Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 120 (2023) 177–182

www.elsevier.com/locate/procedia

56th CIRP Conference on Manufacturing Systems, CIRP CMS '23, South Africa

Integrating Multiple Perspectives in Manufacturing Planning and Control: The Daydreaming Engine Approach

Martin Benfer[a]*, Oliver Brützel[a], Leonard Overbeck[a], Sina Peukert[a], Aydin Nassehi[b], Gisela Lanza[a]

[a]wbk Institute of Production Science, Karlsruhe Institute of Technology (KIT), Kaiserstraße 12, Karlsruhe 76131, Germany
[b]University of Bristol, Bristol BS8 1TR, United Kingdom

* Corresponding author. *E-mail address:* martin.benfer@kit.edu

Abstract

In a world that requires swift responses to a volatile environment, companies need to integrate multiple perspectives seamlessly into their planning processes. While there has been tremendous progress in model-based manufacturing planning and control, both in terms of methods and computing capabilities, the knowledge in many companies remains fractured within their functional departments. This fracture can, for example, lead to production network planning being oblivious to information regarding potential sales development or mismatches in order allocation and logistics planning. These issues may be attributed to the specificity of different heuristics, simulation, optimisation and metamodels and missing avenues to transmit information beyond a singular planned configuration. To overcome this issue, this contribution presents the concept of a daydreaming engine, permits and encourages interaction between different models of manufacturing systems. In a daydreaming engine, the models can request information from each other on-demand, external changes can trigger automatic replanning, and users can utilise daydreaming models to create and explore even unlikely scenarios automatically. Using the daydreaming framework, users can determine possible reactions to scenarios and prepare factors necessary for a quick reaction. The components and modes of interaction within this engine are discussed in detail, and the first steps towards an industrial application are presented. This paper contributes to intelligently connecting physical and virtual production environments using various quantitative models and expertise from several domains.

Keywords: Daydreaming; Modeling; Digital Twin; Production Systems; Model Interaction;

1. Motivation

Manufacturing planning and control comprises all activities associated with ordering and allocating tasks in industrial value creation processes to available resources and acquiring and configuring those resources. To manage the complexity of manufacturing planning and control, companies have created different functional departments concentrated on specific tasks like logistics planning and control, production network planning, layout planning and order scheduling. In past decades, digital models have increasingly been used to make decisions in these complex tasks [1, 2]. More recently, these models are being connected to live data sources, to provide faster planning and address the volatility of today's world [3]. These models that are continuously synchronised with the system they represent and used for decision-making in those systems are called digital twins [4]. While developing digital twins in manufacturing planning and control is promising, two main challenges are not yet fully addressed. First, even though stochastic and scenario-based models exist, the uncertainty regarding future developments in real life scenarios is challenging to capture [5]. Second, many tasks are ambiguous and can be understood differently depending on the perspective [6]. While different functional departments in producing

10.1016/j.procir.2023.08.032

companies allow the consideration of several perspectives, the interaction between those perspectives is often limited to the exchange of singular agreed-upon decisions and not the full scope of possibilities considered in the planning tasks.

To overcome these issues, this contribution proposes the concept of the daydreaming engine, which enables the intelligent interaction of multiple model-based decision-making processes. The concept of the daydreaming factory, defined as "using […] a digitised representation of a system based on engineering models, conducting reveries by generating multiple scenarios and gathering insights from the potential outcomes", is incorporated to address planning and learning in uncertain situations [7]. A system of interconnected models must acknowledge different interactions between organisational functions. Additionally, it must be compatible with a continuously developing landscape of decision support tools. Therefore, this contribution seeks to answer the following research questions:

1) How can multiple models owned by different parts of an organisation be integrated to enable cross-functional information exchange?
2) How can such models interact flexibly to reflect the nature of interactions between different parts of organisations throughout their process landscape?
3) How can a system of models be implemented in large organisations while prioritising utility?

The rest of this contribution is structured as follows. Section 2 provides an overview of relevant related work. Section 3 details the daydreaming engine concept. Section 4 provides an overview of an ongoing industrial application and discusses specific use cases. Finally, section 5 provides a conclusion and an outlook to future research.

