IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received September 3, 2019, accepted October 2, 2019, date of publication October 30, 2019, date of current version January 3, 2020.

*Digital Object Identifier 10.1109/ACCESS.2019.2950507*

# A Feature-Based Framework for Structuring Industrial Digital Twins

**JUUSO AUTIOSALO, (Student Member, IEEE), JARI VEPSÄLÄINEN, (Student Member, IEEE), RAINE VIITALA, AND KARI TAMMI, (Member, IEEE)**
Department of Mechanical Engineering, Aalto University, 02150 Espoo, Finland

Corresponding author: Juuso Autiosalo (juuso.autiosalo@aalto.fi)

This work was supported in part by the Business Finland under Grant 8205/31/2017 "DigiTwin," and in part by the Academy of Finland under Grant 313675 "TwinRotor".

**ABSTRACT** Digital twin is a virtual entity that is linked to a real-world entity. Both the link and the virtual representation can be realized in several different ways. However, the ambiguous meanings associated with the term digital twin are causing unnecessary miscommunications as people have different interpretations of what can be accomplished with it. To provide clarity around the concept, we introduce a general approach to analyze and construct digital twins in various applications. We identify the common features of digital twins from earlier literature and propose an analysis method that compares digital twin instances based on these features. The method is used to verify the existence of the features and can be further enhanced. We formulate the observations to a feature-based digital twin framework (FDTF) to universally define and structure digital twins. The framework consists of three main principles: i) the idea that all digital twins consist of a definite set of features, ii) the features can be used to compare digital twin instances to each other, and iii) the features can be combined via a data link feature to construct future digital twins more efficiently. As key contributions, we found that the features can be identified in existing digital twin implementations and the feature combinations of the implementations are diverse. We suggest that the features should be leveraged to provide clarity and efficiency in digital twin discussion and implementation. We further propose a general procedure for building digital twins.

**INDEX TERMS** Digital twin, enterprise systems, Industrial Internet of Things, cyber-physical systems.

## PROPOSED TERMINOLOGY

DT = digital twin
DTF = digital twin feature
DTI = digital twin instance
DTC = digital twin class
DTB = digital twin block
FDTF = feature-based digital twin framework
FEDA = feature-based digital twin analysis

## I. INTRODUCTION

Internet of Things (IoT), cyber-physical systems (CPS) and digital twins (DT) are considered to form the next generation of digitalized industry among other recent trends. While the definition of each of these concepts is more or less vague, the ambiguous nature of the DT term seems to be creating a particularly vast amount of confusion. The existing DT implementations are fundamentally different from each other, adjusting to the needs of each use case and created with a wide variety of tools. Especially, there seems to be an unfruitful competition between modeling oriented and information management oriented views of the DT concept. The former view spurs from deeply technical engineering issues with the aim to mimic the exact physical environment whereas the latter focuses more on semantic connections and seamless information flow. The division appears between the disciplines of information technology and engineering.

The main purpose of a DT is to act as the single source of information for its real-world counterpart. It links different systems on product level and it is used to structure, monitor and exploit data. The added value comes from linking the information of multiple features and systems to assist and update one another in real-time while providing the information to the user conveniently from a single access point.

The associate editor coordinating the review of this manuscript and approving it for publication was Shiqiang Wang.

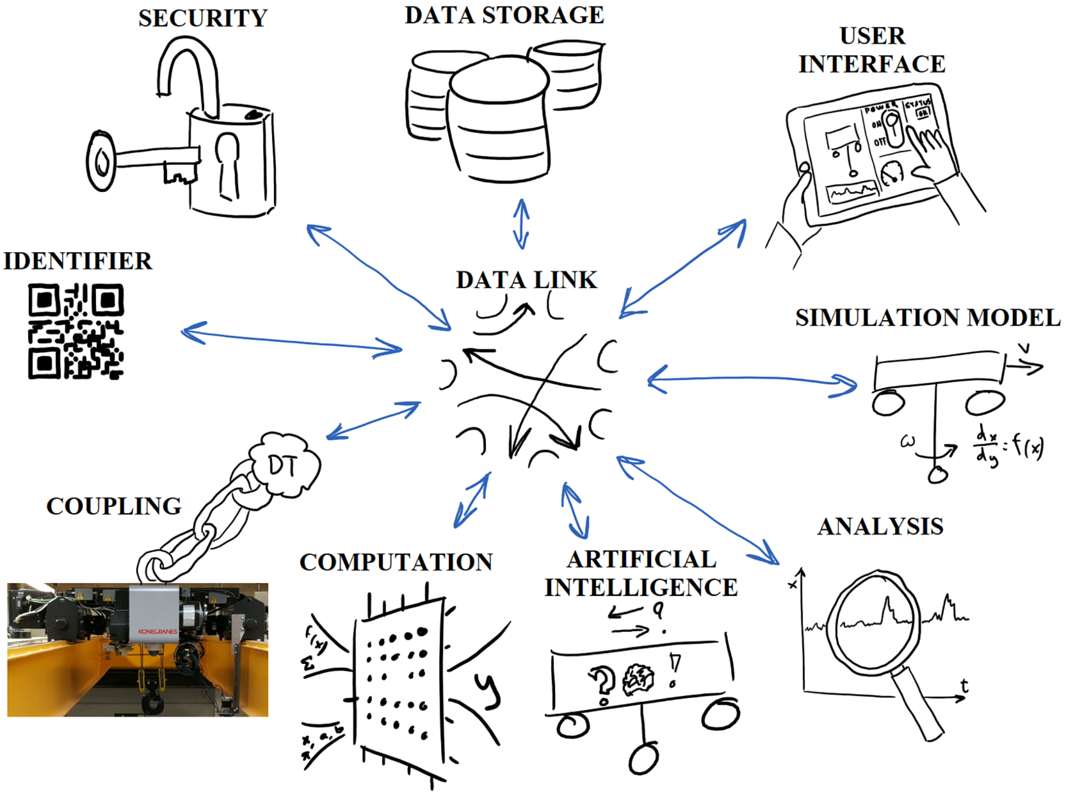

**FIGURE 1.** Conceptual ideal of feature-based digital twin framework.

The digital twin related literature is clearly fragmented, which brings up the need to answer three fundamental research questions: 1) What is a digital twin? 2) How to compare digital twins? 3) How to build a digital twin? The current inability to answer these generally makes the full potential of the DT concept invisible. To overcome this issue, this paper presents a novel abstract level definition that systematically conveys the whole potential by introducing a general structure for DTs. This paper is written and the framework is developed to reflect the mechanical engineering perspective. The scope of the study is to advance two aspects that are especially important in machine design, product information availability and closed-loop product lifecycle management, by introducing a general structure for DTs. Thanks to the general approach, the implications may prove useful in other fields as well. The generality of our approach also differentiates this study from other studies; we link the mechanical engineering viewpoint of DTs to other fields, especially to information technology.

We present three main contributions in this paper. First, we systematically identify common features of a DT, including a general link feature between them (Fig. 1). Second, we analyze the presence of these features in existing publications. Finally, we formulate the features into a novel framework that can be used to categorize existing digital twin implementations and as general guidance for implementing future digital twins. Furthermore, the paper unifies the use of the term "digital twin" and the framework can be used as a design tool to analyze and define a DT system. The framework intends to combine and simplify existing ideas and phenomena, rather than to introduce new complexities.

The paper is structured as follows: Section II presents the background and related work from literature, followed by Section III that describes the used research methods. Section IV introduces the ten identified features of DTs and Section V presents the execution of the FEDA method to compare seven DT implementations. Section VI presents the FDTF framework and Section VII discusses its implications. Section VIII concludes the study.

## II. BACKGROUND AND RELATED WORK

According to current consensus in scientific literature, the term "digital twin" was coined 2010 in a draft strategic roadmap of NASA [1], even though the term was used months earlier by Puig and Duran to describe the digital avatar of a human [2]. The term was also visible in figures of Nicolai *et al.* [3]. The term digital twin does not have a unanimous definition and previous researchers have also used different phrases for similar concepts [4]. Hence, we do not limit our investigation to a single term while reviewing the origins of the digital twin concept.

Virtual representations of real objects have existed for ages; they just have not had a link that would connect the two. For example, the first occurrence of the exact word pair "virtual counterpart" on the Scopus database points to a research published in 1994 that twinned real and virtual rooms to study human perception [5]. Similarly, NASA has been claimed [6], [7] to have used a mirrored system of a space shuttle during the Apollo program missions, although in this case, the twin was physical. The older links between physical product and its digital representation, such as a simulation model, were implemented and maintained manually. Even the original proposal for the World Wide Web from 1989 [8] mentions physical objects as potential nodes of the information network that we today consider as the Internet. With the introduction of *radio frequency identification* (RFID), the link became more feasible and the term *internet of things* was introduced [9].

The basic concept of accompanying a real object with a virtual counterpart has existed from the very early days of IoT [10]. Soon after, similar concepts were presented by different parties: Grieves [11], [12] used term mirrored spaces model, Främling *et al.* [13] developed product agents, and Hribernik *et al.* [14] introduced product avatars.

Grieves focused mainly on the high-level Product Lifecycle Management (PLM) concept in his two books [15], [16]. He worked with multiple aerospace organizations to bring together Systems Engineering and PLM [17]. Later, Schluse *et al.* [18] combined Systems Engineering with digital twin to enable complete system-level simulations.

In the aerospace industry, digital twin has been defined as a tool to analyze wear and fatigue as accurately as possible during the lifetime of an aircraft [19]. The analysis is enabled by the concept of *digital thread* that ties multidisciplinary models together to form one master model of the whole aircraft. These high-accuracy models require a huge amount of computational resources and the development effort required for creating them is enormous. West and Blackburn [20] estimated that the cost of developing such a robust digital twin model for next-generation aircraft would equal to the cost of the Manhattan Project.

The amount of DT focused publications has increased exponentially in the last few years and many parties have made digital twin instances of their own. Negri *et al.* [21] reviewed the term and focused on acknowledging the roles of DTs in the manufacturing field. They listed usages that vary from detecting and predicting failures to general lifecycle management and virtual commissioning. Tao *et al.* [22] reviewed state-of-the-art of DTs in industrial context from 50 publications, 8 patents, and company releases, concluding that prognostics and health management is the most popular application area along with production, design and other areas.

