Towards Actionable Cognitive Digital Twins for Manufacturing

Jože M. Rožanec[1,3,4][0000-0002-3665-639X], Lu Jinzhi[2][0000-0001-5044-2921], Aljaž Košmerlj[1], Klemen Kenda[3,4][0000-0002-4918-0650], Kiritsis Dimitris[2][0000-0003-3660-9187], Viktor Jovanoski[3,4], Jan Rupnik[1,3], Mario Karlovčec[3][0000-0003-4480-082X], and Blaž Fortuna[1,3][0000-0002-8585-9388]

[1] Jožef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia
[2] EPFL SCI-STI-DK, Station 9, CH-1015 Lausanne, Switzerland
[3] Qlector d.o.o., Rovšnikova 7, 1000 Ljubljana, Slovenia
[4] Jožef Stefan International Postgraduate School, Jamova 39, 1000 Ljubljana, Slovenia

Abstract. Digital Twins (DTs) mirror physical assets and can be enriched with software layers that provide different capabilities. In the case of actionable cognitive twins (CTs), algorithms provide behavior (make DTs actionable) and a knowledge graph (KG) adds cognitive capabilities. In this paper we present a new ontology that models a shop-floor DT, capturing background knowledge regarding shop-floor assets and actors, data sources, algorithms (with emphasis on artificial intelligence (AI)) and decision-making opportunities as well as their relations. This ontology can be used to enhance DTs with cognitive capabilities and instantiated to a KG to provide meaningful context to data and algorithm outcomes, enhancing decision-making suggestions. We describe this through two use cases for an automotive parts manufacturing plant in Europe.

Keywords: Actionable Digital Twin · Knowledge Graph · Smart Shop Floor

1 Introduction

Most goods we consume are manufactured in manufacturing plants. These are organized in buildings with shop-floors - areas devoted to machines and tools operated by workers to produce goods as established in production plans by their leaders and managers according to expected demand. The increasing digitization of all aspects of manufacturing allows for greater optimization of the production process and is becoming a requirement for competitiveness. A part of this digitization process is the elaboration of digital twins (DT).

A DT can be defined as "a virtual model of a real product, process or service that can monitor, analyze and improve its performance" [19] as well as to "derive solutions relevant for the real system" [2]. This definition is extended to consider a systemic perspective, by composing DTs into higher abstraction levels [20]. With each abstraction level, we gain new context by getting insights into relationships between elements and other information relevant to that level. Such a systemic abstraction is the shop-floor. DTs are designed in such a way that they encapsulate meaningful data (properties) and behavior (operators exposed through protocols, which make them actionable [11]) and are specific to them. By doing so, responsibility is delegated to the component with most proximity and knowledge to a given problem.

Many authors realize the potential of semantics in the domain of DTs since this approach proved to be effective in many contexts in the past [16, 15]. Boschert et al. [3] describe how semantic technologies can be leveraged in the NextDT paradigm to connect multiple DTs into a single value network and make use of operational data with DTs to offer a wide variety of services. Kharlamov et al. [12] identify four challenges that, in their opinion, should be solved to take full advantage of semantic models in the DTs context. These challenges are how to deal with high-volume streaming and historical data in a semantic context, provide integration of semantic models with analytical solutions, semantically link simulations to specific use-cases and how to learn semantic models over time. Cho et al. [4] understand that one of the main issues of using an ontology in the context of DTs is that the model should be up-to-date to provide value on decision making. They propose an approach based on Gaussian Mixture Models (GMMs) to identify patterns of incoming data and understand if it can be classified into existing classes or it provides new knowledge that should be included in the ontology. Banerjee et al. [1] developed a pipeline to extract semantic relations from sensor data, focusing on features that can be built based on the type of incoming data and information provided by an ontology model to insert them into a KG where relations can be inferred and knowledge queried using a semantic querying mechanism. Another approach was developed by Zehnder et al. [22] who proposed Industrial Data Streams as a novel approach to model DTs by abstracting data streams into virtual sensors and labeling them with semantic tags that allow to describe data characteristics and provide information on how can be grouped and used.

