

Received May 21, 2020, accepted June 1, 2020, date of publication June 5, 2020, date of current version June 19, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3000437

# A Requirements Driven Digital Twin Framework: Specification and Opportunities

**JAMES MOYNE**<sup>1</sup>, (Member, IEEE), **YASSINE QAMSANE**<sup>1</sup>,  
**EFE C. BALTA**<sup>1</sup>, (Graduate Student Member, IEEE),  
**ILYA KOVALENKO**<sup>1</sup>, (Graduate Student Member, IEEE),  
**JOHN FARIS, KIRA BARTON**<sup>1</sup>, (Member, IEEE), AND  
**DAWN M. TILBURY**<sup>1</sup>, (Fellow, IEEE)

Department of Mechanical Engineering, University of Michigan, Ann Arbor, MI 48109, USA

Corresponding author: James Moyne (moyne@umich.edu)

This work was supported in part by the National Science Foundation (NSF) under Project NSF-1544678, and in part by the National Institute of Standards and Technology (NIST) under Award 70NANB19H090.

**ABSTRACT** Among the tenets of Smart Manufacturing (SM) or Industry 4.0 (I4.0), digital twin (DT), which represents the capabilities of virtual representations of components and systems, has been cited as the biggest technology trend disrupting engineering and design today. DTs have been in use for years in areas such as model-based process control and predictive maintenance, however moving forward a framework is needed that will support the expected pervasiveness of DT technology in the evolution of SM or I4.0. A set of requirements for a DT framework has been derived from analysis of DT definitions, DTs in use today, expected DT applications in the near future, and longer-term DT trends and the DT vision in SM. These requirements include elements of re-usability, interoperability, interchangeability, maintainability, extensibility, and autonomy across the entire DT lifecycle. A baseline framework for DT technology has been developed that addresses many aspects of these requirements and enables the addressing of the requirements more fully through additional specification. The baseline framework includes a definition of a DT and an object-oriented (O-O) architecture for DTs that defines generalization, aggregation and instantiation of DT classes. Case studies using and extending the baseline framework illustrate its advantages in supporting DT solutions and trends in SM.

**INDEX TERMS** Digital twin, industry 4.0, modeling, prediction, smart manufacturing.

## I. INTRODUCTION

The evolution of manufacturing during the new smart manufacturing (SM) or Industrie 4.0 era is really part of a continuum that has existed for many decades punctuated by greater integration, vertically and horizontally, across the manufacturing ecosystem; big data trends in the “5 ‘V’ areas of *volume*, *velocity*, *veracity* (data quality), *variety* (data merging), and *value* (including better and better use of analytical techniques); more distributed and coordinated intelligence especially using internet technology; and improvements in capabilities and use of virtual representations of components and systems [1]–[5]. The terms “smart manufacturing” and “Industrie (Industry) 4.0” are often used interchangeably. In this paper we will refer to these concepts collectively

The associate editor coordinating the review of this manuscript and approving it for publication was Liang-Bi Chen<sup>1</sup>.

as smart manufacturing. The capabilities provided by SM solutions are commonly organized as tenets of SM, with one of many depictions shown in Figure 1 [6]. Among the tenets, digital twin (DT), which represents the capabilities of virtual representations of components and systems, has been cited as the biggest technology trend disrupting engineering and design in 2020 [7]. DTs are defined in [8] as software representations of components, assets, systems, and processes that are used to understand, predict, and optimize performance in order to achieve improved business outcomes. Considering this DT definition it can be argued that solutions in use today such as model-based process control (MBPC) and predictive maintenance (PdM) use DT technology [9]–[12], with the DT clients or users of the DT capability ranging from low level equipment, components and processes up through high level manufacturing execution systems (MESs) and enterprise resource planning (ERP) systems.

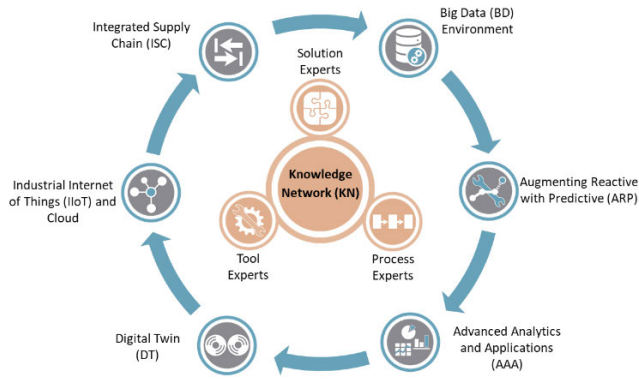


FIGURE 1. One depiction of the tenets of smart manufacturing [6].

The state-of-the-art of DTs provides insight into definitions and solutions in the DT space, but also places requirements on a future DT framework that should incorporate these existing solutions. Moving forward, the importance of DT technology will continue to grow, driven by needs such as tighter objectives of throughput, yield and cost reduction, mass personalization in manufacturing, faster and smarter verification and validation of systems, higher levels of reconfigurability, and tighter integration across the entire manufacturing eco-system including the consumer space [13]–[15]. Over the longer-term, aspects of the vision for DT have been defined for specific industries (e.g., microelectronics [1], [16]) and future greenfield manufacturing environments [4], as part of an overall SM vision.

It is clear from an analysis of these trends and vision aspects that DTs will play an increasingly important role in manufacturing with a flexible DT infrastructure being an integral part of all manufacturing systems. Unfortunately, lack of consistent definitions of capability, structure, and form prevents us from being able to easily re-use, scale, extend, verify, validate, integrate, interchange, and maintain DT technology, approaches and solutions. Also, it does not allow us to develop a clear roadmap for achieving a DT vision. A framework for DT creation through operation and maintenance is needed that supports re-usability, extensibility, interoperability and interchangeability today and greater levels of autonomy in the future.

This paper proposes a baseline framework for DT technology that leverages the knowledge gained from the development of existing DT solutions and incorporates the requirements placed on DT technology by SM trends and the ultimate DT vision. The framework includes a definition of an object-oriented (O-O) architecture for DTs that incorporates the bottom-up knowledge gained from practical development and implementation of today’s DT classes, while addressing requirements such as re-usability, extensibility, interoperability, interchangeability and autonomy incurred by SM trends and the ultimate SM vision. Specifically, following this introduction and a list of acronyms used throughout the paper (Table 1), background information is provided on DT definitions, existing DT capabilities in manufacturing, and DT trends and vision. This is followed by an analysis of the issues

TABLE 1. List of acronyms.

Acronym	Meaning
AGI	Artificial General Intelligence
AI	Artificial Intelligence
AM	Additive Manufacturing
APC	Advanced Process Control
AAS	Asset Administration Shell
Cpk	Process Capability
DT	Digital Twin
ERP	Enterprise Resource Planning
I4.0	Industry’ (Industry) 4.0
KPI	Key Performance Indicator
MBPC	Model-Based Process Control
MES	Manufacturing Execution System
O-O	Object Oriented
PdM	Predictive Maintenance
R2R	Run-to-Run (Control)
ROC	Receiver Operating Characteristic
RUL	Remaining Useful Life
S/D	Scheduling/Dispatch
SDC	Software Defined Control
SECS	SEMI Equipment Communication Standard
SM	Smart Manufacturing
SME	Subject-Matter-Expert or Subject-Matter-Expertise
V&V	Verification and Validation

preventing the realization of the DT continuum, from today’s solutions through the ultimate DT vision. DT framework requirements are then derived, and a baseline DT framework is proposed, consisting of a concise DT definition and an O-O architecture that meets these requirements. A summary analysis of these requirements and framework attributes is then provided, and gaps that remain are identified; building on the baseline specification that fills these gaps would provide a roadmap for realizing a next generation practically usable DT framework. Case studies and framework extension examples are also provided summarizing previous works by the authors and detailing a case study solution for semiconductor manufacturing and a framework extension for maximizing DT benefit. This paper concludes with a summary of the contributions, an overview of the current state of DT technology with respect to the baseline framework, and a discussion of key challenges that must be addressed to realize a practical DT framework that supports current solutions through the ultimate DT vision.

## II. BACKGROUND

### A. DIGITAL TWIN DEFINITIONS

Digital twins were anticipated by David Gelernter’s 1991 book *Mirror Worlds* [17]. In the manufacturing sector, the concept of DT was first introduced by Michael Grieves, then of the University of Michigan, in his presentation about Product Lifecycle Management (PLM) in 2002 [18]. The author presented a conceptual model for a virtual, digital representation equivalent of a physical product that includes the real space, virtual space, and the data and information flow between the two spaces.

Since then, the “digital twin” concept has been defined in many ways in academia and industry. Qamsane, et al. provides a literature search of common DT definitions and associated efforts [19]. These include contributions from NASA and the US Air Force Research Laboratory [20], General Electric (GE) [8], the Industrial Internet Consortium [21], and a number of academic sources [22]–[25].

To date, no consensus has been reached on a DT definition in manufacturing. The DT label is often applied to any capability that replicates some aspect of a system, e.g., simulation of anything in manufacturing. It is the opinion of the authors that most DT definitions either implicitly or explicitly state the following properties of a DT:

- It is some level of replica of a real thing.
- It exists in the cyber world, i.e., it is a software entity.
- It has a purpose of impacting an aspect of the environment in which its real counterpart exists, in a positive way, usually by serving one or more DT clients. This purpose can be broad or very refined and the impact mechanism can vary widely.
- It uses models to achieve its purpose.
- It incorporates some level of subject-matter-expertise (SME, which will also be used to refer to “subject-matter-expert” in this paper) in the solution. This could be as simple as defining the problem, or as complex as being an integral part of the model solution. Some efforts illustrate that the combination of SME and data provides more effective, robust and usable solutions in many manufacturing domains, than purely data-driven solutions [6], [9].
- It uses data to maintain some type of synchronization with its real counterpart. In most definitions, this data is collected in an operation environment.

## B. DT USES TODAY AND CURRENT TRENDS

DT technology has been implemented in several ways in industry. Oftentimes the technology is not referred to as “digital twin”, and, in many cases, it has existed long before the term “digital twin” was coined. For example, Model-Based Process Control (MBPC), virtual metrology, virtual sensing, sensor fusion, and model-based PdM are technologies commonly used in manufacturing today that align with most of the DT properties listed above to achieve an objective such as increase availability and efficiency of equipment, extend useful life, and reduce life cycle cost [11], [26]–[30]. DTs are used today in many industrial areas such as product design, production, and health monitoring [31]–[41]. A summary review of many of these efforts is provided in [19].

DT development processes and implementations today are usually not highly coordinated (e.g., as in a group of MBPCs implemented across a set of similar process chambers in semiconductor manufacturing [42], [43]); however, aspects of reusability of DT components are being explored such as model forms and development and implementation procedures. Additionally, examples have begun to appear in

the literature exploring the benefits of combining DTs and developing DT frameworks. For instance, Plattform Industrie 4.0 released a specification report about Asset Administration Shells (AAS) [13], [14], which are virtual digital representations of SM components. The report provides the AAS specifications and structure that ensures interoperability and exchange of information about SM components in a meaningful way between partners in the SM value chain. Additionally, some of the works of the authors have been focused on extending the baseline framework developed in this paper to practical implementations; an overview of some of these efforts is provided in Section VI.A.

A common element of most DT solutions today is the use of a wide variety of models and analytics. These are chosen based on the purpose that drives the creation of the DTs, and the availability of both SME and analytical resources and tools. The modeling approaches for DTs in and outside of manufacturing, while wide-ranging, can be thought of as *providing intelligence* so that each DT can perform its intended function in a satisfactory manner in a specified environment over an acceptable period of time. A review of *DT intelligence* in solutions as it is available and used today provides many insights. The first is that analytics and SME both provide contributions to the solution, and the complementary use of these is key to the quality and robustness of the DT solution [6]. While the relative contributions of each vary from DT type to type and application to application, both are always required.

The second insight is that a domain of applicability must be defined for a DT to be effective. This is often specified with a set of context such as “equipment ‘X’, process ‘Y’, for product ‘Z’, under conditions ( $a_1 - a_n$ )”. This context is usually determined using a combination of SME and analytics information, and the applicability of the solution is limited to the domain indicated by the context definition. This has an impact on both the human and cyber intelligence in that they may be limited or only applicable in that context-defined area. As an example, if the cyber intelligence is artificial intelligence (AI), which we use here to refer in general to virtual systems that perform tasks using some level of analytics such as machine learning (ML), the solution would generally fall into the narrow AI rather than Artificial General Intelligence (AGI) category [44], [45]. In other words, the developed solutions provide directed intelligence of a specific type in a specific context area utilizing specific data and behavior.

A third insight is that effective DT implementation must include a mechanism to maintain a level of intelligence in the solutions so that the desired benefit can continue to be provided. This usually means that a method of DT maintenance must be part of the solution, and this method must support an update of the DT intelligence to understand and adapt to changes in the application environment [46], [47].

In summary we can thus make the following statements on the state-of-the-art and trends for DTs in manufacturing:

- Many manufacturing industries are already successfully employing DT components, though these components have not been referred to as DTs until very recently.
- Most of the DTs in use in factory operation are dedicated DTs, each with a specific purpose such as predicting remaining useful life or optimizing product quality. A specific class or type of DT commonly has a specific objective (e.g., process optimization), pre-definition of operation environment (e.g., equipment, process, and other context), and defined methods or guidelines for development and deployment.
- DTs such as general equipment and software simulators are also available, with their purpose less directed (i.e., the capability being improved is not as succinctly specified, is not directly considered in the design of the simulator, or the capability is one of many to which the simulator can be configured to address), and capability boundaries usually not as well-defined [48]. These DTs are generally not used during actual manufacturing (e.g., they might be used for off-line analysis or planning); however, DTs that are more dedicated to a specific purpose, as described above, might be built from specific configurations using the general simulators and then applied during manufacturing.
- The quality, throughput and cost pressures in some industries have led to DT advancements and a requirement that DTs be an integral part of factory solutions in many areas.
- There is often little coordination of DT technology between the DT application areas. As an example, model-based process control (MBPC) and model-based PdM literature bases rarely overlap.
- Manufacturing is beginning to explore and benefit from abstracting and combining DT solutions [49], [50].

A review of the state-of-the-art and trends also points to clear gaps in technology that, if filled, could lead to more effective solutions. One important gap is that most DT approaches today lack mechanisms that convey elements of prediction quality, such as prediction uncertainty and model accuracy, with respect to the application environment. There are exceptions including virtual metrology solutions that use a prediction accuracy metric such as reliance index to optimize MBPC gain or metrology sampling rates, and prediction confidence in maintenance prediction [51], [52], or PdM solutions that incorporate the cost of false and missed positives into the optimization of the prediction threshold at which action is taken [53]; these solutions illustrate the importance of incorporating prediction uncertainty aspects into the application environment. Another important gap is the lack of commonality of structure and behavior among DT types. Providing this commonality would lead to improved opportunities for combining DT capabilities and integrating them into manufacturing systems.