Regardless of the active research on the field, the DT related literature seems to lack a systematic approach and is short of reference implementations and frameworks [23]. More precisely, there is no consensus for a generic way of modeling DTs regardless of multiple implementations, and the cyber-physical fusion lacks an universal framework [22]. Insights on how different types of digital instances or blocks should be linked to each other have not been presented. In other words, a systematic approach is needed to define the digital twin as an entity that consists of features and their connections.

## III. METHODS

The research was initiated by the lack of unified methods to compare and construct DTs. As an initial measure, we present existing and proposed terminology. To identify further structure in the clutter of DT related literature, we relied on the well-established Grounded Theory. The Grounded Theory is a set of guidelines to develop new data-based methods that characterize the data in a simple and concise matter. We propose a new holistic scoring system for digital twins to analyze their structures in a quasi-quantitative way. To test and validate the proposed approach, we applied the method to a set of DT implementations and performed correlation analysis on the results.

### A. TERMINOLOGY

The terminology in the current digital twin related literature can be confusing and even contradictory, as can be observed from the list of DT related terms and abbreviations shown in Table 1.

**TABLE 1.** Digital twin related terms and abbreviations found in literature. Evidently, some of the presented terms and abbreviations are unstandardized or contradictory and are therefore not used in this paper. The descriptions can be found in their sources; explanations are left out for conciseness.

| *Term* | *Abbreviation* | *Used in* |
|---|---|---|
| Experimentable Digital Twin | EDT | [18], [24] |
| Digital Twin Shop Floor | DTS | [25], |
| Digital Twin Instance, Prototype, & Environment | DTI, DTP, DTE | [12] |
| Digital twin-driven product design | DTPD | [26], |
| Airframe digital twin | ADT | [27]–[29], |
| Digital thread | DT | [30], [31] |
| Digital twin | DTw | [32]–[34], |
| Digital twin workshop | DTW | [35] |
| Digital Twin-driven Smart ShopFloor | DTSF | [36], |
| Next Generation Digital Twin | nexDT | [37]. |

An exception in Table 1 are the definitions introduced by Grieves and Vickers [12]. We wish to build on their work and generalize the terminology to the fit needs of engineers while they are describing DTs. To fulfill this goal, some nuances must be updated to ensure the fit to future directions of the DT concept, including the framework presented in the present study.

We propose the following terminology:

–*Digital twin* (DT) is a virtual entity that is linked to a real-world entity. It describes a planned or actual real-world object with the best available accuracy. The information can be distributed among different systems, but the pieces of information should be linked to each other to form one coherent entity. The term digital twin serves as a common noun for any kind of digital twin object. To provide an accurate description, additional attributes must be used.

–*Digital twin instance* (DTI) represents the virtual counterpart of a specific real-world object. The object can be a physical product, human, city, process, or event; anything that benefits from being accompanied by a virtual representation or servant. As a DTI is the single common interface for the data of a real-world object, a DTI must be constantly available on the Internet to ensure the constant flow of data. Each DTI also has a unique identifier that can be used to connect to it from anywhere around the world. The definition of DTI in [12] is consistent with our definition.

–*Digital twin block* (DTB) is a sub-system of a DTI. A DTB is an independent software entity that can be connected to other DTBs to form a DTI. Building blocks for digital twins have been called for in [38] and developed in [39]. Boschert *et al.* [37] described DT as a collection of selected digital artifacts.

–*Digital twin class* (DTC) is a tool to create DTIs, similarly to how *object-oriented programming* uses classes as tools to create object instances. This idea has been presented earlier by *Hribernik et al.* [14] and DTC means the same as their *parent product avatar*. DTC has also similarities with the concept of DT prototype defined by [12], although the DT prototype emphasizes the activities of the development phase of new products, whereas DTC emphasizes the creation of DTIs.

–*Digital twin feature* (DTF) is a common noun for different types of technical functionalities of DTs. It is an additional layer of abstraction that facilitates the transition from functional requirements to technical implementation. Further implications and examples of DTFs are provided throughout the present study. Similarities can be found with the DT characteristics described by El Saddik [40].

–*Network of digital twins* represents a network that leverages DTIs as nodes. As DTIs are constantly available and provide links to DTBs, the network of DTs offers a connection to any real-world object at any given time. This type of network yields the classical IoT problems that physical products very often have limited communication capabilities and their data is hidden in proprietary systems. The concept of a network of DTs has been presented earlier in [41].

### B. GROUNDED THEORY

Grounded Theory is used as the methodology for developing the novel methods introduced in this study. Grounded Theory was originally presented by Glaser and Strauss [42] and is conveniently summarized with recent trends by Saunders *et al.* [43]. The basic idea of the Grounded Theory methodology is to produce theories that are grounded on data. Core elements include categorization of data, finding relations between the categories, and finally integrating the categories to develop a theory. Grounded Theory demands simultaneous data collection, analysis, and theory development right from the initiation of the research process.

This study does not focus on following the sophisticated Grounded Theory procedures as strictly as possible. We leverage only the fundamental origins of Grounded Theory rather than any of its specific procedures. This approach is somewhat justified by [42] and [43]. In practice, we use Grounded Theory to draw basic conclusions about the nature of DTs in current literature.

We employ the Grounded Theory methodology as follows. The main data used for the research are the DT use cases and implementations found in the literature. Other related literature is used as supportive data. The data were gathered and analyzed concurrently with theory development. Instead of pursuing statistical coverage, data sampling and theory saturation procedures used in this study aim to prototype a new theory. Theory development was initiated with the categorization of data, which led to the identification of DTFs. Relations between these categories eventually led to the discovery of two theories, an analysis method and a general framework.

### C. FEATURE-BASED DIGITAL TWIN ANALYSIS

As a result of following the Grounded Theory, we present a simple yet novel method, *feature-based digital twin analysis* (FEDA), for comparing and categorizing DT implementations. The usage of the method consists of two major phases. First, a DT is analyzed to define the implementation level of each DTF as numerical value $v_i$. Second, a holistic score $Hs$ for the DT is calculated based on the previously assigned grades $v_{1\ldots i}$. The holistic score is defined as

$$Hs = (\Sigma w_i \cdot v_i / \Sigma w_i \cdot v_{max}) \cdot \ Sc, \tag{1}$$

where $w_i$ is the preferred weight factor of each feature, $v_i$ is the numerical value given in the range $0\ldots v_{max}$ for each feature of a DT implementation, $v_{max}$ is the maximum value of each $v_i$, and $Sc$ determines the scale of $Hs$ as range $0\ldots Sc$.

FEDA provides two outputs. First, the set of values $v_i$ forms a feature profile for each DT. The profile can be assessed as such when dealing with a low number of DTs, it can be drawn graphically (e.g. as radar chart) for visually intuitive representation, or further analysis (e.g. correlation analysis) can be performed especially for large data sets. Secondly, FEDA delivers the execution level of the analyzed DT implementation as the single numerical value $Hs$. The current method provides user-specific grades and can be further refined to give universal grades.

To apply the FEDA method, follow these five steps:

Step I: Identify the DT features that will be included in the analysis.

Step II: Define the grading method for the features, including the maximum value $v_{max}$ for the grades $v_i$.

Step III: Assign the grades for the features of a DT implementation.

Step IV: Define weight factors $w_i$ values for each feature and the scale $Sc$.

Step V: Calculate the holistic score $Hs$.

### D. CORRELATION ANALYSIS

Correlation is the normalized linear similarity between two factors. The normalization bounds the maximum value to 1 and minimum to −1. The value 1 corresponds to a strong positive relationship, meaning that the factors go hand in hand across cases; if one factor is high, so is the other and so on. With a strong negative correlation, the relationship is inverse. However, even if two factors are correlated, it does not guarantee a causal relationship. To prove causality, further support for the claim is needed based on isolated tests, experience or previously proven dependencies, such as the laws of physics. Correlation analysis is utilized here as a tool to understand the relations and hierarchy between DTFs. Inconsistent correlation relations between DTFs or ones that cannot be argued to have causality can be considered independent of other DTFs in the DT application.

## IV. DIGITAL TWIN FEATURES

Existing DT implementations and use cases found in the literature have revealed several distinct categories of technical functionalities, which we refer to as digital twin features (DTFs). Even though the features are clearly present in many implementations, publications that distinctively list DTFs are difficult to find. Nevertheless, we can find many other types of lists on DTs and we used these lists as base material for our study. To showcase the state of the literature, we here present four DT related lists.

1. Rios *et al. [44]* presented five most important topics for the digital twin of a product in the aeronautical sector as: “*product identifier, product lifecycle, product information, product configuration, and product models*”.
2. Tao *et al.* [45] identified challenges of the DT concept to be in the following aspects: “*intelligent perception and connection, virtual modeling, running simulation and verification, digital twin data construction and management, digital twin-driven operation technology, smart production, and precision service*”.
3. Schroeder *et al.* [46] list the most relevant topics for creating a digital twin as “*identification, data management, product models, human computer interface, and communication*”.
4. El Saddik [40] described seven digital twin characteristics: “*unique identifier, sensors and actuators, AI, communication, representation, trust, and privacy and security*.”

These lists, along with other DT literature, support the existence of DTFs presented in this section. Based on literature, we introduce the following list of distinguishable features that can exist in the DT of a single product:

A) data link,
B) coupling,
C) identifier,
D) security,
E) data storage,
F) user interface,
G) simulation model,
H) analysis,
I) artificial intelligence, and
J) computation.

The features are depicted in Fig. 1. We do not claim this to be an exhaustive list or the only correct way to categorize DT functionalities. Instead, it is a result that emerged from our extensive data exploration and can be used as a reference list of DTFs.

The listed features may not exist at the same hierarchical level. For example, a simulation model is a high-level application that builds on other features and provides usable insights, whereas computation is a low-level attribute to be exploited by other features. Nevertheless, there are reasons why each of these is presented as a DTF. The present study attempts to convey those reasons to the reader by first describing the features and then providing use cases on how the feature division can be leveraged to benefit technical solutions.

### A. DATA LINK

The *data link* feature stems from the very essence of the DT concept. The basic idea of DT is quite straightforward, linking a physical thing to a digital thing. However, the structure of the idea is more challenging to define. Several definitions have been proposed [1], [4], [6], [7], [12], [19], [21], [24], [27], [44]–[51] and each of them is valuable in their respective use cases. However, these definitions serve their research case which leads to an unintentional lack of generalization.

To generalize the structure of DT and to interconnect individual DTFs, a “motherboard” of DT has to be defined. We call this “motherboard” the data link. The purpose of a data link is to act as a hub for all information that is related to the physical twin. The data link feature connects digital things to each other and leaves the digital-physical connection for the coupling feature defined in the next subsection.

