A Conceptual Model for Digital Shadows in Industry and Its Application

Fabian Becker[1], Pascal Bibow[1], Manuela Dalibor[2], Aymen Gannouni[3],
Viviane Hahn[4], Christian Hopmann[1], Matthias Jarke[5,6], István Koren[7],
Moritz Kröger[8], Johannes Lipp[5,6], Judith Maibaum[4], Judith Michael[2(✉)],
Bernhard Rumpe[2], Patrick Sapel[1], Niklas Schäfer[4], Georg J. Schmitz[9],
Günther Schuh[4], and Andreas Wortmann[10]

[1] IKV - Institute for Plastics Processing, RWTH Aachen University,
Aachen, Germany
{fabian.becker.sg,pascal.bibow,patrick.sapel}@ikv.rwth-aachen.de
[2] Software Engineering, RWTH Aachen University, Aachen, Germany
{dalibor,michael,rumpe}@se-rwth.de
[3] Chair of Information Management in Mechanical Engineering,
RWTH Aachen University, Aachen, Germany
aymen.gannouni@ima.rwth-aachen.de
[4] Laboratory for Machine Tools and Production Engineering (WZL),
RWTH Aachen University, Aachen, Germany
{v.hahn,j.maibaum,n.schaefer}@wzl.rwth-aachen.de
[5] Chair of Information Systems and Databases, RWTH Aachen University,
Aachen, Germany
{jarke,lipp}@dbis.rwth-aachen.de
[6] Fraunhofer Institute for Applied Information Technology FIT,
Sankt Augustin, Germany
[7] Chair of Process and Data Science, RWTH Aachen University, Aachen, Germany
koren@pads.rwth-aachen.de
[8] Chair for Laser Technology, RWTH Aachen University, Aachen, Germany
moritz.kroeger@llt.rwth-aachen.de
[9] ACCESS e.V. at the RWTH Aachen, Aachen, Germany
g.j.schmitz@access-technology.de
[10] Institute for Control Engineering of Machine Tools and Manufacturing Units
(ISW), University of Stuttgart, Stuttgart, Germany
wortmann@isw.uni-stuttgart.de

Abstract. Smart manufacturing demands to process data in domain-specific real-time. Engineering models created for constructing, commissioning, planning, or simulating manufacturing systems can facilitate aggregating and abstracting the wealth of manufacturing data to faster processable data structures for more timely decision making. Current research lacks conceptual foundations for how data and engineering models can be exploited in an integrated way to achieve this. Such research demands expertise from different smart manufacturing domains to harmonize the notion space. We propose a conceptual model to describe digital shadows, data structures tailored to exploit models and data in smart manufacturing, through a metamodel and its notion space. This conceptual model was established through interdisciplinary research in

the German excellence cluster "Internet of Production" and evaluated in various real-world manufacturing scenarios. This foundation for an understanding helps to manage complexity, automated analyses, and syntheses, and, ultimately, facilitates cross-domain collaboration.

Keywords: Digital shadow · Conceptual model · Smart manufacturing · Internet of production

1   Introduction

Digital transformation shapes our world: from communication, to transportation, to medicine, to manufacturing, documents and processes become digital to enable smarter data analyses, better integration of stakeholders, and more efficient automated information processing. In manufacturing, long-living cyber-physical production systems produce tremendous data that can be exploited to reduce downtime, consumption of resources, and increase manufacturing agility towards lot-size one [29]. Analyzing and processing this data fast enough demands its appropriate abstraction and meaningful aggregation. Current manufacturing environments leverages different and redundant IT-system silos comprising domain-specific data and models [23]. Therefore, access, and analysis of production data is difficult [21]. Often, models of the involved systems and processes (*e.g.,* structure, behavior, knowledge) can be exploited to give meaning to that data and provide a foundation for adaptive Digital Twins (DTs), the digital representations of a cyber-physical system.

