessing the opaque data in information systems. The DTPP must be economically integrated into this environment to support decisions by using the data from information systems and by supporting decision making sys-

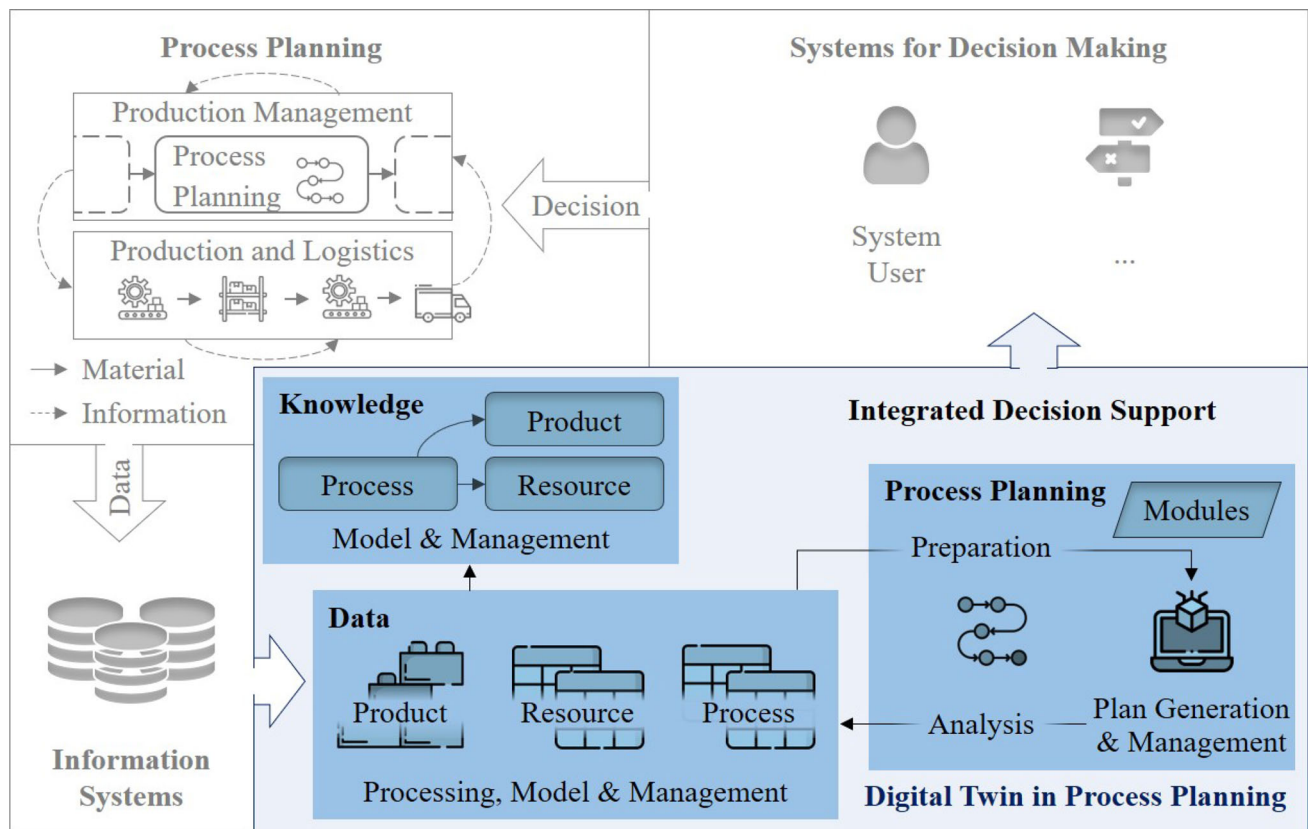


Fig. 5 Concept of the digital twin in process planning concept

tems. We used all the requirements and the system boundary to derive the DTPP concept.

As part of the DT in order processing, the essential DTPP components are a knowledge representation (1), a central database (2), a simulation-based process planning unit (3), and interfaces to the company's systems for integrated decision support (4), see Fig. 5 (Wagner et al., 2021b). The graph-based knowledge representation (1) captures process planning knowledge in a semantic form so that it can be read and understood by both humans and machines. A complementary additional data storage (2) captures data, such as CAD files or time series data, that is not suitable for storage in graph-based knowledge representations. This allows existing data storages to be incorporated into the concept. The resulting data and knowledge transparency supports the generation of process plans within the process planning unit (3). This unit executes the phases of process planning and considers the execution of the resulting process plans through simulating production and logistics. Finally, appropriate interfaces (4) to information and decision making systems ensure integrated decision support and management of process planning. The following sections describe the four building blocks.

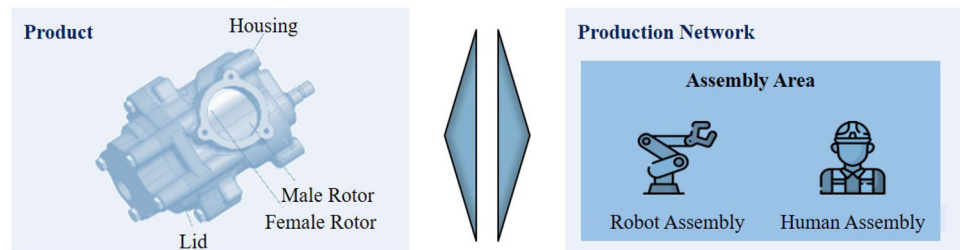
The DTPP has already solved several use cases to test its feasibility. It identified the assembly plans for a brick-based product assembled a robotic cell demonstrator at the Institute

for Machine Tools and Industrial Management (*iwb*) at the Technical University of Munich (TUM) and for an inverter within the production line of an automobile manufacturer. In addition, the process plan for a compressor element of the company Atlas Copco, which is to be produced on a demonstrator, was created. The strategic research center Flanders Make developed this demonstrator, called Infracflex. Infracflex is a flexible assembly line consisting of reconfigurable hexagonal cells (Uzunozmanoglu et al., 2021). The assembly area consists of a human and a robot assembly (simplified example). A compressor element consists of a housing, two rotors, and a lid screwed on top (see Fig. 6). The assembly process includes pick, place, and screw operations. This use case is explained to illustrate the functionality of each component in this paper. At the end of each component description, a paragraph illustrates the component implementation and the software used.

Knowledge

Due to the lack of transparency and implicit expert knowledge in process planning, the DTPP represents knowledge in the form of ontologies (see R5.3). An ontology systematically specifies concepts and vocabularies that describe a domain or part of the real world through classes and the relation-

Fig. 6 Compressor element for the Infracflex assembly use case



ships between these classes (Poli et al., 2010). Due to its high semantic expressiveness, the ontology exchanges knowledge in a digitized and formal format between application programs and humans (Zedlitz, 2013). Compared to databases and their data models, ontologies store semantics/meaning instead of just data through their schema understandable by humans and machines. Ontologies are typically built modularly on different hierarchical levels. The DTPP knowledge representation includes a mid-level ontology, which generally describes process planning, and a domain ontology, which concretizes the previously defined domain. Mid-level ontologies represent common concepts and simplify the mapping between underlying domain ontologies, but are more precise than the generic top-level ontologies (Obrst, 2010).

The DTPP mid-level ontology is based on the triple PPR concept. Existing domain ontologies of the three elements are used to achieve a validated modeling of process planning. Figure 7 visualizes all resulting classes as represented by rectangles. Object properties (represented by arrows) connect two classes. An unfilled arrow represents a subclass relationship, and filled arrows represent process planning specific object properties. The ontology follows the phases of process planning: it maps all inputs, intermediate results, and outputs of process planning. The classes and relations are color-coded according to their affiliation with the inputs or intermediate results of the phases, as shown in Fig. 1. All classes and relationships are explained in the following paragraphs according to the phases of process planning. Different fonts are used to highlight classes and relationships. The compressor element and the resources of Infracflex (second use case) further illustrate the use of the mid-level ontology. Figure 8 visualizes the instances of this process planning example and highlights the classes in the same color as in Fig. 7.

Specification and requirements analysis (P1) examines the most important process planning input: the Product to be produced. The Product concept represents all the products produced or consumed during production. A Product considers different parts and final products that *consistOf* of several subproducts. They *haveProperties*, called *ProductProperties*, such as a specific mass or geometry. In the second use case (see Fig. 8), the Product to be preassembled denotes RotorHousing (core mod-

ule of a compressor element), which *consistsOf* the inputs Housing, MaleRotor, and FemaleRotor.