Moving forward we can say that the industry is already successfully deploying DT components, but there is a need to leverage this base of important research and develop-

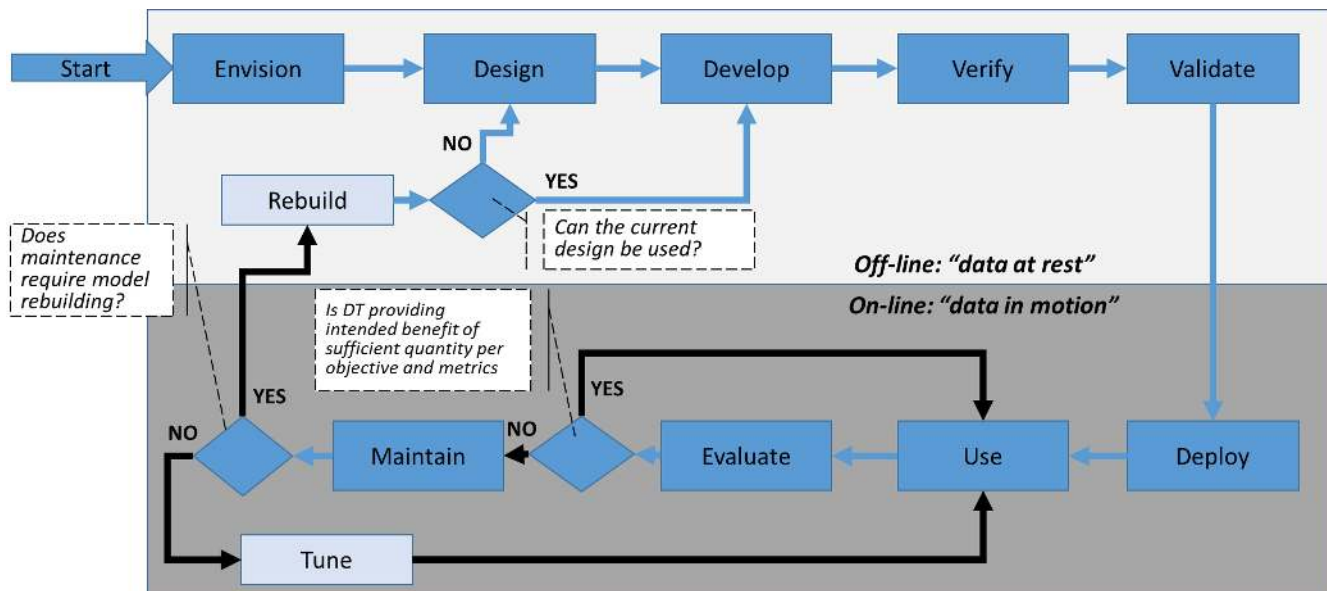
ment through a unified framework that supports DT creation, extension, exchange, reuse, and integration, while allowing for maximizing of DT intelligence.

### C. THE DT LIFECYCLE

Regardless of the DT type and implementation area, effective implementation requires good practices for solution design, development, verification, validation, deployment and maintenance. As a consequence, effective DT technologies today either implicitly or explicitly support a *DT lifecycle* in order to provide a level of guaranteed capability over a time period in a defined environment. A high-level view of a common DT lifecycle is shown in Figure 2 illustrating steps in the lifecycle process and key decision points. Additional description of the steps and transitions in the lifecycle is provided in the Appendix. Practical examples can be found in [6], [11], [19], [21], [30], [54]. The lifecycle can usually be broken down into two main phases: off-line development and on-line deployment and maintenance.

In the *off-line development phase*, data, analytics and SME are usually brought together to envision, design, develop, and verify and validate (V&V) DT solutions prior to deploying on-line. The off-line data (often called “data at rest”) is usually historical data, which is used along with SME and analytics to understand the application environment (e.g., data quality, ability to merge data, and level of supervision or the degree to which “output” data is available to relate to input data of the data set), determine the feasibility of building a DT that can provide benefit (noting that it can be determined at this phase that building a usable DT is not feasible, e.g., due to poor data quality), develop candidate solutions, and quantitatively V&V these solutions. Verification is the process of determining whether a design meets a set of requirements, specifications, and regulations, whereas validation is the process of determining whether a design meets the needs of the user [55]. In the context of DT, verification could include verifying base capabilities against standards and specifications, quantifying metrics on domain of applicability, with validation including validating the quality of DT output across this domain, determining the expected benefit such as financial benefit of applying the solution, and determining the robustness of the solution across methods for solution maintenance [6].

In the *on-line deployment and maintenance phase*, the qualified off-line solution is deployed, used, continuously evaluated, and maintained (the maintenance phase of the DT lifecycle is also referred to as “service” in the literature). The deployment includes the integration of the DT capability into the existing system, including DT data, interfaces, services and behavior, so that the DT capability is effectively used by DT clients. Once deployed, the DT uses select run-time data from its operation environment (often called “data in motion”) to assess the state or condition of aspects of the environment and make recommendations, thereby fulfilling its intended purpose. Continued effective use of the DT requires maintenance of the DT capability. This often



**FIGURE 2.** High-level view of DT lifecycle. Additional description of steps and transitions is provided in the Appendix. Note that blue arrows represent the expected flow through the lifecycle; if issues are encountered at a step, any previous step could be re-visited. See Table 3 for further description of steps and transitions.

includes continuous evaluation of the DT to determine if and when the DT should be updated. The updates could be implemented on-line (e.g., with model tuning) or might require the DT to be taken off-line (e.g., for model rebuilding). DT model maintenance is an often underappreciated and weak area in the DT lifecycle [21], [30], [46], [56], [57].

**D. DTS OF THE FUTURE: NEAR-TERM AND LONGER-TERM**

The DT roadmap and vision in manufacturing are often chronicled as part of the larger SM or I4.0 vision. One common aspect of the SM vision in which DTs play an important role is the movement from reactive to predictive to prescriptive operations in manufacturing application environments [4], [58], [59]. DTs will contribute to providing the indications, predictions, and prescriptions, respectively, in these environments. Thus, many DTs are expected to evolve from providing reactive type capabilities, such as anomaly detection, to providing more predictive capabilities, such as PdM, and eventually prescriptive contributions, such as recommendations for downtime avoidance. Note that while this general trend will be ongoing in SM, there will always be a need for reactive and predictive DT capabilities. For example, there will always be a need to detect anomalies that cannot be (accurately) predicted or avoided, and there will be a need for reactive systems to reduce false negatives of predictive systems [50].

Longer term, an aspect of the DT vision in manufacturing is that DTs exist for all real items, including physical systems, processes, and parts, as a dynamic representation of those items with real-time synchronization (see [1] for a depiction of this vision with respect to systems and processes). This covers not only the ability of the DT to predict the future state of the item, but also to extend the current representation of

that item (e.g., user interface visualization) in such a way that the representation in the past and present can be extended into the future [16].

Utilizing DTs to provide recommendations for existing and newly developed manufacturing systems is an aspect of the vision for DTs. Approaches for designing production lines utilizing DTs for optimizing system performance are proposed in the literature [19], [32], [39], [40], [60], [61]. Similarly, the perspective of simulation for current and future DTs is discussed in [48].

From the DT intelligence perspective, DT technology will continue to leverage advancements in analytics to maximize DT intelligence. This will include improving existing analytics capabilities, but also embracing new capabilities such as what happened with the emergence of deep learning solutions in big data environments over the past decade [62], [63]. Even with these analytics improvements, it is expected that SME will continue to play an important role in DT solutions, with improved methods for AI-SME interaction and integration, and a movement towards a “no knowledge left behind” DT vision aspect [1], [6].

In summary, DT roadmaps and vision generally point to an increased proliferation of and reliance on DT technology. The DTs will be more integrated (vertically and horizontally), will eventually need to be dynamically created, configured, and verified and validated, and operate in an integrated environment that extends beyond the “four-walls” to the entire ecosystem [1], [4], [5]. The DTs will take advantage of advancements in analytics, data, and understanding to provide higher quality and more expansive capabilities. They will continue to use but will better integrate SME. The DTs and their capabilities in future environments will be much more coordinated with additional benefits arising

from the integration of DT information and capabilities vertically and horizontally [40], [49], [50], [58], [59], [64]–[66]. These requirements of support for proliferation, integration, coordination, flexibility and adaptation suggests the need for a DT framework to support typical roll-out software capabilities such as re-use, interoperability and interchangeability, but also more advanced and futuristic software ideas such as automatic DT creation, validation, maintenance and learning [59], [67].

### III. DIGITAL TWIN FRAMEWORK REQUIREMENTS

DT framework requirements must be derived from and satisfy the general understanding of the definition of DTs, DTs in use today, near-term trends in DTs, and DTs of the future including the ultimate vision of DT frameworks in SM. The framework requirements from these sources are presented in the remainder of this section and summarized in Table 2. Note that in this section the term “class” is used to define a type of DT, with an “instance” being a single occurrence of that class. These terms will be more formally defined and used in Section IV.

#### A. DT FRAMEWORK REQUIREMENTS DERIVED FROM A GENERAL DT DEFINITION

The most basic requirement of a DT that can be derived from its definition in manufacturing is that it must be a digital replica of something, where that something could be a physical asset (e.g., machine, component or production part), process (e.g., a drilling process), or a system (e.g., a production system for a part). It is also clear that the DT must have at least one purpose in the manufacturing system, developed or configured in order to provide information that can be used by one or more DT clients to realize some type of improvement in the system. Further, in serving its purpose it must detect, predict or prescribe something. Another key requirement is that, in order to maintain its “twin” status it must be dynamic, responding to changes in its physical counterpart so that it can continue to represent that physical counterpart with some degree of accuracy. This is often stated as a requirement that the DT be synchronized with its physical counterpart, with the level of synchronization (frequency and accuracy) being a function of the purpose of the DT in its environment.

#### B. DT FRAMEWORK REQUIREMENTS DERIVED FROM DTS IN USE TODAY

As noted in Section II, there are many examples of DTs operating in manufacturing today. These DT capabilities are modular with boundaries or the relationship between the DT and the system with which it interacts clearly defined. The DTs adhere to a DT lifecycle that includes envision, design, develop, V&V, deploy, use, evaluate and maintain steps. The DTs generally use some type of narrow DT intelligence capability, usually determined by context, to provide a specific function when provided with a specific type of information. This process can be purely statistical or data-driven, however the application of the DT usually requires the marrying

of analytics with SME [6], [30], [54]. As these solutions must be part of any DT framework, that framework must support the evolution of these capabilities as opposed to providing a revolutionary, disruptive approach to DT organization. Also, while the DT classes in use are quite disparate, one key requirement is that each provides a quantifiable net value-add to some aspect of the manufacturing system. In other words, the benefit of DT correct operation (whether detection, prediction, or prescription) must outweigh the cost of incorrect operation.

#### C. DT FRAMEWORK REQUIREMENTS DERIVED FROM DT APPLICATIONS IN THE NEAR FUTURE AND CURRENT TRENDS

As noted in Sections I and II, DT proliferation will continue to increase, and the cost of development, validation, integration and maintenance of DTs will rapidly become intractable unless certain requirements are met in a DT framework. DT classes must become increasingly capable and accurate, leveraging big data ‘V’ improvements (see Section I and [1]) such as improved analytical approaches. Solution spaces will emerge that require the integration of DTs with other instances, with different DT classes, and with non-DT capabilities. Further, as the SM concepts proliferate beyond the “four-walls” of the manufacturing facility to the entire ecosystem, DT solutions must follow. These trends of proliferation and integration place requirement attributes on a DT framework that are common to many manufacturing advancements such as field-bus networks and reconfigurable manufacturing systems [68], [69]. Many of these “ility” requirement attributes (such as modularity, interoperability, interchangeability, flexibility, scalability, re-usability, and diagnosability) are applied to the DT framework resulting in the following requirements:

*DT re-usability:* DT solutions must become more portable and re-usable so that the “develop once use many” approach can be better leveraged. This also results in improved scalability of the DT solution. While the re-usability concept can quickly become quite complex with the degree of re-usability definable, techniques such as non-threaded MBPC and analytical clustering can be leveraged to determine if model development efforts in different context environments can be leveraged [10], [46], [70], [71].

*DT interoperability:* DT solutions must be able to interoperate with other DT instances, DT classes, and non-DT capabilities such as DT clients. As an example, DT fleet solutions, which leverage interoperability among DT instances, have been proposed for a range of applications including process matching [42], [66], integration across different levels of the manufacturing hierarchy [10], [49], [67], and augmentation of performance monitoring and anomaly detection methods [9], [11], [72].

*DT Interchangeability:* The proliferation of DTs requires that they be modular so that they can be more easily assessed, updated, or even replaced. This interchangeability can be realized with standardized definitions of DT structure,

**TABLE 2. DT framework requirements delineated by category. References cited provide motivation for the requirements.**

#	Requirement
<b>1</b>	<b>Requirements Related to DT Definition and DTs in Use Today [19], [16]</b>
<b>1a</b>	The DT capability is modular; boundaries and relationship with the interacting physical twinned system are clearly defined.
<b>1b</b>	The DTs use models to replicate an aspect of an asset, process, system or product.
<b>1c</b>	The DT solution must also include some computational engine that supports the model and provides the required capability to the DT client.
<b>1d</b>	The DT must be able to use some form of narrow DT intelligence that allows the DT to provide its capability in a specified application domain [45].
<b>1e</b>	The DT must be able to incorporate analytics and SME in a complementary fashion to realize intelligence in solutions.
<b>1f</b>	The DT must deliver its capability in terms of one or more quantifiable metrics.
<b>1g</b>	The accuracy of the DT in providing these quantifiable metric(s) must be ascertainable and quantifiable.
<b>1h</b>	The DT must deliver a quantifiable net value-add in its application environment.
<b>1i</b>	The financial benefit of correct operation of the DT in its environment as well as the cost of incorrect operation in that environment must be ascertainable and quantifiable [53], [6].
<b>1j</b>	The framework must support an evolution rather than revolution of capabilities, especially supporting the evolution of existing capabilities to align with the ultimate DT framework vision.
<b>1k</b>	The framework must support the entire DT lifecycle from envisioning and design through development, validation, deployment and maintenance [11], [30], [21], [54].
<b>2</b>	<b>Requirements Related to DT Application in the Near Future [1], [58], [59], [4], [50], [34], [25], [81]</b>
<b>2a.1</b>	<i>Re-usability:</i> DT solutions must be portable, re-usable and scalable.
<b>2a.2</b>	<i>Re-usability:</i> Degree and method of re-usability (e.g., data translation, sub-set of metrics) must be definable.
<b>2b.1</b>	<i>Interoperability:</i> Multiple instances of the same DT class must be allowed to interact in a coordinated fashion.
<b>2b.2</b>	<i>Interoperability:</i> Integration of and coordination between instances of different DT classes must be supported.
<b>2b.3</b>	<i>Interoperability:</i> Integration of and coordination between DT and non-DT components must be supported.
<b>2c.1</b>	<i>Interchangeability:</i> Interchangeability of different instances of the same DT class must be supported.
<b>2c.2</b>	<i>Interchangeability:</i> The framework must support standardized definitions of DT structure, baseline minimum abilities, quantifiable capabilities metrics, exposed interfaces, services provided, and behavior exhibited.
<b>2d.1</b>	<i>V&amp;V capability:</i> Standardized, reusable and quantifiable V&V processes must be supported [55].
<b>2e.1</b>	<i>DT maintainability:</i> The minimum required DT output quality (see requirement 1g) to provide its intended capability must be quantifiable. Diagnosability of lack of sufficient quality of DT output (e.g., outside of threshold limits) should be identifiable in a time-critical fashion.

baseline abilities, quantifiable capabilities metrics, services provided, interfaces, and behavior exhibited [1], [15], [73].