The idea of a data link has been implemented previously for example by using MQTT [52]. We here describe our intention of how the data link should be implemented to reach both enough technical functionality and worldwide popularity. We position the link between DNS and specific protocols such as OPC UA, MQTT, REST, SOAP, and O-MI. One main requirement of the data link is that it has to be compatible with the existing internet browsers. This compatibility will promote the adoption of DTs as more people will be able to use them with a browser they are familiar with.

The data link allows the transformation from the grid type communication to the star form communication by adding a product agent in the middle [14]. The change from a grid network to a star network with an Internet-enabled agent in the middle was introduced earlier in the field of logistics [53]. Regarding operations between multiple data links, compatibility with the advancements of the semantic web [54] should be achieved. To ensure applicability among diverse use cases, specialized ontologies have been developed e.g. in the smart energy sector [55] and for smart manufacturing [56].

### B. COUPLING

We use the term *coupling* to represent the connection between a physical product and its DT. The physical product is undoubtedly an essential factor when dealing with any DT. However, there is currently no strong consensus on whether a physical product is actually a part of a DT or not. Tao *et al.* [26] proposed that a *digital twin mode* consists of three parts, one being the physical entity. Zheng *et al.* [23] also list the physical space as one of the three main components of a DT system. Many researchers [45], [52], [57]–[59] refer to the physical product as the physical twin. This choice of words indicates that the real-world counterpart is parallel to the DT, not a part of it. The current study follows this view.

Despite being separate from the actual DT, the physical product is such an important topic that it is clearly entitled to have a presence among the features of DT. In fact, the connection to the physical product has been given so much value that it can make the distinction between a regular simulation model and a DT. Hence, we describe the *coupling* as a feature of a DT.

The coupling is a two-way interface between the physical product and the DT. Through the coupling, the physical twin delivers data to the DT or the DT may control the physical product. The gateway between the physical product and the DT is enabled by an identifier.

### C. IDENTIFIER

The identifier of a DT is divided into two basic categories: physical identifier and digital identifier. Physical identifiers represent the identifier in physical space, linking the physical space to the digital space. Hence, physical identifiers enable local access to the DT, serving as a gateway between the physical products and their DTs. As a statement of the importance of the physical identifier, the IoT concept originated from the development of certain physical identifier technology, the RFID tag [10], [60], [61]. This technology enabled a new way to link the digital and physical worlds.

As a distinction to the physical identifier, a digital identifier is a way to connect a DT to a network. The digital identifier has two requirements. It should i) be unique at a sufficient level, optimally globally, and ii) enable access to the DT from any part of the DT network, optimally the Internet. The most prominent alternative for the digital identifier is the uniform resource identifier (URI) [62], which is the parent category of the common URL addresses.

URIs are being commonly used by DOI and Arxiv, along with other providers. The DOI is mainly used in scientific literature, although the standard ISO 26324:2012 [63] describes it as a general identifier for any kind of digital, physical, or abstract object.

### D. SECURITY

The generic domain of computer security is a well-established field with plentiful literature and standards. A survey [64] over 45 years ago lists the basic concepts that are still valid today. However, the specific area of cybersecurity has emerged only fairly recently. As a distinction to traditional computer security, cybersecurity focuses on the challenges that arise from highly connected information systems (i.e. the cyberspace). Chou *et al.* [65] classified cyberspace related security risks in different categories: inherent risks, technology & policy weaknesses, unauthorized intruders and legal issues. The importance of these challenges has been acknowledged at the governmental level, as e.g. the UK, the USA, Finland, Germany, and the Netherlands have established their own cybersecurity organizations [66].

Even though the field of cybersecurity has advanced recently, the cyber-*physical* security required by DTs is a newfound research topic. Humayed *et al.* [67] reviewed the possible threats and vulnerabilities of upcoming technologies. They noted that the CPSs are especially vulnerable due to their heterogeneity. Proper implementation guidelines or standards are missing. Though, the U.S. Department of Homeland Security is actively working towards secure cyber-physical systems [68].

Safe and consistent operation are vital requirements when adopting any digital system, and the DTs are no exception. The DTIs of the future operate deep in cyberspace and therefore must leverage cybersecurity in addition to traditional computer security. For DTs to become truly secure, the security aspects must be embedded to the DT itself, following the *security by design* principle.[1] To identify the appropriate security level for each DT use case, a risk analysis should be performed. Lagus [69] conducted a preliminary investigation for the information security requirements of a digital twin, including a risk assessment for an overhead crane at Aalto University premises.

### E. DATA STORAGE

DTs store data in a variety of methods and locations. While the approaches for small amounts of data differ significantly, large amounts of data are stored in specific databases to enable fast and easy access. The current database implementations store data as block-based, file-based, or object-based formats to keep information in an accessible and systematic order. Along with the lower-level data storage technologies, also the higher-level data model technologies influence how the data can be utilized. Current higher-level data models can be divided into two categories, SQL and NoSQL stores [70]. The main requirement for a suitable data storage method is to be able to communicate through the DT data link.

[1] Security by design principle states that security is not a separate block that you can add to a system afterward without affecting the original design choices of the system. Instead, the whole system should be designed using solutions that support security. To demonstrate the principle, we can compare the security of a normal car and a tank. Concerning bulletproofness, a normal car is totally insecure, whereas a tank has been designed to be bulletproof from the beginning. We can convert the normal car into an armored vehicle, but without changing the original design choices, the bulletproofness will not reach the same level as that of a tank. Security in cyberspace is much more complex than bulletproofness of a vehicle, but the security by design principle holds.

### F. USER INTERFACE

*User interface* (UI) provides human users the possibility to interact with the DT. Many user interfaces have been presented for the digital twin [46], yet in the end, the UI design is always case-specific and driven by the needs of its users. The DT UI is personalized for each user group, depending on their needs and permissions.

A simple yet efficient example of a DT user interface is a web page, as suggested (for virtual counterpart) already in the year 2000 [10]. A web page is an excellent UI in its ubiquity, as it can be accessed with a smartphone. However, web sites are limited in tasks that require a more three-dimensional perception and convenient use of both hands. To address these needs, head-mounted displays have been used: a robot was controlled via a DT that was visualized in virtual reality [71], and maintenance instructions were displayed with augmented reality [72].

### G. SIMULATION

Simulation is used in different ways across industries. A simulation model describes the visual, graphical, and/or numerical essence of a physical product or a system in either steady-state or dynamic form. Traditionally, simulation models have been used to provide artificially generated data, to approximate real-life behavior in a time and cost-efficient way. Simulation tasks include virtual commissioning [73] and virtual prototyping [74]. Furthermore, the different models can interact in a multidisciplinary simulation, which was demonstrated by Brandstetter and Wehrstedt [75].

Some industry and academic players may label sophisticated simulation models as DTs, which seems to have caused confusion on what exactly is the difference between them. To address this confusion, Boschert and Rosen [7] and Grieves and Vickers [12] state that the link to the lifecycle of the physical counterpart is an essential property of a DT.

### H. ANALYSIS

A DT can be used as a tool to perform analyses on the data that is available about the real object. The data can come from the monitoring of the physical product or from simulations. Analyses, such as correlation or sensitivity analyses, are given to the user or the artificial intelligence feature of the DT for decision making.

### I. ARTIFICIAL INTELLIGENCE

*Artificial intelligence* (AI) is the part of DT that makes autonomous decisions based on data and analyses. The distinction to machine learning (ML) is that AI is used for decision making, whereas ML is a set of algorithms and models that can be used to process data for AI or a user. Therefore, many aspects of ML belong to the previously described analysis feature.

AI enables a DT to be a self-active object in the cyberspace, i.e. an intelligent DT. The difference between a passive DT and an intelligent DT is similar to the difference of a regular car and an autonomous one. A regular car requires a driver, while an autonomous car can drive on its own. Analogously, a regular DT always needs a user to perform the tasks, while a DT enhanced with AI can perform decisions on its own. To serve the physical product and its user, the intelligent DT can, for example, i) continuously analyze the condition of the physical product, ii) order maintenance visits, iii) trigger alarms, and iv) stop the operation of the physical product in emergencies [12].

### J. COMPUTATION

Computation is a low-level feature that solves mathematical tasks to generate data. Computation is either local when the system relies on edge computing, or global when it is performed remotely. The location of the computation is important considering the time criticality of different processes. Global computation hubs provide almost unlimited resources for data processing, whereas localized computing enables low latency when needed, e.g. in control loops.

## V. FEATURE-BASED DIGITAL TWIN ANALYSIS

We implement the feature-based digital twin analysis (FEDA) method to demonstrate the presence of the features in existing DT related publications. Hence, the purpose of the implementation is to show that the method is feasible and can be carried out for a variety of DT cases. We would like to note that this implementation does not act as a complete proof of the validity of the method, but rather is an example of how to use the FEDA method. Further validation for the method is to be performed over time.

This study leverages the method as a tool to verify the existence of the presented features and provide material to the FDTF presented later in Section VI. The purpose of the current analysis implementation is to identify the typical feature profiles of state-of-the-art DT implementations, to identify the DT implementations that best fulfill the FDTF concept. Choices during the analysis implementation are made to serve these purposes. Seven DT implementations from existing literature are selected for analysis. These cases range from robotic arms to a bending beam test bench.

### A. FEDA IMPLEMENTATION

We follow the steps defined in Section III, Subsection C.

**Step I**

We use the features presented as subsections of Section IV.

**Step II**

We set a manual grading method that estimates how much each feature is present in the implementation. The numerical value $v_i$ is selected for each feature as follows:

–0: the feature is not present
–1: the feature is mentioned
–2: the feature is clearly present and documented
–3: the feature is implemented exceptionally thoroughly

Hence, the maximum value $v_{max}$ is defined as 3.

### Step III

We manually assign grades for each feature of each DT implementation. Each of the authors performed the evaluations separately, with the averages of the grading results shown in Table 2. The average difference between the grade of each individual respondent and the overall average grade of all respondents is 0.5. The reader can observe the reliability of the evaluations by conducting the evaluations themselves as the data is openly available.

