IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 15, 2021, accepted March 8, 2021, date of publication March 15, 2021, date of current version March 25, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3065971

A Methodology to Develop and Implement Digital Twin Solutions for Manufacturing Systems

YASSINE QAMSANE, (Member, IEEE), JAMES MOYNE, (Member, IEEE), MAXWELL TOOTHMAN, ILYA KOVALENKO, (Member, IEEE), EFE C. BALTA, (Graduate Student Member, IEEE), JOHN FARIS, DAWN M. TILBURY, (Fellow, IEEE), AND KIRA BARTON, (Member, IEEE)

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

Corresponding author: Yassine Qamsane (yqamsane@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 Digital Twin (DT) is an emerging technology that has recently been cited as an underpinning element of the digital transformation. DTs are commonly defined as digital replicas of components, systems, products, and services that receive data from the field to support intelligent decision-making. Although several frameworks for DT application in manufacturing have been proposed, there is no systematic methodology in the literature that supports the development of scalable, reusable, interoperable, interchangeable, and extensible DT solutions, while taking into account specific manufacturing environment needs and conditions. This paper introduces a DT solution development methodology as a generic procedure for analyzing and developing DTs for manufacturing systems. The methodology is based on the well-known System Development Life Cycle (SDLC) process and takes into consideration: (1) the specificity of DT characteristics and requirements, (2) an understanding of the manufacturing context in which the DTs will operate, and (3) the object-oriented aspects required to achieve DT capabilities of scalability, reusability, interoperability, interchangeability, and extensibility. A case study illustrates the advantages of the proposed methodology in supporting manufacturing DT solutions.

INDEX TERMS Digital twin, smart manufacturing, industry 4.0, modeling, lifecycle management.

I. INTRODUCTION

Manufacturing is undergoing a digital transformation driven by high degrees of automation and digital technologies in combination to achieve the convergence between the physical and digital worlds. Emerging manufacturing strategies such as Smart Manufacturing (SM) in the US and Industry 4.0 (I4.0) in Europe and Asia promote the application of new information technologies in manufacturing to achieve this physical-digital convergence. SM and I4.0, often used interchangeably, aim at converting data acquired from different locations in the manufacturing ecosystem into manufacturing intelligence to support and improve decision making quality, efficiency, cost, and agility [1]–[3].

Among the many tenets of SM and I4.0, ‘‘Digital Twin’’ (DT) technology offers a tremendous opportunity to realize the physical-digital convergence. A DT is a purpose driven virtual representation of a physical asset, system, process, product, service, or person, that acts as a mirror of the real world to provide the means to detect, predict, or prescribe some aspect of the desired performance in order to improve the mirrored physical environment [3]–[8]. In particular, DTs have been extensively investigated in manufacturing, allowing manufacturers to leverage existing and emerging technologies for improving the performance of manufacturing systems. Recent efforts have focused on DT frameworks that provide the ability to easily reuse, scale, extend, verify, validate, integrate, interchange, and maintain DT technology, approaches, and solutions. However, these frameworks do not provide a systematic methodology for the design and development of practical DT solutions that fulfill the above-mentioned abilities.

To address this issue, the contribution of this paper is a systematic methodology for practical DT solution development. This methodology uses the baseline requirements-driven framework proposed in [8] and extends it by taking into

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

account specific manufacturing environment requirements. Because there are often multiple ways to realize DT solutions that meet specified requirements, this methodology supports the creation and comparison of multiple solutions. The proposed methodology complies with the System Development Life Cycle (SDLC) [9]–[12], a well-known management process that defines the stages involved in taking a system from envisioning to initial deployment and testing. Following SDLC guidelines, the systematic methodology proposed in this paper allows for effective off-line development (i.e., prior to on-line deployment) of practical, implementable, purpose-driven DT solutions and solution frameworks.

The rest of the paper is organized as follows. Section 2 provides background on the DT research area and briefly introduces the SDLC development process and the benefits of using such a process to develop effective DT solutions. Section 3 details the proposed SDLC-based DT development methodology. Section 4 showcases the effectiveness of the proposed methodology by applying it to a manufacturing example. Finally, Section 5 summarizes the contributions of this paper and presents some future research avenues for this work.

II. BACKGROUND

A. DIGITAL TWIN

Digital Twin (DT) is an emerging technology that has recently been cited as an underpinning element of an organization's digital transformation (see, e.g., [13], [14]). The growth of DTs, particularly in manufacturing, has been enabled thanks to the advancements in SM and I4.0 concepts such as the Industrial Internet of Things (IIoT) and Big Data. DTs have been defined in many ways in the literature. It is commonly cited that the DT terminology was first coined by Michael Grieves, then of the University of Michigan, in his presentation about Product Lifecycle Management (PLM) in 2002. Grieves defined a DT as 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 [4]. Another widely accepted DT definition is that of NASA and the US Air Force Research Laboratory, in which a DT is defined as "an integrated multiphysics, multiscale, probabilistic simulation of an as-built system that uses the best available physical models, sensor updates, fleet history, etc., to mirror the life of its corresponding physical twin" [15].

Currently, there is no consensus on a common DT definition in manufacturing. However, most DT definitions have the properties that are captured in the definition in [8]. Namely DTs either implicitly or explicitly provide a virtual replica of a physical thing (e.g., system, asset, process, person, etc.), reside in the cyber world, provide a service to a DT user to improve an aspect of the environment in which its physical twin exists, uses models to achieve its purpose, incorporates some level of Subject Matter Expertise (SME), and uses data (collected in an operation environment) to maintain some type of synchronization with its physical counterpart. The DT definition in [8] will be used in the methodology developed in this paper.

DTs are being applied in different fields, especially in manufacturing, and are showing great potential. Developing and implementing DT solutions builds a foundation to realize SM. For instance, DTs are used for rapid individualized design of automated flow-shop manufacturing systems (see, e.g., [16]); automatic reconfiguration in the case of design deficiencies (see, e.g., [17]) or disruptions during production (see, e.g., [7]); and predictive maintenance to monitor and predict the performance and condition of equipment during operation (see, e.g., [18]). A literature search of DT definitions and use-cases could be found in [19]–[21].