This paper focuses on a new proposed concept, actionable Cognitive Twins (CTs) supporting decision-making in manufacturing systems, particularly on shop-floor compositions. The main contribution of this paper is the development of a novel ontology that models shop-floor DTs with entities that describe physical assets and actors as well as how data is ingested into the digital counterpart, leveraged by algorithms and AI, and how are their outcomes linked to advice on potential actions that can be taken to mitigate observed issues to help on decision making. We describe how it could complement an existing KG by describing two use cases of actionable shop-floor CTs for decision making in the context of a manufacturing plant of automotive parts located in Europe.

The rest of the paper is organized as follows. We first describe our approach to the actionable shop-floor cognitive twin (CT) for decision makings concept and their composition in Section 2. In Section 3, we describe a use case on how the CT was developed for an automotive components manufacturing plant in Europe.

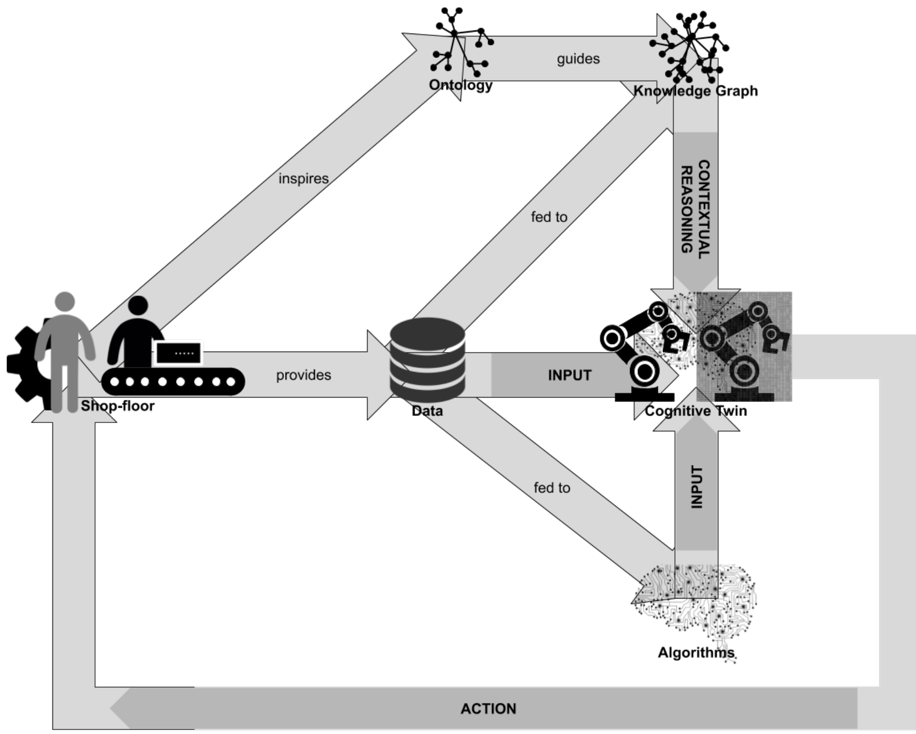

Fig. 1. Cognitive twins supporting decision-making for manufacturing by actions

Finally, we discuss CT concepts in the case study and offer the conclusions with a summary in Section 4.

2 Cognitive Twin

In this paper, we describe an actionable CT to support decision-making for manufacturing systems, as shown in Fig. 1. Regular DTs are created as digital representations of physical entities which the main differences between CTs and DTs are shown in the previous paper [9]. The CTs can be enhanced with behaviors provided by algorithms, AI models and KG models, which make them actionable. Physical and digital entities, data sources, as well as algorithms and AI models, can be abstracted into an ontology model. Thus, each CT has their own ontology for their own environments and the ontology is developed based on a unified specification with more high level abstractions. Cognition capabilities are provided by making use of the KG and AI, although their functions differ. Based on different use cases, the cognitive capabilities can learn the historical behaviors of physical entities and historical data of digital entities in order to provide the decision-makings for the operational physical entities.

The KG takes advantage of knowledge encoded in the ontology regarding shop-floor physical entities and their relationships that can be used to contextualize results obtained by algorithms on specific instances. On the other side, AI algorithms consume available data and provide some response to a proposed problem (e.g. production forecasting or anomaly detection). A subset of these algorithms (machine learning (ML) algorithms) not only consumes data but learns from it capturing knowledge regarding dynamics reflected in incoming data. For the cases we describe in this paper, we make use of the Web Ontology Language (OWL) to develop the shop-floor CT ontology.