The Process concept includes several subclasses that define operations that Resources perform to produce Products. The class Process distinguishes *ProcessTypes* and *ProcessNodes* as subclasses. Process identification and sequencing (P2) creates instances of both classes. A *ProcessType* *consumes* one or more input Products to *create* usually one output Product. Depending on the part connections in the CAD file, these *ProcessTypes* *haveProperties*, such as a screwing or joining, force modeled as *ProcessProperties*. All possible *ProcessTypes* sequences are then modeled by *ProcessNodes* according to the principle of precedence graphs. Each *ProcessNode* *isTypeOf* a specific *ProcessType*. *ProcessNodes* are related to each other. Using the relation *requires*, each *ProcessNode* points to its previous *ProcessNode(s)*. In our example, two *ProcessTypes* were identified from the RotorHousing CAD file: First, the operators combine Housing and MaleRotor into the MaleHousing within the *ProcessType* PlaceMaleRotor. Second, you can assemble the MaleHousing and FemaleRotor into the final Product RotorHousing. There is only one sequence because the *ProcessType* PlaceFemaleRotor requires a MaleHousing as input. The two *ProcessNodes* P1 and P2 *areTypeOf* the two *ProcessTypes* and P2 *requires* the previous execution of P1.

Based on the resulting *ProcessNodes*, resource selection (P3) assigns the Resources capable of performing the *ProcessNodes*. The Resource concept describes all the resources in the production. A Resource *isConnectedTo* another one, which means that they transfer Products from one to the other. In addition, one Resource can *beIn* another, such as a production line within an area. Furthermore, a Resource has a certain *ResourceSkill* set, such as a certain load capacity or screwdriving skills, linked by the relationship *hasSkill*. The DTPP selects a resource by matching the *ResourceSkills* with the *ProductProperties* and *ProcessProperties* using the matches relationship. A *ProcessResourceNode* represents a possible combination of a *ProcessNode* that *performs* a process and the Resources that the process

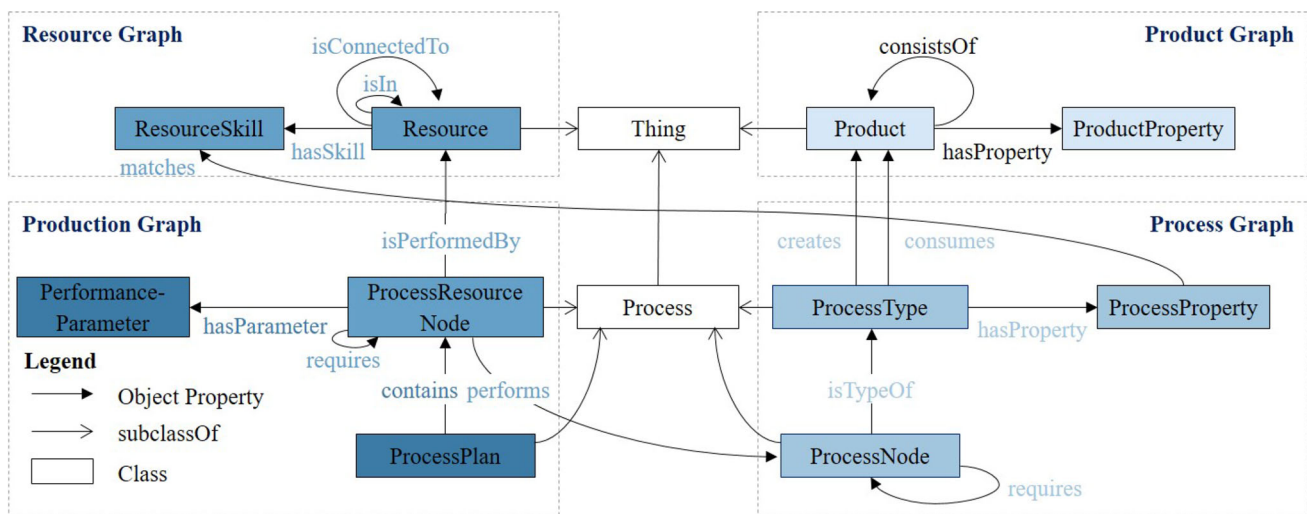


Fig. 7 Mid-level knowledge model of process planning

isPerformedBy. The Infracflex production consists of two assemblies as Resources, where the HumanAssembly *isIn* the AssemblyArea and *isConnectedTo* the Robot Assembly (and vice versa) within the same area. Due to the property skill comparison (not visualized), P1 and P2 can only *bePerformedBy* the HumanAssembly due to missing joining capabilities of the RobotAssembly.

The phase performance parameter determination (P4) assigns PerformanceParameters such as cost, time, and quality (e.g., tolerances) to all ProcessResource Nodes through the *hasParameter* relationship. Based on the parameters, the fourth phase selects appropriate Process ResourceNodes, resulting in an optimal ProcessPlan. ProcessPlans are the result of process planning. A ProcessPlan *contains* a set of ProcessResource Nodes. For the compressor assembly, each ProcessPlan to produce a compressor element contains the same two ProcessResourceNodes to assemble the RotorHousing (independent of determining process parameters not shown): P1onHumanAssembly and then P2onHumanAssembly.

In the following, the modeling of the Product with components and ProductProperties is called a product graph. Using this information, process identification and sequencing generates a process graph including ProcessNodes with associated ProcessTypes. The linked Resources with their capabilities represent the so-called resource graph. The ProcessResourceNodes with PerformanceParameters further extend the process graph and are part of different ProcessPlans. They are all stored within the production graph.

The detailed domain ontology of process planning extends all concepts and their relations and details, in particular the scope of process planning, by concretizing the

mid-level ontology of Fig. 7. For example, libraries of ProcessTypes and ProcessNode consider logistics, assembly, and manufacturing and detail them granularly using standards such as ISO and DIN (see R1.3). This results in subclasses such as Assembly(Node), Joining(Node), and Screwing(Node). The domain ontology further specifies different types of Products, such as parts and finished products, at a granular level (R1.1). It also includes a hierarchy of Resources as subclasses (R1.2): ProductionNetwork, Factory, Area, and ProductionLine, as well as several types of ProductionLines: AssemblyLine, ManufacturingLine, and LogisticElement. In addition to detailing the three main concepts, the representation predefines possible ResourceSkills, ProductProperties, and ProcessProperties, as well as their match. With granularly defined parameters (R1.4) and the previously described levels of abstraction, process plans of varying granularity result (R1.5). Finally, the ontology provides additional functionality through logic rules that, for example, automatically generate corresponding logistics ProcessNodes and ProcessTypes before and after each Manufacturing or AssemblyProcessNode for transport and supply.

The DTPP consists of a graph database that stores the mid-level ontology and the domain ontology with its instances. The instances are managed by methods for mapping data to ontology instances. The mappings automatically create and update all instances. This includes four output mappings that allow the database and the ontology to store the defined graphs: product, process, resource, and production mapping to generate all ontology classes and relationships in each graph. When the process planning unit generates new graphs, the database stores the output and, through the mappings, the instances within the domain ontology. In this way, the

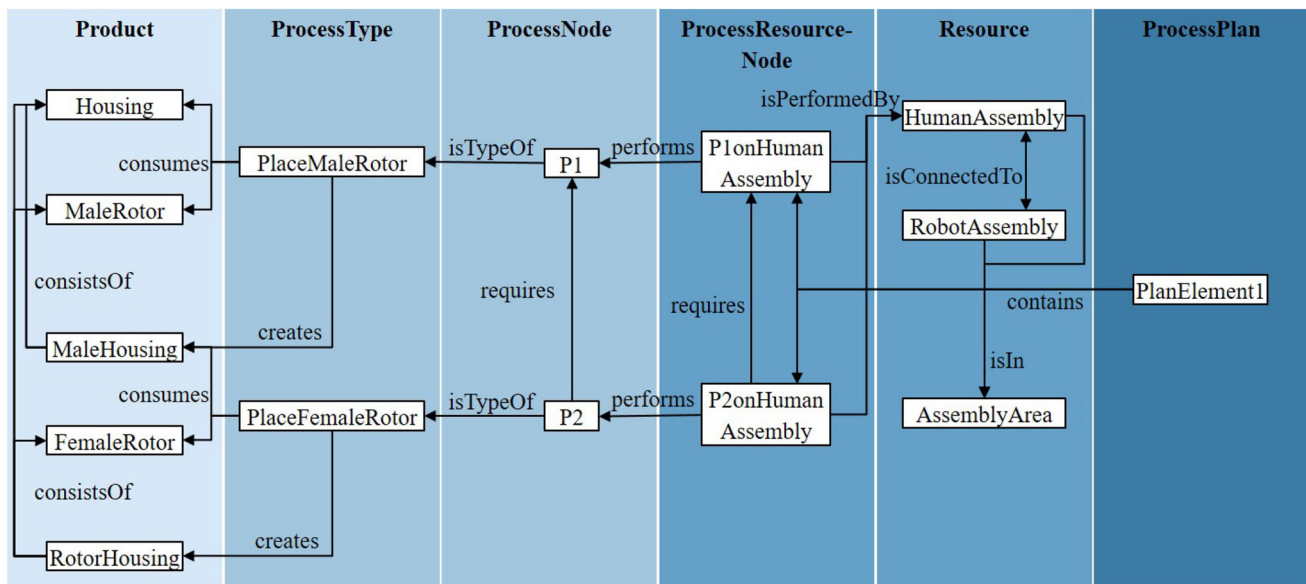


Fig. 8 Mid-level knowledge model for a compressor element

classes and relationships can be created using the data in the database (automatic ontology instantiation). For example, when the database records a new resource, the resource mapping automatically creates a new Resource instance.