The cost-effective implementation of DT technology in manufacturing requires a DT framework that enables reusability, extensibility, interoperability, and interchangeability of DTs. Examples have begun to appear in literature to investigate the benefits of combining DTs and developing DT frameworks across the manufacturing value chain. The ISO 23247 (Digital Twin Manufacturing Framework) standard [22] provides a generic framework for the use of DTs in manufacturing. The standard provides guidelines and procedures for analyzing modeling requirements, defining scope and objectives, promoting common terminology usage, specifying a generic reference architecture that enables instantiating a DT for a specific use-case, and supporting information modeling of the physical system and information synchronization between a DT and its physical counterpart [23]. The Asset Administration Shell (AAS) framework [24] provides specifications and structures that ensure interoperability and exchange of information about I4.0 components in a meaningful way between partners in the manufacturing value chain. Several works have been focusing on extending the AAS framework to practical implementations (see, e.g., [25], [26]). The Digital Twin enhanced Industrial Internet (DT-II) reference framework was proposed in [27] to achieve a unified DT framework including product lifecycle level, intra-enterprise level, and inter-enterprise level.

The works in [24]–[27] provide specifications on the structure of a DT framework, but not specifications on how to design or develop a DT. Thus, any requirements regarding reusability, interoperability, interchangeability, and scalability have been left to the development process, which has currently not been specified. The introduction of a DT development process would complement existing framework specifications and ensure the development of effective DT solutions, while taking into account the requirements for reusability, interoperability, interchangeability, and scalability. Therefore, this paper presents an approach to develop practical DT solutions that use a systematic SDLC-based DT development methodology. The approach extends on our recently proposed baseline framework [8] that includes a method for the generalization and combination of DTs under a class structure using Object Oriented (O-O) constructs. The first O-O construct is generalization/specialization, which is a relationship in which objects of the specialized element (the child) are substitutable for objects of the generalized element (the parent). This relationship allows for the extrapolation of capabilities usually from general to specific [8], [28]. The second O-O construct is aggregation, which is 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'') [8], [28].

Nevertheless, the baseline framework needs to be extended in a manner suitable for individual application environments. This paper presents a systematic methodology for extending the baseline framework of [8] to realize practical and implementable DT solutions.

B. SYSTEM DEVELOPMENT LIFE CYCLE

Effective implementations of DTs entail good practices for solution design, development, verification, validation, deployment, and maintenance. Therefore, 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 [8]. In applying this lifecycle, the focus in practice to-date has generally been on demonstrating ''one-off'' solutions; thus, there are issues with extensibility and reusability of these solutions across different scenarios and different locations within the manufacturing ecosystem. The assessment of the viability of DTs in ''one-off'' solution development processes is commonly ad-hoc, informal, and improvisational. Thus, the quality of DT solutions development and management lacks systematic methods to support typical roll-out software capabilities such as reusability, interoperability, interchangeability, and extensibility.

A number of methods for systematic design have been developed in fields in and outside of manufacturing. One method that has been extensively used in the software development community is System Development Life Cycle (SDLC). SDLC is a process that describes the sequence of activities or stages that a given system goes through from envisioning to initial deployment and testing [9]–[12]. Typically, SDLC involves six key stages, namely: (1) the planning stage, which consists of determining the reasons to develop the new system/software in order to achieve the organization's strategic goals and objectives, (2) the requirements and analysis stage, which consists of gathering the requirements for the system and conducting an analysis that specifies the functionality that the system should deliver to address the requirements, (3) the design stage, which takes the results of the analysis stage and transforms them into a conceptual model that can be implemented, (4) the development stage, which consists of executing the design in terms of specific technologies such as programming languages, and turning it into a working system, (5) the testing stage, where the implementation is validated to ensure that it meets the requirements for the system, and (6) the deployment stage, in which the system is deployed to the end user once it has been tested and validated [9]–[12].

An SDLC methodology is a formalized approach for implementing the SDLC stages, i.e., it is a list of steps and deliverables [29]. A number of different SDLC methodologies (e.g., waterfall, V-shaped, iterative, spiral, big bang, and agile, etc.) are used today to guide systems and software developers in realizing their projects [10], [29]–[31]. Each SDLC methodology is unique based on the order and emphasis it places on each SDLC stage. SDLC methodologies are designed to address specific needs of specific projects.

While there are several frameworks for DT application in the literature, there is a lack of application of systematic techniques that leverage well-known approaches such as the SDLC methodologies to achieve scalable, reusable, interoperable, interchangeable, and extensible DT solutions. The SDLC process can be used to detail the steps involved in a DT lifecycle. The DT lifecycle (see Figure. 1) can be decomposed into two main phases, namely off-line development and on-line deployment and maintenance [8]. The off-line development phase is concerned with activities to envision, design, develop, verify, and validate DT solutions prior to their on-line deployment. This paper focuses on Phase I of the DT lifecycle, i.e., the off-line development phase. This phase leverages off-line data (often called ''data at rest''), or historical data, with SME insights and applied analytics to understand the manufacturing environment, determine the feasibility of building a DT solution that can provide benefits, develop candidate DT solutions, and quantitatively assess these solutions. A qualified off-line DT solution is then deployed, used, continuously evaluated, and maintained in the on-line deployment and maintenance phase. To address the lack of a systematic approach that supports DT development throughout the lifecycle, we propose an SDLC-based DT development methodology that details the off-line DT lifecycle activities (see Figure. 1) that ensures a successful handoff of effective DT solutions to the on-line deployment and maintenance phase in manufacturing environments.

The proposed methodology considers: (1) the specificity of DT objectives, characteristics, and requirements, (2) an understanding of the manufacturing context in which the DTs will operate, and (3) the object-oriented aspects required to achieve DT capabilities of scalability, reusability, interoperability, interchangeability, and extensibility.

III. THE SDLC-BASED METHODOLOGY FOR DEVELOPING PRACTICAL DT FRAMEWORKS

This section presents a systematic SDLC-based DT development methodology that enables the realization of practical DT solutions using the baseline framework of [8]. The proposed methodology provides general guidelines with defined processes for DT development.