Research has produced a vast number of publications on DTs in manufacturing [9,25,30]. These DTs are conceived ad-hoc, for specific real-world applications and related data structures, *e.g.,* for CNC machining [15], injection molding [12], monitoring [6], or fatigue testing [7]. There is *no conceptual foundation* for describing, abstracting, aggregating, and relating the data shared between DTs. To mitigate this, we have conceived a conceptual model [16] of Digital Shadows (DSs), describing data structures capturing the quintessential concepts of manufacturing processes: data-traces including data-points and metadata, different kinds of engineering models, data-sources, and related assets as well as purposes for creating DSs. This model has been defined and refined through a series of interdisciplinary workshops in the Internet of Production (IoP)[1] cluster of excellence and evaluated in various manufacturing scenarios, *e.g.,* injection molding [3], pressure die casting, factory planning, ultra-short pulse ablation.

Overall, the contributions of this paper are (1) a novel conceptual model of DSs in manufacturing including its metamodel and notion space and (2) its practical application on a manufacturing scenario and its impact.

Outline. Section 2 introduces preliminaries before Sect. 3 presents our conceptual model. Afterwards, Sect. 4 shows its practical application on a real-world manufacturing scenario. Sect. 5 discusses related research. Sect. 6 concludes.

---

2   Background

One major difficulty when speaking of models, DS and DT is that these terms are often not clearly distinguished and even used interchangeably. We differentiate these terms based on the information flow between the digital and the physical object [10]. While a model and the physical object it represents are synchronized only through manual updates, a DS follows the physical object based on an automatic data flow. Accordingly, a DS is defined as *a set of contextual data-traces and their aggregation and abstraction collected concerning a system for a specific purpose with respect to the original system* [3]. A DT extends the definition of the DS by automatically influencing the physical object as well.

DSs do not fully represent, but provide a purposeful view of the observed object or process, which is referred to as its asset. Accordingly, they are created with a specific focus and by selection and aggregation of data that may originate from heterogeneous sources. The context for the respective data-traces is given by the metadata. The combination of the metadata with referenced models enables data structuring required for subsequent semantic processing. DSs must therefore contain domain-specific knowledge in suitable form. This allows for task-specific analysis and enrichment of underlying models with relevant data, which thereby enables knowledge- and real-time-based decision making in production.

The DS concept is investigated by means of a conceptual model for DSs. The conceptual model consolidates a set of concepts, which are presented as elements in linguistic format. The notion space [16] clarifies the meaning of all elements of DSs and their relationship on its basis.

3   Digital Shadow Metamodel and Notion Space

To achieve the benefits of a standardized information architecture of relevant data in production such as knowledge-based decision making across domains, a metamodel for DSs is proposed. The corresponding metamodel and its elements are presented in Fig. 1. The concepts of the DS are further detailed in the following descriptions, which define the notion space for the metamodel.
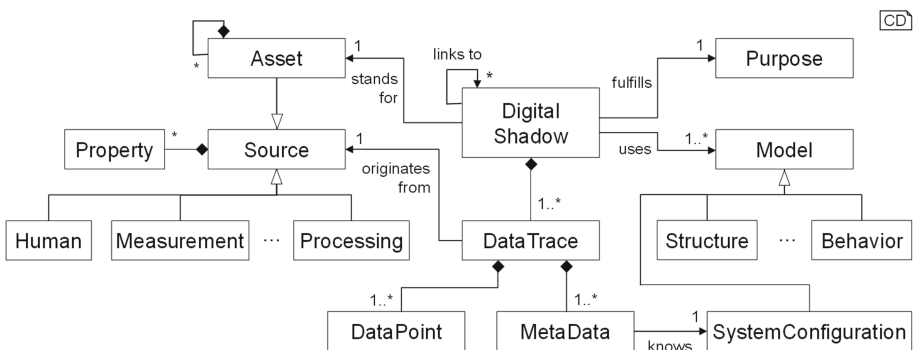

Fig. 1. The metamodel of the DS in UML class diagram (CD) notation.

Asset. According to DIN SPEC 91345 [2] and DIN ISO 55000, "an asset is an item, thing or entity that has potential or actual value to an organization" [1]. An `Asset` can be physical as well as virtual and always fulfills a specific role within a system. Becoming an asset is always concomitant with development, engineering, measurement, construction, or a manufacturing process, regarding the asset's type and role [26]. Physical assets are, *e.g.,* machines, components, and tools. Virtual assets are, *e.g.,* plans, mechanical drawings, standards, or metamodels. A combination of several assets results in a new asset, *e.g.,* by connecting multiple machines to a production cell and vice versa [20]. By characterizing the asset with suitable attributes that reflect its properties, the organization and stakeholders establish the asset in the virtual world so it is applicable as a source [2]. Each DS is associated with exactly one asset which represents the described system. This asset (system) can comprise subsystems of the type `Asset`.