**TABLE 2. Results of Step III: The averages of our best estimates for the level of presence of each feature in the seven DT implementations. Columns A – J correspond to the features in Section III.**

| *DT* | *A* | *B* | *C* | *D* | *E* | *F* | *G* | *H* | *I* | *J* |
|---|---|---|---|---|---|---|---|---|---|---|
| [18] | 2.0 | 2.5 | 0.0 | 0.0 | 0.8 | 1.3 | 2.8 | 1.3 | 0.3 | 1.8 |
| [52] | 2.8 | 2.3 | 0.0 | 0.0 | 0.5 | 1.8 | 1.5 | 1.5 | 0.0 | 1.3 |
| [76] | 0.8 | 2.0 | 0.3 | 0.0 | 0.5 | 1.3 | 2.3 | 2.5 | 0.0 | 1.3 |
| [46] | 1.6 | 1.3 | 2.3 | 0.8 | 1.4 | 2.5 | 1.3 | 1.3 | 0.5 | 2.0 |
| [59] | 1.5 | 1.8 | 0.8 | 0.5 | 1.8 | 1.0 | 0.5 | 0.8 | 0.8 | 0.8 |
| [77] | 1.8 | 1.8 | 0.3 | 0.5 | 1.5 | 2.5 | 1.8 | 1.3 | 1.1 | 1.0 |
| [78] | 1.5 | 1.3 | 1.8 | 0.3 | 0.8 | 1.8 | 2.3 | 1.0 | 1.0 | 1.0 |

### Step IV

We define a flat weight factors $w_{i1}$ to serve as a neutral comparison between the cases. To find implementations that are particularly strong in the data link feature, but also provides strong performance across all features, we define weight factors $w_{i2}$. The weights and scale are shown in Table 3. This affects the holistic score to favor cases that receive a good score for the data link. For additional analysis between the cases, we define weights $w_{i3...9}$ as the grades presented in Table 2, i.e. $w_{i3} = [2.0, 2.5, 0, 0, 0.8, 1.3, 2.8, 1.3, 0.3, 1.8]$ and so on. This allows comparing DT cases to each other on their own scale.

**TABLE 3. The weights and scale of our current FEDA implementation.**

| *Feature* | | *Weight ($w_{i1}$)* | *Weight ($w_{i2}$)* |
|---|---|---|---|
| A | Data link | 1 | 3 |
| B | Coupling | 1 | 1 |
| C | Identifier | 1 | 1 |
| D | Security | 1 | 1 |
| E | Data storage | 1 | 1 |
| F | User interface | 1 | 1 |
| G | Simulation | 1 | 1 |
| H | Analysis | 1 | 1 |
| I | Artificial intelligence | 1 | 1 |
| J | Computation | 1 | 1 |
| Holistic score scale (*Sc*): 5 | | | |

### Step V

We calculate the holistic scores according to Eq. 1. The results are presented in Table 4 and Table 5.

**TABLE 4. Holistic scores calculated with Eq. 1 from the gradings presented in Table 2 and weights and scale presented in Table 3.**

| *DT use case* | *$Hs_1$* | *$Hs_2$* |
|---|---|---|
| Schluse *et al.* [18] | 2.1 | 2.3 |
| Haag and Anderl [52] | 1.9 | 2.4 |
| Grinshpun *et al.* [76] | 1.8 | 1.7 |
| Schroeder *et al.* [46] | 2.5 | 2.5 |
| Abramovici *et al.* [59] | 1.7 | 1.8 |
| Mohammadi and Taylor [77] | 2.2 | 2.3 |
| Sierla *et al.* [78] | 2.1 | 2.2 |

**TABLE 5. Holistic scores calculated similarly as in Table 4 but for implementation-specific weight factors $w_{i3..9}$.**

| *DT* | *$Hs_3$* | *$Hs_4$* | *$Hs_5$* | *$Hs_6$* | *$Hs_7$* | *$Hs_8$* | *$Hs_9$* |
|---|---|---|---|---|---|---|---|
| [18] | 3.3 | 3.2 | 3.1 | 2.2 | 2.3 | 2.6 | 2.4 |
| [52] | 2.9 | 3.2 | 2.7 | 2.0 | 2.3 | 2.4 | 2.2 |
| [76] | 2.7 | 2.6 | 3.0 | 2.0 | 1.8 | 2.1 | 2.1 |
| [46] | 2.6 | 2.7 | 2.6 | 2.8 | 2.5 | 2.6 | 2.7 |
| [59] | 1.8 | 2.0 | 1.7 | 1.6 | 2.0 | 1.8 | 1.6 |
| [77] | 2.7 | 2.8 | 2.6 | 2.3 | 2.4 | 2.7 | 2.4 |
| [78] | 2.4 | 2.4 | 2.4 | 2.4 | 2.1 | 2.3 | 2.5 |

## B. FEDA DISCUSSION

In Step III of the FEDA process, the presence of each DTF in different use cases is turned to numerical values. To easily comprehend the data and the differences between the use cases, a stacked bar chart is displayed in Fig. 2. We can roughly divide the amount of overall presence of features into three categories: i) high amount of implementation is represented by coupling, simulation, user interface (UI) and data link, ii) moderate presence exists for analysis, computation and data storage, and iii) low amount of presence goes to identifier, artificial intelligence (AI) and security. These results indicate that the groups with high and moderate presence are clearly established, whereas the three features with low scores are more questionable.

Only the features coupling and UI have the score 1 or higher in all of the seven DT use cases. This supports the basic notion that a digital twin must be linked to a real-world counterpart. The high presence of UI indicates that the current DTs are directly used by humans instead of being self-active components of the cyberspace. High scores of data link and simulation show that they are prominent features, but not required in all cases.

The identifier, AI, and security features of the selected use cases are rudimentary or do not exist. Identifier and AI exist in several cases, which indicates that they have been recognized as essential parts of future DTs, but the research to include them is in its infancy. Especially the lack of security underlines that these are prototypes, although it may also be.

Correlations, i.e. statistical associations between DTFs of the investigated use cases are presented in Table 6. The numbers are more reliable with a higher amount of data. Negative correlations are of less interest in this analysis, although as an interesting lift, data link and simulation tend to appear

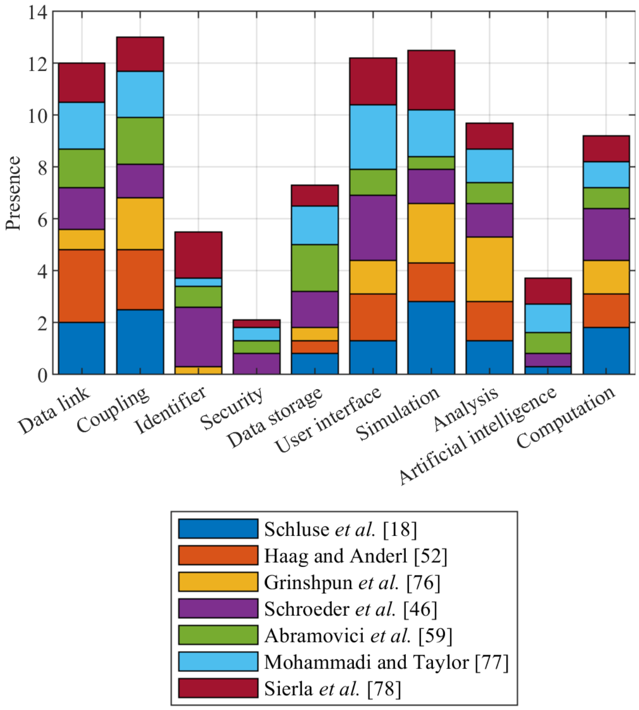

**FIGURE 2.** Step III grading results displayed as a stacked bar chart per feature.

**TABLE 6.** Correlation matrix of DTFs in the studied use cases. Positive (red color and positive number) number means that the features appear together in the use cases and negative (blue color and negative number) means that the features occur separately.

| | Data link | Coupling | Identifier | Security | Data storage | UI | Simulation | Analysis | AI | Computation |
|---|---|---|---|---|---|---|---|---|---|---|
| Data link | 1.0 | | | | | | | | | |
| Coupling | 0.2 | 1.0 | | | | | | | | |
| Identifier | -0.6 | -0.8 | 1.0 | | | | | | | |
| Security | 0.3 | -0.1 | -0.3 | 1.0 | | | | | | |
| Data storage | 0.3 | -0.4 | 0.5 | 0.0 | 1.0 | | | | | |
| UI | -0.3 | -0.2 | 0.3 | -0.7 | -0.1 | 1.0 | | | | |
| Simulation | -0.8 | 0.1 | 0.4 | -0.3 | 0.1 | 0.3 | 1.0 | | | |
| Analysis | 0.3 | -0.4 | 0.4 | 0.2 | 0.5 | -0.6 | -0.4 | 1.0 | | |
| AI | 0.0 | -0.8 | 0.5 | 0.5 | 0.4 | -0.4 | -0.3 | 0.7 | 1.0 | |
| Computation | -0.4 | -0.2 | 0.4 | -0.5 | 0.3 | 0.0 | 0.2 | 0.3 | 0.2 | 1.0 |

in different cases for some reason. The only strong positive link is between analysis and AI. They may be difficult to separate from each other, although we render this as a matter of coincidence due to low amount of AI presence in the cases. Either way, this result along with the larger reliability of the method should be further investigated with a higher amount of data.

The data link focused holistic scores, $Hs_2$, shown in Table 4 suggest that the DT use case presented by Schroeder *et al.* [46] best fulfill the FDTF presented in this paper, closely followed by Haag and Anderl [52], Mohammadi and Taylor [77], Schluse *et al.* [18], and Sierla *et al.* [78].

Lowest holistic score from this FEDA implementation was given to Abramovici *et al.* [59] and Grinshpun *et al.* [76]. The first concentrated on the reconfiguration of products, which is not a clear part of any of the presented DTFs. The latter focused deeply on simulation and analysis, receiving the highest grade in the alternative holistic score $Hs_2$. These two use cases concentrated on matters that were not in the focus of the current FEDA implementation, and it is therefore even an intentional result that some cases receive lower scores to make a distinction to the others.

As can be observed from the selections made in the previous subsection, our implementation of the FEDA method is based on subjective choices. The subjectivity could be reduced with various methods, such as using a higher number of evaluators or developing automatic evaluation methods for Step III, but these are left out of scope for two reasons: 1) The purpose of the FEDA implementation is to act as support for the FDTF concept presented in Section VI instead of being an independent goal of this paper. 2) The features are only presented in this paper and we intend to leave room for discussion on other possible features before developing the analysis method further. If consensus on the features is reached, feature specific evaluation methods can be developed.

