Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 104 (2021) 833–838

www.elsevier.com/locate/procedia

54th CIRP Conference on Manufacturing Systems

An Ecosystem for Digital Shadows in Manufacturing

Christian Brecher[a], Manuela Dalibor[b], Bernhard Rumpe[b], Katrin Schilling[a,*], Andreas Wortmann[b]

[a] *Laboratory for Machine Tools and Production Engineering (WZL), RWTH Aachen University, Aachen, Germany*
[b] *Software Engineering (SE), RWTH Aachen University, Aachen, Germany*

* Corresponding author. Tel.: +49-241-8027449. *E-mail address:* k.schilling@wzl.rwth-aachen.de

Abstract

Digital Shadows are data structures precisely tailored to support decision making in domain-specific real-time that promise tremendous potential to reduce time and cost in manufacturing. They are often engineered ad-hoc, for single specific applications, without considering their aggregation, combination, or reuse. This lack of foundations hampers a joint understanding of Digital Shadows that prevents joint research as well as collaboration and exchange of Digital Shadows across enterprise boundaries. Based on interdisciplinary research, we conceived a conceptual model of Digital Shadows that can guide their engineering, combination, and reuse. This not only supports researchers and practitioners in better understanding each other when discussing Digital Shadows but also eases the engineering of compatible and exchangeable Digital Shadows.

Keywords: Digital Shadow; Smart Manufacturing; Cyber-Physical Production System; Industry 4.0

1. Motivation

Speaking of a "data image" of manufacturing processes or systems often refers to the concept of *Digital Twins (DT)*, which has been described in numerous works for various applications. On the one hand, these works represent interesting enablers for new concepts in manufacturing; on the other hand, they are very diverse, complex, and quite difficult to transfer on industrial cases. Even finding a universally valid as well as precise definition is still not possible: Many definitions in literature are very vague (e.g., a “virtual representation” [1]) or not feasible (e.g., an “exact copy” [2] or “complete mirror in the cyber space of the physical prototype” [3]) – since being “digital” implies that a DT must be subject to abstractions; a “complete” DT is a highly idealistic and most likely not practical idea. This lack of consensus makes it difficult for industrial users to implement the scientific work.

A very similar, but less complex concept is the *Digital Shadow (DS)*. It describes the aggregation of data with a specific purpose and provides the foundation to derive data-based insights, either for the human user, for an analytical algorithm or a simulation. As well as the DT, the concept of DS targets the potentials for efficiency gains through data but provides a more low-effort implementation for industry. The challenge on implementing the concept of DS is to collect, aggregate and provide the relevant data for a specific need; the components and concepts required for this task will be presented in this paper.

Based on interdisciplinary research in the “Internet of Production” excellence cluster, we devised a conceptual model of DS and their ecosystem that can be applied to a variety of domains to better structure, understand, exchange, and exploit manufacturing data. Sec. 3 introduces our concept of DS, before Sec. 4 details the components of their ecosystem and Sec. 5 concludes with a roadmap on smart decision making with DS.

Thus, this paper complements the conceptual perspective to an industrial use case of an automotive supplier, which serves as an example (see Sec. 2.1) and has been described from a more practical perspective in [4].

2212-8271 © 2021 The Authors. Published by Elsevier B.V.

10.1016/j.procir.2021.11.140

2. Background

Digital Shadows [5,6] are a concept of collecting, aggregating, and abstracting manufacturing data for specific purposes to enable decision making based on this data while still being computationally efficient through domain-specific data reduction. For this purpose, DS comprise various kinds of data (e.g., measurement data, simulation data, models) from different sources that are abstracted and aggregated in application-specific ways to fulfill their purpose. They serve as a basis for, e.g., process optimization and data mining.

The *Internet of Production* is a DFG excellence cluster at RWTH Aachen University that pursues the vision of interdisciplinary manufacturing based on continuous data exchange, integrated development, and cross-domain validation. For a first conceptual model of DS, their use, and capabilities, we conducted a survey in the project through which we reached out to its 200 researchers of 25 departments who conduct research in a variety of domains, including artificial intelligence, computer science, innovation research, labor science, mechanical engineering, and manufacturing technology.