2. Related Work

Quantitative models already have a long history of application in industrial production. Such models are always a simplification of a specific system designed for a defined purpose. According to the expressivity of their results, they can be categorised as descriptive, analytical, predictive and prescriptive models [8]. For planning and control of production systems, descriptive models are used to structure information, for example, as dashboards or data models [9]. Analytical models link system behaviour to external influences and educate the decision-maker [10, 11]. There are many applications for predictive models. Such models are used to plan factory layouts, determine bottlenecks, test control algorithms, and guide investment decisions in new lines [1]. On a network level, they can assess risks, forecast demands and plan networks. The most common implementation techniques are discrete event simulation and agent-based simulation, but many other specific techniques exist. Prescriptive models predict future behaviour and provide a preferred decision alternative. They are used for problems where the decision situation can be entirely formalised, such as scheduling decisions, route planning within and between plants, allocation of production orders, and even investments [2].

The issue of enabling more intelligent decision-making using models has been at the forefront of several research streams. The cyber-physical production system describes a sensor-equipped system that interacts continuously with a range of software systems and, for example, supports scheduling [10]. Digital twins have received much attention and describe the comprehensive model-based representation of systems synchronised with their physical counterparts [4, 12]. In some research, the term digital twin is also extended into model-based control systems, i.e., systems where the makeup of the physical counterpart is automatically shaped by the digital model [3]. Such systems have been realised mostly on a machine or line-specific level, though concepts for use in entire supply chains exist [13]. In research on managerial decision-making, a focus has been on providing collaborative and distributed decision-making [14]. Different decision support systems have been proposed, even with a modular structure, that allow the implementation of multiple different models [8, 15]. Finally, the term daydreaming factories was introduced to refer to intelligent multi-model systems that continuously use dormant computing capabilities to improve production systems [7]. Previous research has studied the interaction between different models [16, 17], though that research has been primarily descriptive, focused on specific types of models, and not on the realisation of such interconnected systems. Overall, while several approaches to improve decision making in manufacturing planning and control exist, a methodology to connect multiple different models intelligently is still missing.

3. A Daydreaming Engine Concept

The daydreaming engine is a system to coordinate the interaction between different decision-support tools within an organisation. It interacts with multiple models described in section 3.1 and consists of a database detailed in section 3.2 and multiple modules, as shown in Figure 1. It allows for different model interaction types discussed in section 3.3. This daydreaming engine is not a monolithic multipurpose solution; thus, implementation will likely occur in a brownfield scenario. Section 3.4 lays out the implementation procedure, depending on the starting point of a particular organisation.

The first module is a *database* containing data from the production network characterising the system's current state and modelling results characterising different possible scenarios. It is described in more detail in section 3.2.

The *engine controller* represents the centre of the daydreaming engine. It monitors changes in the database and interacts with the models through the pub/sub-event handler and the request-response handler. It also interacts with the meta-modelling module and the calibration engine. The main task of the controller is to prioritise and schedule different model experiments based on the available computational capacity as well as the urgency of the task.

The *pub/sub-event handler* facilitates model interaction using the publish/subscribe principle by raising change events and notifying the subscribed models. The *request-response handler* allows calling specific model instances in specific settings to gather cross-functional knowledge. Both modules are discussed in more detail in section 3.3.

Metamodelling, or surrogate modelling, describes using simplified, computationally efficient models to approximate the results of more complex models [18]. Those models can allow for the timely provision of results for specific inputs and the creation of efficient prescriptive models from predictive models [18]. In the Daydreaming Engine, the *meta-modelling module* coordinates the training of metamodels based on specific optimisation or simulation models.

The calibration and validation of models is crucial for effective model-based decision-making. In the daydreaming engine, this task is managed by the *calibration engine*, which ensures that modelling results match reality as closely as possible. The calibration engine monitors whether the model results match the results of the real system or other more detailed models. It uses specific interfaces for different models and types of calibration data to provide on-demand calibration services. When specific validity thresholds are defined, it may demand calibration of specific models, thus separating the responsibility for model use and validity checking.