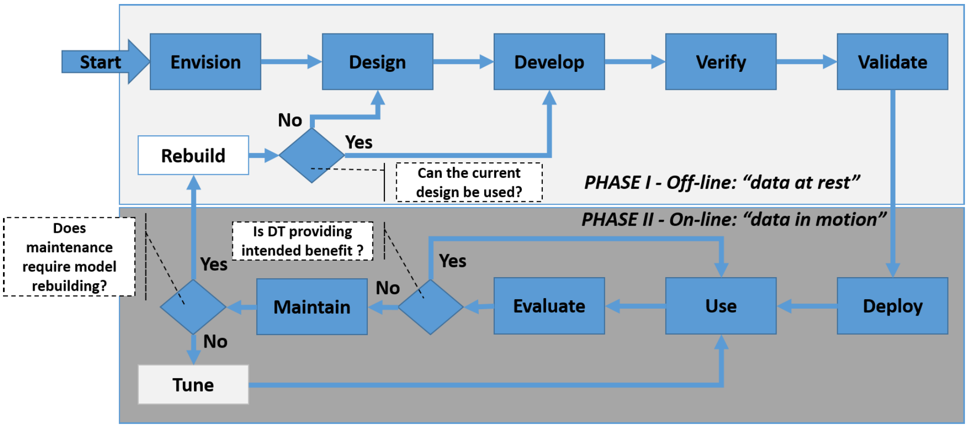

FIGURE 1. High-level view of DT lifecycle [8]. Phase I, off-line development, leverages "data at rest" to envision, design, develop, verify and validate a new DT. Phase II, on-line deployment and maintenance, deploys the DT in the manufacturing environment, using "data in motion" with the outputs of the DT being evaluated regularly, maintained as needed, and updated when required (by returning to Phase I). This paper focuses on Phase I.

A. INTRODUCTION: TOWARDS A METHODOLOGY FOR PRACTICAL DT FRAMEWORKS DEVELOPMENT

As mentioned earlier, this paper introduces an SDLC-based DT development methodology to address the lack of a systematic approach that supports DT development throughout the lifecycle. Key factors driving the SDLC-based DT development methodology include:

- Complying with SDLC guidelines for solution envisioning through design, development, and initial deployment and testing so that solution requirements are consistently addressed.
- Separating the design process into conceptual and final (software) tasks so that more expensive software resources are not incurred until a conceptual design is agreed upon.
- Including go/no-go style decision points throughout the process so that key decisions can be made before additional resources are committed.
- Supporting the creation and comparison of multiple solutions, because there are often multiple ways to realize DT solutions to meet specified requirements.
- Addressing key inquiries about developing and implementing a DT solution such as: What technologies are needed to create a DT? What supporting technologies already exist? How can these technologies be leveraged?
- Developing solutions that can be part of a DT framework to facilitate capabilities such as scalability, reusability, interoperability, interchangeability, and extensibility [8].

B. TERMINOLOGY AND DEFINITIONS

The following is a summary of the main O-O concepts and DT related terminology that will be used in this section to present the SDLC-based development methodology.

- *Attribute*: Property or characteristic of an entity [32].
- *Augmentation*: Adding extra models, algorithms, or data to existing DT resources to provide an intended DT contribution.
- *DT contribution*: Output and quality-of-output metrics that a DT delivers to address a particular manufacturing need [8].
- *DT class*: A type of DT that describes a set of DT objects that share the same attributes, operations, methods, relationships, and semantics in order to deliver a specific capability for the DT user [8], [32].
- *DT lifecycle*: Series of stages a DT undergoes throughout its life. It includes off-line development stages, i.e., inception, design, development, and verification & validation; and on-line deployment stages, i.e., deployment, use, evaluation, and maintenance [8].
- *DT object*: Entity with a well-defined boundary and identity that encapsulates state and behavior [8], [32].
- *DT O-O hierarchy model*: General O-O conceptual model that structures the DT classes that constitute the entire DT solution.
- *DT resource*: Set of models, data, and analytics algorithm that a DT solution uses to provide its contribution.
- *DT solution*: A means for solving a particular manufacturing need through the use of one or more DT classes.
- *DT solution alternative*: Because there are often multiple ways to realize DT solutions to meet specified requirements, a DT solution alternative is a specific type of DT solution that can address a manufacturing need in a particular way.
- *Entity*: Particular thing in the manufacturing ecosystem, such as an asset, system, component, process, product, person, etc. (adapted from [32]).
- *Method*: Implementation of an operation, which specifies the procedure associated with an operation [32].
- *Operation*: Service that can be requested from a DT object to affect behavior [32].
- *Service*: Specific work performed by a DT object [32].

C. THE SDLC-BASED DT DEVELOPMENT METHODOLOGY

In this subsection, DT development is detailed at each SDLC stage (i.e., Planning, Requirements and Analysis, Design, Development, and Testing). As part of this detail, the elements from the SDLC process that are specific to DT development are highlighted.

1) STAGE 1: PLANNING

The objective of the planning stage is to determine if there is a need to enhance some aspect of the manufacturing ecosystem and if that need could be addressed through the application of a DT solution. This stage is SME-centric; an SME describes a manufacturing need driven by strategic objectives or performance requirements not being met or need to be improved. In this stage, a baseline assessment is carried out to (1) identify specific areas where the performance does not meet requirements and determine why these requirements are not being met (i.e., what problems are occurring or anticipated), and (2) identify if these requirements could be achieved through the application of a DT. Note that issues such as data or other resource availability are generally not addressed at this stage. As shown in Figure 2, the DT lifecycle is initiated only if the identified manufacturing need is determined to be addressable through the application of a DT solution as defined in [8]. A set of high-level assumptions and constraints is identified for the manufacturing need and one or more DT solution alternatives are determined. Each recommendation will consist of a set of concepts and methods that identifies the contribution of the DT in terms of what it will provide to address the manufacturing need.