**TABLE 2. (Continued.) DT framework requirements delineated by category. References cited provide motivation for the requirements.**

#	Requirement
<b>2.e.2</b>	<i>DT maintainability:</i> the DT should be updated to continue to provide sufficient output quality if that level of maintenance is a requirement for the DT (e.g., as a form of scheduled maintenance, or as a consequence of the DT providing lack of sufficient output quality, see requirement 2.e.1).
<b>2e.3</b>	<i>DT maintainability:</i> DT improvement support must include solution model tuning as well as model re-building or replacement; this capability should trend toward being fully automated.
<b>2f.1</b>	<i>DT capability and accuracy:</i> DT solutions must be able to use evolving analytics techniques, including improvements on existing techniques and novel new techniques.
<b>2f.2</b>	<i>DT capability and accuracy:</i> DT solutions must support structured and automated integration of analytics and SME information.
<b>2g.1</b>	<i>DT extensibility:</i> DT framework must be extensible to support DT solutions across the entire SM ecosystem.
<b>2g.2</b>	<i>DT extensibility:</i> DT framework must address security requirements, including data partitioning and IP security required for DT operation across the entire SM ecosystem [76], [82].
<b>2h.1</b>	<i>Sustaining a DT technology community:</i> DT framework must provide a common DT definition, taxonomy and other mechanisms that allow the community to collaborate on DT technology, from DT fundamental research through applied research, development, deployment, and maintenance
<b>3</b>	<b>Requirements Related to DT Application Over the Longer Term [4], [1], [20]</b>
<b>3a</b>	DT framework must support virtual counterparts across the entire SM ecosystem (including the full supply chain), that can be used for detection, prediction, prescription and analysis of all aspects of operation. [77], [79], [78].
<b>3b</b>	Automation of DT creation, validation and integration must be supported; generally, automation of the entire DT lifecycle must be supported, except for aspects of envisioning and design [59], [83].
<b>3c</b>	DT framework must support an evolution from narrow intelligence towards more general intelligence with fewer context restrictions (e.g., narrow AI to AGI) [45], [44].
<b>3d</b>	DT framework must support the marrying of SME and analytics as a continuing integral part of DT capability and evolution [6].

*DT V&V capability:* As DTs will increasingly be an integral part of the critical manufacturing processes, their capability must be verified and validated prior to accepting them for use in application environments. Today the V&V process is largely ad hoc, and heavily dependent on the type of DT. For example, MBPC and PdM DT instances are often verified and validated individually on historical data sets. In the future, more standardized, reusable and quantifiable V&V processes must be part of the DT technology capability.

*DT maintainability:* An often-under-estimated requirement of DTs is that they be maintainable over a useful period of time. As an example, robust predictive measurement techniques are usually part of an overall measurement strategy that includes an actual physical measurement capability.

The physical measurement is done infrequently with the data used to update or “tune” the predictive measurement model in an on-line fashion [26]. If the error between prediction and actual is too large or persistent, maintaining the prediction model in the DT may require off-line rebuilding of that model [46], [74]. Identifying when to maintain a DT is an aspect of diagnosability.

*DT capability and accuracy:* DTs will have to become increasingly capable and accurate as part of SM evolution. This requires that the DTs be able to best use the evolving analytical capabilities, whether they be improvements in existing techniques due to big data advancements, or emergence of new techniques, such as the appearance of deep learning [9], [62], [63]. Note that this analytics evolution does not imply that the importance of the SME will be lessened, but rather that the analytics-to-SME integration will become more structured and automated [6].

*DT extensibility:* The SM trend towards integration across the entire ecosystem will require that DTs be extensible across that ecosystem. This means that DT capabilities operating within the “four-walls” of the factory today might in the future incorporate data outside the facility. A good example here is the extension of MBPC or yield prediction beyond the four-walls where the control is to field-yield targets rather than just process targets or even factory-wide yield targets [6], [75]. Unfortunately, this extensibility requirement is associated with many challenges, the most significant of which is maintaining data partitioning and IP security throughout the manufacturing ecosystem [76]. Techniques such as block-chain have been proposed to address aspects of this challenge; however, the scope of the issue will require innovative data partitioning, traceability and translation techniques combined with multi-industry cooperation, standardization and best practices [77]–[80]. While the requirement of ecosystem security is included as a DT technology requirement, it is not addressed in detail in this paper.

*Sustaining a DT technology community:* The increasing and more stringent requirements on DT technology will only be realizable if there is more cooperation between the various DT application areas and the technology community in general. An effective DT framework thus must provide a common taxonomy and other mechanisms that allow the community to collaborate on DT technology, from DT fundamental research through applied research, development, deployment, and maintenance.

#### **D. DT FRAMEWORK REQUIREMENTS DERIVED FROM DT APPLICATION NEEDS OVER THE LONGER TERM**

As noted in Section II, while there are several aspects of the DT vision being proposed including DT solutions in futuristic greenfield environments, common themes exist that can be used to distill DT framework requirements [1], [4]. The first is that there will be a virtual counterpart to the entire SM ecosystem, used for detection, prediction, prescription, and analysis of all aspects of operation. The requirements of a

DT framework to address this theme include those identified in the previous sub-section, namely re-usability, interoperability, and interchangeability. However, this proliferation theme also places requirements on the automation of the DT creation, validation and integration process. More broadly, the majority of the DT lifecycle process from creation through maintenance (see Figure 2) must become fully automated, with manual DT efforts limited to envisioning, innovation, and responding to unforeseen situations.

Another theme that is postulated is that DTs will trend away from narrow intelligence (e.g., narrow AI) towards general intelligence (e.g., AGI [44], [45]) as the analytics’ fields advance. It is the opinion of the authors that this is not a revolution but rather a general evolution towards less narrow intelligence solutions, and the framework must support this evolution.

#### **IV. APPLYING THE REQUIREMENTS TO REALIZE A DIGITAL TWIN FRAMEWORK**

In this section the requirements summarized in Table 2 are applied to distill baseline minimum required attributes of a digital twin framework. This baseline DT framework includes a definition of a DT and a method for the generalization and combination of DTs under a class structure using O-O constructs. The baseline framework derivation will be followed with a summary of how these framework constructs address requirements identified in Table 2, and a discussion of remaining gaps that must be addressed to realize a practical framework. We will then provide case studies and extensions in Section VI to illustrate the importance and practical use of these framework constructs.

While this framework does not address all the requirements in Table 2, it provides a baseline framework that can be extended in a manner suitable for individual application environments so that all requirements can be achieved through additional specification. For example, Qamsane, et al. provides a methodology for realizing a practical framework from this baseline framework [67]. Additionally a recently proposed framework that leverages the DT concept to ensure interoperability and exchange of information about SM components is the Asset Administration Shell (AAS) [13]. In this framework each AAS is structured to enable communication with other AASs and to allow for generalization and aggregation of AASs. While the AAS framework meets some of the requirements put forth in Table 2, there are still extensions to AAS components that must be developed to realize all the requirements proposed herein. A good approach to realizing a practical DT framework then would be to use the baseline framework presented in this section and extend that framework to address practical issues of the specific application environment.

##### **A. DIGITAL TWIN DEFINITION PROPOSAL**

We define a DT as a purpose-driven dynamic digital replica of a physical asset, process, system, or product. A DT is driven by a need for improvement of the manufacturing



environment (e.g., reduce unscheduled downtime, generate a production plan, improve quality, etc.). A DT is a digital replica of an aspect of a manufacturing system; it is usually not a complete replica of an entire system or component. It provides a capability in terms of detection, prediction and/or prescription so that it can deliver on its intended purpose and provide a value-add capability to a DT information client in the SM ecosystem. It is limited in scope, applicability and capability to its intended purpose and defined application environment. This application environment is usually defined by context. An example might be MBPC where the purpose is to improve process quality, and the application environment is the production operation associated with a particular part at a specific process.

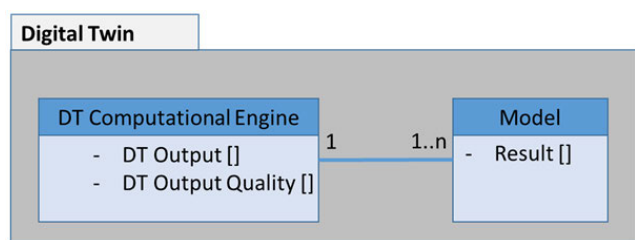


FIGURE 3. A DT representation illustrating the relationship between the modeling resource and computational engine in a DT.

The DT provides its capability by combining a modeling resource with a computational engine, as shown in Figure 3. The modeling resource is used to emulate some aspect or aspects of the physical “twin”. The modeling resource can consist of one or more models. As noted above, these models generally use analytics technology to define behavior within a particular environment defined by context. The model type can range from completely physical through phenomenological to purely statistical (data-driven) with an important goal of DT modeling and intelligence gathering approach being that it supports the incorporation of data and SME in solutions to strive for a “no knowledge left behind” solution [6].

The DT computational engine is used to support the model in terms of accuracy and alignment with its real counterpart, and provide the required output (i.e., deliver on the DT purpose) to the DT client. The alignment requires that the computational engine addresses the management of updating the model with new data from the physical system, and maintaining the model so that it continues to provide information of necessary quality to support the DT delivering on its intended purpose. The synchronization level (e.g., rate and quality) of the DT with its physical counterpart is a function of the DT’s intended purpose and the application environment. This is referred to as “time critical” in this paper to indicate that the timing is a function of the application environment. “Real-time” is not used here as the term is vague and may be misleading. Synchronization requirement examples include “every X <timeperiod>”, and “within <timeperiod-1> following the occurrence of an event ‘a’ and <timeperiod-2> thereafter until the occurrence of event ‘b’”.

The computational engine provides the necessary computational resources so that the DT can deliver on its intended DT capability (as shown in Figure 3). These resources include coordinating the use of one or more models and ensuring that the models have the necessary data and context to maintain synchronization with the physical counterpart, as described above. The DT capability is delivered by the DT producing the following information in a manner suitable for the DT client:

- One or more metrics that quantify the DT output as it relates to the DT purpose in the application environment for the DT client. The metrics could vary greatly and could include a binary indication of an event (e.g., fault), a continuous parameter related to prediction of a future event (e.g., RUL), a recommendation for reconfiguration (e.g, a new product route), or a prescription for future avoidance of an event (e.g., a change in machine operational parameter ranges). The metric is usually related to some aspect of the desired performance such as a key-performance-indicator (KPI).
- One or more metrics that quantify the quality or believability of the DT output (detection, prediction or prescription) as it relates to the intended purpose for the DT client. The DT utilizes data to estimate or predict some aspects of its real counterpart. The quality of the output metrics might relate to the quality of the detection, prediction, or prescription of an event (e.g., probability that the event has/will/can be prevented from occurred/occur/from occurring), and the quality of the prediction horizon (e.g., a confidence interval on the predicted occurrence of a future event and the profile of the indicator leading up to that predicted event). Other quality-of-output metrics might relate aspects of “how”, “where”, “when”, “why” or “to what extent” associated with the output. Some aspects of the quality of the DT output can be expressed with a receiver-operating-characteristic (ROC) curve, which is created by plotting the true positive (detection/prediction/prescription) rate against the false positive/alarm rate [84]. Model quality can be ascertained in-part from the shape of the ROC including the area under the curve, while confidence intervals can also be displayed on the plot.

The DT is required to deliver a net value add in its application environment (see Table 2, requirement 1h). This means that the DT output and output quality information must be utilized effectively to determine the net value add of the implementation of the capability.

**B. BASELINE DIGITAL TWIN OBJECT-ORIENTED FRAMEWORK**

Many of the DT framework requirements summarized in Table 2, such as re-usability, extensibility, interoperability and interchangeability, are common to manufacturing disciplines (e.g., reconfigurable manufacturing systems [69]) and

other areas ranging from software to network control systems (e.g., [68]), equipment representations and even social sciences. Researchers in these areas generally turn to some form of “object-oriented (O-O) technology” as an aid to define structure, behavior, and relationships between types of “things”. Booch et al., [85], is often cited as the essential reference for O-O technology though there is a large volume of literature covering definitions, taxonomy, languages (such as Unified Modeling Language-UML), etc. of O-O [86], [87]. While there is some variability in O-O technology and concepts, we will follow a common approach specified in [86]. In the remainder of this sub-section, we will provide an overview of important constructs that we will use in this taxonomy, apply the DT requirements to realize a baseline O-O framework, and provide a brief example to illustrate how the constructs work together.

### 1) INTRODUCTION: O-O CONSTRUCTS AND TERMINOLOGY

The following is a summary of the main O-O concepts that will be used in this section to present the DT O-O framework. Please refer to [85], [87] for a comprehensive treatment of these concepts.