We envision four components that define actionable CTs for shop-floor decision makings:

1. **Ontology and Knowledge Graph**: the ontology captures background knowledge about entities and their relationships in the physical world as well as their digital counterpart. It is instantiated in a KG, which brings cognitive capabilities to the DT.
2. **Data**: recorded information about shop-floor assets, actors and operations.
3. **Algorithms**: operators that provide specific behavior and capabilities to DTs. AI models provide cognitive capabilities as well.
4. **Actions**: decision-making opportunities suggested to users based on insights obtained through analytics and algorithms ran on the DT.

3 Case Study

3.1 Motivation

Shop-floor is the area of a manufacturing plant where production takes place. It contains the machines required for production and is the place where workers operate them or manage the production process. Relevant Key Performance Indicators (KPIs) to the shop-floor are Operational Equipment Effectiveness (OEE)[14] and Overall Process Efficiency (OPE)[7] among others. OEE (as seen in equation 5) measures parts produced on a machine versus its theoretical maximum capacity; while OPE (as described in equation 6) measures parts produced versus the theoretical maximum capacity, regardless the cause preventing to achieve full performance, considering not only machine inefficiencies but the process as a whole.

$$Availability = \frac{TotalHoursPlanned - LostTime}{TotalHoursPlanned} \tag{1}$$

$$Speedrate = \frac{Actual\ Machine\ Speed}{Design\ Machine\ Speed} \tag{2}$$

$$Qualityrate = \frac{Number\ of\ Good\ Products}{Total\ Products\ Made} \tag{3}$$

$$Utilization = \frac{Scheduled\ Time}{TotalTime} \tag{4}$$

$$OEE = \mathit{Availability} \times \mathit{Speed\ Rate} \times \mathit{Quality\ Rate} \tag{5}$$

$$OPE = OEE \times \mathit{Utilization} \tag{6}$$

Problems we consider in shop-floor are related to the optimization of these KPIs. The success of the proposed approach can be measured and compared against other shop-floor control systems regarding improvements over these KPIs. In this paper we will focus on two problems:

- **Problem 1: anomaly detection**: what anomalies do occur during production? How do they impact the existing production process?
- **Problem 2: production planning**: how do we re-schedule existing production plans based on factors such as early or late terminations, or lack of skilled workers? How do we mitigate potential issues that could affect operational up-time such as lack of required materials or skilled workers?

We illustrate our approach with a real scenario of a global enterprise that has more than 30 manufacturing plants worldwide. In our example, we focus on a single plant located in Europe and dedicated to the manufacturing of discrete components for the automotive industry. For our scenario, we consider two lines (Line 1 and Line 2), each of them has a machine tool (Machine 1 and Machine 2, respectively) to produce the same products (Product 1, Product 2, Product 3). Each of these production lines performs injection molding and takes care of plastic milling for good product termination. There are many workers (Worker 1, Worker 2, Worker 3, Worker 4, Worker 5) with the right competencies to operate the line (Competence 1) - for this example, one worker is required per line per shift (Shift 1, Shift 2, Shift 3). Each worker has certain seniority for a given competence (e.g.: can be junior, semi-senior or senior), which determines if certain guidance may be required. Additionally, a worker can choose which shifts would usually suit them. For the case we present, all workers are willing to work at Shift 1 and Shift 2 and for demand purposes, there is no need to work on Shift 3.

Information regarding planned quantities for a given product, workers assigned to the line, materials in stock are recorded in the Enterprise Resource Planning (ERP) system. The amount of produced units and scrap are recorded by a Manufacturing Execution System (MES) and confirmed by managers at the end of each shift. This information is later recorded in the ERP software as well. Information regarding workers, their attendance and competences are registered in a Human Resources (HR) software. All pieces of software containing relevant information regarding DTs are connected to them through data sources.