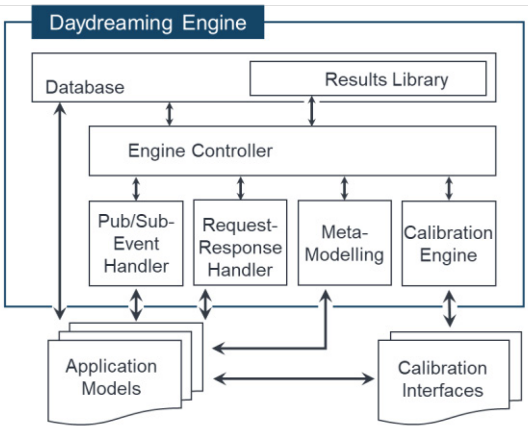

Figure 1: Architecture of the Daydreaming Engine

3.1. Models & Perspectives

The models considered in this architecture are predictive and prescriptive. A predictive model f_{pred} transforms a set of parameters, describing the configuration $P_{C,s}$, of the system s the state of the environment $P_{E,s}$ and model parameters $P_{M,s}$ into the predicted system behaviour, which consists of the system's predicted state z_s and relevant outputs y_s:

$$f_{pred}(P_{C,s}, P_{E,s}, P_{M,s}) = (z_s, y_s) \quad (1)$$

A prescriptive model performs a transformation f_{prescr} on the set of fixed system parameters $P'_{C,s}$, the state of the environment $P_{E,s}$, model parameters $P_{M,s}$ and desired outcomes $d(Y)$, returning the selected set of system state parameters $P^*_{C,s}$ and the associated system behaviour (z^*_s, y^*_s):

$$f_{prescr}(P'_{C,s}, P_{E,s}, P_{M,s}, d(Y)) = (P^*_{C,s}, z^*_s, y^*_s) \quad (2)$$

Each of the models is assumed to be a partial digital twin, i.e. a subset of the configuration parameters $P_{C,A,s} \subset P_{C,s}$ and a subset of environment parameters $P_{E,A,s} \subset P_{E,s}$ is determined automatically absent input from the user. The parameter sets provided manually by the users are $P_{C,M,s} \subset P_{C,s}$ and $P_{E,M,s} \subset P_{E,s}$ respectively. The considered models are 'owned' by one department or function χ of an organisation. For example, logistics optimisation models may be owned by the logistics department. That department is the primary user of the model. Any type of model μ may have multiple instances m for each system s and time-dependent versions v_s of that system.

The proposed concept concentrates on models used for planning purposes in production systems. Relevant departments comprise all organisational functions concerned with configuring production lines, systems and networks and allocating production volumes to those systems. Different planning tasks can occur within those system levels, which consider different system aspects at various levels of abstraction and are measured using a range of KPIs [19, 20]. Depending on the organisation using the concept, the tasks may be concentrated in a few or split among multiple departments.

3.2. Database

One of the challenges of digital twins of production systems is the connection of relevant data sources within the organisation [8]. In contrast to digital twins of single products or resources, where sensors can be more or less directly integrated with a digital model, this approach is impractical for production systems, as the necessary update frequency of the models is much lower, and the number of different direct data sources that would have to be connected is prohibitive. Instead, data can be acquired from information systems such as MES, ERP, CRM and SCM. Often additional data sources for master data are necessary. Furthermore, data may be preprocessed for use in models and stored as files. Model results and plans often only exist as files accessible to specific users.

A shared data model for planning tasks in production systems is used to overcome this problem, as proposed in [8]. The data model represents production networks and their characteristics in an object-oriented form.

This generic model can be adapted to fit the specific requirements of an employing organisation. In the daydreaming engine, the data model is instantiated by defining one or multiple data sources providing information for each object type ω. Each model type μ defines which data it accesses automatically as part of the automatic data set for a model $P_{A,m}$ using an allocation function a_μ that also lets the users define the examines system s, the considered starting time t_0 and the scenario γ.

$$P_{A,m} = a_\mu(s_m, t_{0,m}, \gamma_m) \quad (3)$$

In the database, each object o may be different based on the considered time and scenario. This allows the database to store information regarding past system states that the models can use for calibration.

Each property value $v_{a_o,t,\gamma,u}$ in the database is thus specific to the property ϕ_o of the object it belongs to, the time t it is valid at, the scenario γ, and the update u it was set by. With any update u, the updating time t_u and the information source i_u are saved. Data preprocessing is done during updating and can be defined individually for any information source. The information sources can be classic information systems and results from models. The database thus includes a results library for the different models.