- *Object*: A concept, abstraction or thing with crisp boundaries and meaning for the problem at hand.
- *Class*: A description of a set of objects that share the same attributes, operations, relationships, and semantics.
- *Instance*: A concrete manifestation of an abstraction; an entity to which a set of operations can be applied and that has a state that stores the effects of the operations.
- *Inheritance*: The mechanism by which more-specific elements incorporate the structure and behavior of more-general elements.
- *Generalization hierarchy (IS-A)*: A specialization/generalization relationship in which objects of the specialized element (the child) are substitutable for objects of the generalized element (the parent).
- *Aggregation hierarchy (HAS-A)*: A “has-a” relationship, meaning that an object of the whole has objects of the part. Aggregation is used to model a “whole/part” relationship, in which one class represents a larger thing (the “whole”), which consists of smaller things (the “parts”).

### 2) INTRODUCTION: TOWARDS A BASELINE DT O-O FRAMEWORK

As noted earlier, an ultimate DT O-O framework will incorporate information requirements from many sources including practical applications, industry-specific needs, and standards [1], [14], [67], [88], [89]. In this paper only a baseline of minimum framework specifications is presented, to address the requirements summarized in Table 2. Complete and more practically deployable frameworks would be based on this proposed baseline specification and provide a more restrictive set of specifications in order to better address aspects of the DT requirements.

### 3) DT OBJECT CLASS

A DT class is a type of DT that delivers a specific capability for the DT client. The scope of this capability is defined by the DT output purpose metric, thus all *DTs belonging to the same class must share at least one common output metric*. An example is RUL. DTs belonging to the same class might utilize different numbers and types of modeling approaches and might report additional output purpose metrics. DTs belonging to the same class might utilize and report different quality-of-output metrics.

*DTs belonging to the same class must have the same general defined scope of applicability*. This scope might be very broad as in “all rotating equipment” or might be very focused as in “pumps of brand ‘X’, model ‘Y’, operating in an environment with parameter ranges ‘Z’”. The scope might be defined qualitatively or quantitatively. The scope is usually defined and refined using contextual information about the equipment, components, process, people, environment, etc. Some refinement might occur at the instance level. For example, the scope of the class might be all pumps of brand ‘X’, model ‘Y’, but the instance might further refine the application environment to process ‘A’.

*DTs belonging to the same class must have common defined behavior*. This behavior should include a method (e.g., defined service) for reporting the output metric(s), quality-of-output metric(s), and any other behavior required for the DT to deliver on its intended purpose. This behavior also includes specifications and any restrictions on ability to instantiate off of the class, which will be discussed later in this section. DTs belonging to a class might have additional specifications on behavior, but these specifications cannot conflict with the common defined behavior. An example of this type of additional specification is a Harel hierarchical state-chart that defines common behavior in superstates with additional behavior defined in substates [90]. Common DT class behavior is highly dependent on the purpose that the DT has and the capability it is expected to deliver, as described in the DT definition.

Note that the naming of a class should provide some indication of purpose and scope of applicability. An example might be “RUL for rotating equipment”. There are no requirements provided here for class naming as existing implementations, trends, expertise, analytics, etc. will likely lead to colloquial naming of many application classes such as “PdM” or “MBPC”.

### 4) GENERALIZATION HIERARCHY AND INHERITANCE

A DT hierarchy defines how the DT structure, the services it provides, and the behavior it exhibits, can be re-used, combined, shared, or otherwise aggregated or inherited between DT classes or instances. As noted above, generalization allows for the extrapolation of capabilities usually from general to specific.

In a DT hierarchy, *sub-classes of a DT class must provide the DT output metric or metrics defined as required in the super-class. Sub-classes of a DT class must also comply with*

the common behavior in the super-class. The scope of applicability of a sub-class must fall within the scope of the super-class, but it does not have to be the same as the superclass. In fact, in many application environments the main purpose of sub-classing will be to refine or constrict the application environment, e.g., from “all rotating equipment” to “rotating pumps”.

Note that in many implementations sub-classing can be used to further define output and quality-of-output metrics or further define behavior. However, this is not a requirement of sub-classing.

### 5) AGGREGATION HIERARCHY

Aggregation is a specification that allows for the combination of DT object instances. The combination is associated with some physical relationship between the application environments of the individual instances and is often associated with some aspect of common purpose among the DT object instances. A practical example of aggregation is the combination of DTs that have the same specific purpose and output metric (e.g., RUL), where the DTs relate to the individual components in a larger system (e.g., to express the RUL of that system). The relationship between the components might be more tenuous as in “a count of all DT event indications on the plant floor”. Regardless of the relationship between DT components in an aggregation, a requirement of any DT aggregation is that membership in the aggregation must be specified in terms of purpose and scope. Examples include “must belong to a PdM class providing a RUL output metric for one or more components or sub-systems of ‘X’”, or “must be a DT instance that delivers a capability for factory ‘Y’”.

In some aggregations, the existence of the component DTs is dependent on the existence of the parent aggregate, or the existence of the parent aggregate requires the existence of some or all of the child components. A requirement of the aggregation is the need for existence of the parent and child DTs in the aggregation be specified. This requirement could be different for different components in the aggregate, e.g., “RUL DT for the fan must exist while RUL DTs for the heating elements are optional”.

As the aggregate itself is a DT, it must comply with the DT and DT class definition. Thus, a DT aggregate must have a defined common output metric, scope of applicability, and common behavior. Note that these specifications might or might not be the same as one or all of the DTs in the aggregate. As an example, the aggregate DT might report RUL of a machine by using its computational engine to analyze the component RUL information reported by the DTs in the aggregate and determine the RUL of the machine. A scheduling DT might use an aggregate of these same PdM-RUL DTs and provide an output metric of optimized schedule by incorporating downtime horizons in an optimization algorithm. Supporting an aggregation of DTs requires that they communicate in some way to exchange information ranging from internal parameters (e.g., modeling outputs) to performance and quality outputs. This is often accomplished

in a formalized way using associations between DTs as objects or between objects within a DT [86].

### 6) INSTANTIATION AND IMPLEMENTATION

Using the DT definition and requirements specified above, a wide variety of DT object hierarchies can be realized incorporating both generalization and aggregation. Depending on how these hierarchies are realized, instantiation of actual DT implementations could be specified as being allowed to occur at any place in the hierarchy or restricted to be below a certain level in the generalization portion of the hierarchy such as “only in the leaves of the generalization hierarchy”. For example an approach to generalization for additive manufacturing (AM), i.e., 3-D printing, which includes fused deposition modeling (FDM) and selective laser sintering (SLS) approaches, is shown in Figure 4. This example illustrates that a common approach to DT instantiation in a generalization hierarchy might be instantiation off the sub-class if it is specified, with instantiation off the super-class if the sub-class is not specified. A good example of this approach being used in industry today is the Common and Specific Device Models (CDM and SDMs respectively) specified in the semiconductor standard for network devices [68]. The requirements and restrictions for instantiations within a defined DT hierarchy must be specified. If instantiation of a superclass is allowed, then instantiation of all subclasses of that superclass must also be allowed as this is an aspect of class behavior.

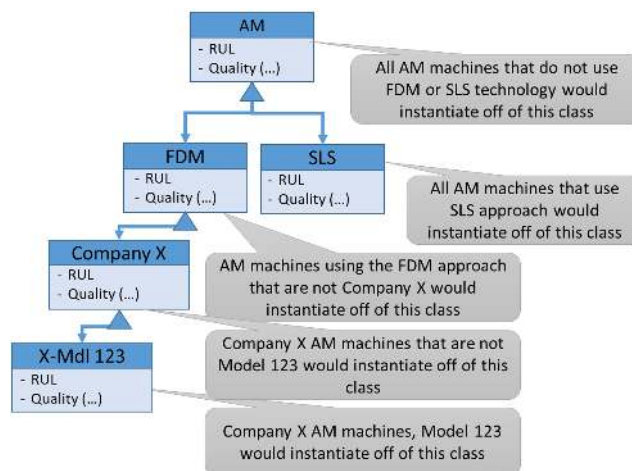
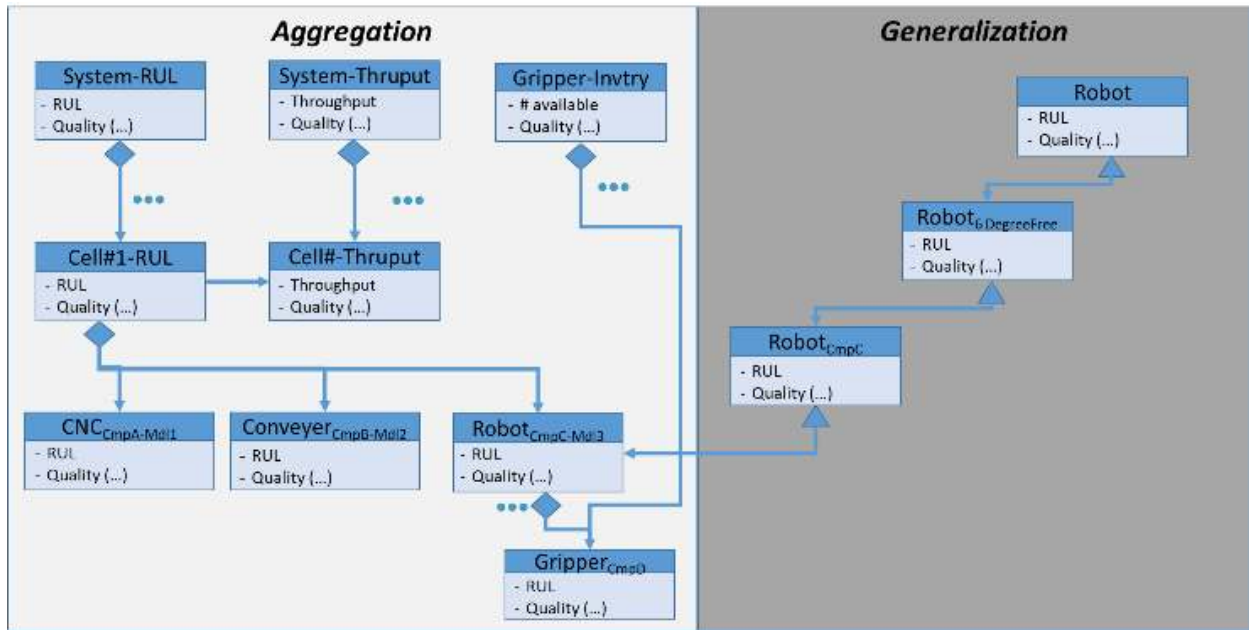


FIGURE 4. Example of a generalization hierarchy of an additive manufacturing (AM) PdM DT class supporting instantiation at multiple levels.

### C. EXAMPLES

With the baseline of minimum attributes of a DT framework defined, many examples can be realized that illustrate the opportunities for re-use, integration and interoperability and interchangeability. An example that might be representative of a DT solution for an assembly operation, combining both generalization and aggregation, is provided in Figure 5. The aggregation and generalization subclassing paths (e.g., from machine general type (Robot), to specific type (6 degree



**FIGURE 5.** Example of a DT implementation hierarchy illustrating generalization and aggregation (“Invtry” == “Inventory”; “Cmp” == “Company”; “Mdl” == “Model”).

of freedom), to producing company, to model number are examples only; the actual subclassing path would depend on the application environment, DT class characteristics, and specification developer choices. The figure is only meant to represent a portion of the generalization and aggregation hierarchies (and possibilities). For example, a generalization hierarchy is only illustrated for a RUL DT class.

A more detailed treatise of an extended DT O-O framework implementation that uses this baseline is provided in [67]. In this implementation, additional DT hierarchy specifications are provided that support a specific implementation; these illustrate how the basic specifications presented herein can be used as a template to realize practical DT implementations that better meet the requirements summarized in Table 2.

## V. MAPPING OF REQUIREMENTS TO BASELINE DT FRAMEWORK CAPABILITIES: GAPS AND OPPORTUNITIES

The DT baseline framework specification presented in Section IV addresses important aspects of most of the requirements presented in Table 2. Importantly, the framework does not invalidate or prevent the attainment of any of the requirements.

An analysis of the framework specification also reveals gaps in meeting some of the requirements; these gaps might be met with additional specifications, standards, implementation best practices, etc., to realize practical solutions, but are associated with specification choices that might not be unique to realizing an effective solution [1], [13], [15], [67].

The gaps that could be filled with additional specifications or research and development include:

- *Improving integrability, interoperability and interchangeability of objects within the DT* through definition of the interaction between models and with the computational engine in the DT object. This could include a framework for DT models [91], [92]. The specification would likely have to support a variety of different modeling and computation approaches, but could probably be much more refined for specific DT classes and application environments.
- *Improving interoperability, interchangeability, reusability, etc. of DTs throughout the entire DT lifecycle* with detailed specifications for support of the DT at each stage of the lifecycle and its transition between these stages. The specifications should be divided into off-line or “data at rest” components to support model development, V&V, and on-line or “data in motion” components to support model deployment, use and maintenance [6]. The framework presented herein is focused on the DT operation and maintenance during the on-line portion of the lifecycle. Structured methods for off-line DT development and V&V have been proposed, which could be added to the framework [9], [11], [30].
- *Improving interoperability between DT and non-DT components* with specifications that define the interaction between the DT service and clients to deliver the DT capability in an increasingly flexible environment. These could be defined in terms of the DT data structures, interfaces, services and behavior that are exposed to DT clients [1], [73]. The support for interaction could include definition of a language, data structures, interfaces, services and behavior of both DT and non-DT

components. These specifications could be added at any part of the DT hierarchy.

- *Facilitating and improving DT V&V lifecycle component capabilities* with specifications that govern V&V approaches and quantify V&V results. V&V are critical steps in the DT lifecycle and will become more important and require automation as DTs proliferate (see below).
- *Improving the maintainability component of the DT lifecycle* with methods and specifications for DT maintenance as part of lifecycle on-line (e.g., model tuning), as well as off-line (e.g., model rebuilding) activities [30], [46]. Generally speaking, automation of the on-line maintenance capability should be supported today, while automation of the off-line capability is a longer-term objective (see below).
- *Improving consistency, understanding and assessment of DT benefit through agreed upon metrics* for communicating the DT capabilities as well as the aspects of the accuracy of the DT [51]. This might be achieved via a standardization process.
- *Improving the DT quantified benefit and the understanding of that benefit* with extensions to the methods for calculation and optimization of financial net value-add, to address items such as sensitivity analysis and opportunity cost. An example of how this can be done is provided in the next section.
- *Facilitating more structured, automated and optimized methods for integration of analytics and SME* throughout the DT lifecycle process (Figure 2) [4], [6], [9]. This improvement will leverage advancements in analytics that allow them to proactively ask for and use SME information, optimally balance analytics and SME information, and be able to assimilate SME information when provided asynchronously. Also, best practices for DT-to-analytics integration at various stages in the DT lifecycle could be specified at different points in the DT generalization hierarchy.
- *Addressing security of DT data and intellectual property* with standards as well as technology advancements. As DTs will often be employed in a very heterogeneous environment in terms of applications and users throughout the manufacturing ecosystem, security of DT data, models, computational approaches and output will all have an inherent element of propriety that will have to be managed. As noted earlier, this topic is outside the scope of this paper, however efforts are underway to address issues of this type [1], [76], [82], [93].
- *Providing a platform for research and technology advancement* by extending the baseline specification to agree on standard DT terminology, classes, areas, etc., that would support collaboration. Standards and other specifications derived from a combination of technology assessment, deployment information and consensus building could provide a framework for better coordination and proliferation of technology advancement.