The DT contribution consists of producing and providing quantifiable output metrics that can be used to address a manufacturing need as well as quality-of-output metrics that quantify the quality or believability of these DT output metrics [8]. A set of metrics to be used to evaluate and make decisions about the specific manufacturing need is identified by SMEs. The different DT solution alternatives are proposed to evaluate these metrics. Note that the values of these metrics may vary from one DT solution alternative to another. DT output metrics are usually related to the evaluation of some Key Performance Indicators (KPI) used to monitor, analyze, and/or optimize the manufacturing environment. Examples of metrics include a binary indication of an event (e.g., fault), a continuous indicator for the prediction of a future event occurrence (e.g., Mean Time Before Failure (MTBF)), a recommendation for reconfiguration (e.g, a new schedule for a job), and a prescription for future mitigation of an event (e.g., a control action such as a change in a machine parameter). The ability to quantify the quality or believability of the DT output metrics provides an indication of the DT solution's quality and accuracy. The quality-of-output metrics might relate to the quality of the detection, prediction, or prescription of an event (e.g., probability that the event: has occurred / will occur / can be prevented from occurring), and the quality of the prediction horizon (e.g., a confidence interval on the

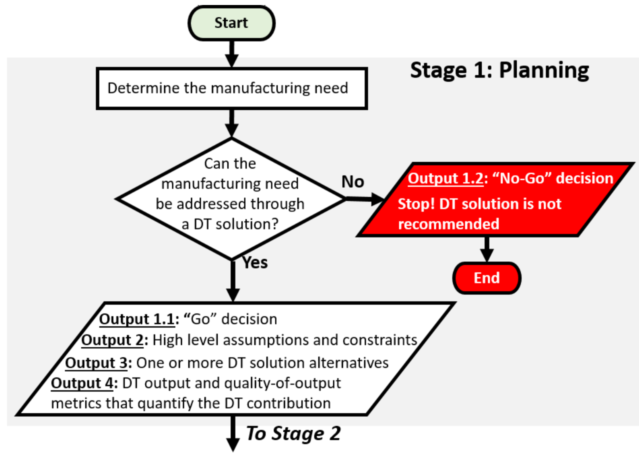

FIGURE 2. The planning stage flowchart.

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 to aspects of "how", "where", "when", "why" or "to what extent" associated with the DT output metric [8].

Once the DT contribution has been determined, high level descriptions of DT solutions that will provide values for these metrics are envisioned. The descriptions are developed by SMEs and serve to describe how the concepts and methods identified in this stage address a manufacturing need by providing values to these DT output metrics.

The outputs of this stage are: (1.1) a "Go" / (1.2) "No-Go" recommendation of whether or not a DT solution could be used to evaluate and make decisions about a specific manufacturing need, (2) a set of high-level assumptions and constraints on the identified manufacturing need, (3) one or more alternatives, each consisting of a set of concepts and methods that identify how to address the need through a DT solution, and (4) DT output and quality-of-output metrics that quantify the DT contribution.

2) STAGE 2: REQUIREMENTS AND ANALYSIS

The requirements and analysis stage is initiated to study and analyze the requirements for the DT design and development activities. In this stage, the scope and capabilities of the envisioned DT solution, defined by the DT contribution (output and quality-of-output metrics), are analyzed by exploring a set of possible technical solution alternatives to meet the manufacturing need. The DT solution alternatives should be analyzed from qualitative (initially) and (then) quantitative DT lifecycle perspectives. A qualitative analysis investigates the feasibility of a range of possible solution alternatives based on SME observations and understanding of resources, as well as data quality requirements (e.g., availability, accessibility, volume, velocity, and veracity) [2], [33]. The quantitative analysis studies feasible solution alternatives in terms of technical capability and performance measures using mathematical calculations and statistical analysis. Serializing qualitative assessment ahead of quantitative reduces the resource investment requirement, especially in cases where a DT solution approach is determined to not be qualitatively feasible. Additionally, the qualitative and quantitative assessments typically narrow the possible technical DT solution alternatives to a few potential, desirable solutions that should continue into the next DT lifecycle stages.

As shown in Figure 3, the analysis stage consists of the following steps:

Step 2.1: Qualitative Assessment. The first step of the requirements and analysis stage consists of carrying out a qualitative assessment of the DT solution alternative by analyzing the resources, data, and data quality requirements. In this step, the existing resources (DTs, models, analytics software, and data) already in use within the system are identified. Typically, resources come from widely disparate places in the system, therefore, it is necessary to define the boundaries for resource acquisition (e.g., just because something is there does not mean it can be used). Security requirements for protecting sensitive data often play a role here, thus any security considerations that may affect access to resources must be identified [34]. Note that security of a DT framework that uses high-value data is an important issue to be addressed. For example, blockchain is a technology that may have the potential to address important security issues [35]. Blockchain provides a secure data trail, in which transactions between the many data sources and applications that are associated with a DT, belonging to the same network are stored in a secure, verifiable, and permanent way. While security is a key requirement of a DT framework, it is not addressed in detail in this paper.

Based on the existing resources and resource boundaries, it is determined whether the DT solution alternative: (i) can be directly provided by the existing accessible DT resources (i.e., DTs, models, and analytics algorithms and software), (ii) can be derived by combining two or more existing accessible DT resources, (iii) can include an augmentation of one or more existing accessible DT resources with additional new DT resources, or (iv) requires a completely new DT solution. In all cases, the O-O constructs of generalization and aggregation (see Section II-A) are used to define how DT resources (existing and envisioned) are reused, combined, and shared, etc.

An assessment should be made as to whether generalization can be applied to facilitate the realization of the DT solution alternative. As an example, a DT for a similar purpose developed for a solution or solution component from a different manufacturer or in a different application environment might be able to be generalized to a level where aspects of the DT can be reused for a solution alternative as part of a DT framework. As another example, a DT generalization hierarchy might already exist for this solution type as part of a DT framework and the solution alternative or a component of it might be realized as an instantiation of an existing class, or through sub-classing an existing DT class. If the DT solution alternative or a portion of it is an augmentation of existing resources, the aggregation construct is used to combine existing and additional augmentation resources. In the case of envisioning a new DT solution, the generalization/specialization and aggregation constructs are used to consider reusing resources among solutions and what the DT might contribute as a generalized or aggregate component of future DTs. Such constructs provide a vision as to how to reuse DTs in the current and future DT landscape.

If the DT contribution is determined to be able to be provided with sufficient quality either directly or by combining existing accessible DTs, models, or analytics software, then, a quantitative assessment of these resources is initiated in step 2.2. If the DT contribution requires augmentation with other resources, then these resources are qualitatively analyzed to determine if they provide sufficient quality. If the combination of resources provides a DT contribution with sufficient quality, then a quantitative assessment is initiated in step 2.2. Any discrepancies between the needed and available resources should be documented

As part of this step, data quality is assessed in terms of the "five Vs" of big data (Volume, Velocity, Veracity, Variety, and Value) [2], [33], all of which are important to the quality of a DT solution.