Due to its wide distribution and intractability, both ontologies were initially developed using the Web Ontology Language in Protégé. The DTPP stores the instantiated ontology in one of the most popular graph DBMS systems: GraphDB from Ontotext. This tool stores the triples and provides tools to synchronize the knowledge model. Using OntoRefine, multiple JSON files map the data in the database to the ontology to support the creation of the knowledge representation. Using the mappings, new data from the database automatically instantiates and updates the ontology when inserted in the predefined format. The JSON mappings are executed using a Python script that retrieves the data from the database, executes the JSON mappings, and stores the new triples as ttl files in GraphDB.

Data

Because some data types, such as CAD files, cannot be stored in a graph database, and because large amounts of data can be accessed more efficiently, the DTPP includes a central database (see R5.1). The database must contain any data not covered by the ontology, with the ontology pointing to the missing data through virtualization. For example, the domain ontology for the Products to be produced contains the relationship *storedAt*, which records the location of the CAD file in the database. If desired, a company can also develop a central database storing all process planning data. This can be useful when the overview of all process planning data

becomes fragmented due to the presence of heterogeneous data repositories. Then, the database stores all the data and the ontology captures the corresponding meaning. The ontology serves as an explanatory knowledge storage tool that materializes a lot of data through mapping and virtualizes the missing data. In this case, the benefits of both storage concepts can be leveraged, although additional resources are required for redundant information in both the ontology and the database. In both cases, companies need to provide input data for process planning, either within the ontology and the database or solely in the database, where most of it is subsequently mapped to the ontology

Figure 9 illustrates all DTPP input and output data within a simplified data model, indicating optional (redundant) and required data within the database. It ensures that the database provides the data necessary for the operation of the DTPP (R6.2). The inputs follow the PPR model (R2.1): As product data, the CAD file (mandatory in the database) and the product properties are particularly relevant. The figure shows an example of the CAD file of the compressor element and the mass of its rotors. For some processes, additional process properties are required to identify the process. This includes, for example, the joining force required to join rotors and housings. In addition, the database stores large amounts of historical data (historical process resource combinations), which is used later by the process planning modules to predict performance parameters (mandatory in the database for more efficient access). The connected resources and all resource skills form the final required input from the resource perspective. For the Infracflex demonstrator, the production network data includes how the different resources, such as the robot and human assembly, are connected. As an example of a

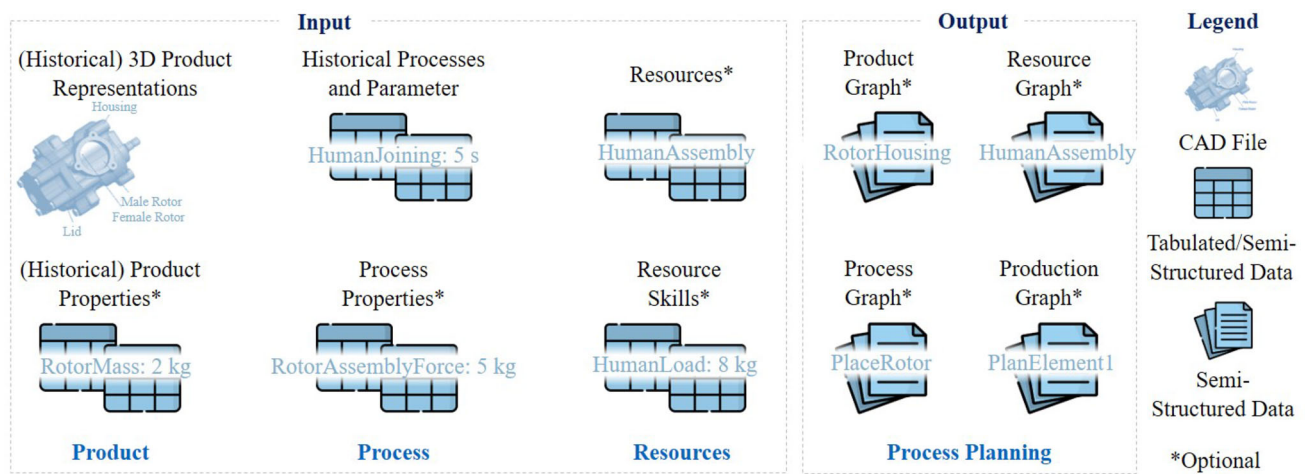


Fig. 9 Simplified data model of the digital twin in process planning

resource skill, robot assembly has a maximum load. The process planning (intermediate) outputs (R2.2) consider the resulting product, resource, process and production graph (see Sect. “Knowledge”). For the centralized data storage, the outputs can also be stored in the database and mapped to the knowledge representation.

A database must manage all the data defined in the simplified data model. The data type is critical in selecting an appropriate database type and its database management system. In Fig. 9, different symbols visualize the three different data types. A CAD file usually represents the 3D product. Historical data includes textual information that can be stored as tabulated data with a consistent syntax or alternatively in a semi-structured data format such as JSON or XML that defines a class structure (i.e., does not require a schema). Since process planning links data to information, semi-structured data formats such as JSON are an appropriate way to store all four graphs (see Fig. 7) in the database within a predefined class structure and data formats for each graph. Because of the different data types, a non-relational database manages the DTPP data. For the DTPP, a document-oriented database can store all the required data types.

For high performance and support, the document oriented database MongoDB is used to store the DTPP data. The implementation imports the CAD model as a STEP file and the historical data as CSV files. All DTPP output graphs are first stored in JSON files. Python scripts then map their schema to the knowledge model as described in Sect. “Knowledge”.

Simulation-based process planning

This section describes how the DTPP generates multiple process plans (see R5.2) based on the process planning phases described (P1 - P4). A process planning module has been

developed for each phase based on the requirements defined in Sect. “Requirements”. Figure 10 illustrates all four process planning modules, their functionality (3) and their interaction with the knowledge representation (1), the database (2), and the system user as part of the system context (4) (numbering analogous to Fig. 5). The figure also illustrates the technologies used to implement the DTPP components. The modules receive input from the database or ontology and output the resulting graphs to the graph database. When the database stores all the data, the required input information for the modules is first mapped to the ontology and then used by the modules (not visualized in Fig. 10). In addition, the database then stores the results (all four graphs) as semi-structured data before mapping them to the ontology (represented by a paper clip in the figure). The following sections describe each process planning module, its functionality, inputs and outputs, user interaction, and implementation technologies.

Specification and requirement analysis

The first process planning module starts with the analysis of the product to be produced (see R3.1). This includes the identification of the individual product characteristics that are relevant to production. The properties are identified through an analysis of the CAD model (e.g., 2 kg weight of the MaleRotor) and additional ProductProperties provided by the product designer (e.g., gripping points, screw torque). The CAD model and the ProductProperties are input data as described in Fig. 9. All (sub)product properties add value to process planning by providing more information for the final derivation of process requirements. This procedure lets the DTPP handle changes in the product. It does this by making a new product properties using the new CAD file or product details. This new graph becomes the starting point for all the next planning steps