- *Extending the DT O-O specification to the entire SM ecosystem.* The boundaries of manufacturing will become increasingly blurred with new manufacturing paradigms incorporating optimization throughout the SM ecosystem from raw material production all the way through the end customer experience [4], [5]. DT technology will provide benefits throughout this ecosystem, with many of these arising from the coordination of DT instances (see Section VI for examples). While the framework presented herein facilitates this extension and coordination, much more specification will be required to realize a practical framework across the entire ecosystem.
- *Automating all aspects of the DT lifecycle.* The proliferation of and increased reliance on DTs across the entire manufacturing ecosystem eventually will require automation across most stages of the DT lifecycle, from design through maintenance. Many of the lifecycle steps such as evaluation and on-line maintenance are already automated in some applications, however automation of other stages, such as design, development, V&V and off-line rebuilding represent ongoing technical challenges. Additionally, automation of SME-to-analytics interaction in the various stages of the DT lifecycle will be required [4], [6].
- *Accommodating analytical trends in DT technology.* The analytics frontier is dynamic and unpredictable. Evolutionary and revolutionary (disruptive) technologies must be accommodated, or, at minimum, any DT framework should not prevent the use of these technologies. Expected directions in this space include the move towards less narrow context-restricted solutions (e.g., AGI), and more automation of the role of analytics throughout the DT lifecycle.

## VI. CASE STUDIES AND EXTENSIONS

Using the baseline DT framework to address practical manufacturing problems is explored in this section. A literature review of previous works of the authors that provide DT application case studies and framework extensions related to the baseline framework developed in this paper is presented in Section VI.A, with references to more in-depth descriptions. A case study of process-capability aware scheduling and dispatch is then presented illustrating the combination of DTs at different levels in the manufacturing hierarchy. This section concludes with a description of a framework extension that supports optimization of DT benefits (e.g., financial) in a defined application environment.

### A. DT FRAMEWORK CASE STUDIES AND EXTENSIONS

There have been several other previous works of the authors that illustrate the utility of a DT baseline framework by providing extensions and case studies of use. A reconfiguring scenario utilizing multiple DTs of a manufacturing cell to optimize cell throughput is presented in [94]. A system-level reconfiguration scenario utilizing a DT framework within a

software-defined control (SDC) architecture [81] to understand the current system state and optimize system metrics between throughput and quality is presented in [95]. An aggregation and generalization scheme for a system-level DT in a DT framework is presented in [65]. An SDC architecture utilizing a DT framework for fleets of additive manufacturing machines (AM fleet) to perform anomaly detection and reconfiguration is presented in [66], [96], and to perform fleet-process matching using MBPC is presented in [42], [43]. Following this line of work, a reference architecture for the implementation of DTs for AM with implementations on off-the-shelf systems is presented in [97]. Additionally, DT-based run-to-run controllers with a case study on system-level chamber matching utilizing a DT framework for semiconductor manufacturing is illustrated in [66]. In [4] the long-term vision on the role of DTs in a greenfield manufacturing environment of the future is presented. This includes a discussion of the use of agent-based decentralized approaches [83], [98] in conjunction with existing and new centralized approaches for a futuristic DT framework that would continuously extend and improve itself to address the manufacturing needs of the factory of the future.

### B. CASE STUDY: PROCESS-CAPABILITY AWARE REAL-TIME SCHEDULING AND DISPATCH

This case study in the manufacturing domain of semiconductor wafer fabrication highlights the need to incorporate instances of different DT classes to provide benefits above and beyond what is provided by the individual DTs [49]. Each of the DTs in this case study complies with aspects of the DT definition, and their collaboration can be facilitated by the DT O-O framework presented herein. An overview of some of the major tasks in the lifecycle process for the DTs in this case study is provided in the Appendix.

In the semiconductor (also referred to as “microelectronics”) manufacturing environments, semiconductor “wafers” are processed in a fabrication facility or “fab” using complex combinations of process steps such as deposition, lithography patterning and etch to create precise repeated patterns called “die” on each wafer [99]. These “front-end” processes are often required to produce angstrom-level precision patterning with yields > 99% while operating at near 100% capacity. Maintaining these stringent operating conditions requires augmenting precise equipment and process design and configuration with adaptive cyber or software mechanisms to optimize operation. At the lower levels, a communication protocol called SEMI Equipment Communication Standard (SECS) is used to provide standardized data collection and actuation across production and metrology equipment in order to understand equipment, process and product context, state and quality [71], [73], [100]. A form of MBPC called “run-to-run (R2R) control”, which is implemented widely in fabs, uses this communication environment to tune and maximize process capability or “ $C_{pk}$ ” at each process [101].  $C_{pk}$  is a common metric used in manufacturing to convey process capability. It is a quantitative indication

of the closeness to process target as well as variability of the process output [102]. At higher levels a suite of fully automated systems work together using a combination of SEMI standardized and proprietary protocols to maximize the production capability of the fab. ERP systems coordinate and prioritize production orders, MESs determine production flows, bill-of-materials, etc. to execute orders, while real-time scheduling/dispatching (S/D) systems consolidate data from fab MES and ERP systems to understand production needs (e.g., orders, priorities, deadlines) and capabilities. This understanding is then used to dynamically route groups of wafers called “lots” in order to optimize production according to a cost/profit function in the face of ever-changing production capability information such as queue lengths, equipment cycle times, and unexpected downtime.

It has been argued that if more information from the wafer fabs was used in S/D systems, better decisions could be made in determining production schedules [103]–[105]. For instance, the incorporation of process capability in S/D would help align high-value, high-precision products with high-capacity tools to minimize scrap and improve profit.

In this case study, an approach to S/D is deployed that uses information from multiple sources, including process capability estimates from process control DTs, to enable a S/D DT to propose optimal production schedules, updated dynamically in a time-critical fashion, for a lithography bay of twelve processes from which to select (lithography is often a bottleneck process in semiconductor manufacturing [105]). The solution is illustrated in Figure 6. It employs a rule-based S/D DT to make appropriate S/D plans (production schedules) and updates these plans in response to events in the fab. In this case study, a product called RTD®<sup>1</sup> is used to provide this rule-based configurable “real-time” S/D decision-making capability for a single process type [104]. A model of the fab layout defines process flow and opportunities for S/D decisions. The S/D DT then evaluates various production capabilities using a weighted normalized cost function to make decisions to determine which equipment instance, among a bank of possibilities, to select for a particular wafer process.

The cost  $c$  of equipment  $j$  is defined as:

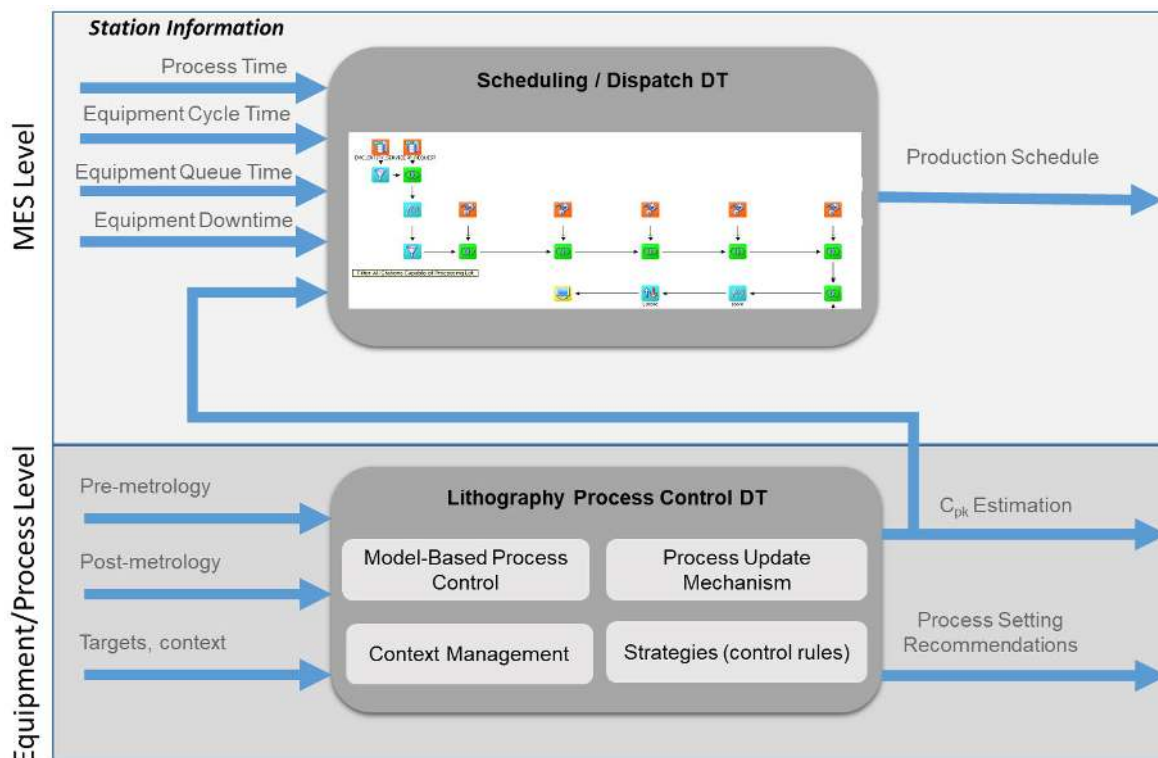
$$c_j(t) = \sum_i \frac{w_i}{\sum_i w_i} f_{ij}(t) \quad (1)$$

where  $f_{ij}$  are the factors taken into consideration, gathered from applications at the MES and ERP levels, and  $w_i$  are their respective weights.

The S/D optimization process consists of the following steps which are executed by the S/D system after every service request by wafer lots:

- 1) Determine set of candidate tools, defined as operational tools that can perform the required task.
- 2) Remove candidate tools whose queues exceed a user-defined limit (e.g., do not consider tools with more than ten lots in queue).

<sup>1</sup>RTD® is a registered trademark of Applied Materials Corporation.



**FIGURE 6.** Solution environment illustrating the information flow and the integration of information from a lithography process control DT (bottom) into a S/D DT (top). The graphic in the S/D DT is an RTD® screenshot of the S/D DT. It incorporates the outputs from the individual DTs (see Figure 7) and executes the S/D optimization process.

- 3) Compute a score for each candidate tool following equation (1).
- 4) Choose tools with the lowest cost for processing by solving an integer programming problem.

As shown in Figure 7, the factor values  $f_{ij}$  in (1) can be determined by individual DTs and then aggregated by the S/D DT solution to determine a production schedule that provides an optimal throughput strategy. This figure only shows the aggregation portion of the O-O hierarchy. The O-O structure for these DT classes is based on the baseline framework. Extensions such as specific class definitions and divisions between DT capabilities would depend on how the solutions are realized in the DT lifecycle. For example, the tool process control DT is derived from a baseline process control DT class, and the bay quality DT, which is derived from a baseline quality DT class, incorporates an aggregate of process control DT class outputs. In an alternate extension of the baseline framework, the bay S/D DT class might incorporate these outputs directly, circumventing the need for a baseline quality DT class.

The solution does not usually incorporate process quality in the S/D DT because that information is not readily available at the MES and ERP levels. However at lower levels in the fab, control infrastructure equipment process control DTs use MBPC models and methods and provide two outputs to support the DT purpose: (1) process setting recommendations (called “recipes”) prior to each wafer production “run”,

and (2) quantitative estimates of process capability reported the  $C_{pk}$  metric. Note that DT output quality metrics are not provided in this solution. In this case study, the S/D DT is augmented to incorporate process capability information from the process control DTs by including these process control DTs in the aggregation (see dashed lines in Figure 7) and adding the  $C_{pk}$  information to the S/D DT cost function model (1) and Figure 6. The results are shown in Figure 8. In this experiment, the output of the S/D DT is evaluated in a simulated fab operation environment. In this environment it is assumed that there are (1) three products of different production value being produced, (2) factors of cycle time, queuing time and  $C_{pk}$  impacting profit from the perspective of S/D, and (3)  $C_{pk}$  estimates are accurate. Standard six sigma conversion tables are used to determine wafer die scrap. Scrap for product 2 is shown in the figure due to its high value. Scrap calculations assume 600 die per wafer [49]. The results illustrate the impact of different weights of the  $C_{pk}$  factor on the overall profit of the fab. A typical S/D DT implementation that does not incorporate the process control DT information would produce a production schedule recommendation equivalent to a  $C_{pk}$  weight of zero. As the  $C_{pk}$  factor is increased, the fab profit increases and then decreases illustrating the tradeoff of throughput and quality. From this analysis an S/D DT production schedule output can be calculated that is optimized to the fab profit.