It is worth noting that the user-specific weight factors accentuate subjectivity in a positive manner, providing customized results for each user. Hence, the holistic scores calculated in Step V are not a general evaluation of the use cases. To reach generally applicable evaluations, objective, mathematically justified methods of implementing Steps I-IV should be developed. The mathematical grading methods would also greatly increase the validity of the analysis method as well as potentially enable automatic categorization of DT related articles.

In this study, a mathematical definition of categorization could not be included for two reasons. First, there is no clear method to analyze the implementations reliably. For example, it would be possible to count words for each feature, but the selection method for related words is not clear and demands an effort that is outside the possibilities and scope of this study. Second, even though a set of related words could be defined, the quality of the implementation cannot be reliably defined with the number of words. Quality could be taught to machine learning algorithms, but this would require an extensive amount of high-quality training data that is not available. Also, the training should be done to each feature separately. Because of these reasons, this study relies on qualitative methods on defining the presence of features.

The current implementation of the FEDA method contains numerous points of improvement. Nevertheless, it fulfills its purpose of this publication by showing that dividing a DT to different features is a feasible approach.

## VI. FEATURE-BASED DIGITAL TWIN FRAMEWORK

Emerging from the discovery of DTFs, we propose a general *feature-based digital twin framework* (FDTF) as a set of design guidelines for DTs. We merge the discovery of DTFs with the findings of existing literature. The framework mainly builds on the product-centric information management concept presented by Holmström *et al.* [53] and Hribernik *et al.* [14], who applied the structure to enhance information logistics between stakeholders of a product. The benefits of this approach in logistics context were discussed by Rönkkö *et al.* [79], embodying in a better quality of service and enabling new functionalities. Our approach is to bring the benefits of product-centric information structure to DT context by connecting the DTFs with the data link feature, as depicted in Fig. 1.

The idea of dividing a DT into multiple building blocks has been presented before in several publications. Boschert and Rosen [7] described DT to include information from multiple systems. Canedo [41] presented a DT consisting of multiple nodes and edges. Later, Datta [38] proposed that the connected blocks should be included in an open repository and flashed the idea that blockchain technology can provide trust in the repository. DebRoy *et al.* [50] and Knapp *et al.* [39] developed building blocks of digital twins for additive manufacturing. These works show a clear demand for building blocks, which we call DTBs.

The DTBs are located in separate systems and therefore need data interfaces between each other. These interfaces are called APIs and the current software industry is increasingly relying on them when building web-based applications. The information systems are also turning from large monolithic local applications to collections of small services, often referred to as microservices.[2] The main benefits of migrating to microservices architecture include maintainability and scalability [80]. The development of APIs and microservices feed each other's growth as microservices communicate via APIs. RESTful API has developed into an industry standard, and the distribution of microservices has become easy with software delivery tools such as Docker containerization technology [81]. Containers are becoming popular also in fields outside isolated web application engineering, such as in IoT [82] and in supporting reproducible research [83].

We propose using microservices as DTBs. The high amount of DTBs and APIs between them potentially results in an increasingly complex communication structure. The complexity is controlled by the star style communication structure presented by the product-centric information management concept. The star structure allows full connectivity between DTBs, i.e. nodes, by adding a broker node in the center.

The alternative unstructured, naturally adaptable grid-style structure requires individual connections between every pair of nodes. The maximum number of connections for grid-style connections is *n(n-1)/2* for *n* nodes as dictated by graph theory. Hence, the number of connections for grid-style is equal to the amount of nodes at $n = 3$ and raises substantially with an increasing number of nodes, with 6 connections at $n = 4$, 10 at $n = 5$ and so forth. For star-style structure, the amount is simply $n$, with the additional broker node in the middle. Fig. 3 illustrates the difference between star and grid style connections with $n = 5$.

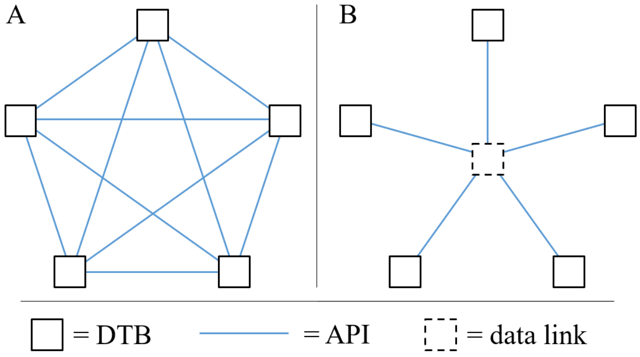

**FIGURE 3. Visualization of API connections between DTBs for n = 5 for grid structure (A) and star structure (B).**

The star-style structure provides practical benefits in addition to lowering the number of connections. The broker node, i.e. the data link, provides a list of the other nodes from a single access point, offering a convenient summary of the capabilities of a DTI. When DTIs communicate with each other, the single access point enables the network of DTs to develop into a similar decentralized network that the Internet currently is. Difference is that DTIs and their real-world counterparts act as the nodes of the network, fulfilling the grand vision of IoT.

The differences between concepts and reality divide the FDTF into two dimensions. The conceptual dimension displays the futuristic goal of the framework which is not achievable in the present but requires technical and communal advancements to reach reality. The situation is similar to the beginning of the Internet where the advantages of distributed networks were presented in 1964 [84], but it took decades to reach the current situation where the Internet is massively important in our daily lives.

In contrast to the future seeking conceptual dimension, the realization dimension concentrates on what should be done now. It is dependent on the capabilities of existing systems as it has to deliver benefit. The current systems have developed for a long time and acquired customized features, providing high operational efficiency. Hence, realization dimension balances between operational efficiency and pursuing the supposed benefits of the conceptual dimension.

### *A. CONCEPTUAL DIMENSION*

The basic idea of the conceptual dimension of FDTF is presented in Fig. 1 and also in simplified style in Fig. 4. It links the DTFs of Section IV together in star style connection.

[2] Microservice is a software architecture style gaining traction in the IT systems field. There is no exact technical definition of what is a microservice. Microservice is a small separate service that runs on its own and implements a single service that it provides to other services via APIs, as opposed to large systems that implement a wide variety of different functions. The advantage of the microservice style architecture is that one service stays small and can be easily updated. Microservices change the focus of systems engineering from one machine to the interoperability of multiple machines.

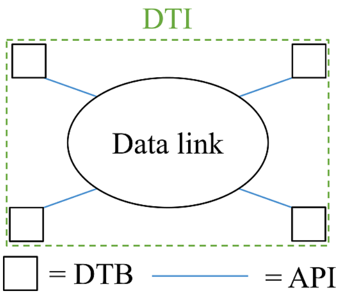

**FIGURE 4.** Simplified illustration of the conceptual dimension of FDTF.

The concept portrays that each DTI consists of DTBs which fulfill exactly one feature and the DTBs are connected together with a data link. Hence, each DTF is fulfilled by a separate microservice. These services are joined together with APIs, exchanging data from feature to another locally or over the Internet.

The framework is analogous to a *personal computer* (PC). Components of a personal computer are like the DTFs. The CPU corresponds to the computation feature, the hard drive to the data storage, and the motherboard is the data link that enables the information flow between the features. While the significance of the features vary and they may overlap, it is imperative that they are connected together. The connected DTFs form explicitly defined DTIs which are linked together as a network of DTs, similarly as PCs and other devices have formed the Internet. Furthermore, as the Internet consists of a diverse set of devices, also the network of DTs consists of different kinds of DTIs.

A set of features can be selected and combined to form the appropriate DTI for each use case. The most useful combinations are brought together into DT classes (DTCs) that enable the easy creation of DTIs, similarly to as object-oriented programming uses classes to make new objects. The code found in the DTC initializes all the necessary software components required for the new DTI, eliminating the monotonic manual labor that is otherwise required for creating DTs. With intelligent use of DTCs, the adoption of the DT concept eases the workload of people instead of becoming another system that needs to be maintained. Hence, DTCs deserve specific effort as they pave the way for DTs to become business as usual.

The basic procedure of creating a DTI is described as follows.

1. Define the functional requirements of the DTI based on the underlying use case.
2. Determine the necessary DTFs of the DTI based on the functional requirements.
3. Select the DTBs that fulfill the corresponding DTFs.
4. Configure the DTBs to the use case.
5. Deploy the DTBs to form an operational DTI.
6. Optionally create a DTC based on the DTI.

Step 6 is ideally included in the previous steps as a standard mode of operation. With a readymade DTC, the procedure is reduced to only steps 4 and 5.

### B. REALIZATION DIMENSION

Current enterprise systems are built for specialized tasks and they fulfill their tasks well. The systems are mostly built as monolithic applications, entailing that their update cycle is somewhere from weeks to months. Techniques for converting monolithic software to microservices-based architecture are being developed [85], [86] and a mission-critical banking system has been migrated to microservices architecture [87]. The microservices-based architecture provides natural support for the principles of conceptual dimension better, but also monolithic enterprise systems can be used as DTBs as long as they have APIs.

Most existing enterprise systems implement more than one DTF due to their current usage profile. Therefore each DTB implemented with legacy systems fulfills many features, which creates a clear distinction to the principle of the conceptual dimension which claims that one DTB equals one DTF. This is not an issue but it means that only some small part of a monolithic enterprise system is used for each DTI, creating situation depicted in Fig. 5. The DTI is depicted in green, with fuzzy boundaries of which parts of the ESs belong to the DT and which do not. The connections between the systems are drawn in blue and they must be precisely defined for the DTI to work as intended. Hence, there is no problem with using ESs this way. Problems may arise when changes need to be done, as changes will impact other users of the ES. Therefore service breaks must be scheduled according to a vast amount of demands, and customization increases the complexity of the whole system.