Data is ingested into the DT software though data sources (Datasource 1c, Datasource 1s, Datasource 1v, Datasource 2c, Datasource 2s, Datasource 2v) which for this example report production capacity, scrap and velocity for Machine 1 and Machine 2. Data of listed data sources is consumed by Algorithm 1 and Algorithm 2 which perform some cognitive function, for example, to understand if some anomaly is detected and propose actions (Action 1, Action 2) which may be taken regarding Machine 1 or Machine 2 to mitigate the issue detected and reduce impact on Line 1 and Line 2, respectively.

3.2 Use-case overview

In order to solve the problems stated above, we developed a shop-floor CT as a piece of software that mirrors the physical shop-floor and consists of four components: ontology and KG, data (historic and current values), algorithms and actions (suggested decisions that can be made to fix an issue). The current implementation allows to, at runtime, define data sources for components, specify relationships as well as decide on which algorithms or analyses are desired in each case. Action items are suggested based on encoded knowledge and the semantics and context of a given component.

Ontology and Knowledge Graph

The CT has a KG that stores information regarding workers, lines, and plants as well as their interactions and decision-making opportunities under different circumstances. This encoded knowledge provides semantic context to inputs and outputs (processed data, triggered events and insights) obtained through the analysis performed by different modules of the shop-floor DT. The KG is built by custom mapping data from tables and enriched with semantic knowledge captured in an ontology that describes shop-floor entities. Data is matched to those entities in order to create specific instances. The ontology may be used, in a future, to interface with other KGs that share this same convention.

Data

Data allows mapping a physical asset to its digital counterpart. In order to obtain and feed it to the system, the software provides data source abstractions to connect to an industrial manufacturing ERP, human resources management software, Internet of Things (IoT) interfaces and other sources of data. Each integration may have a different data velocity and the data sources are aware of that. One such example is data regarding production orders, their execution through shifts, goods stock, and delivery. Information regarding produced goods is introduced into the ERP system after each shift: line leaders report on partial progress and managers confirm information regarding production after each shift. This data is shortly after ingested into the platform and disseminated to subscribed modules for further processing. In every case, data points are persisted into a database. This allows accessing current or older states and configurations so that can be mirrored in the software or used for simulation purposes or train machine learning models. It also allows us to create and update specific KG instances, to have an up-to-date shop-floor representation.

Algorithms

Our shop-floor digital twin conceives algorithms as operators that provide specific behavior and capabilities to the DT representation.

Regarding Problem 1 an anomaly detector [10] runs algorithms for stream analysis, searches for anomalous behavior and alerts on them as well as on multiple observed anomalies that are considered to be correlated. Examples of such anomalies are high or low levels of produced goods regarding an expected quantity range, higher than expected levels of scrap and operational or technical downtimes regarding a specific machine or line. High or low levels of produced goods compared to the expected ones negatively impact the production process either by increasing stock costs or putting at risk agreed delivery deadlines to clients. Higher than expected scrap levels impact production costs and even production schedules since a greater amount of material is required to produce the desired goods and thus material stocks need to be reviewed based on this fact. Finally, technical downtimes affect not only production schedules but also imply costs of skilled workers paid for dead time.

Events regarding detected anomalies can activate other contextual functions based on relationships exposed in the KG. Such cases are further analysis and simulations to understand how downtimes may affect termination dates and provide expected ones as well as potential deviations.

To solve issues related to Problem 2, a production planning software module makes use of probabilistic ML and heuristics to assist supervisors when creating a production schedule in order to ensure workers are assigned to lines of their own and for which they have the required competences. Heuristics also validate constraints regarding legislation or shift preferences and ensure they are respected. The software module regularly reviews existing production plans, analyzes required materials and skilled workers to handle them and provides insights about what is needed.

Actions

Insights obtained from the modules described above are put into context within the knowledge graph, which also provides decision-making opportunities. This can be considered as advice so that by taking action in the physical world, detected issues can be mitigated. Such an example is advice provided when analysis of production plans reveals not enough materials are stocked to meet production requirements or that workers with required competencies are still to be assigned or may not be available for planned dates. The worker is advised to check if stock of material may exist but was not properly recorded in the ERP software or to issue material orders to get the required stock in time and avoid operational downtimes. Issues regarding worker assignment to production lines may be fixed either by making the corresponding assignments or reordering the production plans. If a shortage of workers with a certain competence is regularly observed, the issue may be mitigated by hiring people with the required competencies as well as by training some of the current workers to acquire it.