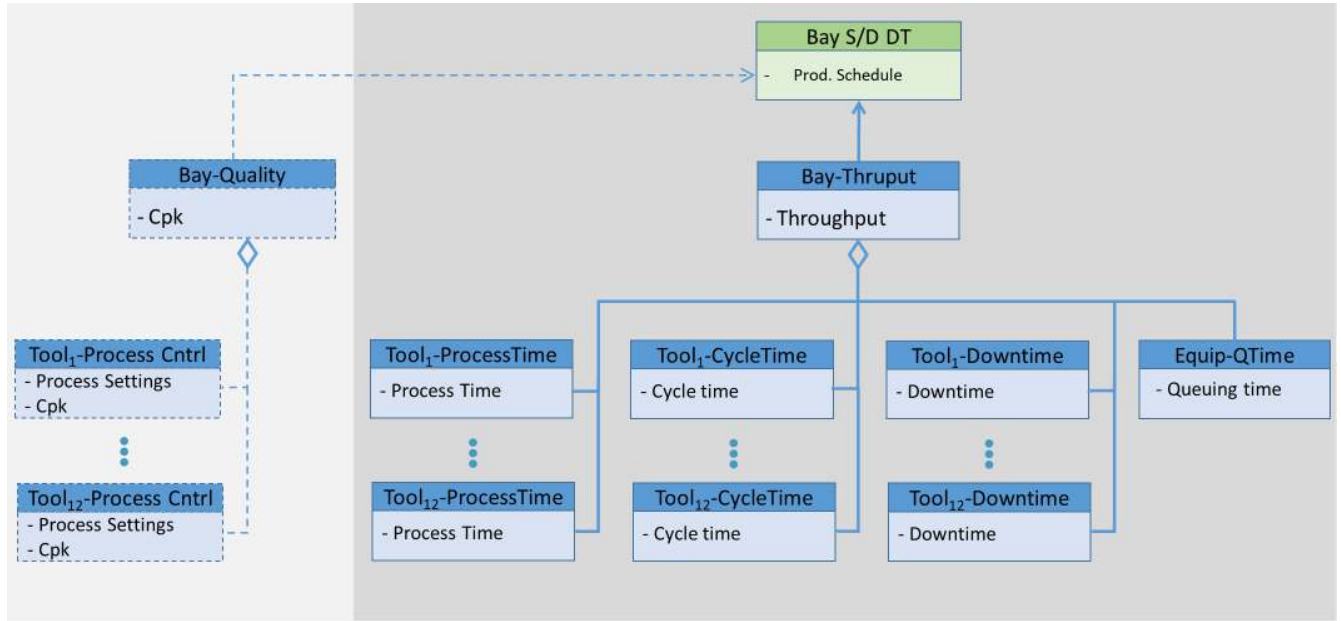


FIGURE 7. Illustration of one conceptualization of a DT framework for the process-capability aware real-time scheduling and dispatch solution.

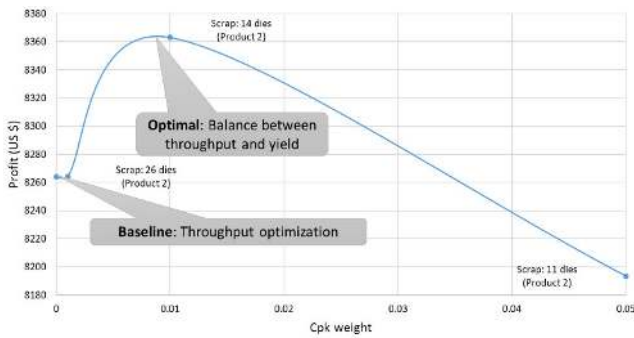


FIGURE 8. Illustration of the effect of Cpk weight on profit and scrap.

This solution illustrates the benefit of the DT framework from the perspectives of consistent DT definition and structure (e.g., for incorporating information from a bank of similar process control DTs into a single S/D DT), collaboration between DTs, and quantification of DT output. The framework also points to areas for improvement in this case study solution including the need for quantification of the quality of the  $C_{pk}$  estimates provided by the process control DTs as well as the decisions made by the S/D DT.

**C. EXTENSION: METHOD FOR OPTIMIZING DT BENEFIT IN A DEFINED APPLICATION ENVIRONMENT**

A DT is required to deliver a net value add in its application environment, and the benefit must be ascertainable and quantifiable (see Table 2, requirements 1h and 1i). This net value add is often expressed as a financial benefit. A good example is the case study just presented where optimal benefit is expressed as a balance between throughput and yield (Figure 8). In most DT environments, determining and optimizing this financial benefit also requires understanding

the quality of the DT detection, prediction, or prescription output, and the corresponding application environment cost of acting on correct and incorrect detection, prediction or prescription information provided by the DT. A practical DT framework thus might use an extension that provides for optimization of the DT benefit in the application environment.

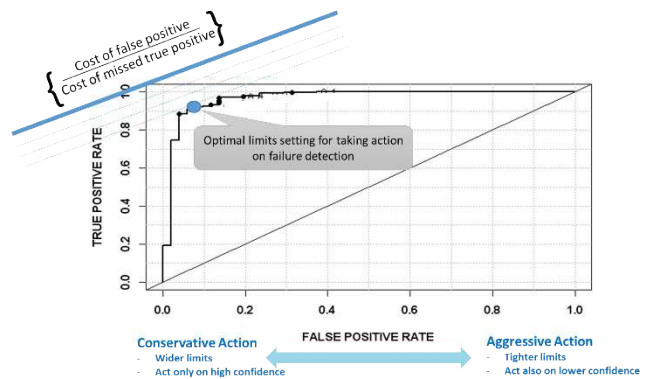


FIGURE 9. Use of a ROC curve to determine the operating point and net value add of a DT detection, prediction or prescription capability.

An example of such an extension is shown in Figure 9. Here the ROC for a DT detection, prediction or prevention capability is used to determine an operating point for acting on the DT output (e.g., determining a parameter value “limit” at which to act on the prediction of a failure event). The ratio of the cost of a false positive (alarm) and missed positive *in the application environment* is also indicated (as a straight line). For example, the cost of a false positive might be the total cost of a scheduled (but unneeded) downtime while the cost of a missed positive might be the cost of an unscheduled downtime minus the cost of a scheduled downtime. The point where the ROC has the same slope as the cost ratio is the ideal



TABLE 3. DT lifecycle steps, common transitions and example application.

Step	Step description	Output → Next State	Example: Case Study of Section VI.B
<b>OFF-LINE</b>			
<b>Envision</b>	SME-centric step: Determine high-value-problem, requirements definition, objective, metrics of success (name and values), domain/context, application environment and interactions, resource validation (existing solutions, off and on-line data, SMEs, interactions), roadmap, etc.	Successful envisioning → <b>Design</b>	Problem: Lost profit due to non-optimal S/D decisions. Objective: Improve S/D through incorporation of product quality into S/D decisions at lithography. Metric of success: profit of fab. Resources available: extensible S/D and MBPC DTs, integration environment, SMEs, equip. availability for test, funding, etc.
<b>Design</b>	Determine solution architecture, data collection environment, requirements refinement, modeling approach, solution maintenance methodology. Perform what-if analysis, off-line data analysis for demonstrating design, etc.	DT design → <b>Develop</b> Design questions and concerns → <b>Envision</b>	See Section VI.B for design details.
<b>Develop</b>	DT model development (perform variable down-selection, model training and general evaluation), DT functionally development, software development (integration and maintenance solutions), etc.	DT solution → <b>Verify</b> Solution development issues → <b>Design</b>	RTD® rules update to extend equation (1). Aggregation of MBPC DT C <sub>pk</sub> outputs to realize Bay-Quality DT. General evaluation in simulation environment, e.g., See Figure 8
<b>Verify</b>	Verify that solution addresses problem and meets needs identified by objective, off-line testing, standards compliance.	DT solution verified → <b>Validate</b> Verification issues → <b>Develop</b> or <b>Design</b> (depending on type and severity of issues)	Verify SEMI standard data collection and interface requirements. Test on off-line data and quantify results against objectives, etc.
<b>Validate</b>	Using off-line data and analysis, validate that particular solution meets needs of user in defined application environment, performance, quality assurance.	DT solution validated → <b>Deploy</b> Validation issues → <b>Develop</b> or <b>Design</b> (depending on type and severity of issues)	Validate capability qualitatively and quantitatively in a user environment. Provide specific evaluation metric values, performance validation, quality assurance, etc.
<b>ON-LINE</b>			
<b>Deploy</b>	Integration and integration strategies / business and control rules, test, evaluate, etc.	Deployment test successful → <b>Use</b> Deployment test failure → <b>Envision, Design, Develop, Verify</b> or <b>Validate</b> (depending on type and severity of failures or issues)	Update existing RTD® rules. Update Process Control DTs and install Bay-Quality DT. Update integration to existing non-DT systems. Evaluate in “advise-only” mode. Deploy in full operational mode and evaluate.
<b>Use</b>	DT is providing intended benefit of sufficient quantity.	Evaluation based on timing, event occurrence, etc., as defined by solution → <b>Evaluate</b>	Solution is providing improved profit in fab by incorporating process quality information in lithography bay S/D decision-making.
<b>Evaluate</b>	Periodically assess whether DT is continuing to provide intended benefit of sufficient quantity.	YES, DT continues to provide intended benefit of sufficient quantity per objective and metrics → <b>Use</b> NO, DT is not providing intended benefit of sufficient quantity per objective and metrics → <b>Maintain</b>	Is S/D providing a profit level above threshold defined in solution design? Are Process Controller DTs providing process C <sub>pk</sub> values within limits identified in solution design (this could be considered as an evaluation criterion for the aggregate DT or relegated to the individual process control DT maintenance designs).
<b>Maintain</b>	If DT is not providing intended benefit of sufficient quantity per objective and metrics, assess the problem and solution opportunities.	Maintenance can be achieved with on-line adjustments to solution → <b>Tune</b> Maintenance requires off-line adjustments to solution → <b>Rebuild</b>	Assess capability of tuning vs. rebuilding of all DTs in solution as defined in the solution design.
<b>Tune</b>	Address DT maintenance through on-line update of DT such as update or “tuning” of one or more DT models.	Tuning successful → <b>Use</b> Tuning not successful → <b>Maintain</b>	In S/D DT, weights of factors in equation (1) are tuned. Process Controller DT models are tuned (as defined in the design)
<b>Rebuild</b>	Assess opportunities for off-line maintenance of DT. Determine if current design is adequate.	Assessment is that current design can be used → <b>Develop</b> (i.e., address maintenance through rebuild of aspect of DT such as retraining one or more models). Assessment that current design can no longer be used → <b>Design</b> (e.g., update of solution maintenance methodology).	In S/D DT, weights of factors in equation (1) are re-calculated based on more recent off-line data and any changes in context requirements; S/D rules may be changed. Process control DT models are rebuilt (as defined in the design) based on more recent data and any changes in context requirements.

operating point for acting on the DT event indication, with difference between cost and benefit at that operating point

being the net value add that the DT should be expected to provide in this application environment [9], [53].

## VII. SUMMARY

Digital twin technology, which is an integral part of the evolution of smart manufacturing evolution, has actually been in use for decades in manufacturing. Solutions such as MBPC, PdM and virtual sensing/metrology all fit the general DT definition as they utilize virtual representations of some aspect of a system to provide quantifiable benefit. As we move forward in DT technology evolution we must embrace these existing knowledge, technology and solution bases while accommodating near-term DT requirements such as re-usability, interoperability, interchangeability, V&V, maintainability, and extensibility, along with longer-term and visionary requirements such as automatic creation, V&V, integration, and proliferation across the entire ecosystem. Further we must provide an environment for a sustainable DT technology community that supports common definitions, taxonomy and other mechanisms for collaboration.

While there have been many important efforts focused on providing DT definitions and frameworks, this paper uses a requirements-based approach to derive baseline components of a framework upon which practical frameworks and solutions can be built. This baseline framework includes a DT definition and O-O specification that addresses important aspects of requirements of DT solutions in operation today, in the near future, and over the longer-term including the DT vision. Key attributes of this framework include (1) a DT definition as a purpose-driven dynamic digital replica of a physical asset, process, system or product; (2) a DT structure that includes one or more modeling resources coordinated via a computational engine; (3) a DT output that includes metric(s) that quantify the DT output as it relates to the DT purpose, and metric(s) that quantify the quality or believability of the DT output; and (4) a DT O-O structure that supports both generalization and aggregation subclassing structures to facilitate re-usability, interoperability, interchangeability, maintainability, and extensibility. Extending this baseline framework to practical solutions requires addressing gaps identified in fully addressing the derived requirements. Key among these include (1) providing specifications for interaction between components within DTs, between DTs, and between DTs and non-DTs, across the entire DT lifecycle; (2) providing better mechanisms for facilitating analytics-to-SME interaction; and (3) providing mechanisms for automation of all aspects of the DT lifecycle including DT creation and V&V. Addressing these gaps will require making (perhaps capability limiting) choices instituted via standards, best practices and other specifications.

Moving forward will require a focus on consolidation of DT research, development and implementation. Bringing these great bodies of DT research and development together will require advancements in technology, but also development of standards and other specifications that will serve to align DT efforts without compromising capability or creativity. Specific deficiencies in DT technology today that will have to be addressed to realize longer-term

solutions include providing more robust and automated solutions for DT maintenance, automating DT creation, V&V, and analytics-to-SME incorporation processes, providing solutions for data sharing that preserve content to support DT operation while protecting intellectual property, and realizing DT solutions more collaboratively vertically and horizontally across the entire manufacturing ecosystem.

## APPENDIX

### DT LIFECYCLE STEP AND TRANSITION DESCRIPTION

A high-level view of the DT lifecycle is provided in Figure 2. Additional detail on the lifecycle steps and transitions is provided in Table 3, columns one through three. Note that step descriptions can vary widely depending on the DT purpose and application environment. Column four provides additional detail on the case study of Section VI.B illustrating some of the major tasks that might be executed at each step in the lifecycle.

## ACKNOWLEDGMENT

The authors would like to thank the Integrated Manufacturing Process Automation and Control Technologies (IMPACT) and Barton-Tilbury Groups at the University of Michigan.