Conceptual modeling [7] is an abstraction technique in which complex systems are represented through their quintessential concepts and relations to improve understanding of the subject that the model represents. Conceptual models can describe structure, behavior, or a combination by the use of modeling techniques, such as UML Class Diagrams (CDs) [8] for the description of structural aspects or BPMN [9] for behavioral aspects. In the following, we will use CDs and UML Object Diagrams (ODs) [8] to describe our conceptual model of Digital Shadows.

2.1. Industrial use case

In addition to our research work, the conceptual model was applied on a real-world use case from an automotive supplier, who produces various glass panes for cars. Along the production line, raw data is collected from machines or sensors and published via edge devices. One example is the thermal processing of the glass. Here, the speed of the conveyor belt on which the individual panes are transported, and the temperature of the furnace are relevant information for quality monitoring. Furthermore, the id of each pane is scanned when it enters and leaves the furnace chamber. The raw data is published from the sensor or control system via an MQTT broker.

Now, these data can be interpreted from different perspectives: How many panes have passed the furnace in a certain period of time? What was the temperature profile of the furnace? What temperatures has a particular pane gone through? To answer these questions, the respective data are aggregated as DS and stored in a database. They form ready-made information packages that can answer a specific question about a process or product.

3. On the Nature of Digital Shadows

The DS always refers to a system of interest. A *System* is a set of elements that interact with each other and form a whole; the system could be technical or biological, physical, or non-material (e.g., software), real or theoretical (e.g., as a simulated model). At the moment of observation, system can exist in parallel with the DS, it can exist no longer (e.g., for a historical view) or not yet (e.g., as prediction) in the considered state.

DS are created with a specific *Purpose*, e.g., representations, regarding the system of interest. We specifically focus on the production domain, thus, considering especially DS of production machines. A *Cyber-Physical Production System (CPPS)* is a composition of human resources, production equipment, and aggregated products towards which it establishes one or several cyber-physically formulated interaction interfaces [10]. In the context of the (Industrial) Internet of Things, this is referred to as an *Asset* [11].

3.1. Digital shadow (instances)

DS instances, which we call "Digital Shadows" for short, are sets of contextual data traces, their aggregation and abstraction collected concerning a system for a specific purpose with respect to the original system. DS combine data from various sources. Consequently, a DS may contain historical information about the system that the system itself has cleared in the meantime because it does not necessarily remember all its historical behavior, activities, and structural changes. A DS does not have to reflect the observed system entirely but can include abstractions that reduce the available data to a limit that can be processed in time. One CPPS can have many different DS describing various aspects of the system in different detail and at different times. *DS instances* are defined by *DS types*.

3.2. Digital shadow types

Different *DS types* prescribe different DS for different purposes, e.g., predictive maintenance or adaptive control. Each DS type encapsulates a subset of the required data for the specific information need driven by its purpose. Depending on the asset's status when the DS is created, the encapsulated data differs, but the structure remains; the type defines which data and from which origin should be part of the DS. The DS type thus serves as a "construction plan" for DS instances. DS can be enriched with contextual information (such as metadata about time, location, and purpose). Since the necessary metadata is purpose-oriented, it cannot be defined generically but must also be part of the DS type.

3.3. A conceptual model of digital shadows

To characterize the DS components more precisely, we devised the conceptual model illustrated in Figure 1. The conceptual model specifies which concepts and relations a data structure requires, such that it can serve as a DS: A *Digital Shadow* consists of a set of contextual *Data Traces*. These can encapsulate various values, e.g., changing over time. The