For each DT solution alternative defined in the planning stage, step 2.1 provides (1.1) a "Go" / (1.2) "No-Go" recommendation on whether to continue to pursue development of the DT solution alternative; (2) In the case of a "no-go" recommendation, the discrepancy between needed and available resources should be documented so that the DT solution evaluation could be reassessed at a later time, e.g., if other DT solutions are determined to be infeasible or underperforming; (3) a set of resources that must be accessed to realize the solution; (4) an indication of whether the solution is feasible: if the minimum set of resources needed to realize the DT solution alternative can be accessed, the solution alternative is determined to be feasible, otherwise it is determined to be not feasible; and (5) high-level O-O models that describe the reuse of existing DT resources through generalization and aggregation constructs. This also provides an understanding of any development that might be needed to augment existing resources.

Step 2.2: Quantitative Assessment. This step quantitatively evaluates metrics of success of each of the DT solution alternatives with a "Go" decision identified in the output of step 2.1 by assessing the data quality resources. In this step, SMEs use analysis and quality metrics to determine the probable quality range of the candidate DT solution. This requires an evaluation of aspects of quality of data resources in terms of availability, accuracy, consistency, context richness, integrity, and timeliness, and how this data quality impacts the quality of the DT output. This evaluation also assesses the effort of increasing the quality of the output of the DT solution alternative, as well as explores methods for gaining insights into the data relevant to the issue that the DT is trying to address. The analysis often consists of evaluating historical data, i.e., "data

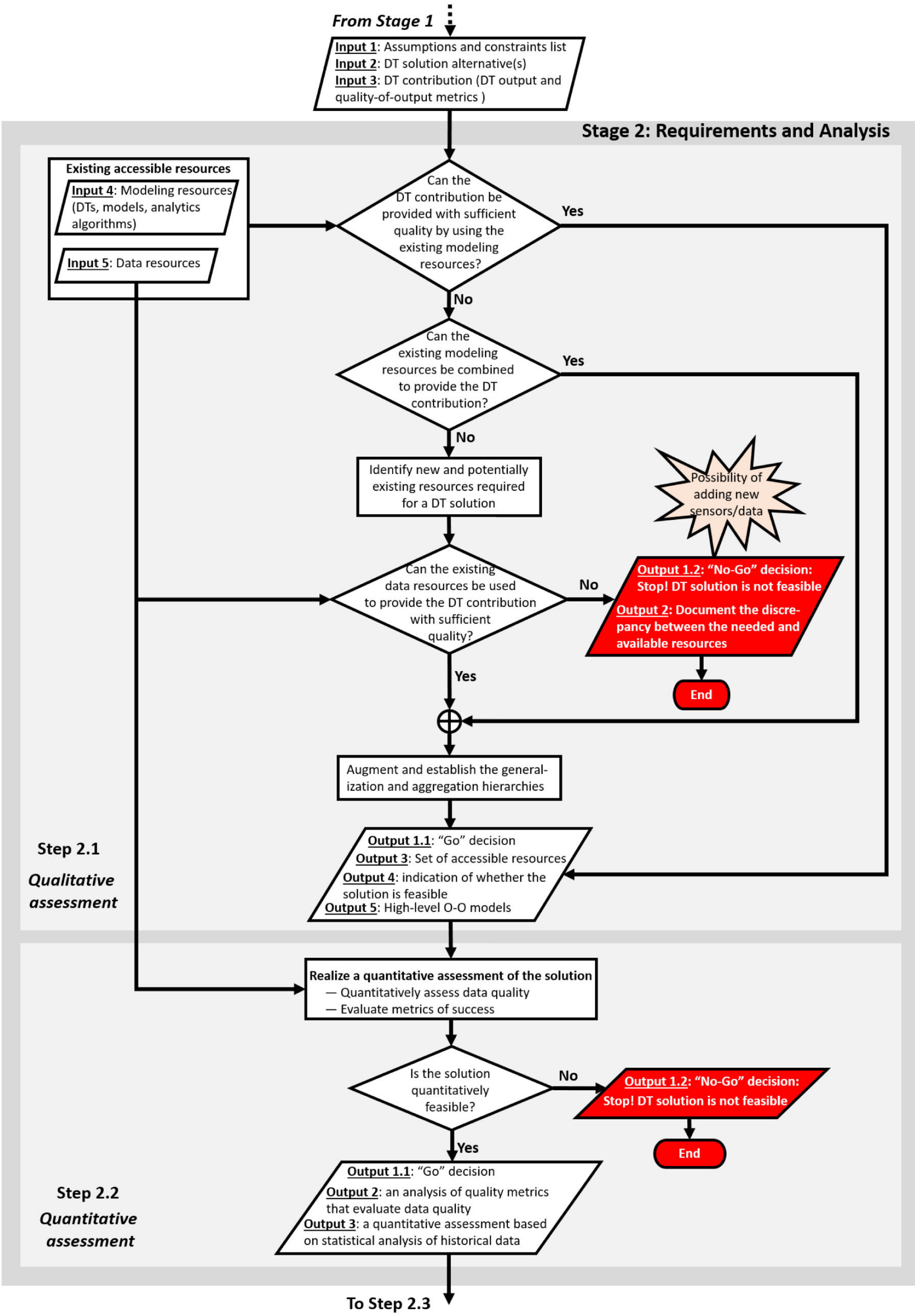

FIGURE 3. The requirements and analysis stage flowchart.