## REFERENCES

- [1] Factory Integration. (2020). *International Roadmap for Devices and Systems: 2020*. [Online]. Available: <https://irds.ieee.org>
- [2] J. Moyne, J. Samantaray, and M. Armacost, "Big data capabilities applied to semiconductor manufacturing advanced process control," *IEEE Trans. Semicond. Manuf.*, vol. 29, no. 4, pp. 283–291, Nov. 2016.
- [3] J. Davis, T. Edgar, J. Porter, J. Bernaden, and M. Sarli, "Smart manufacturing, manufacturing intelligence and demand-dynamic performance," *Comput. Chem. Eng.*, vol. 47, pp. 145–156, Dec. 2012.
- [4] J. Moyne, E. Balta, I. Kovalenko, Y. Qamsane, and K. Barton, "The digital twin in the manufacturing ecosystem of the future," in *Proc. 1st Int. Workshop Smart Manuf. Modeling Anal.*, Apr. 2019. [Online]. Available: <http://sdc-mfg.engin.umich.edu/roleofdtgreenfield/>
- [5] E. C. Balta, Y. Lin, K. Barton, D. M. Tilbury, and Z. M. Mao, "Production as a service: A digital manufacturing framework for optimizing utilization," *IEEE Trans. Autom. Sci. Eng.*, vol. 15, no. 4, pp. 1483–1493, Oct. 2018.
- [6] J. Moyne, "Bringing subject matter expertise and analytics together in solutions for microelectronics smart manufacturing: Consensus, domains, and approaches," in *Proc. APC Conf. XXXI*, San Antonio, TX, USA, Oct. 2019. [Online]. Available: <http://apconference.com>
- [7] B. Marr. The 5 Biggest Technology Trends Disrupting Engineering and Design in 2020. *Forbes Magazine*, Nov. 2019. [Online]. Available: <https://www.forbes.com/sites/bernardmarr/2019/11/22/the-5-biggest-technology-trends-disrupting-engineering-and-design-in-2020/#5d754a3a676a>
- [8] General Electric, "The digital twin: Compressing time to value for digital industrial companies," Gen. Electr., Boston, MA, USA, White Paper, 2018. [Online]. Available: [https://www.ge.com/digital/sites/default/files/download\\_assets/The-Digital-Twin\\_Compressing-Time-to-Value-for-Digital-Industrial-Companies.pdf](https://www.ge.com/digital/sites/default/files/download_assets/The-Digital-Twin_Compressing-Time-to-Value-for-Digital-Industrial-Companies.pdf)
- [9] J. Moyne and J. Iskandar, "Big data analytics for smart manufacturing: Case studies in semiconductor manufacturing," *Processes*, vol. 5, no. 4, p. 39, Jul. 2017. [Online]. Available: <http://www.mdpi.com/2227-9717/5/3/39/htm>
- [10] R. Good, D. Balasubramanian, and I. Krumanocker, "Non threaded run-to-run control: Four key advantages for high-mix manufacturing," in *Proc. APC Conf. XXXI*, San Antonio, TX, USA, Oct. 2019. [Online]. Available: <http://apconference.com>
- [11] J. Iskandar, J. Moyne, K. Subrahmanyam, P. Hawkins, and M. Armacost, "Predictive maintenance in semiconductor manufacturing: Moving to fab-wide solutions," in *Proc. 26th Annu. Adv. Semiconductor Manuf. Conf. (ASMC)*, Saratoga Springs, NY, USA, May 2015, pp. 384–389.

- [12] 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)*, 2016, pp. 208–214.
- [13] X. Ye and S. H. Hong, "Toward industry 4.0 components: Insights into and implementation of asset administration shells," *IEEE Ind. Electron. Mag.*, vol. 13, no. 1, pp. 13–25, Mar. 2019.
- [14] Plattform Industrie 4.0, "Details of the asset administration shell. Part 1—The exchange of information between partners in the value chain of Industrie 4.0 (Version 1.0)," Federal Ministry Econ. Affairs Energy, Berlin, Germany, Working Paper, Nov. 2018. [Online]. Available: <https://www.plattform-i40.de/I40/Redaktion/EN/Downloads/Publikation/2018-details-of-the-asset-administration-shell.pdf?blob=publicationFile&v=9>
- [15] E. Y. Song, M. Burns, A. Pandey, and T. Roth, "IEEE 1451 smart sensor digital twin federation for IoT/CPS research," in *Proc. IEEE Sensors Appl. Symp. (SAS)*, Sophia Antipolis, France, Mar. 2019, pp. 1–6.
- [16] J. Moyne and R. Stafford, "User interface with visualization of real and virtual data," U.S. Patent 8 612 864, Dec. 17, 2013.
- [17] D. Gelernter, *Mirror Worlds: Or: The Day Software Puts the Universe in a Shoebox...How It Will Happen and What It Will Mean?* New York, NY, USA: Oxford Univ. Press, 1991.
- [18] M. Grieves and J. Vickers, "Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems," in *Transdisciplinary Perspectives on Complex Systems*. Cham, Switzerland: Springer, 2017, pp. 85–113.
- [19] Y. Qamsane, "Digital twin in manufacturing: A review of definitions and applications," Univ. Michigan, Ann Arbor, MI, USA, Tech. Rep., May 2020. [Online]. Available: <https://sdc-mfg.engin.umich.edu/digitaltwinreview/>
- [20] E. Glaessgen and D. Stargel, "The digital twin paradigm for future NASA and U.S. Air force vehicles," in *Proc. 53rd Struct. Dyn. Mater. Conf. Special Session Digit. Twin*, Honolulu, HI, USA, Apr. 2012, pp. 1–14.
- [21] P. S. Van Schalkwyk and S. W. M. Lin, "A short introduction to digital twins," *Ind. Internet Consortium J. Innov.*, 2019. [Online]. Available: <https://www.iiconsortium.org/news/joi-articles/2019-November-Joi-A-Short-Introduction-to-Digital-Twins.pdf>
- [22] R. Stark and T. Damerau, "Digital twin," in *CIRP Encyclopedia of Production Engineering*. Berlin, Germany: Springer, 2019, pp. 1–8.
- [23] 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.
- [24] K. Ding, F. T. S. Chan, X. Zhang, G. Zhou, and F. Zhang, "Defining a digital twin-based cyber-physical production system for autonomous manufacturing in smart shop floors," *Int. J. Prod. Res.*, vol. 57, no. 20, pp. 6315–6334, Oct. 2019.
- [25] K. Y. H. Lim, P. Zheng, and C.-H. Chen, "A state-of-the-art survey of digital twin: Techniques, engineering product lifecycle management and business innovation perspectives," *J. Intell. Manuf.*, pp. 1–25, Nov. 2019.
- [26] A. A. Khan, J. R. Moyne, and D. M. Tilbury, "An approach for factory-wide control utilizing virtual metrology," *IEEE Trans. Semicond. Manuf.*, vol. 20, no. 4, pp. 364–375, Nov. 2007.
- [27] J. Feng, X. Jia, F. Zhu, J. Moyne, J. Iskandar, and J. Lee, "An online virtual metrology model with sample selection for the tracking of dynamic manufacturing processes with slow drift," *IEEE Trans. Semicond. Manuf.*, vol. 32, no. 4, pp. 574–582, Nov. 2019.
- [28] Z. Liu, N. Meyendorf, and N. Mrad, "The role of data fusion in predictive maintenance using digital twin," in *Proc. AIP Conf.*, 2018, vol. 1949, no. 1, Art. no. 020023.
- [29] P. Aivaliotis, K. Georgoulas, and G. Chryssolouris, "The use of digital twin for predictive maintenance in manufacturing," *Int. J. Comput. Integr. Manuf.*, vol. 32, no. 11, pp. 1067–1080, Nov. 2019.
- [30] S. Boschert and R. Rosen, "Digital twin—The simulation aspect," in *Mechatronic Futures*. Cham, Switzerland: Springer, 2016, pp. 59–74.
- [31] 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.
- [32] 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.
- [33] J. Leng, H. Zhang, D. Yan, Q. Liu, X. Chen, and D. Zhang, "Digital twin-driven manufacturing cyber-physical system for parallel controlling of smart workshop," *J. Ambient Intell. Humanized Comput.*, vol. 10, no. 3, pp. 1155–1166, Jun. 2018.
- [34] 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.
- [35] J. Liu, H. Zhou, X. Liu, G. Tian, M. Wu, L. Cao, and W. Wang, "Dynamic evaluation method of machining process planning based on digital twin," *IEEE Access*, vol. 7, pp. 19312–19323, 2019.
- [36] G. Warwick, "GE advances analytical maintenance with digital twins," *Aviation Week Space Technol.*, pp. 10–19, Oct. 2015.
- [37] A. M. Lund, K. Mochel, J.-W. Lin, R. Onetto, J. Srinivasan, P. Gregg, J. E. Bergman, K. D. Hartling, Jr., A. Ahmed, and S. Chotai, "Digital wind farm system," U.S. Patent 15 075 231, Nov. 17, 2016.
- [38] R. H. Guerra, R. Quiza, A. Villalonga, J. Arenas, and F. Castaño, "Digital twin-based optimization for ultraprecision motion systems with backlash and friction," *IEEE Access*, vol. 7, pp. 93462–93472, 2019.
- [39] F. Tao, F. Sui, A. Liu, Q. Qi, M. Zhang, B. Song, Z. Guo, S. C.-Y. Lu, and A. Nee, "Digital twin-driven product design framework," *Int. J. Prod. Res.*, vol. 57, no. 12, pp. 3935–3953, 2019.
- [40] 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, Feb. 2018.
- [41] R. Söderberg, K. Wärnefjord, J. S. Carlson, and L. Lindkvist, "Toward a digital twin for real-time geometry assurance in individualized production," *CIRP Ann.*, vol. 66, no. 1, pp. 137–140, 2017.
- [42] J. Moyne, M. Yedatore, J. Iskandar, P. Hawkins, and J. Scoville, "Chamber matching across multiple dimensions utilizing predictive maintenance, equipment health monitoring, virtual metrology and run-to-run control," in *Proc. 25th Annu. SEMI Adv. Semiconductor Manuf. Conf. (ASMC)*, Saratoga Springs, NY, May 2014, pp. 86–91.
- [43] J. Moyne and J. Iskandar, "Matching process controllers for improved matching of process," U.S. Patent 9 606 519 B2, Mar. 6, 2017.
- [44] J. Kahn, "The quest for human-level A.I.," *Fortune Magazine*, Feb. 2020.
- [45] B. Goertzel and C. Pennachin, Eds., "Artificial general intelligence," in *Cognitive Technologies*. Berlin, Germany: Springer, 2007.
- [46] J. Iskandar and J. Moyne, "Maintenance of virtual metrology models," in *Proc. 27th Annu. SEMI Adv. Semiconductor Manuf. Conf. (ASMC)*, Saratoga Springs, NY, USA, May 2016, pp. 393–398.
- [47] J. Moyne, "Methods and apparatuses for utilizing adaptive predictive algorithms and determining when to use the adaptive predictive algorithms for virtual metrology," U.S. Patent 9 002 492 B2, Apr. 7, 2015.
- [48] G. Shao, S. Jain, C. Laroque, L. H. Lee, P. Lendermann, and O. Rose, "Digital twin for smart manufacturing: The simulation aspect," in *Proc. Winter Simulation Conf. (WSC)*, National Harbor, MD, USA, Dec. 2019, pp. 2085–2098.
- [49] F. Lopez, J. Moyne, K. Barton, and D. Tilbury, "Process-capability-aware scheduling/dispatching in wafer fabs," in *Proc. 26th Adv. Process Control Conf.*, Oct. 2017. [Online]. Available: <http://apconference.com>
- [50] J. Moyne, M. Armacost, B. Williams, and S. Bushman, "Leveraging smart manufacturing solutions at multiple operational levels in microelectronics smart manufacturing: A digital twin case study," in *Proc. Adv. Process Control Conf. XXX*, Oct. 2018. [Online]. Available: <http://apconference.com>
- [51] F.-T. Cheng, C.-A. Kao, C.-F. Chen, and W.-H. Tsai, "Tutorial on applying the VM technology for TFT-LCD manufacturing," *IEEE Trans. Semicond. Manuf.*, vol. 28, no. 1, pp. 55–69, Feb. 2015.
- [52] B. Smarslok, A. Culler, and S. Mahadevan, "Error quantification and confidence assessment of aerothermal model predictions for hypersonic aircraft," in *Proc. 53rd AIAA/ASME/ASCE/AHS/ASC Struct. Struct. Dyn. Mater. Conf.*, Apr. 2012, p. 1817.
- [53] J. Moyne, "Method and apparatus for optimizing profit in predictive systems," U.S. Patent 9 582 828 B2, Feb. 28, 2017.
- [54] D. Kibira, K. Morris, and S. Kumaraguru, "Methods and tools for performance assurance of smart manufacturing systems," *J. Res. Nat. Inst. Standards Technol.*, vol. 121, p. 287, Jun. 2016.
- [55] Cyber-Physical Systems—A Concept Map. Dept. Elect. Eng. Comput. Sci., Ptolemy Project, UC Berkeley, Berkeley, CA, USA, Jun. 2018. [Online]. Available: <https://ptolemy.berkeley.edu/projects/cps/>
- [56] Z. Guo, D. Zhou, Q. Zhou, S. Mei, S. Zeng, D. Yu, and J. Chen, "A hybrid method for evaluation of maintainability towards a design process using virtual reality," *Comput. Ind. Eng.*, vol. 140, Feb. 2020, Art. no. 106227.