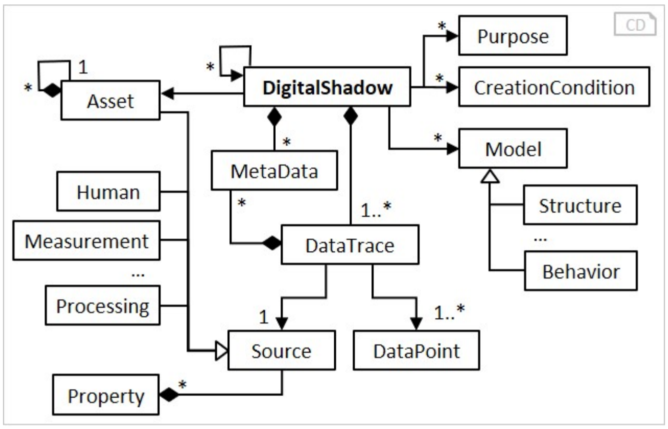

Fig. 1. Conceptual model of Digital Shadows

Data Point class is not further restricted, thus, also allows extensions for storing image data or a 3-dimensional vector representing a location. Each *Data Trace* has a *Source* that can be further described by properties. Sources of data traces can be *Measurements*, *Processings* (for e.g., filtering data) or even the *Human* worker. *Assets* can also serve as sources for data traces, e.g., the speed of a line's conveyor belt from its drive. *Meta Data* can contain any information about data that is relevant for further usage. This may include the timestamp of the measurement, the location or information about data precision. The DS always refers to an *Asset*, e.g., a component of a CPPS, that is described by this DS. The *Purpose* describes why the DS is created. For instance, a DS's purpose may be to sum up quality information for each individual product. DS also rely on models for gaining domain knowledge about e.g., the CPPS, its behavior, or further information about the application context. These are represented by the *Model* class. The *Creation Condition* describes when the DS (instance) should be created. Possible triggers could be a human operator querying the current production state, a clock indicating time interval, or an event from the CPPS, e.g., exceeding a critical threshold or detecting the next object.

The CD (Figure 2, top) is an instance of the conceptual model. It specifies an exemplary DS type that characterizes the reformation of windowpanes and specifies which data should be part of each DS conforming to this type. For another glass pane, the values would differ, but both DS share a common DS type since the same data is relevant for quality analysis. The DS is created every time that the product id changes, as indicated by *NewPane_Trigger*. The *shadowID* can identify the *ProductShadow*, so each instance of this DS type has a unique id. The DS type contains three data traces: the *ProductID_DataTrace*, *Speed_DataTrace*, and *Temp_DataTrace*. Each of these data traces refers to an asset as a data source. The classes of these assets also contain a topic attribute that refers to the MQTT topic of the data stream.

While the *ProductID_DataTrace* contains only one data point, the *Speed_DataTrace* and *Temp_DataTrace* can also contain multiple speed and temperature values. Figure 2 also shows a DS instance (bottom) that conforms to the DS type and is represented as an OD. The DS is created for one glass pane with the product id "00123" and contains three temperature values "21.03", "21.75", "22.07" and one value for speed "44"