Sources. As the DS is always considered within a context, all data that is regarded as part of it must be connected with its primary source. Therefore, each `DataTrace` must be associated with one distinct `Source`. A `Source` is composed of any number of defined properties, that specify at least the name, the data type, and the unit of the source's data. Sources can be of manifold kinds, which peculiarities shall not be limited in the context of the metamodel. In our metamodel, the `Asset` is a `Source`. While being the considered system within the DS, an asset can serve as a source (*e.g.,* for process limits due to machine specifications) for its own and also other DSs. Other source types could be human inputs via human-machine interfaces, sensors delivering measurement data, or any type of algorithms and simulations delivering specific information. A simple example is the error-acknowledgment by a machine operator. The source is a human and the property a Boolean. A more complex one is an injection molding machine as a source of type asset with diverse properties. Those could vary being specifications like a screw diameter as a float with the metric unit 'mm", actual values like the current clamping force as a float with the unit "kN", or simply the machine name as a string. Those are just a few examples to demonstrate the variety of possible sources and thus the cross-domain applicability of this model.

DataTrace. The core element of the DS is data that describes the matter of concern about the given asset. The DS consists of contextualized `DataTraces` as a subset of accessible data consisting of one or more `DataPoints` and their respective `MetaData`. These data-sets can be numerical values, lists, or complex data objects and are in production, *e.g.,* motion and state-dependent data.

In injection molding, for example, the data-trace may refer to data about the pressure signals or melt temperature measurements of a cavity sensor closely linked to the specific volume of a plastic material under processing conditions. Another possibility may be data about screw movements. The injection molding machine, therefore, measures and controls the screw position at each increment in time. The respective data-trace subsequently corresponds to the screw position as motion data. Whereas the first data-trace indicates the actual material behavior at processing conditions, the second data-trace refers to the actual machine behavior while processing. Both are relevant subsets of the accessible data and

can be analyzed independently in direct comparison to underlying models of an ideal system. However, they can also be correlated with each other and thereby generate information about the correlation of temperature and pressure rising at specific screw movements.

DataPoint. As elaborated above, the DS may consist of multiple `DataTraces` and subsequently, a multitude of `DataPoints` that originate from several `Sources`. The data-points refer to the data a DS needs to describe the system's behavior following the targeted `Purpose`. It may be accessible by value in case of only small amounts of data or by reference to a database.

MetaData. To enable contextualizing and interpreting the actual data of concern that describe a system's behavior, the DS itself as well as the `DataTraces` hold `MetaData`. The metadata of the DS, therefore, lists on the one-hand side the complete system configuration (system setup) with all relevant master data about system components and subcomponents. Some of those subcomponents may serve as sources for individual data-traces and some might just act as supporting assets to the process of concern. Whereas the `SystemConfiguration` itself is integrated as a `Model`. Further structural, physical, or otherwise relevant models are also integrated to achieve the targeted `Purpose` of the DS.

The metadata of the individual sub data-traces on the other hand only need a reference to the overall system configuration as well as a reference to the process model and the respective asset that serves as the source for the data-points. Furthermore, it lists the settings of the data-source and the parameters that can be found in the data-points. The point identifier finally indicates the one parameter inside the point header that serves for unique identification.

Figure 2 presents an exemplary shadow object about actual process data of an injection unit during injection. The overall system is referenced by an 'uid' as well as the general process model and the injection unit as the `Source` of this `DataTrace`. To keep the trace and thereby also the DS lean, the data specification is separated from the final values accessible as `DataPoints`. To access the respective data, a point identifier from the list of actual values is determined, which might be the actual machine cycle or a timestamp. The `MetaData` finally enables the recombination of several `DataTraces` that, *e.g.,* originate from other processes and DS, to generate new `DigitalShadows` with new `Purposes`.