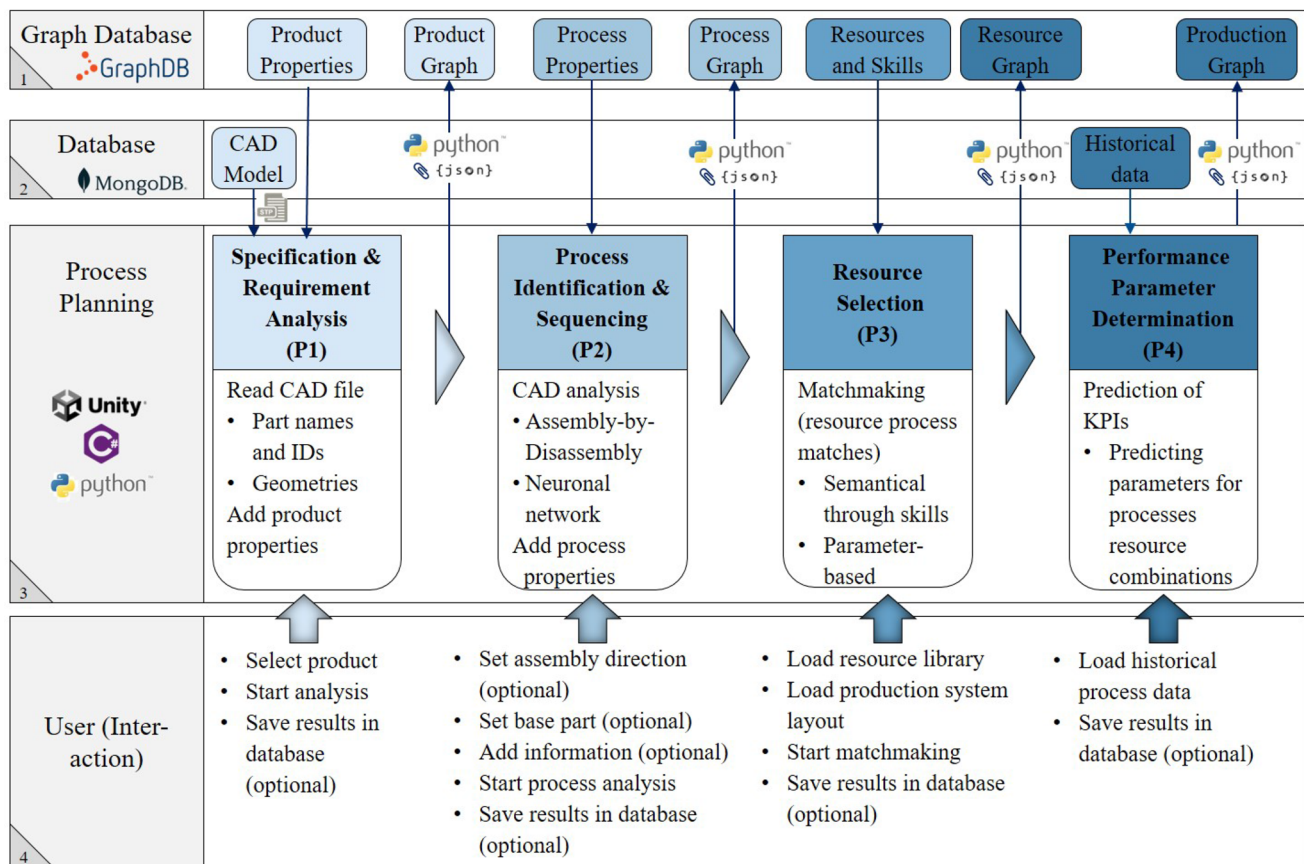


Fig. 10 Pipeline for generating process plans

(see R4.2). The product graph in the knowledge model stores the resulting *Products* and their *ProductProperties* (intermediate outputs of the first module) for later process planning steps shown in Fig. 8. In the Infracflex use case, the product *CompressorElement* *consistsOf* the subproducts *Housing*, *MaleRotor*, *FemaleRotor*, and *Lid*. For each product, product properties are identified with additional information about gripping points, weight, etc. For example, the *MaleRotor* *hasProperty* *Mass* of 2 kg. These properties are the basis for generating process properties within P2. The analysis of the CAD model to generate the product graph and the assignment of the *ProductProperties* to the graph is implemented in the Unity game engine, which uses the C# programming language. The product is imported as an object file and automatically analyzed for its parts and properties. The information is stored in a JSON format and later transferred to the product graph ontology implemented in GraphDB.

Process identification and sequencing

The second module identifies and sequences processes (see R3.2) based on the product graph. The procedures for assem-

bly, manufacturing, and logistics processes are different. Considering assembly processes, a final product consists of several input products to be assembled. An assembly-by-disassembly approach can determine the assembly sequence of an assembled product, similar to Costa et al. (2018) and Michniewicz (2019). A collision clearance simulation disassembles the CAD model in a predefined direction, such as the main disassembly direction (see Fig. 11). Either the user or the product knowledge representation (i.e., product properties) identifies a base part on which the other components are being assembled. This base part remains fixed throughout the disassembly simulation process. The assembly-by-disassembly aims to detect collisions between components during virtual disassembly in the main disassembly direction (if manually defined by the user) and then in the other directions. If collisions occur during disassembly, the approach has detected an invalid sequence of parts to be disassembled. The approach iteratively checks the different disassembly sequences and disassembly directions for collision freedom. Assembly sequences are obtained by inverting the valid disassembly sequences, resulting in assembly *ProcessTypes* and *ProcessNodes*. The *requires* relation connects the *ProcessNodes* representing the arrows

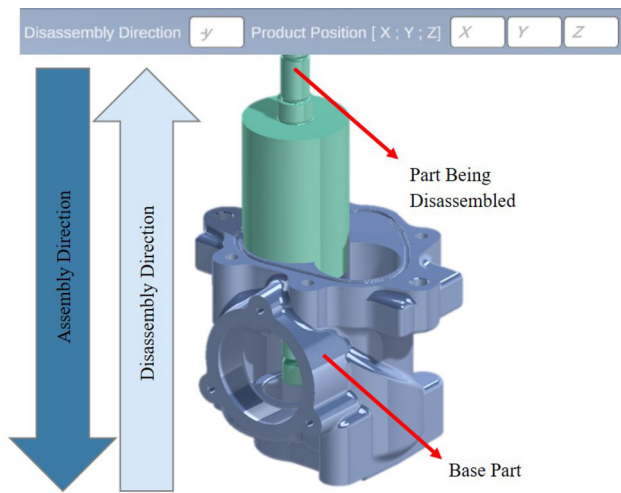


Fig. 11 Assembly-by-disassembly approach

of a precedence graph. Additional *ProductProperties* relevant to the assembly processes (e.g. screws, screw parameters, gripping positions) are analyzed from the first planning step (P1). They allow further specification of the assembly processes (e.g., joining and screwing). If the CAD file does not contain this information, the parameters can be added using predefined *ProcessProperties* via the ontology. In addition, processes such as welding or gluing that cannot be automatically detected by assembly-by-disassembly can be manually inserted. The DTPP thus semi-automatically detects multiple assembly sequences and processes (properties) within a process graph.

As shown in Fig. 8 the *ProcessType* *PlaceMaleRotor* is determined to be a valid process to create the *RotorHousing* after the base part *Housing* is defined. Then the process *PlaceFemaleRotor* is determined to be a valid *ProcessType*. To keep track of the order of the two *ProcessTypes*, they are assigned to two *ProcessNodes*: P1 and P2 (P2 *requires* P1). For *PlaceMaleRotor*, the *ProductProperty* of the *MaleRotor* is transferred to a *ProcessProperty*: *Joining Massof2kg*. In addition, a *ProcessProperty* has been manually added to the ontology: *AssemblyForceOf5kg*. The approach was implemented in the Unity game engine using the C# programming language. It allows the dynamic loading of CAD models as object files and other product information into the environment. The DTPP then executes and simulates the assembly-by-disassembly approach and defines individual process requirements. The resulting process graph can be stored in the database as JSON files and mapped to the ontology described in Sect. “Knowledge”.

Manufacturing processes must be determined for each separate (input) part. Similar to the assembly processes, the *ProductProperties* serve as the main input to determine the manufacturing processes. Different manufac-

turing processes can produce the part depending on the product’s material, size, and other *ProductProperties*, such as heat resistance. In addition, the geometry of the part affects the manufacturing process alternatives. These *ProductProperties* are either extracted from the CAD file or entered manually as described in Section 4.3.1. According to Büchler et al. (2022), a neural network trained with labeled historical or benchmark 2D images of the component and the *ProductProperties* identifies the possible *ProcessTypes*, *ProcessNodes* and resulting *ProcessProperties*, such as the type of manufacturing process or forces (Buechler et al., 2022). The seven images include the six central views of a CAD file (two images along each of the three axes) and the ISO view as input to the neural network. The images are automatically generated from the underlying CAD file and ensure low computation time for the neural network compared to using a 3D CAD file. In addition, the seven images must be labeled with existing and any other feasible alternative technologies. Labeling with alternative processes avoids bias towards conventional processes within the company-specific data. To generate this input database, benchmarks or company-specific CAD files and their corresponding processes can be used. The neural network must then be selected, trained, validated, and compared. Different neural networks are suitable for the classification of manufacturing processes. Based on an exemplary application in car body design, a multi-view convolutional neural network (CNN) with a single CNN1 analyzing each of the seven perspectives and a view pool layer combined with CNN2 bundles all views to generate the output (see Buechler et al. (2022)). The output classifies the finished part according to various possible manufacturing technologies (for example, the geometric characteristics of the rotors are 50 % in favor of casting, 20 % for extrusion, and 30 % for deep drawing). Finally, fuzzy rules decide on the manufacturing technology alternatives based on the information generated from the geometric analysis and the product characteristics. They eliminate geometrically possible processes that do not match the product characteristics.

Contrary to Büchler et al. (2022), KPIs such as costs and the current production network are not considered here, as the subsequent process planning phases consider them in more detail than the fuzzy rules. The process identification and sequencing finally results in a combined process graph for each product by connecting its final *ManufacturingNode* to the *AssemblyNode* that uses the manufactured part as input via the *requires* relationship. All of the manufacturing process identification was implemented using Python with data mining and machine learning libraries such as scikit-learn, pandas, and fuzzylogic.