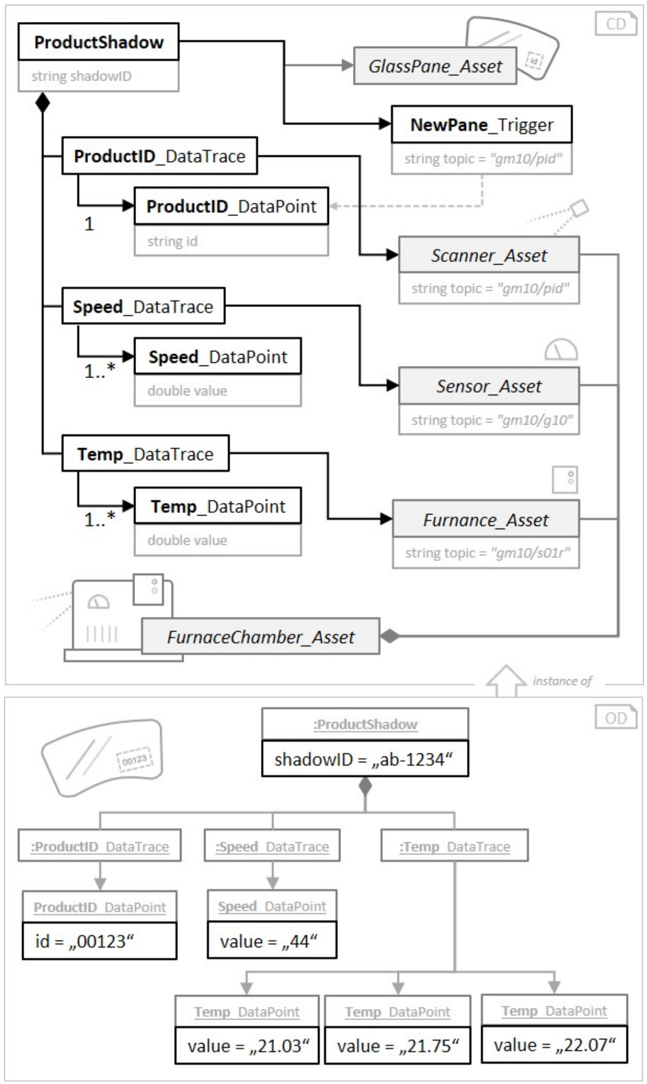

Fig. 2. Exemplary DS Type (top) and DS Instance (bottom) for a data set of a glass pane, mapping the temperatures and speed of the conveyor belt to each individual pane.

(Note that the data stream provides raw values without any unit or other context; the modeling of the data is discussed in [4]). Other properties are omitted for clarity. Data can also be entered by the operators via user interfaces; the individual data origin depends on the use case.

The values displayed in the DS instance in Figure 2 are values originating from a glass pane's real production process. They are captured via sensors within the CPPS. Summarizing, the conceptual model of DS specifies which concepts and relations are used to describe DS types. The DS type describes the components of one concrete that can be instantiated with different data values at runtime. Thus, it prescribes the structure of a Digital Shadows for one particular use case.

The DS Type is an instance of the DS conceptual model. At runtime, there can be different DS conforming to the DS type with different values. The technical realization of DS includes different infrastructure elements. There has to be data acquisition and storage, a communication infrastructure, and a software component that creates DS based on the DS type every time the specified trigger occurs. The data acquisition and storage collects the data necessary to create DS and also provide historical values. The DS caster elevates the notion of DS to an actual useful concept that adds value to the production by providing information that would otherwise not have been available to users of the CPPS. This infrastructure is defined separately to decouple the concept of DS type and DS (instance) from the technical realization.

4. A Conceptual Ecosystem for Digital Shadows

The basic requirement for DS is data that sufficiently describes the (production) system in terms of its intended purpose. Such data can be generated and provided by the asset itself, e.g., from the sensors or the control unit. It may also come from other systems that generate data concerning the system of interest; this can be simulations, management software or design tools, as well as humans in various roles providing information. The selection of relevant data is closely related to the question that the data is intended to answer. So, the selection can be defined within the scope of an automated analysis by its programmer, or it is arranged by a data analyst on demand. For example, the practical paper [4] on our industry example describes a method that supports the selection of requested data traces in an editor tool for data analysts to explore root causes of failures.

The concept of the DS does not prescribe a technical realization of data provision: Data can be fetched directly via interfaces of the individual apps or devices, or it may first be aggregated from various sources in one (or more) repositories such as a data lake. The challenge is to balance data economy with great flexibility for future, even not yet defined, data analysis cases. Figure 4 illustrates how the DS is generated from data.

4.1. Digital shadow caster

The *DS Caster*, a software component, instantiates the DS: The blueprint for this task is provided by the DS type, defining which data is aggregated in which way. Additional domain knowledge can be added for this purpose through models; these can be physical correlations, a forecasting model or reference process. Depending on the purpose or the individual scenario, the instantiation of the DS can appear in different shapes: The instance of the DS can be a live reflection whose content continuously changes with the state of the real asset. This enables the online observation of the asset from a purpose-driven perspective and is usually volatile. However, the caster can also generate individual *DS Instances*, initiated (as well as terminated) by a trigger. The trigger can be defined periodically, either time-based (e.g., every 10ms, every morning or once a year) or event-driven (e.g., when a value drops below a threshold or each time a new item is detected). This instantiation can happen in real-time, or - if the data originates from an archive - it can be generated (retrospectively) on-demand, e.g., when a case of damage occurs. The DS instance created by the caster can be consumed directly by the asset itself or by other services. Depending on the system architecture, both publish-subscribe concepts as well as querying of a concrete DS by the consumer are applicable. There is a wide range of applications that can be