For past system states one "real" scenario γ_0 exists, but for future states, many different scenarios can occur. The different models can create new scenarios. Each new scenario is characterised by a parent scenario ψ_γ, a time of deviation $t_{D,\gamma}$, and a likelihood of occurrence l_γ, with

$$l_{\hat{\gamma}} \geq \sum_{\gamma \in R, \psi_\gamma = \hat{\gamma}} (l_\gamma) \quad (4)$$

where R denotes a consistent scenario space owned by a particular department or function χ. The creation of different scenario spaces allows the departments to plan confidentially and ensures the viability of planning on very different time horizons. Scenario spaces can start at present or from a point within an existing scenario and diverge from there. Users can specify the likelihood of occurrence for scenarios or set it to 0. Any scenario γ defined by a specific model m is typically only partial, i.e., information on objects and associated properties is missing. Thus, the engine controller seeks to increase the scenario's completeness, prioritising scenarios with a high temporal relevance ρ_γ expressed as

$$\rho_\gamma = \frac{l_\gamma}{t_{D,\gamma} - t} \quad (5)$$

As indicated before, the database is also used for calibration. Calibration describes the adjustment of inner model parameters $p_{M,s,m}$ to more closely resemble the behaviour of the system s. For each type of parameter, different methods can be used. Thus, a calibration interface defining a calibration function $c((z_s, y_s)_\mu, (z_s, y_s)_{cal})$ can be designed for each parameter. The calibration engine can then either provide historic result data $(z_s, y_s)_{cal}$ or call other models or expert inputs as a reference. It does so by using the capacity-based model initialisation presented in section 3.3.

Finally, the database contains sensitive information that is confidential only to parts of an organisation. Therefore, the data model is implemented with access management. Access management is included at several levels. Specific scenario spaces, scenarios, object types, objects, and even properties can require access rights from either models or users so that sensitive information is only passed to users with permission.

3.3. Model Initialisation

A core aspect of the interaction between different models is the initialisation of model-based experiments. Conventionally, experiments are triggered by the model user, either as a single run or for a specified parameter range. If the models are partial digital twins. However, in digital twins, where the models are synchronised with the represented system, users specify parameters not defined by the synchronisation and overwrite parameters defined by synchronisation to explore alternative actions. This is the usual model initialisation mode for planning production systems. Within a daydreaming engine, however, other modes of initialisation are possible. These modes are *request, event*, or *capacity* based, as shown in Figure 2.

In model request-response initialisation, whenever a property value is requested where a model m is registered as a data source, and the value does not yet exist for the required scenario γ or the current update u, the request-response handler queries the registered model to provide it. The model m may also specify parameters sets $[P_C, P_E, P_M]_{m,\gamma}$ necessary for the desired result. In the case of *fully automatic provision*, the entire information necessary for the experiments must be provided to the model either by the requesting entity or the database. In *consensual data provision*, a request is put forward to the model owner, who must approve the request before the model is executed. *Semiautomatic requests* require additional data input from the model owner before data can be provided.

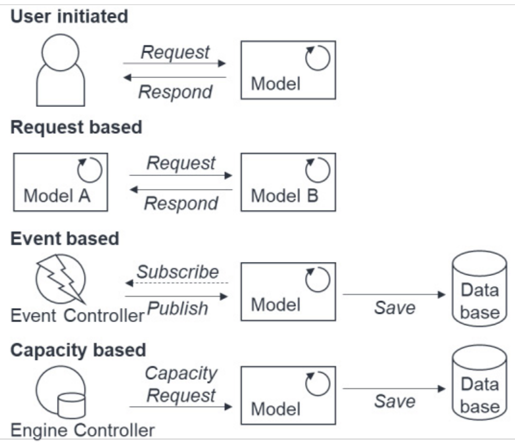

Figure 2: Model Initialisation Modes

Building on the previously described data model, models can also be initiated based on change events ϵ raised by the pub/sub event handler:

$$\epsilon = \left(o, \delta, \gamma, P_{\delta,o}^{old}, P_{\delta,o}^{new}\right) \quad (6)$$

where o is the changed object, δ the type of change, γ the scenario which has changed, and $P_{\delta,o}^{old}$ and $P_{\delta,o}^{new}$ the set of parameter values before and after the change. Models may subscribe to events using a subscription rule σ