Models. `Models` are a central constituent of `DigitalShadows`. According to Stachowiak [24], models (1) consist of a mapping to an original object, that the model represents, (2) are reduced to the relevant aspects and abstract from details of the original, and (3) have a pragmatism that lets them replace the original in certain scenarios. Models may map to the DS's `Asset` (the physical thing that it represents). In this scenario, the `Model` adds information about the `Asset` to the `DigitalShadow` and helps evaluate/understand the data that this asset provides. For instance, a SysML Block Definition Diagram (BDD) [27] can represent the asset's structure, a BPMN model [28] describes an established process, or a simulation model [8,17] provides information about the expected system behavior at a specific time. Models may also represent the context in
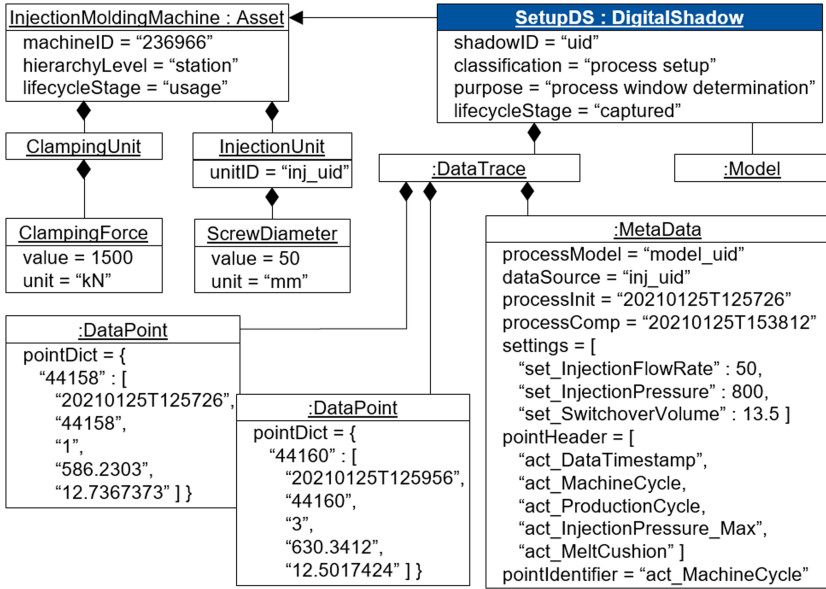
Fig. 2. Structure of an instantiated DS object in injection molding

which the DS is created or in which its asset operates. Models about the context help to understand why `DataPoints` change and relate these changes to events that happen in the operating context. Models also specify the Digital Shadow itself. They define which elements should be part of it, *i.e.,* it is beneficial for the intended purpose. A model's purpose also varies. Models describe the DS itself, thus, serve as the construction plan. They specify its asset and hint how the data-traces are interrelated. Further, when describing the context, models qualify specific values or value changes in the DS, *e.g.,* models can help to decide whether a sensed value can be considered as good or bad.

Purpose. Purpose in a general understanding is "what the DS is supposed to do". DSs must be tailored to a specific `Purpose`, because they neither act on their own nor do they interact directly with the regarded system (see Sect. 2). The purpose definition is a basis for correct cross-domain interoperability as well as a precise understanding of the DS in future use. A DS is designed for its exact purpose. Regarding complex purposes, it is possible to associate several thematically connected DS more comprehensively. To achieve said `Purpose` the functionalities are defined within the `Models` or can be realized through `DataTraces` from sophisticated simulation or processing sources. These functionalities can vary from simply monitoring a heartbeat of a referred system to very complex data analyses, *e.g.,* AI-based predictions. The format of the `Purpose` has no further specifications but the requirement to define it clearly and understandable.

4 Application Example

In the following section, we demonstrate the usability of the introduced meta-model with one real-world application example by adding application-specific instances to the generic classes. We will illustrate a production planning process in injection molding domain to outline the metamodel's usability.