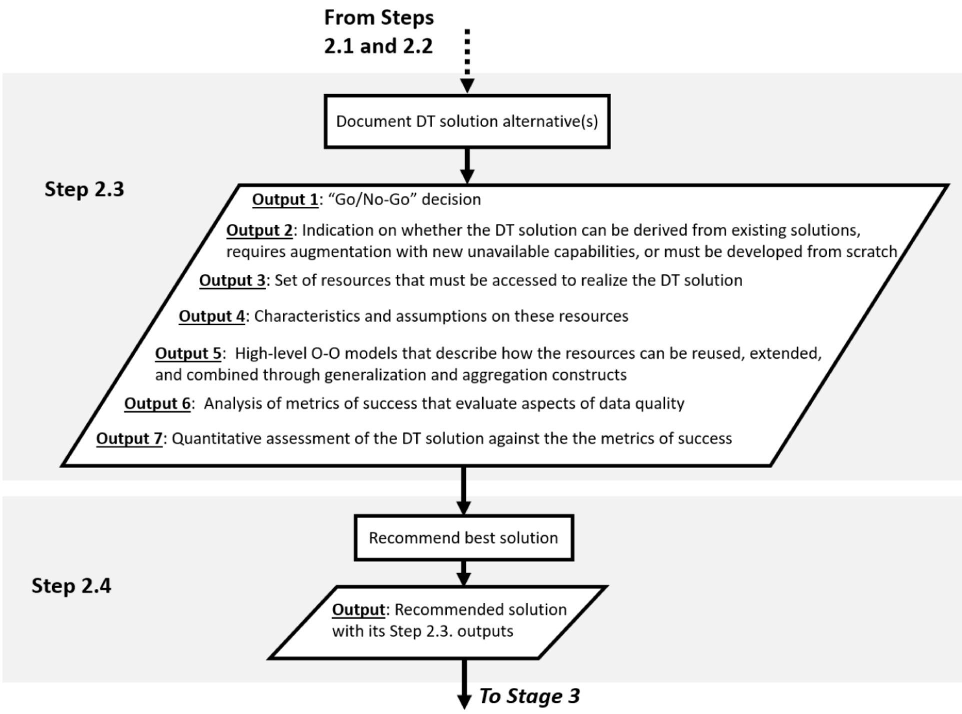

FIGURE 4. The requirements and analysis stage flowchart (continued).

at rest'' using mathematical calculations and statistical analysis. Descriptive analysis, which is an exploratory analysis of historical data, is used to quantitatively describe the main features of the data. Data mining and statistical methods can be used to reveal data characteristics, recognize patterns, and identify relationships between data objects [36].

The following is an example of a process for gaining successful insights from data to support DT development: (1) visualize the data and gain simple insights via the SME, (2) derive simple descriptive statistics, (3) perform univariate analysis on each variable, (4) create derived variables and metrics if necessary, and (5) find the correlation between the variables [37]. Techniques like Receiver Operating Characteristic (ROC) and optimization analysis can also be used to measure the quality of data against the capability of the DT and thus quantitatively assess the DT solution alternative [8], [38].

An important aspect of quantitative assessment is determining the ''financial benefit'' of the application of the DT. A DT is expected to provide a range of benefits when producing an output that improves the system (e.g., a correct indication, prediction or prescription advice). However, because it is a digital approximation, it may occasionally provide incorrect information including a false positive (false alarm) or missed positive (missed alarm). The impact of both correct and incorrect outputs of the DT must be assessed, a financial model developed, and the quality of the DT output applied to that financial model to determine the overall financial impact and impact range of the DT.

For a better understanding and to gain insights from data, interactions with SMEs are a necessary and integral part of this analysis process. For example, incorporating SME feedback provides additional insights on data collection (what data to aggregate? what data to use? what data to trust?), data treatment (how to partition? what features to extract? how to strengthen the data?), and feature selection (key and causal parameters, unsupervised / semi-supervised / supervised and the migration to more supervised environments, independent vs. dependent parameters / features, hyperparameters) [39].

The outputs of this step are, for each DT solution alternative identified as a " Go" in step 2.1: (1.1) a "Go" / (1.2) "No-Go" recommendation on whether to continue to pursue the design and development activities of the DT solution alternative; (2) an analysis of quality metrics that evaluate aspects of data quality in terms of availability, accuracy, consistency, context richness, integrity, and timeliness; (3) a quantitative assessment of the DT solution alternative against the defined metrics of success, based on mathematical calculations and statistical analysis of historical data.

Step 2.3. Document DT solution alternatives. The DT solution alternatives that satisfy the requirements of both the qualitative and quantitative assessment steps are documented in step 2.3.

The output of this step is a document that lists the set of feasible DT solution alternatives. The document includes: (1) a "Go/No-Go" recommendation on whether to continue to pursue the design and development activities of the DT solution alternative; (2) an indication on whether the DT *solution can be derived from existing solutions, requires augmentation with new or unavailable capabilities, or must be developed from scratch; (3) the set of resources that must be accessed to realize the solution; (4) the characteristics and assumptions on these resources; (5) high-level O-O models that describe how the resources can be reused, extended, and combined through generalization and aggregation constructs; (6) metrics of success of the DT solution alternative based on the quantitative analysis of the DT resources; and (7) a quantitative assessment of the capability of the DT solution alternative with respect to the metrics of success.*

Step 2.4. Recommend best DT solution alternative. In this step, the feasible DT solution alternatives (if any) that satisfy the qualitative and quantitative assessment requirements are compared with respect to decision criteria (e.g., technology used, cost, etc.). At least two alternatives should be compared: the alternative of doing nothing versus the alternative of anticipating the benefit and the likely effects of taking action through a DT solution. The best DT solution alternatives that satisfy these requirements are sorted and the best-approved one is recommended and is an input to the next stage of the DT lifecycle.

The output of this step is a list of feasible DT solution alternatives (including the "do nothing" alternative) sorted based on decision criteria and a recommendation of the best solution based on that decision criteria.

3) STAGE 3: DESIGN

The design stage defines a design of a DT solution as part of a DT framework based on the recommended alternative that will meet functional, data, and interaction requirements. The functional requirements are determined in the planning stage. They consist of concepts that identify how the DT solution alternative can address the manufacturing need through DT output and quality-of-output metrics (i.e., the DT contribution). Data requirements are determined in Stage 2: Requirements and Analysis (Section III-C2) where the data resources for the DT solution alternative are qualitatively and quantitatively assessed to determine if these data resources can be used to provide the DT contribution with enough quality. On the other hand, interaction requirements are defined as part of this design stage. They represent how the objects that the DT solution is representing are interconnected. These objects are abstracted in DT classes that deliver a specific DT contribution for the DT user [8]. The requirements are mapped into a global DT O-O hierarchy model that defines the DT classes, their attributes, operation, methods, and relationships. The global DT O-O hierarchy model also defines how DT resources for the overall DT solution are combined and shared, and how they can be extended and reused.