Finally, the DTPP takes into account logistic processes by adding *LogisticNodes* that refer to *LogisticProcesses* before the *ProductionNodes* through the

relationship *requires*. This includes *FeedingNodes* for all consumed raw materials or supplies and *TransportNodes* for consumed products with previous production stages. Logic rules within the domain ontology automatically generate all *LogisticProcesses* and *-Nodes*.

Resource selection

To select resources (see R3.3), the DTPP first analyzes the production network using the company's input (e.g., production system and resource information) and generates a skill model that defines each resource's functionalities (e.g., screwing and joining possibilities). This phase also identifies the material flow available in the production network (e.g. unidirectional or bidirectional production). It stores this information via *Resources* and *ResourceSkills* in the knowledge graph within the resource graph, see Sect. "Knowledge". If resources or skills change during work system planning, the DTPP captures these changes through changes in the ontology (or the central database). The resource graph is then automatically updated when the resource selection module is executed (see R4.1). As shown in Fig. 8, the ontology can describe the *AssemblyArea* consisting of *HumanAssembly* or *RobotAssembly*, and each resource inherits different skills (e.g., *Joining* with a maximum *LoadOf8kg* for the *HumanAssembly*).

The resources are then allocated through a three-step match-making. The identified processes of P2 place requirements on the production resources through *ProcessProperties*. These requirements can be described in machine-readable semantic terms and with the help of parameters. In addition, the DTPP has to describe the *ResourceSkills* in the same taxonomy to automatically generate allocations/matches. In production planning, the concept of capabilities or skills has been introduced (Hammerstingl & Reinhart, 2018; Gonnermann et al., 2020). DIN standards and VDI guidelines describe processes and skills. In addition, input and output parameters specify process requirements and resource skills. The first match-making step consists of a semantic match-making between the process requirements identified by P2 and the resource skills described in this phase. The DTPP automatically detects several possible allocations, such as robot load (i.e., skill) with assembly force and joining mass (i.e., process requirements). The second step consists of parameter match-making which evaluates each allocation for compatibility by determining for each *ProcessProperty* *matching ResourceSkills* (e.g., *HumanLoadOf8kg* is greater than and therefore *matches* with the sum of *JoiningMassOf2kg* and *RotorAssemblyForceOf5kg*). After validating all the requirement skill combinations, the DTPP detects multiple processes with assigned resources within *ProcessResourceNodes*. This allows

automatic identification of appropriate *Resources* for the *ProcessTypes* and *ProcessNodes* to generate valid *ProcessResourceNodes* for assembly, manufacturing and logistics. If there is a *ProcessNode* that no *Resource* can execute, the DTPP stores all *ProcessRequirements* that do not have a match. This allows work system planning and research and development to see which *ProductProperties* are causing problems and which *ResourceSkills* are missing, allowing reconfiguration. Figure 8 shows the final result of the Infracflex match-making (i.e., the *HumanAssembly* can perform P1 due to its skills (see *PlonHumanAssembly*). The *RobotAssembly* only has *Screwing* skills and therefore does not meet the *Joining* requirement of P1 and P2. Based on the resource and skill inputs, the DTPP can automatically specify the material flow within the production network. Not only value-adding (primary) but also non-value-adding *LogisticProcessTypes* (secondary) processes such as transportation and feeding are matched. The requirement-skill comparison is implemented in a C# environment to identify suitable process resource combinations. A JSON interface provides input to the system, including process requirements and resource skills of a specific production network (i.e., layout). A Levenshtein distance and parameter comparison approach matches the requirements to the skills semantically and parameter-based (C# package). The result can then be exported again as a JSON file and transferred to the ontology via resource mapping (see Fig. 10 and Sect. "Knowledge").

A material flow simulation validates the material flow between *ProcessResourceNodes*. It links each *ProcessNode* to its *Resource* and passes the product to the next *Resource*. For valid material flow links between two *ProcessResourceNodes*, the DTPP adds the relationship *requires* at the level of the *ProcessResourceNodes*, e.g., *PlonHumanAssembly requires P2onHumanAssembly*. Using this simulation, the DTPP can validate supply and transportation processes, or delete them if the resource allocation does not require a transportation process. *PlantSimulation* and *Python* proved to be suitable software for the implementation.

Performance parameter determination

As a result of the resource selection step (P3), several possible process resource assignments have been identified. These serve as input to the performance parameter determination step (P4). The variety of the resulting different process plans can be significant due to the large solution space induced by the number of different process sequences and resource allocations. Therefore, the goal of this step is to predict the corresponding KPIs of the process resource allocations in order to evaluate the quality of the match and to use this evaluation to select optimal process plans (see R3.4). In the

following, a predictive module predicts the KPIs while the prescriptive model selects the optimal process plan.

Within the predictive module, the most important KPI to determine is the processing time, which is used in subsequent planning steps such as line balancing and scheduling. Quality and cost KPIs extend the time in the decision making regarding optimal process plans. The DTPP predicts these KPIs for each process resource combination. In the domain ontology, the *PerformanceParameters* KPI_Time, KPI_Costs, and KPI_Quality are added to each individual *ProcessResourceNode* through the relationship *hasParameter*. Three approaches are used for KPI prediction, depending on the historical process planning data available, such as the KPI_Time of 5s for the *HumanAssembly* performing a similar task, such as P1. A supervised machine learning approach predicts KPIs most accurately when historical data is available in the database. For a small amount of data, a simulation-based approach is appropriate. If no historical data is available, the DTPP can only predict KPIs based on a fuzzy logic approach.

The well-known CRISP-DM (Chapman et al., 2000) guided the development of the supervised machine learning approach. In the first step, the business problem to be solved is defined before the underlying data is observed and evaluated. Then, the predictive module preprocesses the available data. Next, machine learning models are built and their performance is evaluated. Finally, developers select the best model and deploy it across the system. Regression models play an important role in process planning because they can predict continuous variables.

A multi-body simulation model calculates the KPIs of processes performed on specific resources in terms of their cost, quality, and time in the simulation-based approach. Historical data parameterizes the simulation model, including performance parameters for each *ProcessResourceNode*, to ensure data-driven decision making and avoid invalid assumptions. For example, the time simulation of a pick and place process can be simulated by defining the *ProcessParameter* Max_Speed in the ontology and knowing the resource trajectory. Note that for this approach, a multi-body simulation of the corresponding resources must first be modeled, resulting in significant manual effort.

The fuzzy logic approach developed is based on the Mamdani fuzzy system (Mamdani, 1974). It consists of three main steps and quantifies expert knowledge to predict KPIs. First, the approach fuzzifies numerical features describing the combination of process resources and translates them, for example, from 'Automation_Degree = 85/100' to 'Automation_Degree = high'. Expert interviews are used to derive so-called fuzzy sets. In the second step, another interview analysis defines a fuzzy rule base. It consists of several if-then statements and uses the fuzzified input values from the first step. An example rule might be

"IF Automation_Degree = high AND Part_Weight = low THEN KPI_Time = low". Based on these rules, the approach infers new patterns, and thus parameters for new process resource allocations, resulting in fuzzy output values. The final step translates the fuzzy output values back into numerical output values, called defuzzification. So the whole process ends up translating the fuzzy KPI_Time back into a numerical number, such as 26s. Note that depending on the numerical input variables, different rules from the rule base will be active, resulting in different rule weights and the defuzzified numerical output value.

The supervised machine learning and fuzzy logic approaches are implemented in Python. Both approaches consume feature sets of process resource assignments as raw data from the database. The underlying models in both cases predict the KPIs and push them into the ontology for each *ProcessResourceNode*. Further research needs to investigate the implementation of multi-body simulation. As a result, this step generates KPIs for different combinations based on historical data, a multi-body simulation or expert knowledge.

The prescriptive module aims to select the final process plan based on the identified parameters to meet the users' objectives. Once the relevant KPIs of the *ProcessResourceNodes* have been predicted, the decision makers need assistance in selecting a process plan that meets their needs. In the prescriptive module, the underlying decision problem to be solved is a problem of allocating processes to resources. Depending on the size of the problem, different possibilities can be distinguished. A trivial approach can be used to select the most appropriate process plan for relatively small solution spaces. In such a case, the predictive module calculates KPIs for all possible combinations of process resources and selects the most appropriate combination. A mathematical optimization problem solver selects the most appropriate process plan for a relatively large solution space. The DTPP adapts the problem's objective function, which is a weighted sum of the predicted KPIs, to the decision maker's needs. The prescriptive model also defines several constraints to find valid process plans while maximizing the objective. Operations research solvers can find heuristic or optimal solutions to mathematical optimization problems, such as Hashemi-Petroodi et al. (2022). Another approach to address large solution spaces is to use novel machine learning techniques, such as reinforcement learning. In reinforcement learning approaches, an intelligent agent interacts with the DTPP and is rewarded for executing appropriate process plans. The predicted KPIs define the value of the process plan. A prerequisite for the use of reinforcement learning is the definition of process plan selection as a sequential decision process.