Figure 3 shows the metamodel enhanced with instances for the production planning process of an injection molding production. In this application example, the objective for a production controller is to set an optimal schedule. Finding the injection molding machine (IMM) where the expected rejection rate for part X is minimal is the `purpose` the controller aims for. Because production planning processes are often highly complex, we assume that the shopfloor comprises only two injection molding machines (A and B) that only differ in the rejection rate for part X. Further, no other constraints are given. To support the controller in the decision, the DS must provide the machine ID (IMM-ID) of the injection molding machine, where the expected rejection rate for part X is the lowest. This necessary information comes from multiple `assets`, namely the Manufacturing Execution System (MES) containing multiple data sources, and the machines.
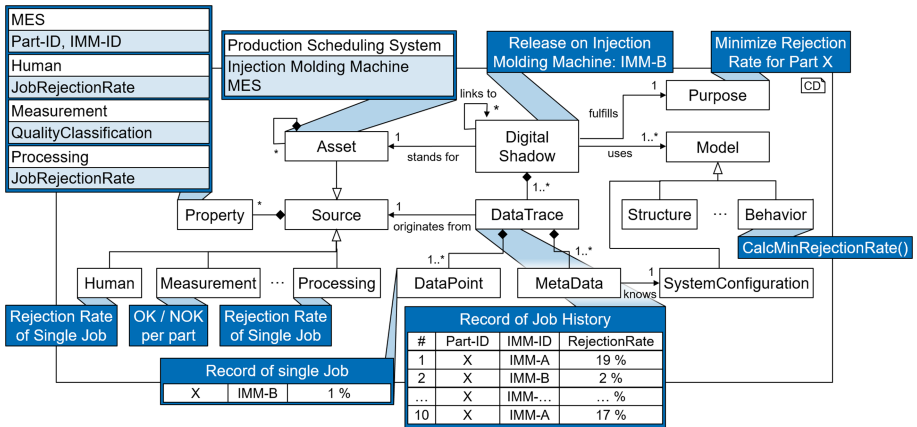

Fig. 3. Metamodel enhanced with specific instances for the planning process.

Relevant *attributes* are Part-ID, IMM-ID, and the current rejection rate of machine A and machine B for part X. The MES as `Source` provides the Part-ID and IMM-ID. The gathering of the rejection rates was made, on the one hand, manually by a human operator via plant data acquisition for a single, finished job on machine A. The relating attribute in the `Property` class for the human `Source` calls JobRejectionRate. On the other hand, sensors at machine B inspect every single part and automatically classifies the approved part as OK, while rejected parts are classified as NOK (not OK). After the job on machine B is terminated, it directly proceeds the JobRejectionRate and forwards

the information to the MES. The MES enhances the corresponding job record with the received rejection rate. Hence, a record of a single job represents one single `DataPoint`. After combining all single data points to a `DataTrace`, a mathematical `model` (CalcMinRejectionRate) first computes the mean rejection rate for machines A and B. Subsequently, the model selects the machine ID concerning the minimum rejection rate and provides the result to the controller. The controller finally sets the optimal schedule that presumably minimizes the number of rejects of part X. One benefit is a possible faster termination of the job when producing on IMM-B instead of IMM-A. That leads to a reduction of the waiting time for the following jobs. Besides, in comparison to IMM-A, less input of resin on IMM-B is necessary to manufacture the desired quantity of part X. Consequently, material costs are saveable.

5   Related Work

By now, there exists no conceptual model for DSs from the computer science perspective and following the conceptual model definition from Mayr and Thalheim [16]. However, Digital Shadows play an important role in smart manufacturing and some of the concepts are already defined in other contexts.

Quix, Hai and Vatov [19] present a conceptual view on a metadata-model related to metadata extraction and management in data lakes. Liebenberg and Jarke present the AI modeling and data management aspects of DSs in the IoP as a generalization of database view conceptualizations [13]. Our conceptual model for DSs goes beyond that and considers further context information such as the source of data, the relation to assets and the connection to engineering models.

Loucopoulos et al. [14] present a conceptual meta model for Cyber-Physical Production Systems which emphasizes information sharing and analysis aspects at a broad requirements engineering level, without going into the level of standardization and detail presented in this paper.

Bravo et al. [4] present a meta model for manufacturing which focuses on the resources, execution, planning, the product, and the client. Their approach does not consider the assets of the physical object, the purpose for data collection, and aggregation or metadata.