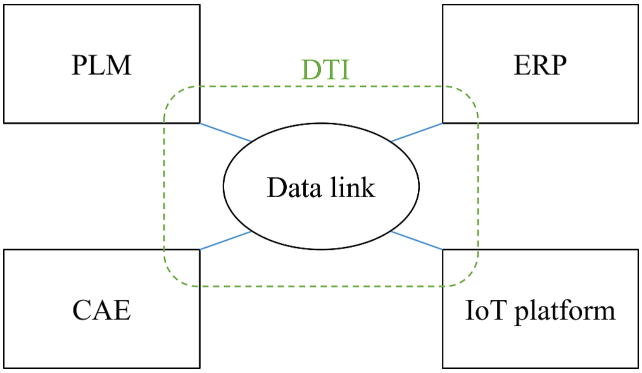

**FIGURE 5.** Illustration of a DTI created with monolithic systems.

Microservices enable the situation shown in Fig. 6 as an alternative to the monolithic architecture style. Some ESs, e.g. database, are still included in monolithic fashion (i), whereas other systems are structured as collections of microservices that are organized in various ways (ii and iii), and the DTI has also its own microservices (iv). The situation (i) can be appropriate when a monolithic system has a clear readymade feature that the DTI can use. However, the monolithic style does not support customizability and its development cycles are long. The grid-style architecture (ii) may exist in a complicated microservice system where the blocks are connected to each other from multiple directions, meaning

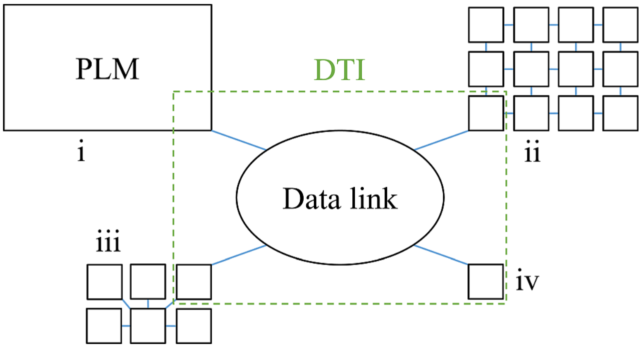

**FIGURE 6.** Illustration of a DTI created with a diverse set of systems.

that the microservice is used by several other services. Changing the microservice may be a complicated process because of multiple dependencies. The star style connection (iii) depicts a situation where multiple services are centrally managed, but dedicated for single DTI. This allows more customizability, but restrictions derive from the central management system. A standalone microservice (iv) provides complete customizability in favor of the DTI, as there are no dependencies to other services. However, administration work is higher per DTI. Each of the styles have their advantages and disadvantages, but the adoption rate of microservices is proving their feasibility.

One of the biggest challenges in the realization efforts of FDTF is the lack of standardization and lack of knowledge and skills of the end-users who have the domain knowledge to develop truly meaningful DTIs. The lack of standardization makes it impossible to learn and use the new technology, as each system is different and requires dedicated learning effort. The new paradigm of using multiple systems requires new way of thinking that concentrates on the APIs, and the APIs need to become even more user-friendly than now. However, these are issues that are solved by simply elapsing time as increasing demand makes the new technology more user-friendly.

The previously described ways to implement DTIs, i.e. following the FDTF concept, are more time demanding than strict need-based implementation when building just one DTI. The advantages of FDTF are envisioned to take effect when an organization has a strategic approach to build and develop DTs in the long term. The benefits are similar to those of the modularity used in various physical products.

### C. COMPARISON

The conceptual dimension of FDTF provides high-level structure for DTs and proposes a basic procedure for creating DTIs. This approach can be used to disseminate knowledge and it gives a long term goal to pursue. The realization dimension focuses on the present situation of ESs, explaining their current state and direction. The realization dimension describes how existing systems can be leveraged to pursue the conceptual goals. The two dimensions of the FDTF have different approaches in many aspects. We summarize the differences in Table 7.

**TABLE 7.** Comparison between conceptual and realization dimensions of the FDTF.

| *Aspect* | *Conceptual* | *Realization* |
|---|---|---|
| *Background* | Academic research on IoT and data management | Practical progress of software technology |
| *Purpose* | Providing long term goal | Guiding current development |
| *Goal* | Building sustainable digital infrastructure | Solving current problems in a future proof way |
| *Theoretical emphasis* | Intangible concepts: DTFs and data link | Actual software components: DTBs and APIs |
| *Time frame* | 5+ years | 0-5 years |
| *Opportunities* | Revolution of data management | Enhanced operational efficiency |
| *Strategic position* | Long term development | Operational tasks |
| *Suggested actions* | Education on DT data linking concept | Adopting microservice and API technologies |
| *Relation between dimensions* | Guides the reality dimension | Follows concept dimension where possible |

To summarize the framework we define the following statements as the basis of the FDTF. 1) Each DTI has a one-to-one correspondence to its real-world counterpart, enabling product-centric information management. 2) The creation of a DTI starts by defining use-case-specific *functional requirements* which are converted to *technical functionalities* (DTFs) and finally built as a *technical implementation* (collection of DTBs). 3) The future DTI is a modular entity that is made of DTBs and connected together via APIs.

## VII. DISCUSSION

This study set out to clarify the obscurity around the DT concept. This is a difficult but called-for task and the success can be evaluated by observing the answers to the three research questions.

The first research question "What is a digital twin?" can be answered by first finding the common elements of previous definitions, accompanied with the newly found structure of FDTF. We formulate our definition in two sentences: "Digital twin is a virtual entity that is linked to a real-world entity. Digital twin consists of various features that are selected and customized to serve the needs of diverse use cases." This is a general definition that catches the fundamental idea and then adds the notion that DT implementations are diverse, whereas other definitions are characterized by some specific use case.

The second question "How to compare digital twins?" can be answered by identifying features that can be evaluated on a similar scale. We formulated the evaluation process as the FEDA method we can assign numerical values for DT implementations. The method allows the user to choose their own set of DTFs and weightings which consequently lead to the fact that the FEDA method currently provides DT categorization based on personal preferences. However, the method is left general enough to be further improved, reaching objective categorization and grading for DTs.

Nevertheless, the FEDA method raises further questions. Are the features presented in this paper the universal set of features? How can we compare each individual feature objectively? Which are the most important features? We argue that the answers to these questions are subjective by nature with current knowledge, but with further studies and discussion, more elaborate and objective answers can be reached.

We open the discussion by suggesting that the data link is the most important feature as it provides unseen interoperability and scalability. The data link offers a revolutionary enhancement to the potential of DTs thanks to its ability to enable the network of DTs and enabling modular structure with DTBs. When the data link feature is implemented properly, APIs provide clear boundaries to other features, expediting their development as the expert of each feature can develop their DTB independently. Furthermore, we propose our set of ten features as the universal set of features. All of the features are not equally present in current DT implementations, but we see the less used features are fundamental to future development.

The third question ''How to build digital twins?'' has been independently answered by multiple academics and industry personnel from both conceptual and technical perspectives. A general technical answer cannot be currently provided because of high diversity between the implementations. As a conceptual answer to this question, we propose the procedure and guidelines described in FDTF (Section VI). Dividing the framework into conceptual and realization dimensions highlights the difference between long-term DT ambitions and the current capabilities of ESs. The vision of DT cannot be yet implemented efficiently, but the description of the conceptual dimension helps companies to future-proof their ESs.

From the wide perspective, the data-linking-oriented concept of FDTF aims to fix the current situation where the information of products is not available as it is scattered across different systems that are so complex that one person cannot easily master. The dispersion makes achieving the overall status of a product an unnecessarily time-demanding task, even though each system is perfectly capable of fulfilling their intended main purpose. Our proposed solution connects the data from multiple systems to make the information of a single product instance available from a single interface. Hence, the benefits of the FDTF lie in the enhanced and automated information flow between multiple parties, including software, hardware, and humans. The improved information flow further induces multiple benefits, e.g. time savings in information fetching, higher efficiency due to optimal parameters, and new applications thanks to machine-to-machine communications.

Finally, we would like to draw attention to a common misconception that has appeared during the process of preparing this paper. Digital twin is not a technology, but rather an idea or philosophy that can be realized with many different technologies. The digital twin concept belongs to the semantic layer rather than to the technology layer.

## VIII. CONCLUSION

Current research on the DT concept is fragmented and ridden with misconceptions. This work presents a novel way to structure and compare DTs by first identifying features that exist in current implementations, formulating the FEDA method to analyze the implementations, and presenting the FDTF framework on how to leverage these features to design future DTs. To enable communication, we introduce a well-overdue unified terminology for the DT based on previous work and our current findings.

The novel FEDA method is a tool to categorize DT implementations by comparing the presence of features in them. Seven existing implementations were evaluated and a correlation analysis was carried out to study the dependencies between DTFs. No arguable correlations between the features which indicates that the identified features are independent of each other.

FDTF is a high-level guideline on how to design future DTs. The framework builds on our novel definition of the data link feature which connects the information flow of all the other features. The three main potential benefits of the framework include: i) establishing a universal structure across diverse DT implementations, ii) dividing the DT into blocks that can be easily added or removed, and iii) enabling easy access to all available product information via single data link.

The DT features (DTFs) describe the contents of a DT at a functional level. To provide guidelines for realization, we describe initial insights on how current enterprise systems can be used as DT blocks (DTBs) for building DT instances (DTIs). DTFs and DTBs operate in different dimensions: the DTBs may consist of multiple DTFs.

Planned future work includes automated categorization method built for the FEDA method, implementing a DT for the engineering process of an industrial crane, and an example implementation of the data link feature.

## ACKNOWLEDGMENT

The authors would like to thank the industrial partners of the DigiTwin project and the industrial steering group of the TwinRotor project for providing valuable practical insights during the study. The authors thank Emil Kurvinen for pointing out the need for a grading method of DTs and Tuomas Tiainen for the discussions on the essence of DTs. The authors would like to thank Ville Klar for everything.

## REFERENCES

[1] M. Shafto, "DRAFT modeling, simulation, information technology & processing roadmap," Technol. Area, NASA, Washington, DC, USA, Tech. Rep., Nov. 2010. [Online]. Available: https://www.nasa.gov/pdf/501321main_TA11-MSITP-DRAFT-Nov2010-A1.pdf

[2] J. Puig and J. Duran, "Digital twins," in *Proc. 4th Int. Multi-Conf. Soc., Cybern. Inform. (IMSCI)*, 2010, pp. 28–31.

[3] T. Nicolai, F. Resatsch, and D. Michelis, "The Web of augmented physical objects," in *Proc. 4th Int. Conf. Mobile Bus. (ICMB)*, Jul. 2005, pp. 340–346.

[4] J. Autiosalo, "Platform for industrial Internet and digital twin focused education, research, and innovation: Ilmatar the overhead crane," in *Proc. IEEE 4th World Forum Internet Things (WF-IoT)*, Feb. 2018, pp. 241–244.

[5] J. P. Hale, II, and M. L. Dittmar, "Virtual reality as a human factors design analysis tool for architectural spaces—Control rooms to space stations I: Objective measures," in *Proc. Hum. Factors Ergonom. Soc. Annu. Meeting*, vol. 38, no. 4, pp. 275–279, 1994.

[6] R. Rosen, G. von Wichert, G. Lo, and K. D. Bettenhausen, "About the importance of autonomy and digital twins for the future of manufacturing," *IFAC-PapersOnLine*, vol. 28, no. 3, pp. 567–572, 2015.