The final module of the DTPP selects process resource combinations that best meet the desired KPIs of the deci-

sion makers. The union of all required process resource combinations results in the overall process plan. To implement the module, the KPIs of each process resource combination must be queried and loaded from the ontology into the optimization procedure, either a mathematical model or a reinforcement learning approach. The procedure selects the most suitable process plan or several suitable process plan alternatives. It pushes them back to the ontology into an instance of a *ProcessPlan* consisting of multiple *ProcessResourceNodes*. For the *CompressorElement*, the *ProcessPlan* *PlanElement1* was run with the best *ProcessParameters* including, for example, *PlanHumanAssembly* with a *KPI_Time* of 5 s. Future research includes further implementation of the module. Detailed models and implementations from the literature can form the basis for several approaches, e.g., Gonnermann et al. (2022).

Interfaces for integrated decision support

In order for the DTPP to be used for automated decision support in process planning (see R5.4), it must be integrated into the company by integrating the systems and phases described in Fig. 3. Data transfer between the separate systems is difficult due to different data formats, varying information systems, custom software, and limitations posed by commercial software (Kunath & Winkler, 2018). The following sections first describes the connections to all relevant systems, as shown in Fig. 5. This includes how the DT automatically receives its input data from enterprise information systems (see Sect. “Interface to information systems”) and how the DTPP automatically provides its outputs to systems for decision making (see Sect. “Interface to decision making systems”) to fulfill R6.1 and R6.3. In addition, Sect. 4.4.3 explains the connection to the phases of order processing shown in Fig. 1 that affect process planning and vice versa.

Interface to information systems

The DTPP needs appropriate interfaces to information systems in order to access all necessary data (as described in 4.2). The essential product, process, and resource data are usually stored in different systems and databases, such as ERP or PLM systems. These multiple heterogeneous sources store the data relevant for process planning in different formats, such as CAD, JSON, CSV, etc. Both DTPP ontologies create a common understanding of the data. On the one hand, data can be virtualized, including a reference to its storage location. On the other hand, it can be integrated directly. Virtualization is particularly useful for large amounts of data, such as historical and time-series data, or data in unsuitable formats, such as CAD files. Specifically, for the CAD file, the format can be transformed if the source format does

not conform to the DTPP format. Model-based definition CAD formats, such as STEP, QIF, or JT, include product and process requirements and product geometry. Transformation processes insert these product and process properties in the DTPP database. At the same time, the non-virtualized data must be integrated into the DTPP ontology (materialization). This requires two steps. First, software interfaces must be defined to retrieve the data from the different systems. Second, mappings from the retrieved data to the ontologies must be set up so that the heterogeneous data can be included in the knowledge graph.

However, the latter task is not easy to implement if the user has limited knowledge about ontologies and semantic technologies. Therefore, it is helpful to create a pipeline that allows the user to integrate the required data into the knowledge graph without much knowledge of semantic technologies or ontologies. Extract transform load (ETL) procedures allow data to be transferred from a source database to a target database. Typically, these underlying information systems provide interfaces that can be used in the ETL processes. The idea is to create unified formats that fit the ontological schema. For the DTPP, the database and its data model define the unified formats for process planning. The DTPP ontology supports the generation of the transformation programs by capturing knowledge about the target database (DTPP database) and the source data model, e.g. through defined synonyms in the ontology. In the ETL pipeline, the first phase requires the user to specify the data sources and create transformations for the data into the unified formats. The preprocessed data can then be automatically mapped to the ontology (and cached in the central DTPP database if required) as described in “Knowledge”. The mappings are created in advance and can be used repeatedly for different data sources. For the implementation of the ETL procedure, general programming languages such as Python provide modules for creating the ETL pipelines. This includes the implementation of connectors to data sources such as ERP systems. For better usability, it also makes sense to use flow-based low-coding editors.

Interface to decision making systems

Due to the variety of products and different stakeholders of the DTPP, the decision making systems need support to use the DTPP effectively and efficiently. For the central decision making system, the user, an interface that considers the capabilities and permissions of the different users of the DTPP is needed to support all DTPP use cases (see Sect. “Use cases”). As functional requirements derived from the use cases, the interface must allow the execution, modification, and analysis of the data used and generated by the system. In addition, the interface must consider user-specific access rights. The interface considers different roles with their activ-

ities and permissions, as well as collaboration between users. Quality requirements such as correctness or usability are also considered as non-functional requirements. Based on these requirements, the user interface concept was developed. After user login, the interface adapts views and access rights according to the user. On the start page, the user can execute and modify the four planning phases and analyze specific planning documents. In addition, various menu bar elements help users to work efficiently and collaboratively. When executing process planning, the input data is selected first. The interface then displays the output data. Subsequently, all data can be modified if necessary. In addition to visualization, the user interface offers speech-based interaction with the system by integrating a chatbot. It can understand written text through natural language processing and, using the ontology of the DTPP, output it as a written response through natural language generation. For other decision making systems, such as visualization systems, inference engines, or AI methods, to access the results of process planning, the data and knowledge are available in standardized formats, and prototypical interfaces exist to facilitate exchange. JavaScript, HyperText Markup Language (HTML), and Cascading Style Sheets (CSS) were used to implement the front-end. For data and information analysis, the user interface includes an interface to MongoDB and GraphDB via various Python scripts. This allows to display the four different graphs as well as numerous input data. A local instance of WebProtege visualizes the ontologies. Interfaces to the individual planning modules are also supported: A WebGL framework and Open CasCade (OCC) transform a three-dimensional product file and visualize it on the web (P1). Both an embedded Unity instance (assembly) and an interface to a Python-based manufacturing process planning system allow the identification and sequencing of the production processes (P2). Finally, a Python-based script creates a local PlantSimulation model that predicts the parameters and selects resources (P3 and P4). The user interface visualizes the output data. The chatbot uses existing natural language understanding, generation and dialog control features of the Rasa chatbot framework. The chatbot accesses the ontology in the ttl file via defined actions and the Python module owlready2. Other decision making systems can access the data using Python MongoDB and GraphDB connectors.

Connection to the phases of order processing

Since several phases are interrelated within order processing, this section explains the relationship of the DTPP to each upstream and downstream phase. Research and development influence process planning with their product CAD files. When a company introduces a new product, it loads the new CAD file into the DTPP database, and DTPP provides a new process plan for that product. In addition, research and

development receives feedback from DTPP for the product being created. If a product cannot be produced with current resources, the DTPP provides the product characteristics that are in conflict. They can use the DTPP to evaluate their product proposals. Work system planning influences process planning by providing the layout of the production network. The DTPP partially covers this phase (see Fig. 3). To consider reconfiguration, the DTPP provides suggestions for arranging resources to perform different processes, e.g., faster or at lower cost. In addition, the DTPP describes the skills that are missing from the production network if the current production network cannot produce a product. Therefore, work system planning must evaluate the reconfiguration proposals across all products and decide how to procure resources with the appropriate skills. Production planning and control uses the resulting process plans to generate a production plan/schedule and apply logic to control production when malfunctions or illnesses occur. To take into account the capacity and availability of resources within this phase, the DTPP provides several appropriate process plans. Within production and logistics, different process plans are executed according to the defined schedule and control strategies. Except for production planning and control decisions, the DTPP considers production and logistics through its simulation and prediction capability. For example, the performance parameters (time, cost, quality) resulting from the execution of the process plan at each process step are the primary input for performance parameter determination to consider real parameters. Some publications already support aspects of the mentioned phases (see Kousi et al. (2021) or Wang and Wu (2020)). Defined information flows between DTs of these phases with the DTPP (DT network) would allow a holistic support of order processing.

Discussion

Finally, the fulfillment of all DTPP requirements (see Section 5.1) evaluates the DTPP concept. Based on the DTPP implementation and its application in the two use cases, Section 5.2 discusses its strengths and limitations.

Requirements evaluation

The DTPP concept is first evaluated against all of its requirements (see Sect. “Requirements”). Table 3 summarizes the fulfillment of each DTPP requirement and which main component ensures this requirement (often many components are responsible). By averaging each building requirement, a DTPP base requirement is evaluated. The database and ontologies provide varying granularity (R1) and realistic representation (R2). The process planning unit simulates/predicts (R3) product requirements, processes, resource

Table 3 Evaluation of DTPP building requirements

Requirement	Fulfillment	Component (Reference)
R1.1 Product	●	Domain Ontology (4.1)
R1.2 Resources	●	Domain Ontology (4.1)
R1.3 Processes	●	Domain Ontology (4.1)
R1.4 Parameters	●	Domain Ontology (4.1)
R1.5 Process Plans	●	Domain Ontology (4.1)
R2.1 Inputs	●	Database (4.2)
R2.2 Outputs	●	Database (4.2)
R3.1 Analyze Product	●	Planning Module 1 (4.3)
R3.2 Identify Processes	●	Planning Module 2 (4.3)
R3.3 Select Resources	●	Planning Module 3 (4.3)
R3.4 Determine Parameters	●	Planning Module 4 (4.3)
R4.1 Production Changes	●	Planning Module 1 (4.3)
R4.2 Product Changes	●	Planning Module 2+3 (4.3)
R5.1 Database	●	Database (4.2)
R5.2 Generate Process Plans	●	Planning Modules (4.3)
R5.3 Domain Knowledge	●	Domain Ontology (4.1)
R5.4 Decision Support	●	Interfaces (4.4)
R6.1 Automation	●	Interfaces (4.4)
R6.2 Correctness	●	Database (4.2)
R6.3 Universal applicability	●	Interfaces (4.4)

allocations, and performance parameters to generate process plans. In addition, it considers life cycle orientation (R4). Finally, the interfaces of the DTPP ensure an architecture adapted to the process planning (R5) and several qualitative requirements (R6). Further software development can improve qualitative requirements such as correctness and universal applicability with suitable interfaces that can be used in many companies.