3.3 Ontology and Knowledge Graph design

In order to develop KG models for the case study, an ontology is defined which describes shop-floor DT entities. The main focus of the constructed ontology is to create a unified description for sharing and reusing knowledge about use cases described considered in this paper. It was constructed following the steps enumerated below:

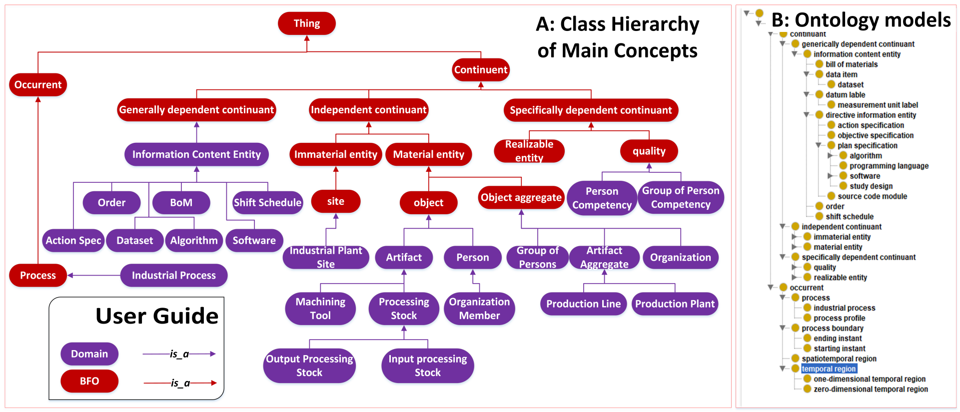

Fig. 2. Knowledge graph modeling for decision-makings in the manufacturing systems

1. Define the use case.
2. Identify the ontology concepts for class hierarchy development in the use case.
3. Identify the interrelationships between ontology concepts.
4. Construct the KG models. Tools such as Protege [18] can be used to this end.
5. Develop Application Processing Interfaces (APIs) for updating KG models and access encoded knowledge.
6. Extend KG models and APIs to other DT domains.

After analyzing the use case, the ontology is developed based on Basic Formal Ontology (BFO) [17] and Industrial Ontologies Foundry (IoF) concepts [13]. The key ontology concepts are introduced in Table 1. Based on the key concepts, object properties are defined to construct the interrelationships between different entities. As shown in Figure 2, the ontology is first clarified by two types: 1) Occurrent, entities that occur or happen; 2) Continuent, entities that continue or persist through time. The occurrent entities include a process that is specified by the industrial process (the operational process in the case study). The red nodes refer to the basic compositions of BFO. The purple nodes refer to the domain-specific definitions for this case study.

Except for occurrent, the continuent entities include: 1) Generally dependent continuant, entity specifically dependent on another if another cannot exist, it does not exist; 2) Independent continuant, referring to the entities existing on themselves; 3) Specifically dependent continuant, a continuent entity depends on one or more specific independent continuants for its existence. In summary, the *industrial process* and *machining tools* are implemented by *orders* with a *Bill of Materials (BoM)* during *shift schedule*. In each *industrial process*, the *process stock* with specific *materials* (object properties) is processed using one *machine tool* operated by a *person* at the *industrial plant site*. The person who has *competency* is under an *organization*. The *machine tools* construct the *production lines* which are compositions of *production plants*. In order to support AI algorithm development, the *dataset* is defined to represent the data generated from the *machine tool*. Such *dataset* support *algorithm* development which is implemented in software. Finally, the software provides *actions* to be taken in the physical world (e.g: to control *machine tools*).

The ontology[8] is modeled in Protege as shown in Fig. 2-B. Concepts are defined as OWL classes. The interrelationships between classes are defined as object properties. Attributes of ontology concepts are defined as data properties. Based on the case study, individuals are developed to represent the information of the case study. As described in Section 3.1, two individuals of *production lines* are defined as *Line1* and *Line2* with attribute *Competence1*. Each of the production lines has its own machine: *Machine1* and *Machine2* defined as individuals of the *machine tool*. All their products refer to individuals of *Output Processing Stock* as *Product1*, *Product2*, and *Product3*. The five workers are defined as individuals of *person* as *Worker1-Worker5* with their own competences (individuals of *person competence*: *Competence1*) and seniority (*person competence seniority: junior,semi-senior and senior*). Each worker operates the *Line1* and *Line2* on individuals of *Shift schedule* as *Shift1-Shift3*.