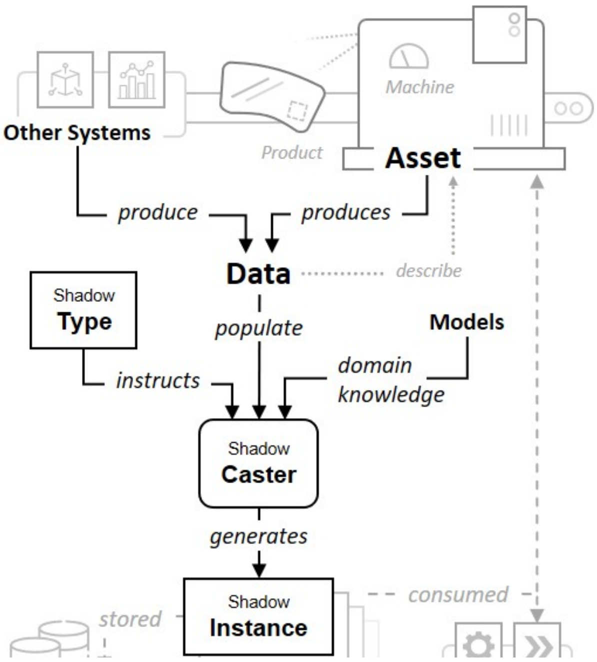

Fig. 3. Workflow of the Digital Shadow system to create and consume DS Instances

considered as consuming services, including a device in the CPPS, software applications, or humans, either in the role of a worker or a data analyst. If the caster generates instances on a continuous schedule, these might also be stored in the data store to be available for their use.

4.2. Asset

An *Asset* in our conceptual model is a system that "has a value for an organization" [11]. As such, it could be a CPPS (e.g., a machine or a component of the production line), the workpiece, the product, or a pure software system as long as data relevant to the purpose of the DS can be collected. Purely physical systems consequently need to be accompanied by another system able to provide that data, e.g., through measurements. The asset may produce data itself as well as there might be other systems, that produce data about the asset of interest.

4.3. Data lake

Data Lakes are big data repositories that store raw data of heterogeneous data sources and provide functionality for on-demand integration with the help of metadata descriptions [12]. Data lakes combine data objects of structured and unstructured form. In addition to data from relational databases they may contain, e.g., emails, images, PDFs but also semi-structured files like CSV, XML, and JSON. One advantage of data lakes is that they are designed to store different types of data and can also handle large amounts of data. In addition, data lakes also provide mechanisms for searching and structuring the data they contain. They are therefore well suited for storing data and engineering models of CPPS and making them available when needed. A well-known realization of the data lake concept is available from Apache Hadoop2. MongoDB also offers a data lake realization which in contrast to HDFS is not open source.

In the infrastructure we envision, a data lake will store historical data, engineering models, metadata, and generated digital shadows, if their generation was computationally intensive.

4.4. Service

A software service refers to a software solution, providing a functionality or a set of software functionalities such a data retrieval or controlling a CPPS with a specific purpose and that is also reusable through different users. In the field of smart manufacturing, many domain-specific services are being developed, such as automated execution of experiments, process monitoring, and predictive maintenance. All of these services require information about the underlying CPPS which they can obtain in the form of DS. The DS caster will produce these DS instances and make them available to the services via an (open) interface.

4.5. Domain knowledge from models

Model-based Systems Engineering is the formalized application of modeling to support system requirements, design, analysis, verification, and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases [13]. As such, the resulting engineering models store information about the CPPS structure, geometries, and intended behavior. This information is relevant as metadata for DS, as it can provide additional information about the origin of measurement data. In addition, parts of the DS infrastructure can be derived from these models. For example, databases, communication infrastructure, or monitoring DS types that combine information about the predefined activities of the CPPS and can serve to detect malfunctioning can be generated from a well-formed engineering model.