Given the research questions, the domain ontology and the database create transparency about process planning (RQ1). The planning modules automate process planning (RQ2) and together with all interfaces form the required DTPP architecture (RQ3).

Application evaluation

The DTPP implementation and the three use cases (compressor element, inverter, and brick-based product) identified the limitations and capabilities of the concept. All use cases are simplified examples with a limited number of resources, products, and process alternatives. The DTPP concept is able to plan processes for more complex products or production networks, considering performance limitations and manual adjustments, because the implementation is only a prototype. For example, inserting the second rotor of the compressor element proved to be problematic because it can only be assembled without collision in a rotating motion, but it is also not a pure screw operation. This complex assembly process

had to be added manually. However, the DTPP has proven to be very robust, generating numerous process plans for different products produced on different systems. In addition, the framework is very flexible because it presents the components of DTPP in a tool-independent way and many commercial or open-source systems are suitable for implementation. In the described implementation, changes such as other data formats than STEP are particularly challenging. Changes in the production system (resources or layout) have to be considered manually in the CAD file. The developed user interface supports the manual effort and allows the selection of product, production system, KPIs and the resulting process plan to be used in production.

As part of the underlying research project, the participants evaluated the knowledge base as mature and valuable in the application of both use cases. The database as a complement achieved high access performance. When generating process plans, many product properties are automatically identified if they are detailed in the CAD file (P1). The DTPP conceptually considers manufacturing and assembly (P2). However, the implementation of the DTPP focuses on assembly by screwing or joining processes. The generation of assembly plans with these processes (e.g. joining and screwing) has been tested and validated in research projects at *iwb*. The integration of manufacturing processes into the DTPP has yet to be tested and validated. The allocation of resources (P3) to previously identified process requirements, based on semantic descriptions and preset parameters, depends on a unified

and sorted description of processes and resource capabilities. The process resource matches allow multiple alternatives of process plans by considering the material flow of the production system layout. The capacity of individual resources and more or less suitable process plans are not yet considered due to the focus on small batch sizes. The determination of process parameters (P4) is conceptually developed based on artificial data (supervised learning based approach) and a simplified expert interview (fuzzy logic based approach), since only rare real data points were available. The validation with real production data, the simulation-based approach, and the entire prescriptive module need to be performed in future research. The automated process planning of the DTPP considers the execution of the process plan in production and logistics at all stages through simulations. The implementation showed that the DTPP interfaces need to be specifically adapted to the information systems or decision making systems. As a result, it was not possible to develop universal interfaces because they are highly dependent on the software used (e.g., SAP and Oracle as possible ERP systems, and PowerBI or Splunk for decision support). A suitable interface could only be developed for humans. In addition, company-specific tools or process planning procedures often already exist. Due to the modular structure of the DTPP (database, ontology and four planning modules), semi-automatic process planning using individual planning modules for dedicated phases or pure knowledge storage through the ontology are also possible.

Conclusion and outlook

Given the shortcomings of CAPP systems in industry and research, this paper presents the concept and implementation of a digital twin in process planning (DTPP). The DTPP framework consists of four central pillars. A database stores and concatenates all necessary data sets for decision making in process planning. A knowledge graph describes the database and adds semantics to the raw data. Simulation-based process planning approaches and assembly-by-disassembly algorithms generate the process plans. Finally, the last element includes interfaces to information systems and decision makers.

All building blocks of the DTPP have been implemented in an industrial use case and demonstrate superior performance, especially in terms of time to complete multiple valid process plan tasks. Despite the high level of functionality, several research questions remain:

- First, the technical integration of a DTPP into an existing company infrastructure remains partly vague, since interfaces to existing software are highly individual in most industrial use cases. There are different data require-

ments, data formats and security measures. Detailed product data is often not available, and product designers do not automatically generate or store information about the product properties that served as the basis for this DTPP approach. Therefore, future research needs to address a change in mindset and product design processes as well as standardized interfaces between software. In addition, the role of the human as the final decision maker in an automated decision support system needs to be further explored. Transparent solutions and user interfaces can play an important role here.

- As a future research goal, the challenging steps of production planning and control also require the support of a digital twin. Planning and controlling production in terms of time, quantity, and capacity is equally challenging for production planners due to the large number of product variants and the wide range of resources with varying capabilities and availabilities.
- Finally, all of the resulting digital twins require predefined information flows in order to be connected via appropriate interfaces to form an intelligent digital twin network that supports the entire order processing.

Acknowledgements This research was funded by the EU project ASSISTANT grant number 101000165. The authors would like to thank all the participants of the ASSISTANT project.

Funding Open Access funding enabled and organized by Projekt DEAL.

Data availability The data supporting the findings in this paper are available upon request from the corresponding author, Sarah Wagner. The data are not publicly available due to privacy restrictions imposed by the companies participating in the research project.