Except for describing the manufacturing systems, the data flow for AI algorithm during the operational process is defined as well. Datasources to ERP for *Machine1* and *Machine2* are defined individuals of data source: *DS1c*, *DS1v*, *DS1s*, *DS2c*, *DS2v*, and *DS2s*. The data sources are used for *Algorithm1* and *Algorithm2* (individuals of *Algorithm*) which is implemented in the *software1* (individual of *Software*) to provide decisions for *Machine1* and *Machine2* as individuals of *Action*: *Action1* and *Action2*.

Table 1. Main Ontology concepts and their attributes

Entity	**Attribute**
Production Plant	id, name
Production Line	id, name, plant id
Process Stock	id, title, materials
Industrial Process	id, name, line id, persons per shift, material id, plant id, delay type, delay time
Machining Tool	id, name
Production order	id, creation date, earliest start date, earliest end date, person time, machine time, clearing time, BoM version
BoM - Bill of Materials	BoM version, component version, input material id, input material quantity, plant id, output material id, output material quantity
Competences	id, name, description
Person	id, name, active
Shift schedule	id, plant id, shift number, week, year, shift start time, shift end time, shift date
Data source	id, title, URL
Algorithm	id, title, algorithm type
Software	id, title
Action	id, title, decision

3.4 Use of Artificial Intelligence

Artificial Intelligence can be considered as a subset of algorithmic functions described above. It involves heuristics and algorithms capable of learning from data in order to achieve a certain objective.

In order to solve Problem 1, the anomaly detector considers streams of data and two algorithms to understand if a data point should be considered an anomaly. The first one is a threshold set in the KG so that any value surpassing it is considered anomalous. The second one uses a t-digest data structure [5] to efficiently compute quantiles and will consider a new value as an anomaly if it corresponds to $q > 0.99$. The KG is leveraged to understand which streams reflect different aspects of the same reality. Considering that information in addition to time proximity of detected anomalies, the software can identify which anomalies may be correlated and provide this insight to the users. Depending on the context, decision-making opportunities regarded as *Action* instances can be retrieved and served to the users as well.

In the case of production planning in Problem 2, we run Monte Carlo simulations [21, 6] based on historic data to perform repeated random sampling based on existing production information in order to deliver most probable termination dates as well as expected deviations due to uncertainty. Simulations are run on a regular basis, taking into account state updates from the physical world in order to provide the most accurate expected status to end-users. Given results are contrasted with expected termination dates in order to understand if production will be finished early, on time or late and provide contextual decision making suggestions based on these insights and leveraging KG encoded knowledge.

4 Conclusion

The increasing digitization of manufacturing processes is leveraged to create actionable cognitive twins, where not only the state of physical assets is mirrored, but algorithms are used to provide behavior through heuristic and AI models. In this paper, we present how actionable shop-floor DTs can be further enhanced with cognition capabilities. We propose an ontology[8] that encodes background knowledge regarding shop-floor entities, their relationship to data sources, algorithms and decision-making opportunities based on algorithm outcomes.

For future work, the current KG module can be enhanced with new entities and relationships required to support new use-cases. In particular, we would like to address demand forecasting, a relevant problem whose outcomes impact the whole production process. In order to achieve that, we may need to enrich the current model with understanding on how particular data should be treated in order to obtain required features and how this features can be fed to train complex ML models as well as provide semantics to contextualize forecasted results within company production lifecycle and global context. We also embrace the possibility of adding a reasoning module, that would bring new capabilities regarding how knowledge captured in KG may be used and augmented.

Acknowledgement

This research was funded by European Union's Horizon 2020 programme project FACTLOG (Innovation Action: Energy-aware Factory Analytics for Precess Industries) under grant agreement number 869951.