$$\sigma = (w_O, \Delta, \Gamma) \quad (7)$$

where w_O is a function to determine the set of considered objects O, Δ is the set of considered changes and Γ the set of scenarios. The model may then be executed with a predetermined set of parameters. This way, models can react to changes in the configuration of the production system, changes in the system's environment, or even changes to future configurations γ based on new results from other models. Different subscription types are possible depending on the users' preferences and the model's availability outside of active usage. *A fully automated subscription* starts model runs immediately upon the published event and returns the results to the database while merely notifying the user. *Consentuable automation* automatically performs specified experiments but requires authorisation to publish the results to the database. *Suggestive automation* proposes experiments to users but requires additional data input or agreement to run experiments.

Lastly, capacity-based initialisation operationalises the concept of daydreaming to use the dormant computational capacity to expand the organisation's knowledge. This is only possible with models that can be queried on demand, running on a server. Knowledge can be expanded through a more comprehensive database or the improvement of models. The engine controller pursues the former, directly expanding and completing the space of examined scenarios described in section 3.2. Furthermore, the engine controller can even expand the number of considered scenarios. An owner may specify a part $\widetilde{\Pi_{E,\mu}}$ of the models parameterspace $\Pi_{E,\mu}$ as open for experiments and define limits and likely distributions for each parameter p_E in $\widetilde{\Pi_{E,\mu}}$. The distributions are then used to either explore likely scenarios or examine unlikely extremes using the domain randomisation method described in [7]. The engine controller can also schedule model runs to support the creation of specific metamodels, following a fitting experimental design approach within the parameter space [18]. Finally, additional capacity may be used to better calibrate models.

3.4. Implementation

The devised daydreaming engine interfaces many models and users, making implementation in one step challenging. Instead, an iterative implementation is proposed, utilising the modularity of the concept. Thus, benefits for the company can be realised at every implementation step. The implementation process encompasses three general streams (i) application development, (ii) data integration, and (iii) support functionality development. The first stream to start is application development, as they provide value even without the daydreaming engine and help integrate enthusiastic users into the development [21]. Subsequently, the second stream is started, implementing the database and connecting the application models. This turns models into digital twins and reduces the effort for model use, making more use cases for the application model desirable. Finally, the modules of the daydreaming engine are implemented as support functionality.

4. Industrial Application & Use Cases

The daydreaming engine is currently being implemented at an automotive supplier. At the time of writing, three applications have been developed, a discrete event simulation (DES) model of the localised production system described in [12], a mixed integer optimisation model for order allocation and investment planning [22], and a capacity coordination tool managing investments for the sites. Additionally, a commercial logistics tool has been connected to the other models.

The simulation model has been developed into a digital twin by integrating multiple live data sources, for example, the company's MES systems. The model contains several production lines producing a common family of products. The model is used for weekly production planning, optimising worker allocation, order scheduling, and bottleneck detection.

The optimisation model is employed in the yearly planning cycle for each product family. It determines the cost-optimal allocation of orders and decisions on line purchases, upgrades for production capacity and capability, production releases, and shift models. The model requires data input from planners, such as the forecasted demand, product features, line properties and capabilities, transport relations, and contractual site-specific allocation premises. Finally, estimates for line purchase, upgrade, and release costs must be given.

The capacity coordination tool is used to monitor capacity at the sites of the overall network, combining planning results from multiple product-family-specific production networks. It tracks cumulated space, employee, and investment requirements and matches them with the existing and planned capacity to coordinate the planning across the networks.

The commercial logistics tool determines ideal delivery routes for both inbound and outbound transport of goods. It integrates warehouses, multiple modes of transport and customs and duties into a heuristic cost optimisation.

Several use cases were developed to demonstrate the utility of the proposed daydreaming engine that take advantage of the opportunities only available by intelligently connecting multiple tools from different perspectives. These use cases were designed in the context of the previously described industrial application and use models presented there.