Ladj et al. [11] propose a framework for a knowledge-based DS including a physical and a virtual system. Their DS is a data and knowledge management system. Data analytics are applied to the database to generate the knowledge base of the DS. The analysis of the physical system by the DS supports the decision process. The proposed DS is self-learning and therefore improves continuously. Their approach describes elements of DSs for the specific purpose of a machine as physical object but it lacks a generic description of DS elements.

Schuh et al. [22] develop a data structure for DSs. The database consists of the organization's knowledge that can be utilized to solve its tasks. The data structure model is based on a generic order fulfillment process. An entity-relationship model describes the relationships between the data, resources, and elements of the order fulfillment. To complete the data structure, information requirements

are modeled and linked with the data. Within [22], no conceptual model is given and concepts such as purpose or engineering models are not considered.

Parri et al. [18] create an architecture for a DT-based knowledge base. They present a metamodel as UML CD describing the knowledge base including the concepts DTs and Meta-DTs. Their metamodel has a focus on digital systems of a company, further elements of DSs like data or models are not considered.

6 Conclusion and Outlook

In this paper, we proposed a conceptual model for DSs and successfully evaluated it in real-world scenarios. Researchers in the smart manufacturing domain process and analyze data from various sources with heterogeneous formats and interfaces. They create models during all lifecycle stages of products, machines, and other entities. One major issue is the insufficient integration between data and models. Our conceptual model tackles that issue.

The impact of the proposed conceptual model of the DS in the production domain is multifaceted. Besides general model-driven benefits such as complexity management and code generation [5], it facilitates cross-domain collaboration by providing a base to define instances, as our real-life application in Sect. 4 showed. Its adequate and compact core enables interoperability on multiple levels, while it allows flexible extensions as well as domain-specific additions based on future requirements. Our application example in the manufacturing domain proof that this DS conceptual model can be a foundation to enable benefits in real-world applications in the short and long term.

References

1. DIN ISO 55000:2017–05, Asset Management - Übersicht, Leitlinien und Begriffe
2. DIN SPEC 91345:2016–04, Reference Architecture Model Industrie 4.0 (RAMI4.0)
3. Bibow, P., et al.: Model-driven development of a digital twin for injection molding. In: Dustdar, S., Yu, E., Salinesi, C., Rieu, D., Pant, V. (eds.) CAiSE 2020. LNCS, vol. 12127, pp. 85–100. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-49435-3_6
4. Bravo, C., Aguilar, J., Ríos, A., Aguilar-Martin, J., Rivas, F.: A generalized data meta-model for production companies ontology definition. Int. J. Syst. Appl. Eng. Dev. **2**, 191–202 (2008)
5. Dalibor, M., Michael, J., Rumpe, B., Varga, S., Wortmann, A.: Towards a model-driven architecture for interactive digital twin cockpits. In: Dobbie, G., Frank, U., Kappel, G., Liddle, S.W., Mayr, H.C. (eds.) ER 2020. LNCS, vol. 12400, pp. 377–387. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-62522-1_28
6. Desai, N., Ananya, S.K., Bajaj, L., Periwal, A., Desai, S.R.: Process parameter monitoring and control using digital twin. In: Auer, M.E., Ram B., K. (eds.) REV2019 2019. LNNS, vol. 80, pp. 74–80. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-23162-0_8
7. Gomez-Escalonilla, J., Garijo, D., Valencia, O., Rivero, I.: Development of efficient high-fidelity solutions for virtual fatigue testing. In: Niepokolczycki, A., Komorowski, J. (eds.) ICAF 2019. LNME, pp. 187–200. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-21503-3_15

8. Gosavi, A., et al.: Simulation-Based Optimization. Springer, Heidelberg (2015). https://doi.org/10.1007/978-1-4899-7491-4

9. Hu, L., et al.: Modeling of cloud-based digital twins for smart manufacturing with MT connect. Proc. Manuf. **26**, 1193–1203 (2018)

10. Kritzinger, W., Karner, M., Traar, G., Henjes, J., Sihn, W.: Digital twin in manufacturing: a categorical literature review and classification. IFAC-PapersOnLine **51**(11), 1016–1022 (2018)

11. Ladj, A., Wang, Z., Meski, O., Belkadi, F., Ritou, M., Da Cunha, C.: A knowledge-based digital shadow for machining industry in a digital twin perspective. J. Manuf. Syst. **58**, 168–179 (2021)