5. Conclusion and Future Work

In this paper, a conceptual model for Digital Shadows was presented, which outlines the necessary concepts. The role and structure of DS types, which represent the "blueprint" for the individual DS instances, was presented. The DS instances are the basis for interpretations by data analysts, e.g., to study root-cause correlations of failures or quality deviations. Furthermore, we presented the system of how these DS instances are generated. The presented conceptual model was discussed in the scientific work with experts from different disciplines and applied in an industrial use case. The challenge was to represent complex use cases (e.g., for AI applications) as well as pragmatic and lightweight implementations. Both aspects were covered by our model, but as a consequence, individual components might be further specified.

Finally, this section discusses five research challenges towards the vision of semantic DS that are automatically gathered, abstracted, and aggregated in the most suitable granularity for a specific purpose.

5.1. Semantic integration of engineering models

Giving semantics in the sense of meaning [14] to the data of Digital Shadows demands understanding them first. The data gathered from production systems always exists in the context of its sensors, the surrounding system, and the environment. In Industry 4.0, this context is increasingly often modeled explicitly [15]. Consequently, properly understanding the data demands relating it to the (AutomationML [16], OWL [17], Simulink [18], SysML [19], UML [8] etc.) models of its context. To make these relations explicit and comprehensible, modeling itself can help. For instance, UML class diagrams, OWL ontologies, or SysML blocks can relate models to data (sources) and describe their context through additional (meta) data, such as employed units, precision, and more.

5.2. Automated generation of (minimal) digital shadow types

Currently, Digital Shadow Types are crafted manually to describe semantically relevant integrated data. For large systems of systems, collecting and properly connecting the data structures is challenging due to the sheer amount of data structures involved. Where a system leveraging Digital Shadows is reified, data acquisition, e.g., through SPARQL [20] queries, and the underlying data structures can be derived automatically. The data parsimony of the resulting data structures then depends on the precision of the reified data collection requirements.

5.3. Data pursuing digital shadows

Currently, a Digital Shadow is a snapshot of system data collected, abstracted, and aggregated for a specific purpose. While each Digital Shadow may link to its predecessor, the latest Digital Shadow can neither be retrieved easily nor reliably if the predecessor link is not implemented. Providing services for each Digital Shadow Type that always make their respective latest Digital Shadows and their predecessors accessible, e.g., as MQTT [21] topics can mitigate that. Using technique from model-driven development [22], such services could be generated. However, regularly publishing a Digital Shadow with all its predecessors may not be data parsimonious. Similar to the former challenge, means to automatically derive the minimal frequency, history, etc. of such data pursuing Digital Shadows need to be conceived.

5.4. Composing and integrating digital shadow

Digital Shadows are data structures linking to models, data traces, assets, and data sources. For parsimonious data collection, Digital Shadows and Digital Shadow Type need to be reusable in different contexts. For instance, a Digital Shadow representing the power consumption of different parts of a manufacturing system could be integrated into a Digital Shadow representing the power consumption of a factory containing that manufacturing system. Consequently, Digital Shadow must support comprising other Digital Shadows, including their models, data traces, assets, and data sources. Where Digital Shadows include data of different quality, from different time frames, or collected at different intervals, they need to be integrated, merged, composed, or embedded carefully to retain the semantic integrity of the result. Otherwise, composing incompatible Digital Shadows might lead to making flawed decisions based on their data. Consequently, also the notion of Digital Shadow compatibility needs to be investigated.

5.5. Connecting digital shadows to data sources

Digital Shadow Types describe logical data structures that refer to data sources. They do not explain how the data shall be obtained, e.g., the technical realization of data access. This separation of concerns serves the purpose of reusing Digital Shadow Types with different systems and in different contexts. This, however, demands connecting the platform-independent Digital Shadow Types to platform-specific data sources. Currently, this mapping needs to be established manually. In a future where systems are modeled holistically, the information about technical access to data sources will be part of the system models and the mapping of Digital Shadows can be derived from these accordingly. All of these functionalities can be used in the future to create context-aware Digital Twins that optimize the production process in CPPS based on data encapsulated in Digital Shadows.

Acknowledgements

Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy - EXC 2023 Internet of Production.