[7] S. Boschert and R. Rosen, "Digital twin—The simulation aspect," in *Mechatronic Futures*, P. Hehenberger and D. Bradley, Eds. Cham, Switzerland: Springer, 2016, pp. 59–74.

[8] T. Berners-Lee. (1989). *Information Management: A Proposal*. Accessed: Feb. 20, 2019. [Online]. Available: https://www.w3.org/History/1989/proposal.html

[9] K. Ashton, *That 'Internet of Things'*, RFID, Alpharetta, GA, USA, 2009.

[10] M. Langheinrich, F. Mattern, K. Römer, and H. Vogt, "First steps towards an event-based infrastructure for smart things," in *Proc. Ubiquitous Comput. Work. PACT*, Jul. 2000, pp. 1–13.

[11] M. W. Grieves, "Product lifecycle management: The new paradigm for enterprises," *Int. J. Prod. Develop.*, vol. 2, nos. 1–2, pp. 71–84, 2005.

[12] M. Grieves and J. Vickers, "Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems," in *Transdisciplinary Perspectives on Complex Systems: New Findings and Approaches*, F.-J. Kahlen, S. Flumerfelt, and A. Alves, Eds. Cham, Switzerland: Springer, 2017, pp. 85–113.

[13] K. Främling, J. Holmström, T. Ala-Risku, and M. Kärkkäinen, "Product agents for handling information about physical objects," Lab. Inf. Process. Sci., Helsinki Univ. Technol., Espoo, Finland, Tech. Rep. TKO-B 153/03, 2003. [Online]. Available: http://www.cs.hut.fi/Publications/Reports/B153.pdf

[14] K. A. Hribernik, L. Rabe, K. D. Thoben, and J. Schumacher, "The product avatar as a product-instance-centric information management concept," *Int. J. Prod. Lifecycle Manag.*, vol. 1, no. 4, pp. 367–379, 2006.

[15] M. Grieves, *Product Lifecycle Management: Driving the Next Generation of Lean Thinking*. New York, NY, USA: McGraw-Hill, 2006.

[16] M. Grieves, *Virtually Perfect: Driving Innovative and Lean Products Through Product Lifecycle Management*. Cocoa Beach, FL, USA: Space Coast Press, 2011.

[17] M. Grieves, *Virtually Indistinguishable: Systems Engineering and PLM*, vol. 388. New Delhi, India: AICT, 2012.

[18] M. Schluse, M. Priggemeyer, L. Atorf, and J. Rossmann, "Experimentable digital twins—Streamlining simulation-based systems engineering for industry 4.0," *IEEE Trans. Ind. Informat.*, vol. 14, no. 4, pp. 1722–1731, Apr. 2018.

[19] E. J. Tuegel, A. R. Ingraffea, T. G. Eason, and S. M. Spottswood, "Reengineering aircraft structural life prediction using a digital twin," *Int. J. Aerosp. Eng.*, vol. 2011, Aug. 2011, Art. no. 154798.

[20] T. D. West and M. Blackburn, "Is digital thread/digital twin affordable? A systemic assessment of the cost of DoD's latest manhattan project," *Procedia Comput. Sci.*, vol. 114, pp. 47–56, Jan. 2017.

[21] E. Negri, L. Fumagalli, and M. Macchi, "A review of the roles of digital twin in CPS-based production systems," *Procedia Manuf.*, vol. 11, pp. 939–948, Jan. 2017.

[22] F. Tao, H. Zhang, A. Liu, and A. Y. C. Nee, "Digital twin in industry: State-of-the-art," *IEEE Trans. Ind. Informat.*, vol. 15, no. 4, pp. 2405–2415, Apr. 2019.

[23] Y. Zheng, S. Yang, and H. Cheng, "An application framework of digital twin and its case study," *J. Ambient Intell. Hum. Comput.*, vol. 10, no. 3, pp. 1141–1153, 2019.

[24] M. Schluse and J. Rossmann, "From simulation to experimentable digital twins: Simulation-based development and operation of complex technical systems," in *Proc. IEEE Int. Symp. Syst. Eng. (ISSE)*, Oct. 2016, pp. 1–6.

[25] F. Tao and M. Zhang, "Digital twin shop-floor: A new shop-floor paradigm towards smart manufacturing," *IEEE Access*, vol. 5, pp. 20418–20427, 2017.

[26] F. Tao, F. Sui, A. Liu, Q. Qi, M. Zhang, B. Song, Z. Guo, S. C.-Y. Lu, and A. Y. C. Nee, "Digital twin-driven product design framework," *Int. J. Prod. Res.*, vol. 57, no. 12, pp. 3935–3953, 2018.

[27] E. Tuegel, "The airframe digital twin: Some challenges to realization," in *Proc. 53rd AIAA/ASME/ASCE/AHS/ASC Struct., Struct. Dyn. Mater. Conf.*, 2012, pp. 7177–7184.

[28] E. Tuegel, "Airframe digital twin: An overview," in *Proc. Aircr. Airworthiness Sustainment Conf.*, Apr. 2012, pp. 2–5.

[29] B. Gockel, A. Tudor, M. Brandyberry, R. Penmetsa, and E. Tuegel, "Challenges with structural life forecasting using realistic mission profiles," in *Proc. 53rd AIAA/ASME/ASCE/AHS/ASC Struct., Struct. Dyn. Mater. Conf.*, 2012, p. 1813.

[30] D. A. Fulghum and G. Warwick, "Industry works to boost security as cyberattacks escalate," in *Aviation Week & Space Technology*, vol. 170, no. 17. New York, NY, USA: Aviation Week Network, 2009.

[31] E. M. Kraft, "The air force digital thread/digital twin—Life cycle integration and use of computational and experimental knowledge," in *Proc. 54th AIAA Aerosp. Sci. Meeting*, 2016, p. 897.

[32] J. V. Zweber, R. M. Kolonay, P. Kobryn, and E. J. Tuegel, "Digital thread and twin for systems engineering: Requirements to design," in *Proc. 55th AIAA Aerosp. Sci. Meeting*, 2017, p. 875.

[33] P. Kobryn, E. J. Tuegel, J. V. Zweber, and R. M. Kolonay, "Digital thread and twin for systems engineering: EMD To disposal," in *Proc. 55th AIAA Aerosp. Sci. Meeting*, 2017, pp. 1–13.

[34] A. J. Zakrajsek and S. B. Mall, "The development and use of a digital twin model for tire touchdown health monitoring," in *Proc. 58th AIAA/ASCE/AHS/ASC Struct., Struct. Dyn., Mater. Conf.*, 2017, pp. 1–16.

[35] F. Tao, M. Zhang, J. Cheng, and Q. Qi, "Digital twin workshop: A new paradigm for future workshop," *Computer Integr. Manuf. Syst.*, vol. 23, no. 1, pp. 1–9, 2017.

[36] H. Zhang, G. Zhang, and Q. Yan, "Dynamic resource allocation optimization for digital twin-driven smart shopfloor," in *Proc. IEEE 15th Int. Conf. Netw., Sens. Control (ICNSC)*, Mar. 2018, pp. 1–5.

[37] S. Boschert, C. Heinrich, and R. Rosen, "Next generation digital twin," in *Proc. TMCE*, 2018, pp. 209–218.

[38] S. P. A. Datta, "Emergence of Digital Twins—Is this the March of reason?" *J. Innov. Manag.*, vol. 5, no. 3, pp. 14–33, 2017.

[39] G. L. Knapp, T. Mukherjee, J. S. Zuback, H. L. Wei, T. A. Palmer, A. De, and T. DebRoy, "Building blocks for a digital twin of additive manufacturing," *Acta Mater.*, vol. 135, pp. 390–399, Aug. 2017.

[40] A. E. Saddik, "Digital twins: The convergence of multimedia technologies," *IEEE Multimed.*, vol. 25, no. 2, pp. 87–92, Apr./Jun. 2018.

[41] A. Canedo, "Industrial IoT lifecycle via digital twins," in *Proc. 11th IEEE/ACM/IFIP Int. Conf. Hardw./Softw. Codesign Syst. Synth. (CODES)*, Oct. 2016, Art. no. 29.

[42] B. G. Glaser and A. L. Strauss, *The Discovery of Grounded Theory: Strategies for Qualitative Research*. Chicago, IL, USA: Aldine, 1967.

[43] M. Saunders, P. Lewis, and A. Thornhill, *Research Methods for Business Students*, 7th ed. Harlow, U.K.: Pearson, 2016.

[44] J. Ríos, J. C. Hernández, M. Oliva, and F. Mas, "Product avatar as digital counterpart of a physical individual product: Literature review and implications in an aircraft," in *Proc. Adv. Transdisciplinary Eng.*, vol. 2, 2015, pp. 657–666.

[45] F. Tao, J. Cheng, Q. Qi, M. Zhang, H. Zhang, and F. Sui, "Digital twin-driven product design, manufacturing and service with big data," *Int. J. Adv. Manuf. Technol.*, vol. 94, nos. 9–12, pp. 3563–3576, 2018.

[46] G. Schroeder, C. Steinmetz, C. E. Pereira, I. Müller, N. Garcia, D. Espindola, and R. Rodrigues, "Visualising the digital twin using Web services and augmented reality," in *Proc. IEEE 14th Int. Conf. Ind. Inform. (INDIN)*, Jul. 2016, pp. 522–527.

[47] E. Glaessgen and D. Stargel, "The digital twin paradigm for future NASA and U.S. air force vehicles," in *Proc. 53rd AIAA/ASME/ASCE/AHS/ASC Struct., Struct. Dyn. Mater. Conf.*, 2012, pp. 1–14.

[48] A. Cerrone, J. Hochhalter, G. Heber, and A. Ingraffea, "On the effects of modeling as-manufactured geometry: Toward digital twin," *Int. J. Aerosp. Eng.*, vol. 2014, Aug. 2014, Art. no. 439278.

[49] T. Gabor, L. Belzner, M. Kiermeier, M. T. Beck, and A. Neitz, "A simulation-based architecture for smart cyber-physical systems," in *Proc. IEEE Int. Conf. Autonomic Comput. (ICAC)*, Jul. 2016, pp. 374–379.

[50] T. Debroy, W. Zhang, S. S. Babu, and J. Turner, "Building digital twins of 3D printing machines," *Scripta Mater.*, vol. 135, pp. 119–124, Jul. 2017.