The key to constructing the global DT O-O hierarchy model is to define the DT classes for the DT objects based on the recommended DT solution. These objects are identified by decomposing the portion of the system that the DT solution is representing into components, sub-components, and so on; splitting the functions of that portion of the system into sub-functions on an increasing level of detail; or by using both forms where appropriate. A methodology to construct the DT O-O hierarchy model is shown in Figure 5. The methodology consists of the following steps:

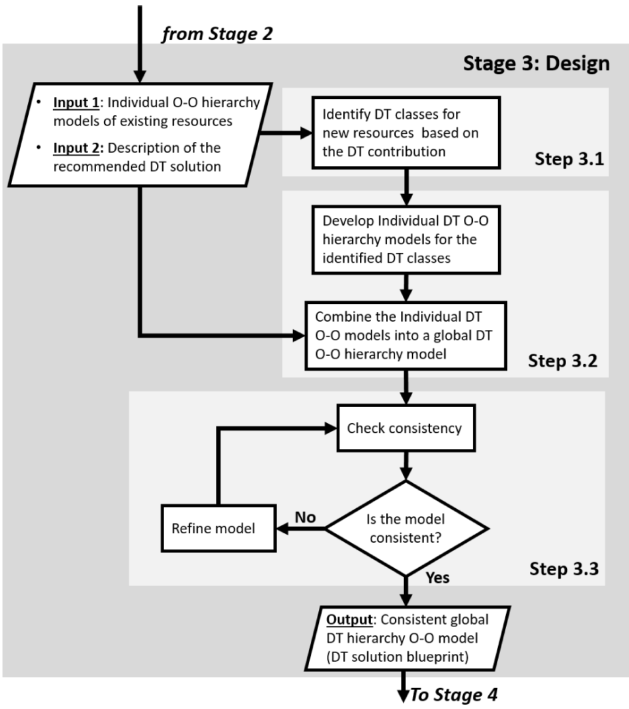

FIGURE 5. **The design stage flowchart.**

Step 3.1. Identify the DT classes: DT classes are determined based on the description of the recommended DT solution, i.e., based on the contribution it should provide (DT output and quality-of-output metrics). DT classes are identified for components of the solution that provide a capability by delivering an output and quality-of-output metrics. For instance, if a DT contribution to address a manufacturing need requires delivering a process quality metric, a DT class that evaluates process capability (Cpk) can be defined for different machining processes in a system. Some DT classes may have already been identified for the existing resources in the system as per step 2.1 in stage 2 or any other DT creation process. These DT classes are leveraged in this step, with the goal being to use existing classes or sub-class of an existing class in the DT framework if possible. Note that DT objects belonging to the same class might utilize different modeling approaches to provide a common DT output but might report different quality-of-output metrics that depend on the modeling approach; this fact could be used to expand the definition of a class to support a DT object in the recommended solution [8]. If an existing class cannot be used for the DT object, a new class is created. The classes are given names that provide some indication of purpose and scope of the DT contribution.

Step 3.2. Incrementally build the DT O-O hierarchy model for the recommended DT solution: Individual DT O-O models are created for the DT objects in the recommended DT solution based on the DT classes defined in step 3.1. If the DT class (the whole) consists of a combination of other DT classes (the parts), the aggregation construct [8] is used to establish an aggregate DT O-O hierarchy model that contains the other DT classes (the parts). If the DT objects of a specialized DT class (sub-class) may be substitutable for DT objects of a generalized DT class (parent class), the generalization/specialization construct [8] is used to establish the individual DT O-O hierarchy model. The computational engine object of the DT object being constructed is used to combine the outputs of the component DTs in an appropriate way so that constructed DT provides the desired capability and output. All of the established individual DT O-O hierarchy models are then consolidated into one cumulative global DT O-O hierarchy model. Unified Modeling Language (UML) relationships between the DT objects are used to establish the cumulative global DT O-O hierarchy model. DT class interrelationships are modeled with meaningful names and multiplicities based on the dependencies between DT objects and the number of DT objects involved.

Step 3.3. Check the consistency and refine the DT O-O hierarchy model of the recommended DT solution: Syntactic and semantic inconsistencies in the cumulative DT class hierarchy model have to be discovered and corrected. The DT O-O hierarchy model is refined by addressing the following syntactic and semantic inconsistencies: (i) classes in the consolidated DT O-O hierarchy model that have different names but represent the same thing, (ii) relationships in the consolidated DT O-O hierarchy model that have different names or descriptions but the same meaning, (iii) attributes within the DT classes that have different names but represent the same thing, and (iv) discrepancy between different parts of the DT O-O hierarchy model. Tools for consistency checking of UML models like [40] can be used in detecting and tracking inconsistencies.

Once the DT O-O hierarchy model is checked, refined as necessary, and verified against inconsistencies, it is then ready to progress to the development phase.

The output of the design stage is a detailed DT O-O hierarchy model that provides the structure of the recommended DT solution, the services it provides, and the behavior it exhibits. The established DT O-O hierarchy model serves as a blueprint that describes how the DT classes or instances within the DT solution can be reused, combined, shared, or otherwise aggregated or inherited.

4) STAGE 4: DEVELOPMENT

The objective of the development stage is to transform the output design stage into a complete working DT solution capable of addressing the manufacturing need established in the planning stage. Note that a DT solution combines models that emulate some aspect(s) of the DT's physical counterpart with a computational engine that supports the model(s) in terms of accuracy and alignment with that physical counterpart [8]. The development stage consists of activities for building DT classes and models. DT model development is the main piece of the development stage of the DT lifecycle.

In this stage, the modeling techniques used to transform the DT O-O hierarchy model into an operating DT are identified and built. This stage consists of formulating the DT models which could include selecting algorithms to use, writing the equations linking system response to inputs and values of system parameters, performing variable down-selection, model training, and general evaluation, etc [41]. The DT solution is then developed by generating software programs that capture the algorithms, equations, SME input, and integrating all the DT components together.