References

[1] Atorf, L., Roßmann, J., 2018. Interactive Analysis and Visualization of Digital Twins in High-Dimensional State Spaces, in: 2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV), pp. 241–246.
[2] Rückert, M., Merkelbach, S., Alt, R., Schmitz, K., 2018. Online Life Cycle Assessment for Fluid Power Manufacturing Systems-Challenges and Opportunities, in: IFIP International Conference on Advances in Production Management Systems, pp. 128–135.
[3] Fei, X., Fengchen, Q., Bing, S., Yuzhu, F., 2018. Digital Twin of Solid Rocket Motor, Problem and Challenge, in: 2018 11th International Symposium on Computational Intelligence and Design (ISCID), pp. 7–11.
[4] Brecher, C., Buchsbaum, M., Müller, A., Schilling, K., Obdenbusch, M., Staudacher, S., Albasatineh, M.C., 2021. Gaining IIoT insights by leveraging ontology-based modelling of raw data and Digital Shadows. The 4th IEEE International Conference on Industrial Cyber-Physical Systems.
[5] Riesener, M., Schuh, G., Dölle, C., Tönnes, C., 2019. The Digital Shadow as Enabler for Data Analytics in Product Life Cycle Management. Procedia CIRP 80, 729–734.
[6] Schuh, G., Häfner, C., Hopmann, C., Rumpe, B., Brockmann, M., Wortmann, A., Maibaum, J., Dalibor, M., Bibow, P., Sapel, P., Kröger, M., Bauer, W., Volk, W., Zäh, M., 2020. Effizientere Produktion mit Digitalen Schatten. ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb 115 (special), 105–107.
[7] Karagiannis, D., Mayr, H.C., Mylopoulos, J., 2016. Domain-specific conceptual modeling. Springer.
[8] Rumpe, B., 2016. Modeling with UML: Language, Concepts, Methods. Springer International.
[9] White, S.A., 2004. Introduction to BPMN. IBM Cooperation 2 (0), 0.
[10] L. Ribeiro, 2017. Cyber-physical production systems' design challenges, in: 2017 IEEE 26th International Symposium on Industrial Electronics (ISIE), pp. 1189–1194.
[11] VDI/VDE, 2016. Industrie 4.0 – Technical Assets. Basic terminology concepts, life cycles and administration models. VDI/VDE Society Measurement and Automatic Control (GMA).
[12] Hai, R., Geisler, S., Quix, C., 2016. Constance: An intelligent data lake system, in: Proceedings of the 2016 International Conference on Management of Data, pp. 2097–2100.
[13] Fisher, A., Nolan, M., Friedenthal, S., Loeffler, M., Sampson, M., Bajaj, M., VanZandt, L., Hovey, K., Palmer, J., Hart, L., 2014. 3.1.1 Model Lifecycle Management for MBSE. INCOSE International Symposium 24 (1), 207–229.
[14] Harel, D., Rumpe, B., 2004. Meaningful Modeling: What's the Semantics of "Semantics"? IEEE Computer 37 (10), 64–72.
[15] Wortmann, A., Barais, O., Combemale, B., Wimmer, M., 2020. Modeling languages in Industry 4.0: an extended systematic mapping study. Software and Systems Modeling 19 (1), 67–94.
[16] Drath, R., Luder, A., Peschke, J., Hundt, L., 2008. AutomationML-the glue for seamless automation engineering, in: 2008 IEEE International Conference on Emerging Technologies and Factory Automation, pp. 616–623.
[17] McGuinness, D.L., van Harmelen, F., others, 2004. OWL web ontology language overview. W3C recommendation 10 (10), 2004.
[18] Dabney, J.B., Harman, T.L., 2004. Mastering Simulink. Pearson.
[19] Friedenthal, S., Moore, A., Steiner, R., 2014. A practical guide to SysML: the systems modeling language. Morgan Kaufmann.
[20] Pérez, J., Arenas, M., Gutierrez, C., 2009. Semantics and complexity of SPARQL. ACM Transactions on Database Systems (TODS) 34 (3), 1–45.
[21] Hillar, G.C., 2017. MQTT Essentials-A lightweight IoT protocol. Packt Publishing Ltd.
[22] France, R., Rumpe, B., 2007. Model-driven Development of Complex Software: A Research Roadmap. Future of Software Engineering (FOSE '07), 37–54.