[51] K. M. Alam and A. El Saddik, "C2PS: A digital twin architecture reference model for the cloud-based cyber-physical systems," *IEEE Access*, vol. 5, pp. 2050–2062, 2017.

[52] S. Haag and R. Anderl, "Digital twin—Proof of concept," *Manuf. Lett.*, vol. 15, pp. 64–66, Jan. 2018.

[53] J. Holmström, K. Främling, J. Tuomi, M. Kärkkäinen, and T. Ala-Risku, "Implementing collaboration process networks," *Int. J. Logistics Manage.*, vol. 13, no. 2, pp. 39–50, 2002.

[54] T. Berners-Lee, J. Hendler, and O. Lassila, "The semantic Web," *Sci. Amer.*, vol. 284, no. 5, pp. 34–43, May 2001.

[55] S. Chun *et al.*, "Designing an integrated knowledge graph for smart energy services," *J. Supercomput.*, 2018, doi: 10.1007/s11227-018-2672-3.

[56] Y. Lu and F. Ju, "Smart manufacturing systems based on cyber-physical manufacturing services (CPMS)," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 15883–15889, Jul. 2017.

[57] B. R. Seshadri and T. Krishnamurthy, "Structural health management of damaged aircraft structures using digital twin concept," in *Proc. 25th AIAA/AHS Adapt. Struct. Conf.*, Jan. 2017, p. 1675.

[58] B. Schleich, N. Anwer, L. Mathieu, and S. Wartzack, "Shaping the digital twin for design and production engineering," *CIRP Ann.*, vol. 66, no. 1, pp. 141–144, 2017.

[59] M. Abramovici, J. C. Göbel, and P. Savarino, "Reconfiguration of smart products during their use phase based on virtual product twins," *CIRP Ann.*, vol. 66, no. 1, pp. 165–168, 2017.

[60] M. Kärkkäinen and J. Holmström, "Wireless product identification: Enabler for handling efficiency, customisation and information sharing," *Supply Chain Manag.*, vol. 7, no. 4, pp. 242–252, 2002.

[61] D. McFarlane, S. Sarma, J. L. Chirn, C. Wong, and K. Ashton, "Auto ID systems and intelligent manufacturing control," *Eng. Appl. Artif. Intell.*, vol. 16, no. 4, pp. 365–376, Jun. 2003.

[62] T. Berners-Lee, R. Fielding, and L. Masinter, *Uniform Resource Identifier (URI): Generic Syntax*, document RFC 3986, 2005. [Online]. Available: https://www.ietf.org/rfc/rfc3986.txt

[63] *Information and Documentation–Digital Object Identifier System*, Standard ISO 263242012, 2012.

[64] P. S. Browne, "Computer security: A survey," *ACM SIGMIS Database, DATABASE Adv. Inf. Syst.*, vol. 4, no. 3, pp. 1–12, Sep. 1972.

[65] D. C. Chou, D. C. Yen, B. Lin, and P. H.-L. Cheng, "Cyberspace security management," *Ind. Manag. Data Syst.*, vol. 99, no. 8, pp. 353–361, Dec. 1999.

[66] S. Backman, "Organising national cybersecurity centres," *Inf. Secur.*, vol. 32, no. 1, pp. 9–26, 2015.

[67] A. Humayed, J. Lin, F. Li, and B. Luo, "Cyber-physical systems security—A survey," *IEEE Internet Things J.*, vol. 4, no. 6, pp. 1802–1831, Dec. 2017.

[68] U. S. Department of Homeland Security. (2018). *Cyber Physical Systems Security | Homeland Security*. Accessed: Jul. 20, 2018. [Online]. Available: https://www.dhs.gov/science-and-technology/csd-cpssec

[69] H. Lagus, "Digitaalisen kaksosen tietoturvavaatimukset," Bachelor's thesis, Aalto Univ., Espoo, Finland, 2018. [Online]. Available: http://urn.fi/URN:NBN:fi:aalto-201806193390

[70] D. Wu, S. Sakr, and L. Zhu, "Big Data Storage and Data Models," in *Handbook of Big Data Technologies*, A. Y. Zomaya and S. Sakr, Eds. Cham, Switzerland: Springer, 2017, pp. 3–29.

[71] H. Laaki, Y. Miche, and K. Tammi, "Prototyping a digital twin for real time remote control over mobile networks: Application of remote surgery," *IEEE Access*, vol. 7, pp. 20325–20336, 2019.

[72] J. Posada, C. Toro, I. Barandiaran, D. Oyarzun, D. Stricker, R. de Amicis, E. B. Pinto, P. Eisert, J. Döllner, and I. Vallarino, "Visual computing as a key enabling technology for industrie 4.0 and Industrial Internet," *IEEE Comput. Graph. Appl.*, vol. 35, no. 2, pp. 26–40, Apr. 2015.

[73] *Virtual Commissioning*, Standard VDI/VDE 3693, 2016. [Online]. Available: http://www.vdi.de/3693

[74] G. G. Wang, "Definition and review of virtual prototyping," *J. Comput. Inf. Sci. Eng.*, vol. 2, no. 3, pp. 232–236, Sep. 2002.

[75] V. Brandstetter and J. C. Wehrstedt, "A framework for multidisciplinary simulation of cyber-physical production systems," *IFAC-PapersOnLine*, vol. 51, no. 11, pp. 809–814, 2018.

[76] G. Grinshpun, T. Cichon, D. Dipika, and J. Rossmann, "From virtual testbeds to real lightweight robots: Development and deployment of control algorithms for soft robots, with particular reference to industrial peg-in-hole insertion tasks," in *Proc. 47st Int. Symp. Robot. (ISR)*, Jun. 2016, pp. 1–7.

[77] N. Mohammadi and J. E. Taylor, "Smart city digital twins," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Nov./Dec. 2017, pp. 1–5.

[78] S. Sierla, V. Kyrki, P. Aarnio, and V. Vyatkin, "Automatic assembly planning based on digital product descriptions," *Comput. Ind.*, vol. 97, pp. 34–46, May 2018.

[79] M. Rönkkö, M. Kärkkäinen, and J. Holmström, "Benefits of an item-centric enterprise-data model in logistics services: A case study," *Comput. Ind.*, vol. 58, nos. 8–9, pp. 814–822, 2007.

[80] D. Taibi, V. Lenarduzzi, and C. Pahl, "Processes, motivations, and issues for migrating to microservices architectures: An empirical investigation," *IEEE Cloud Comput.*, vol. 4, no. 5, pp. 22–32, Sep. 2017.

[81] C. Anderson, "Docker [software engineering]," *IEEE Softw.*, vol. 32, no. 3, pp. 102–c3, May/Jun. 2015.

[82] M. Alam, J. Rufino, J. Ferreira, S. H. Ahmed, N. Shah, and Y. Chen, "Orchestration of microservices for IoT using docker and edge computing," *IEEE Commun. Mag.*, vol. 56, no. 9, pp. 118–123, Sep. 2018.

[83] C. Boettiger, "An introduction to docker for reproducible research," *SIGOPS Oper. Syst. Rev.*, vol. 49, no. 1, pp. 71–79, Jan. 2015.

[84] P. Baran, "On distributed communications networks," *IEEE Trans. Commun. Syst.*, vol. 12, no. 1, pp. 1–9, Mar. 1964.

[85] A. Levcovitz, R. Terra, and M. T. Valente, "Towards a technique for extracting microservices from monolithic enterprise systems," 2016, *arXiv:1605.03175*. [Online]. Available: https://arxiv.org/abs/1605.03175

[86] G. Mazlami, J. Cito, and P. Leitner, "Extraction of microservices from monolithic software architectures," in *Proc. IEEE 24th Int. Conf. Web Services (ICWS)*, Jun. 2017, pp. 524–531.

[87] M. Mazzara, N. Dragoni, A. Bucchiarone, A. Giaretta, S. T. Larsen, and S. Dustdar, "Microservices: Migration of a mission critical system," *IEEE Trans. Services Comput.*, to be published.

**JUUSO AUTIOSALO** was born in 1991. He received the B.Sc. and M.Sc. degrees from Aalto University, in 2015 and 2017, respectively, where he is currently pursuing the Ph.D. degree. He is currently the Project Manager for the DigiTwin project, which is executed in tight collaboration with industry partners. Before focusing on the intersection of the Internet of Things and machine design, he implemented several study projects on mechatronics and wrote his M.Sc. thesis on hydraulic accumulators. His research aims to enable the practical implementation and network effects of digital twins.

**JARI VEPSÄLÄINEN** received the B.Sc. and M.Sc. degrees from Aalto University, in 2014 and 2016, respectively. He is currently pursuing the Ph.D. degree with Aalto University. He was a Research Assistant with the Fluid Power Laboratory. During his doctoral studies, he has also been involved in course planning and assisting in teaching of mechatronics.

**RAINE VIITALA** was born in 1992. He received the M.Sc. and D.Sc. degrees in mechanical engineering from Aalto University, in 2017 and 2018, respectively. In 2018, he visited Technische Universität Darmstadt, researching rotating kinetic energy storages, i.e., flywheels. He has a solid background in experimental large rotor research, including vibration analysis and subcritical vibration, bearing excitations, and roundness measurements and manufacturing for operating conditions. His work can be widely applied in several industrial applications, such as electric motors and generators, turbines, and paper machines. He is involved in developing an AI and simulation enhanced digital twin of a rotor systems. He is instructing four doctoral candidates, three master's thesis students, and two bachelor's thesis students. In addition, he lectures a bachelor level course, mechatronics basics. His doctoral dissertation was acknowledged with the Aalto University School of Engineering dissertation award.

**KARI TAMMI** (M'15) was born in 1974. He received the M.Sc., Lic.Sc., and D.Sc. degrees from the Helsinki University of Technology, in 1999, 2003, and 2007, respectively. He received a Teacher's pedagogical qualification from the Häme University of Applied Sciences, in 2017. He was a Researcher with CERN, the European Organization for Nuclear Research, from 1997 to 2000, and a Postdoctoral Researcher with North Carolina State University, USA, from 2007 to 2008. From 2000 to 2015, he was a Research Professor, a Research Manager, the Team Leader, and other positions with the VTT Technical Research Centre, Finland. He has been an Associate Professor with Aalto University, since 2015. He is currently a Chief Engineer Counselor with the Finnish Administrative Supreme Court. He has authored more than 60 peer-reviewed publications cited in over 1500 other publications. He is a member of the Finnish Academy of Technology. He serves as the Deputy Chair for IFTOMM Finland.