Declarations

Conflicts of interest The authors declare that they have no competing interests.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Abele, E., & Reinhart, G. (2011). *Zukunft der Produktion: Herausforderungen, Forschungsfelder. Chancen*: Hanser.
- AboElHassan, A., & Yacout, S. (2022). A digital shadow framework using distributed system concepts. *Journal of Intelligent Manufacturing*, 34(8), 3579–3598. <https://doi.org/10.1007/s10845-022-02028-6>
- Ajmal, A. (1992). Knowledge based group technology (gt) classification and coding system for the casting industry-methodology and its application. *IFAC Proceedings Volumes*, 25(7), 85–90. [https://doi.org/10.1016/S1474-6670\(17\)52344-7](https://doi.org/10.1016/S1474-6670(17)52344-7)
- Biesinger, F., Meike, D., Kraß, B., & Weyrich, M. (2019). A digital twin for production planning based on cyber-physical systems: A case study for a cyber-physical system-based creation of a digital twin. *Procedia CIRP*, 79, 355–360. <https://doi.org/10.1016/j.procir.2019.02.087>
- Bracht, U., Geckler, D., & Wenzel, S. (2011). *Digitale Fabrik*. Springer. 01.
- Buechler, T., Schulz, M., & Zaeh, M. F. (2022). Automated geometric analysis of metallic components through picture recognition models for manufacturing technology assessments. *Production Engineering*, 17, 1–20. <https://doi.org/10.1007/s11740-022-01146-7>
- Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., & Wirth, R. (2000). *The crisp-dm user guide*. Technical report, The CRISP-DM consortium.
- Chen, Y., Chen, J., Li, C., Xu, J., & Xu, B. (2020). Research on application of digital twin in simulation of equipment maintenance support. *IOP Conference Series: Materials Science and Engineering*, 782(3), 032088. <https://doi.org/10.1088/1757-899x/782/3/032088>
- Cleveland, H. (1982). Information as a resource. *The Futurist*, 16(6), 34–39. https://doi.org/10.1007/1-4020-0613-6_8950
- Costa, C. M., Veiga, G., Sousa, A., Rocha, L., Oliveira, E., Lopes Cardoso, H., & Thomas, U. (2018). Automatic generation of disassembly sequences and exploded views from solidworks symbolic geometric relationships. *IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, 2018, 211–218. <https://doi.org/10.1109/ICARSC.2018.8374185>
- Dankwort, C. W., Weidlich, R., Guenther, B., & Blaurock, J. E. (2004). Engineers' education-it's not only cad. *Computer-Aided Design*, 36(14), 1439–1450. <https://doi.org/10.1016/j.cad.2004.02.011>
- ElMaraghy, H. A. (1993). Evolution and future perspectives of capp. *CIRP Annals*, 42(2), 739–751. [https://doi.org/10.1016/S0007-8506\(07\)62537-2](https://doi.org/10.1016/S0007-8506(07)62537-2)
- ElMaraghy, H., & Nassehi, A. (2019). Computer-aided process planning. *CIRP Encyclopedia of Production Engineering*. https://doi.org/10.1007/978-3-642-20617-7_6551
- Eversheim, W. (2002). *Organisation in der Produktionstechnik 3*. VDI-Verl.
- Gonnermann, C., Hashemi-Petroodi, E. S., Thevenin, S., Dolgui, A., & Daub, R. (2022). A skill- and feature-based approach to planning process monitoring in assembly planning. *The International Journal of Advanced Manufacturing Technology*, 122, 2645–2670. <https://doi.org/10.1007/s00170-022-09931-5>
- Gonnermann, C., Weth, J., & Reinhart, G. (2020). Skill modeling in cyber-physical production systems for process monitoring. *Procedia CIRP*, 93, 1376–1381. <https://doi.org/10.1016/j.procir.2020.03.095>
- Gonnermann, C., Zels, B., & Reinhart, G. (2021). Automatized generation of alternatives for process monitoring in cyber-physical assembly systems. *Procedia CIRP*, 104, 732–737. <https://doi.org/10.1016/j.procir.2021.11.123>
- Hammerstingl, V., & G. Reinhart. (2018). Skills in assembly. Hashemi-Petroodi, S. E., Thevenin, S., & Dolgui, A. (2022). Mixed-model assembly line design with new product variants in production generations. *IFAC-PaperOnLine*, 55(10), 25–30. <https://doi.org/10.1016/j.ifacol.2022.09.363>
- Koch, J., Michels, N., & Reinhart, G. (2016). Context model design for a process-oriented manufacturing change management. *Procedia CIRP*, 41, 33–38. <https://doi.org/10.1016/j.procir.2015.12.111>
- Kousi, N., Gkournelos, C., Aivaliotis, S., Lotsaris, K., Bavelos, A. C., Baris, P., Michalos, G., & Makris, S. (2021). Digital twin for designing and reconfiguring human-robot collaborative assembly lines. *Applied Sciences*. <https://doi.org/10.3390/app11104620>
- Kunath, M., & Winkler, H. (2018). Integrating the digital twin of the manufacturing system into a decision support system for improving the order management process. *Procedia CIRP*, 72, 225–231. <https://doi.org/10.1016/j.procir.2018.03.192>
- Liu, J., Zhou, H., Tian, G., Liu, X., & Jing, X. (2019). Digital twin-based process reuse and evaluation approach for smart process planning. *The International Journal of Advanced Manufacturing Technology*, 100(5), 1619–1634. <https://doi.org/10.1007/s00170-018-2748-5>
- Lu, Y., Liu, C., Wang, K. I. K., Huang, H., & Xu, X. (2020). Digital twin-driven smart manufacturing: Connotation, reference model, applications and research issues. *Robotics and Computer-Integrated Manufacturing*, 61, 101837. <https://doi.org/10.1016/j.rcim.2019.101837>
- Ma, J., Chen, H., Zhang, Y., Guo, H., Ren, Y., Mo, R., & Liu, L. (2020). A digital twin-driven production management system for production workshop. *The International Journal of Advanced Manufacturing Technology*, 110, 1385–1397. <https://doi.org/10.1007/s00170-020-05977-5>
- Mamdani, E. H. (1974). Application of fuzzy algorithms for control of simple dynamic plant. *Proceedings of the Institution of Electrical Engineers*, 121(12), 1585–1588. <https://doi.org/10.1049/PiEE.1974.0328>
- Michniewicz, J. (2019). *Automatische simulationsgestützte Arbeitsplanung in der Montage*. Technische Universität München.
- Negri, E., Fumagalli, L., & Macchi, M. (2017). A review of the roles of digital twin in cps-based production systems. *Procedia Manufacturing*, 11, 939–948. <https://doi.org/10.1016/j.promfg.2017.07.198>
- Obrst, L. (2010). Ontological architectures. In Poli, R., M. Healy, A. Kameas (eds.), *Theory and Applications of Ontology: Computer Applications*. p. 27–66. Springer Netherlands. https://doi.org/10.1007/978-90-481-8847-5_2
- Pfrommer, J., Schleipen, M., & Beyer, J. (2013). Pprs: Production skills and their relation to product, process, and resource. *2013 IEEE 18th Conference on Emerging Technologies & Factory Automation (ETFA)*. p. 1–4. <https://doi.org/10.1109/ETFA.2013.6648114>
- Poli, R., Healy, M., & Kameas, A. (2010). *Theory and Applications of Ontology: Computer Applications*. Netherlands: Springer.
- Rowley, J. (2007). The wisdom hierarchy: representations of the dikw hierarchy. *Journal of Information Science*, 33(2), 163–180. <https://doi.org/10.1177/0165551506070706>
- Schleipen, M., & Drath, R. (2009). Three-view-concept for modeling process or manufacturing plants with automationml. *IEEE Conference on Emerging Technologies and Factory Automation*, 2009, 1–4. <https://doi.org/10.1109/ETFA.2009.5347260>
- Schuh, G., & Gierth, A. (2012). Aachener pps-modell. In Schuh, G., V. Stich (eds.), *Produktionsplanung und -steuerung 1* (4. Aufl. 2012 ed.). VDI-Buch. Springer. <https://doi.org/10.1007/978-3-642-25423-9>
- Schuh, G., Prote, J. P., & Hünnekes, P. (2020). Data mining methods for macro level process planning. *Procedia CIRP*, 88, 48–53. <https://doi.org/10.1016/j.procir.2020.05.009>

- Shakirov, E. F., Kattner, N., Fortin, C., Uzhinsky, I. K., & Lindemann, U. (2021). Reducing the uncertainty in engineering change management using historical data and simulation modelling: a process twin concept. *International Journal of Product Lifecycle Management*, 13, 89–114. <https://doi.org/10.1504/IJPLM.2021.10037271>
- Talkhestani, B.A., Jung, T., Lindemann, B., Sahlab, N., Jazdi, N., Schloegl, W., & Weyrich, M. (2019). An architecture of an intelligent digital twin in a cyber-physical production system. *at - Automatisierungstechnik* 67(9): 762–782. <https://doi.org/10.1515/auto-2019-0039>.
- Tao, F., Cheng, J., Qi, Q., Zhang, M., Zhang, H., & Sui, F. (2018). Digital twin-driven product design, manufacturing and service with big data. *The International Journal of Advanced Manufacturing Technology*, 94(9–12), 3563–3576. <https://doi.org/10.1007/s00170-017-0233-1>
- Uzunosmanoglu, M., Raa, B., Limère, V., Cock, A.D., Singh, Y., Lopez, A.J., Gautama, S., & Cottyn, J. (2021). Aggregate planning for multi-product assembly lines with reconfigurable cells. In A. Dolgui, A. Bernard, D. Lemoine, G. von Cieminski, and D. Romero (Eds.), *Advances in Production Management Systems. Artificial Intelligence for Sustainable and Resilient Production Systems - IFIP WG 5.7 International Conference, APMS 2021, Nantes, France, September 5–9, 2021, Proceedings, Part II*, Volume 631 of *IFIP Advances in Information and Communication Technology*, pp. 525–534. Springer.
- Vishnu, V., Varghese, K. G., & Gurumoorthy, B. (2021). A data-driven digital twin of cnc machining processes for predicting surface roughness. *Procedia CIRP*, 104, 1065–1070. <https://doi.org/10.1016/j.procir.2021.11.179>
- Wagner, S., Milde, M., Barhebwa-Mushamuka, F., & Reinhart, G. (2021). Digital twin design in production. In A.-L. Andersen, R. Andersen, T. D. Brunoe, M. S. S. Larsen, K. Nielsen, A. Napoleone, & S. Kjeldgaard (Eds.), *Towards Sustainable Customization: Bridging Smart Products and Manufacturing Systems* (pp. 339–346). Springer International Publishing.
- Wagner, S., Milde, M., & Reinhart, G. (2021). The digital twin in order processing. *Procedia CIRP*, 104, 863–868. <https://doi.org/10.1016/j.procir.2021.11.145>
- Wang, Y., & Wu, Z. (2020). Model construction of planning and scheduling system based on digital twin. *The International Journal of Advanced Manufacturing Technology*, 109, 2189–2203.
- Zedlitz, J. (2013). *Konzeptuelle Modellierung mit UML und OWL – Untersuchung der Gemeinsamkeiten und Unterschiede mit Hilfe von Modelltransformationen*. Zedlitz, Jesper. Zedlitz, Jesper (Verfasser).
- Zhang, H., Liu, Q., Chen, X., Zhang, D., & Leng, J. (2017). A digital twin-based approach for designing and multi-objective optimization of hollow glass production line. *IEEE Access*, 5, 26901–26911. <https://doi.org/10.1109/ACCESS.2017.2766453>
- Zhang, H., Zhang, G., & Yan, Q. (2018). Digital twin-driven cyber-physical production system towards smart shop-floor. *Journal of Ambient Intelligence and Humanized Computing*, 10, 4439–4453. <https://doi.org/10.1007/s12652-018-1125-4>
- Zhao, P., Liu, J., Jing, X., Tang, M., Sheng, S., Zhou, H., & Liu, Y. (2020). The modeling and using strategy for the digital twin in process planning. *IEEE Access*, 8, 41229–41245. <https://doi.org/10.1109/ACCESS.2020.2974241>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.