4.1. Requested Information – Line Upgrade

This use case involves two models, the discrete event simulation model of local production systems consisting of multiple lines and the mixed integer linear optimisation model for allocating production orders and investment planning. An essential aspect of the optimisation model is the decision to upgrade lines depending on demand and required production capabilities. The planers typically estimate this data and, thus, part of the manual configuration parameter set $P_{C,M}{}'$. By utilising the existing DES of a site-specific production system, the consequences of the upgrade for the resulting line performance can be estimated more accurately. In this case, the optimisation model first identifies a need for more detailed information as it plans the upgraded line and requests data on the new production line, for which the simulation model is registered as a source. The optimisation model specifies the planned order program, the number of available workers and the upgrade. The simulation then provides the resulting performance of the lines based on a set of experiments. With this updated data, the optimisation model is rerun.

Similar interaction modes are used between the allocation optimisation and logistics planning models. For example, the allocation model requests logistics costs from the route planning tool. On the other hand, the capacity monitoring tool requests production costs from the optimisation model.

4.2. External Event – Change in Order Backlog

Various disruptions in global supply chains have shown that external events strongly influence the performance of production systems, and one core aspect of production systems resilience is sensitivity, the ability to replan quickly in the face of changes [23]. For example, sudden changes in the order backlog can drastically alter the optimal allocation plan. To overcome this, the order allocation model is subscribed to the data point orders, specifically orders produced within the model's production network. Upon changes in these orders, the daydreaming engine first checks whether existing scenarios anticipate the change. If not, a data change event is thrown, and the allocation model is executed using the new order backlog. Finally, the results are directly forwarded to the particular owner of the optimisation model to assess whether the action in the real world needs to be taken.

This type of interaction can also be used with different models, for example, changes in available employee capacity in the simulation model or for different parameters, such as changes in energy costs at a particular site.

4.3. Daydreaming –Unlikely Demand Scenarios

Changes in demand significantly impact production planning. The capacity-based daydreaming approach can be used to explore such scenarios. For this purpose, the users mark a selection of the parameter space of the optimisation model $\widetilde{\Pi_E} \subset \Pi_E$ for daydreaming and define a probability density functions for those parameters that encourage edge cases. The daydreaming engine controller then creates new parameter sets $\widetilde{P_E} \in \widetilde{\Pi_E}$ and runs experiments on the optimisation model. The results are saved as specific scenarios. The capacity-based initialisation allows those experiments to be run in model downtimes. The results are explored for noteworthy characteristics, like unsolvability, sudden changes in production costs, or site underutilisation. The limits of the current production networks can be found by connecting those results to the corresponding areas of the parameter space.

5. Conclusion, Discussion & Outlook

This paper presents an approach to integrate multiple functional perspectives through the decentralised interaction of model-based decision support systems in manufacturing planning and control. A systematisation of model interaction modes is presented with an iterative implementation process involving different parts of the organisation. Finally, the first steps towards industrial implementation are shown. The daydreaming engine may make producing companies more resilient by allowing them to quickly plan reactions to changes and understand challenges holistically using perspectives from the entire organisation.

Although the approach is promising, there remain some challenges. This decentralised approach requires large efforts to standardise data exchange formats and define interfaces. Although knowledge can be shared selectively, its protection is a difficult task in such complex systems, and care has to be taken to avoid possible model-based access exploits. Other challenges may arise due to the availability of plans in the form of data that their owner may not contextualise. Furthermore, the implementation in large organisations can become inefficient as the number of stakeholders in a system such as the proposed one grow. Even still, integrating a broad range of different users into a shared system is demanding. An alternative to the presented approach would be to generate a monolithic decision support software that can consider all the different tasks. However, that would limit the specialisation of models and reduce the plurality of perspectives on planning and control problems in manufacturing.

As the presented daydreaming engine is not yet fully implemented, future research should investigate under which circumstances the concept is most suitable. The most promising path to evaluate the concept will be action research.

Acknowledgements

This research was funded by the German Federal Ministry for Economic Affairs and Climate Action (BMWK) through the research project 13IK001ZF "Software-Defined Manufacturing for the automotive and supplying industry”.

References

[1] Jeon, S.M., Kim, G., 2016. A survey of simulation modeling techniques in production planning and control (PPC). Production Planning & Control *27*, p. 360.

[2] Díaz-Madroñero, M., Mula, J., Peidro, D., 2014. A review of discrete-time optimization models for tactical production planning. International Journal of Production Research *52*, p. 5171.