- [57] M. Weyrich, M. Klein, J.-P. Schmidt, N. Jazdi, K. D. Bettenhausen, F. Buschmann, C. Rubner, M. Pirker, and K. Wurm, "Evaluation model for assessment of cyber-physical production systems," in *Industrial Internet of Things* (Springer Series in Wireless Technology), S. Jeschke, C. Brecher, H. Song, D. Rawat, Eds. Cham, Switzerland: Springer, 2017.
- [58] Q. Qi and F. Tao, "Digital twin and big data towards smart manufacturing and industry 4.0: 360 degree comparison," *IEEE Access*, vol. 6, pp. 3585–3593, 2018.
- [59] 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. 48, no. 3, pp. 567–572, 2015.
- [60] H. Zhang, Q. Liu, X. Chen, D. Zhang, and J. Leng, "A digital twin-based approach for designing and multi-objective optimization of hollow glass production line," *IEEE Access*, vol. 5, pp. 26901–26911, 2017.
- [61] S. Weyer, T. Meyer, M. Ohmer, D. Gorecky, and D. Zühlke, "Future modeling and simulation of CPS-based factories: An example from the automotive industry," *IFAC-PapersOnLine*, vol. 49, no. 31, pp. 97–102, 2016.
- [62] M. M. Najafabadi, F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R. Wald, and E. Muharemagic, "Deep learning applications and challenges in big data analytics," *J. Big Data*, vol. 2, no. 1, pp. 1–21, Feb. 2015.
- [63] J. J. Vogel-Walcutt, J. B. Gebirim, C. Bowers, T. M. Carper, and D. Nicholson, "Cognitive load theory vs. Constructivist approaches: Which best leads to efficient, deep learning?" *J. Comput. Assist. Learn.*, vol. 27, no. 2, pp. 133–145, Apr. 2011.
- [64] Y. Lu, C. Liu, K. I.-K. Wang, H. Huang, and X. Xu, "Digital twin-driven smart manufacturing: Connotation, reference model, applications and research issues," *Robot. Comput.-Integr. Manuf.*, vol. 61, Feb. 2020, Art. no. 101837.
- [65] Y. Qamsane, J. Moyne, D. Tilbury, and K. Barton, "A comprehensive digital twin framework for semiconductor manufacturing: Case study—Optimized scheduling/dispatch," in *Proc. Adv. Process Control Conf. XXXI*, San Antonio, TX, USA, Oct. 2019. [Online]. Available: <http://apconconference.com>
- [66] E. Balta, I. Kovalenko, J. Moyne, D. Tilbury, and K. Barton, "Unit manufacturing process models for improved performance monitoring and control of resource fleets in manufacturing systems," in *Proc. APC Conf. XXXI*, San Antonio, TX, USA, Oct. 2019. [Online]. Available: <http://apconconference.com>
- [67] Y. Qamsane, E. C. Balta, I. Kovalenko, J. Moyne, J. Faris, D. Tilbury, and K. Barton, *A Methodology to Develop and Implement a Practical DT Framework for Manufacturing Systems*. [Online]. Available: <http://sdcmfg.engin.umich.edu/practicaldtframework/>
- [68] SEMI E54: *Specification for Sensor/Actuator Network*, Standard SEMI E54.1:2016, 2016. [Online]. Available: <http://www.semi.org/en/Standards/>
- [69] Y. Koren, U. Heisel, F. Jovane, T. Moriwaki, G. Pritschow, G. Ulsoy, and H. Van Brussel, "Reconfigurable manufacturing systems," *Ann. CIRP*, vol. 48, no. 2, pp. 527–540, 1999.
- [70] F. Harirchi, T. Vincent, A. Subramanian, K. Poolla, and B. Stirton, "Implementation of nonthreaded estimation for run-to-run control of high mix semiconductor manufacturing," *IEEE Trans. Semicond. Manuf.*, vol. 26, no. 4, pp. 516–528, Nov. 2013.
- [71] J. Wang, Q. P. He, and T. F. Edgar, "State estimation for integrated moving average processes in high-mix semiconductor manufacturing," *Ind. Eng. Chem. Res.*, vol. 53, no. 13, pp. 5194–5204, Apr. 2014.
- [72] F. Lopez, M. Saez, Y. Shao, E. C. Balta, J. Moyne, Z. M. Mao, K. Barton, and D. Tilbury, "Categorization of anomalies in smart manufacturing systems to support the selection of detection mechanisms," *IEEE Robot. Autom. Lett.*, vol. 2, no. 4, pp. 1885–1892, Oct. 2017.
- [73] J. R. Moyne, H. Hajj, K. Beatty, and R. Lewandowski, "SEMI E133—The process control system standard: Deriving a software interoperability standard for advanced process control in semiconductor manufacturing," *IEEE Trans. Semicond. Manuf.*, vol. 20, no. 4, pp. 408–420, Nov. 2007.
- [74] D. Kurz, C. De Luca, and J. Pilz, "A sampling decision system for virtual metrology in semiconductor manufacturing," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 1, pp. 75–83, Jan. 2015.
- [75] J. Tikkanen, N. Sumikawa, L. Wang, L. Winemberg, and M. S. Abadir, "Statistical outlier screening for latent defects," in *Proc. IEEE Int. Rel. Phys. Symp. (IRPS)*, Anaheim, CA, USA, Apr. 2013, pp. 2E.1.1–2E.1.4.
- [76] J. Moyne, S. Mashiro, and D. Gross, "Determining a security roadmap for the microelectronics industry," in *Proc. 29th Annu. SEMI Adv. Semiconductor Manuf. Conf. (ASMC)*, Saratoga Springs, NY, USA, Apr. 2018, pp. 291–294.
- [77] X. Xu, F. Rahman, B. Shakya, A. Vassilev, D. Forte, and M. Tehranipoor, "Electronics supply chain integrity enabled by blockchain," *ACM Trans. Design Autom. Electron. Syst.*, vol. 24, no. 3, May 2019, Art. no. 31.
- [78] J. Francis, "Enterprise blockchain for supply chain," in *IEEE Courses*. IEEE, May 2019.
- [79] Plattform Industrie 4.0, "Blockchain and the law in the context of Industrie 4.0," Federal Ministry Econ. Affairs Energy, Berlin, Germany, Working Paper, Feb. 2019. [Online]. Available: [https://www.plattform-i40.de/PI40/Redaktion/EN/Downloads/Publikation/blockchain-and-the-law-industrie4.0.pdf?\\_\\_blob=publicationFile&v=5](https://www.plattform-i40.de/PI40/Redaktion/EN/Downloads/Publikation/blockchain-and-the-law-industrie4.0.pdf?__blob=publicationFile&v=5)
- [80] S. Huang, G. Wang, Y. Yan, and X. Fang, "Blockchain-based data management for digital twin of product," *J. Manuf. Syst.*, vol. 54, pp. 361–371, Jan. 2020.
- [81] F. Lopez, Y. Shao, Z. M. Mao, J. Moyne, K. Barton, and D. Tilbury, "A software-defined framework for the integrated management of smart manufacturing systems," *Manuf. Lett.*, vol. 15, pp. 18–21, Jan. 2018.
- [82] K. Harper, S. Malakuti, and C. Ganz, "Digital twin architecture and standards," *IIC J. Innov.*, pp. 1–12, Nov. 2019. [Online]. Available: <https://www.iiconsortium.org/news/joi-articles/2019-November-Joi-Digital-Twin-Architecture-and-Standards.pdf>
- [83] F. Ocker, I. Kovalenko, K. Barton, D. Tilbury, and B. Vogel-Heuser, "A framework for automatic initialization of multi-agent production systems using semantic Web technologies," *IEEE Robot. Autom. Lett.*, vol. 4, no. 4, pp. 4330–4337, Oct. 2019.
- [84] J. A. Hanley and B. J. McNeil, "The meaning and use of the area under a receiver operating characteristic (ROC) curve," *Radiology*, vol. 43, no. 1, pp. 29–36, 1982.
- [85] G. Booch, R. Maksimchuk, M. Engle, B. Young, J. Conallen, and K. Houston, *Object-Oriented Analysis and Design With Applications*. Reading, MA, USA: Addison-Wesley, 2007.
- [86] Object Management Group (OMG). (Dec. 2017). *OMG Unified Modeling Language (OMG UML) Version 2.5.1*. [Online]. Available: <https://www.omg.org/spec/UML/2.5.1/PDF>
- [87] M. Blaha and J. Rumbaugh, *Object-Oriented Modeling and Design With UML*. Upper Saddle River, NJ, USA: Prentice-Hall, 1991.
- [88] CORDIS Project: IoTwins. (2020). *Distributed Digital Twins for Industrial SMEs: A Big-Data Platform*. [Online]. Available: <https://cordis.europa.eu/project/id/857191>
- [89] Y. Zheng, S. Yang, and H. Cheng, "An application framework of digital twin and its case study," *J. Ambient Intell. Humanized Comput.*, vol. 10, no. 3, pp. 1141–1153, Jun. 2018.
- [90] D. Harel, "STATECHARTS: A visual formalism for complex systems," *Sci. Comput. Program.*, vol. 8, no. 3, pp. 231–274, Jun. 1987.
- [91] M. Mani, J. Larborn, B. Johansson, K. W. Lyons, and K. C. Morris, "Standard representations for sustainability characterization of industrial processes," *J. Manuf. Sci. Eng.*, vol. 138, no. 10, Oct. 2016.
- [92] *IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)—Framework and Rules—Redline*, IEEE Standard 1516-2010 (Revision of IEEE Standard 1516-2000) Aug. 2010, pp. 1–38.
- [93] 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.
- [94] Y. Qamsane, E. C. Balta, J. Moyne, D. Tilbury, and K. Barton, "Dynamic rerouting of cyber-physical production systems in response to disruptions based on SDC framework," in *Proc. Amer. Control Conf. (ACC)*, Jul. 2019, pp. 3650–3657.
- [95] Y. Qamsane, C.-Y. Chen, E. C. Balta, B.-C. Kao, S. Mohan, J. Moyne, D. Tilbury, and K. Barton, "A unified digital twin framework for real-time monitoring and evaluation of smart manufacturing systems," in *Proc. IEEE 15th Int. Conf. Autom. Sci. Eng. (CASE)*, Aug. 2019, pp. 1394–1401.
- [96] E. C. Balta, D. M. Tilbury, and K. Barton, "A centralized framework for system-level control and management of additive manufacturing fleets," in *Proc. IEEE 14th Int. Conf. Autom. Sci. Eng. (CASE)*, Aug. 2018, pp. 1071–1078.
- [97] E. C. Balta, D. M. Tilbury, and K. Barton, "A digital twin framework for performance monitoring and anomaly detection in fused deposition modeling," in *Proc. IEEE 15th Int. Conf. Autom. Sci. Eng. (CASE)*, Aug. 2019, pp. 823–829.
- [98] I. Kovalenko, D. Tilbury, and K. Barton, "The model-based product agent: A control oriented architecture for intelligent products in multi-agent manufacturing systems," *Control Eng. Pract.*, vol. 86, pp. 105–117, May 2019.

[99] G. May and C. Spanos, *Fundamentals of Semiconductor Manufacturing and Process Control*. Hoboken, NJ, USA: Wiley, 2006.

[100] (2020). *SEMI E30: Specification for Generic Model for Communications and Control of Manufacturing Equipment (GEM)*. [Online]. Available: <http://www.semi.org/en/Standards/>

[101] J. Moyne, "Run-to-run control in semiconductor manufacturing," in *Encyclopedia of Systems and Control*, J. Baillieula and T. Samad, Eds. London, U.K.: Springer, 2019.

[102] V. E. Kane, "Process capability indices," *J. Qual. Technol.*, vol. 18, no. 1, pp. 41–52, 1986.

[103] C. Yugma, J. Blue, S. Dauzère-Pérès, and A. Obeid, "Integration of scheduling and advanced process control in semiconductor manufacturing: Review and outlook," *J. Scheduling*, vol. 18, no. 2, pp. 195–205, Apr. 2015.

[104] G. K. Agrawal, S. Y. Loh, and A. B. Shebi, "Advanced process control (APC) and real time dispatch (RTD) system integration for etch depth control process in 300 mm fab," in *Proc. 26th Annu. SEMI Adv. Semiconductor Manuf. Conf. (ASMC)*, Saratoga Springs, NY, USA, May 2015, pp. 390–394.

[105] M. Kidambi, S. Krishnaswamy, S. Marteny, J. Moyne, D. Norman, and J. Webb, "Improving lithography throughput and minimizing waste using predictive multi-area scheduling," *Proc. SPIE*, vol. 8324, Apr. 2012, Art. no. 832439.



**JAMES MOYNE** (Member, IEEE) received the B.S.E.E. and B.S.E. in mathematics in 1984, and the M.S.E.E. and Ph.D. degrees from the University of Michigan, Ann Arbor, in 1985 and 1990, respectively.

He is currently an Associate Research Scientist with the Department of Mechanical Engineering, University of Michigan. He is also a Consultant for standards and technology at Applied Materials, focusing on smart manufacturing initiatives.

He has experience in model-based process control, prediction technology (predictive maintenance, virtual metrology, and yield prediction), and smart manufacturing topics, such as big data, digital twin, and analytics. He is the author of a number of refereed publications and holds patents in each of these areas.

Dr. Moyne is the Co-Chair the Factory Integration Thrust of the International Roadmap for Devices and Systems (IRDS), and the Information and Control Committee of SEMI, which is the semiconductor manufacturing standards organization. He's on the advisory board of Pharma (smart manufacturing in pharmaceuticals) and CTO of the Integrated Measurement Association.



**YASSINE QAMSANE** received the M.S. and Ph.D. degrees in electrical engineering, telecommunications, and industrial technologies from Cadi Ayyad University, Marrakech, Morocco, in 2012 and 2018, respectively.

He is currently a Research Fellow with the Department of Mechanical Engineering, University of Michigan, Ann Arbor. His current research interests include digital twins, smart manufacturing, and control systems.



**EFE C. BALTA** (Graduate Student Member, IEEE) received the B.S. degree in manufacturing engineering from the Mechanical Engineering Faculty, Istanbul Technical University, Turkey, in 2016, and the M.S. degree in mechanical engineering from the University of Michigan, Ann Arbor, in 2018, where he is currently pursuing the Ph.D. degree in mechanical engineering.

His research interests include control theory, manufacturing science, additive manufacturing, robotics, optimization, and systems engineering.



**ILYA KOVALENKO** (Graduate Student Member, IEEE) received the B.S. degree in mechanical engineering from the Georgia Institute of Technology, Atlanta, GA, USA, in 2015, and the M.S. degree in mechanical engineering from the University of Michigan, Ann Arbor, MI, USA, in 2018, where he is currently pursuing the Ph.D. degree in mechanical engineering.

His current research interests include control theory and application, with a focus on multi-agent systems, advanced manufacturing, cyber-physical systems, and robotics.



**JOHN FARIS** received the B.S. degree in mechanical engineering from Lawrence Technological University, in 1987, and the M.B.A. degree from Wayne State University, in 1996.

He is currently a Research Area Specialist Senior with the University of Michigan. He has over 30 years of experience at Ford Motor Company, where he developed and launched new vehicles and new facilities in three continents, and developed and launched new organizations and new standards for the Asia Pacific region. His current research interests include digital twins for complete manufacturing systems and virtual commissioning for complete auto body assembly systems.



**KIRA BARTON** (Member, IEEE) received the B.Sc. degree in mechanical engineering from the University of Colorado at Boulder, in 2001, and the M.Sc. and Ph.D. degrees in mechanical engineering from the University of Illinois at Urbana-Champaign, in 2006 and 2010, respectively. She held a postdoctoral research position at the University of Illinois, from Fall 2010 until Fall 2011, at which point she joined the Mechanical Engineering Department at the University of Michigan,

Ann Arbor. She is currently an Associate Professor and a Miller Faculty Scholar with the Department of Mechanical Engineering, University of Michigan. She conducts research in modeling, sensing, and control for applications in advanced manufacturing and robotics, with a specialization in cooperative learning and smart manufacturing. She was a recipient of an NSF CAREER Award, in 2014, the 2015 SME Outstanding Young Manufacturing Engineer Award, the 2015 University of Illinois, Department of Mechanical Science and Engineering Outstanding Young Alumni Award, the 2016 University of Michigan, Department of Mechanical Engineering Department Achievement Award, and the 2017 ASME Dynamic Systems and Control Young Investigator Award.



**DAWN M. TILBURY** (Fellow, IEEE) received the B.S. degree (*summa cum laude*) in electrical engineering from the University of Minnesota, in 1989, and the M.S. and Ph.D. degrees in electrical engineering and computer sciences from the University of California, Berkeley, in 1992 and 1994, respectively.

In 1995, she joined the Mechanical Engineering Department, University of Michigan, Ann Arbor, where she is currently a Professor, with a joint appointment as a Professor of EECS. In June of 2017, she became the Assistant Director for Engineering at the National Science Foundation, while maintaining her position at the University of Michigan. She has published more than 150 articles in refereed journals and conference proceedings. Her research interests include control systems, including applications to robotics and manufacturing systems.

...