12. Liau, Y., Lee, H., Ryu, K.: Digital twin concept for smart injection molding. In: IOP Conference Series: Materials Science and Engineering, vol. 324, p. 012077 (2018)

13. Liebenberg, M., Jarke, M.: Information systems engineering with digital shadows: concept and case studies. In: Dustdar, S., Yu, E., Salinesi, C., Rieu, D., Pant, V. (eds.) CAiSE 2020. LNCS, vol. 12127, pp. 70–84. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-49435-3_5

14. Loucopoulos, P., Kavakli, E., Chechina, N.: Requirements engineering for cyber physical production systems. In: Giorgini, P., Weber, B. (eds.) CAiSE 2019. LNCS, vol. 11483, pp. 276–291. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-21290-2_18

15. Luo, W., Hu, T., Zhang, C., Wei, Y.: Digital twin for CNC machine tool: modeling and using strategy. J. Ambient Intell. Humaniz. Comput. **10**(3), 1129–1140 (2019)

16. Mayr, H.C., Thalheim, B.: The triptych of conceptual modeling. Softw. Syst. Model. **20**(1), 7–24 (2020). https://doi.org/10.1007/s10270-020-00836-z

17. Mujber, T., Szecsi, T., Hashmi, M.: Virtual reality applications in manufacturing process simulation. J. Mater. Process. Technol. **155-156**, 1834–1838 (2004)

18. Parri, J., Patara, F., Sampietro, S., Vicario, E.: A framework for model-driven engineering of resilient software-controlled systems. Computing **103**(4), 589–612 (2021)

19. Quix, C., Hai, R., Vatov, I.: Metadata extraction and management in data lakes with GEMMS. Complex Syst. Inf. Model. Q. **9**, 67–83 (2016)

20. Schmertosch, T., Krabbes, M.: Automatisierung 4.0: Objektorientierte Entwicklung modularer Maschinen für die digitale Produktion. Carl Hanser Verlag (2018)

21. Schuh, G., Gützlaff, A., Sauermann, F., Maibaum, J.: Digital shadows as an enabler for the internet of production. In: Lalic, B., Majstorovic, V., Marjanovic, U., von Cieminski, G., Romero, D. (eds.) APMS 2020. IAICT, vol. 591, pp. 179–186. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-57993-7_21

22. Schuh, G., Kelzenberg, C., Wiese, J., Ochel, T.: Data structure of the digital shadow for systematic knowledge management systems in single and small batch production. Proc. CIRP **84**, 1094–1100 (2019)

23. Schuh, G., Prote, J.-P., Gützlaff, A., Thomas, K., Sauermann, F., Rodemann, N.: Internet of production: rethinking production management. In: Wulfsberg, J.P., Hintze, W., Behrens, B.-A. (eds.) Production at the Leading Edge of Technology, pp. 533–542. Springer, Heidelberg (2019). https://doi.org/10.1007/978-3-662-60417-5_53

24. Stachowiak, H.: Allgemeine Modelltheorie. Springer, Heidelberg (1973)

25. Urbina Coronado, P.D., Lynn, R., Louhichi, W., Parto, M., Wescoat, E., Kurfess, T.: Part data integration in the shop floor digital twin: mobile and cloud technologies to enable a manufacturing execution system. J. Manuf. Syst. **48**, 25–33 (2018). Special Issue on Smart Manufacturing

26. VDI: Industrie 4.0 - Technical Assets - Basic terminology concepts, life cycles and administration models (2016)
27. Weilkiens, T.: Systems Engineering with SysML/UML: Modeling, Analysis, Design. Elsevier (2011)
28. White, S.A.: Introduction to BPMN. IBM Cooper. **2**, 1–26 (2004)
29. Wortmann, A., Barais, O., Combemale, B., Wimmer, M.: Modeling languages in industry 4.0: an extended systematic mapping study. Softw. Syst. Model. **19**(1), 67–94 (2020)
30. Zambal, S., Eitzinger, C., Clarke, M., Klintworth, J., Mechin, P.: A digital twin for composite parts manufacturing: effects of defects analysis based on manufacturing data. In: International Conference on Industrial Informatics (INDIN 2018), pp. 803–808. IEEE (2018)