References

1. Banerjee, A., Dalal, R., Mittal, S., Joshi, K.P.: Generating digital twin models using knowledge graphs for industrial production lines. UMBC Information Systems Department (2017)
2. Boschert, S., Heinrich, C., Rosen, R.: Next generation digital twin. In: Proc. TMCE. pp. 209–218. Las Palmas de Gran Canaria, Spain (2018)
3. Boschert, S., Heinrich, C., Rosen, R.: Next generation digital twin. In: Proc. TMCE. pp. 209–218. Las Palmas de Gran Canaria, Spain (2018)
4. Cho, S., May, G., Kiritsis, D.: A semantic-driven approach for industry 4.0. In: 2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS). pp. 347–354. IEEE (2019)
5. Dunning, T., Ertl, O.: Computing extremely accurate quantiles using t-digests. arXiv preprint arXiv:1902.04023 (2019)
6. Huang*, S.H., Liu, Q., Musa, R.: Tolerance-based process plan evaluation using monte carlo simulation. International Journal of Production Research **42**(23), 4871–4891 (2004)
7. Ivancic, I.: Development of maintenance in modern production. In: Euromaintenance'98 Conference Proceedings. pp. 5–7. CRO Dubrovnik/Hrvatska (1998)
8. Jinzhi, L., Dimitris, K., Rožanec, J.M.: Shop-floor DT ontology (2020). https://doi.org/10.7910/DVN/RMFVWX, https://doi.org/10.7910/DVN/RMFVWX
9. Jinzhi Lu, Xiaochen Zheng, A.G.K.K.D.K.: Cognitive twins for supporting decision-makings of internet of things systems. In: Proceeding of 5th International Conference on the Industry 4.0 Model for Advanced Manufacturing (2020 (In press))
10. Jovanoski, V., Rupnik, J.: Fsada, an anomaly detection approach
11. Kahlen, F.J., Flumerfelt, S., Alves, A.: Transdisciplinary Perspectives on Complex Systems. Springer (2017)
12. Kharlamov, E., Martin-Recuerda, F., Perry, B., Cameron, D., Fjellheim, R., Waaler, A.: Towards semantically enhanced digital twins. In: 2018 IEEE International Conference on Big Data (Big Data). pp. 4189–4193. IEEE (2018)
13. Kulvatunyou, B., Wallace, E., Kiritsis, D., Smith, B., Will, C.: The Industrial Ontologies Foundry Proof-of-Concept Project. In: IFIP Advances in Information and Communication Technology, pp. 402–409 (2018). https://doi.org/10.1007/978-3-319-99707-0_50
14. Nakajima, S.: Introduction to tpm: total productive maintenance.(translation). Productivity Press, Inc., 1988, p. 129 (1988)
15. Paulheim, H.: Knowledge graph refinement: A survey of approaches and evaluation methods. Semantic web **8**(3), 489–508 (2017)
16. Rotmensch, M., Halpern, Y., Tlimat, A., Horng, S., Sontag, D.: Learning a health knowledge graph from electronic medical records. Scientific reports **7**(1), 1–11 (2017)

17. Smith, B., Grenon, P.: Basic Formal Ontology. Draft. Downloadable at http://ontology.buffalo.edu/bfo (2002)
18. Stevens, R., Hancock, J.M.: Protégé. In: Dictionary of Bioinformatics and Computational Biology. John Wiley & Sons, Ltd, Chichester, UK (oct 2004). https://doi.org/10.1002/9780471650126.dob0577.pub2, http://doi.wiley.com/10.1002/9780471650126.dob0577.pub2
19. Tao, F., Sui, F., Liu, A., Qi, Q., Zhang, M., Song, B., Guo, Z., Lu, S.C.Y., Nee, A.: Digital twin-driven product design framework. International Journal of Production Research **57**(12), 3935–3953 (2019)
20. Tao, F., Zhang, H., Liu, A., Nee, A.Y.C.: Digital Twin in Industry: State-of-the-Art. IEEE Transactions on Industrial Informatics **15**(4), 2405–2415 (apr 2019). https://doi.org/10.1109/TII.2018.2873186, https://ieeexplore.ieee.org/document/8477101/
21. Ulam, S.M.: Monte carlo calculations in problems of mathematical physics. Modern Mathematics for the Engineers pp. 261–281 (1961)
22. Zehnder, P., Riemer, D.: Representing industrial data streams in digital twins using semantic labeling. In: 2018 IEEE International Conference on Big Data (Big Data). pp. 4223–4226. IEEE (2018)