[3] Kritzinger, W., Karner, M., Traar, G., Henjes, J. *et al.*, 2018. Digital Twin in manufacturing: A categorical literature review and classification. IFAC-PapersOnLine *51*, p. 1016.

[4] Stark, R., Fresemann, C., Lindow, K., 2019. Development and operation of Digital Twins for technical systems and services. CIRP Annals *68*, p. 129.

[5] Govindan, K., Fattahi, M., Keyvanshokooh, E., 2017. Supply chain network design under uncertainty: A comprehensive review and future research directions. European Journal of Operational Research *263*, p. 108.

[6] Gunessee, S., Subramanian, N., 2020. Ambiguity and its coping mechanisms in supply chains lessons from the Covid-19 pandemic and natural disasters. IJOPM *40*, p. 1201.

[7] Nassehi, A., Colledani, M., Kádár, B., Lutters, E., 2022. Daydreaming factories. CIRP Annals *71*, p. 671.

[8] Benfer, M., Peukert, S., Lanza, G., 2021. A Framework for Digital Twins for Production Network Management. Procedia CIRP *104*, p. 1269.

[9] Tokola, H., Gröger, C., Järvenpää, E., Niemi, E., 2016. Designing Manufacturing Dashboards on the Basis of a Key Performance Indicator Survey. Procedia CIRP *57*, p. 619.

[10] Schuh, A.G., Stich, V., Reuter, C., Blum, M. *et al.*, 2017. Cyber Physical Production Control, in *Industrial Internet of Things*, Springer International Publishing, Cham, p. 519.

[11] Eom, S., 2020. DSS, BI, and Data Analytics Research: Current State and Emerging Trends (2015–2019), in *Decision Support Systems X: Cognitive Decision Support Systems and Technologies*, Springer International Publishing, Cham, p. 167.

[12] Overbeck, L., Brützel, O., Teufel, M., Stricker, N. *et al.*, 2021. Continuous adaption through real data analysis turn simulation models into digital twins. Procedia CIRP *104*, p. 98.

[13] Ivanov, D., Dolgui, A., 2021. A digital supply chain twin for managing the disruption risks and resilience in the era of Industry 4.0. Production Planning & Control *32*, p. 775.

[14] Pienaar, S., Srinivasan, A., Sundaram, D., 2007. Support for Collaborative and Distributed Decision Making, in *Decision Support for Global Enterprises*, Springer US, Boston, MA, p. 51.

[15] Chandra, C., Grabis, J., 2016. *Supply Chain Configuration: Concepts, Solutions, and Applications,* 2nd edn. Springer New York, New York, NY, s.l.

[16] Morgan, J.S., Howick, S., Belton, V., 2017. A toolkit of designs for mixing Discrete Event Simulation and System Dynamics. European Journal of Operational Research *257*, p. 907.

[17] Brailsford, S.C., Eldabi, T., Kunc, M., Mustafee, N. *et al.*, 2019. Hybrid simulation modelling in operational research: A state-of-the-art review. European Journal of Operational Research *278*, p. 721.

[18] Li, Y.F., Ng, S.H., Xie, M., Goh, T.N., 2010. A systematic comparison of metamodeling techniques for simulation optimization in Decision Support Systems. Applied Soft Computing *10*, p. 1257.

[19] Benfer, M., Ziegler, M., Gützlaff, A., Fränken, B. *et al.*, 2019. Determination of the abstraction level in production network models. Procedia CIRP *81*, p. 198.

[20] Stricker, N., Echsler Minguillon, F., Lanza, G., 2017. Selecting key performance indicators for production with a linear programming approach. International Journal of Production Research *55*, p. 5537.

[21] Robertson, T.S., 1967. The Process of Innovation and the Diffusion of Innovation. Journal of Marketing *31*, p. 14.

[22] Brützel, O., Küppers, F., Overbeck, L., Stricker, N. *et al.*, 2021. Eine automatisierungsgerechte robuste Produktionsplanung. Zeitschrift für wirtschaftlichen Fabrikbetrieb *116*, p. 44.

[23] Benfer, M., Verhaelen, B., Peukert, S., Lanza, G., 2021. Resilience Measures in Global Production Networks: A Literature Review and Conceptual Framework. DU *75*, p. 491.