The output of this stage is a complete implementation of the DT solution. The stage output includes DT model development, DT functionality development, and software development (integration).

5) STAGE 5: TESTING

After the DT solution is developed, it is tested against the requirements to make sure that it actually meets specifications and fulfills its intended purpose of addressing the manufacturing need identified in the planning stage. Verification and Validation (V&V) processes may be used for testing purposes. Verification is used to check if the requirements, specifications, and regulations are met and that the DT achieves its intended goals without any bugs or gaps [42], [43]. Validation evaluates if the DT satisfies the needs of the user at the end of the development.

The aim of verification is to ensure that the models used by the DTs accurately represent the requirements and specifications, comply with regulations and standards, and will not misbehave under a very broad range of circumstances. Formal verification methods such as reachability analysis, model-checking, equivalence-checking, and model proving may be used here to verify the consistency and compliance of the developed models with the requirements [44]. Simulations could also be used for verification (e.g., Monte-Carlo simulations).

Validation consists of evaluating the DT model under several conditions, observing its behavior, and inspecting it for defects by comparing its results to experimental evidence. Validation ensures that the DT meets the needs of the user as defined in Stage 2. Typically, historical data is used here for sensitivity analysis to evaluate the effects of changes in inputs and in physical system changes on outputs. An example of an objective here might be to establish a distribution of the closeness of a DT output to its physical counterpart, establish a 95% confidence interval and verify that solution objectives fall within this confidence interval. Simulations could be also used for validation [42]. A state-of-the-art in V&V of machine learning-based safety-critical systems can be found in [45].

The output of this stage is a verified and validated DT solution that is ready to be deployed to the end user. If the V&V does not meet specifications, the DT model and/or DT functionality have to be refined.

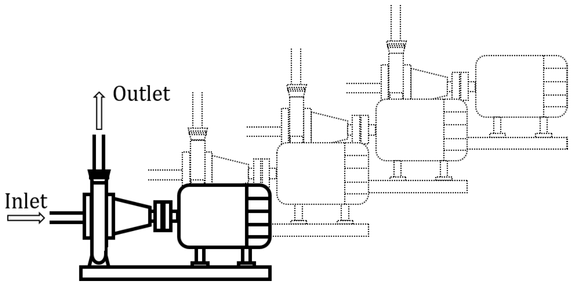

FIGURE 6. The studied pump system consisting of four identical pumps.

IV. CASE STUDY

A. SYSTEM DESCRIPTION

In this section, the SDLC-based DT development methodology presented in Section III is applied to a four pump system (Figure 6) to illustrate the methodology's use and evaluate its effectiveness. The goal is to investigate the benefit of using a DT solution to maximize the production availability of the pump system by avoiding unscheduled downtime. Each pump features four rolling-element bearings, and an accelerometer is installed on each bearing housing to collect vibration data. The study uses a bearing data set from the prognostics data repository hosted by NASA, which was made available by the center of Intelligent Maintenance Systems (IMS) of the University of Cincinnati [46]. More details about the instrumentation used to collect this data set can be found in [47].

B. APPLICATION OF THE SDLC-BASED DT DEVELOPMENT METHODOLOGY

This section explores the application of the SDLC-based DT development methodology by highlighting the actions and decisions at each stage presented in Section III.

1) STAGE 1: PLANNING

(a) *Identifying the manufacturing need*: Pump systems usually have large production volumes, thus, small deviations in capacity and downtime can have substantial economic impacts. A DT solution that provides the ability to determine the condition of the pump system, understand what might be wrong, and thus avoid unnecessary downtime or failures is considered. One contribution of the DT solution for this case study is to provide an estimate of the pump system health state. Therefore, the manufacturing need identified in this case study is to reduce unscheduled downtime resulting from failure by monitoring, assessing, and acting on the condition of the pump system.

(b) *Exploring one or more DT solution alternative to address the identified manufacturing need*: A common approach for reducing the unscheduled downtime that follows equipment failure is to employ a predictive maintenance (PdM) strategy. PdM strategies use equipment state estimates and predictions of future states to inform the management of the optimum time to schedule maintenance. PdM strategies also help locate problems in machinery and help identify the parts that need to be fixed. Hence, a PdM strategy for monitoring the pump system condition is a well-suited DT solution to address the identified manufacturing need. For the sake of brevity, only one DT solution alternative will be discussed here.

(c) *Identifying the DT output and quality-of-output metrics*: Reliable health state estimation is crucial to prevent unscheduled downtime. In this case study, a PdM DT solution will be developed for the purpose of estimating the current equipment health state and predicting the future health state of the pump system of Figure 6. An accurate health state estimate is valuable in many circumstances to flag degrading equipment for repair before complete failure occurs. Another contribution of this DT is a quality-of-output metric that assigns a confidence interval to all potential health state values as well as the probability of future failure in a defined time horizon, given current and historical observations from the physical system.

Note that a true PdM solution is a predictive system and thus provides outputs of (1) a prediction of a future event, (2) a time horizon for that prediction, and (3) quality metrics delineating the confidence in the prediction event (e.g., probability) as well as event time horizon (e.g., confidence limits). A precursor to a complete PdM solution is often a health state estimator DT that can determine that a future failure is "imminent". This health state estimator DT is pursued as an initial PdM solution in this study. Further information that might lead to the development of the full PdM solution is provided in the discussion of Stages 4 and 5.

(d) *Deciding if the DT solution alternative is feasible from a planning standpoint*: PdM DTs using vibration analysis are proven to be an important solution in manufacturing today to increase availability and efficiency of rotating machinery [48], [49]. Using vibration data to monitor the state of pump systems has been demonstrated to be one of the most effective tools for determining their condition [50], [51]. Thus, from a planning standpoint, a PdM DT solution that uses the available vibration data seems to be well-suited to address the identified manufacturing need. A "Go" decision is recommended in this stage to explore additional resources in the Requirements and Analysis stage.

Table 1 Summarizes the outputs of the planning stage for the case study.

2) STAGE 2: REQUIREMENTS AND ANALYSIS

Step 2.1. Qualitative assessment: In this step, an SME performs a qualitative analysis of the DT solution in terms of data environment, DT resource availability, and data quality requirements. The SME's assessment of the recommended PdM DT solution reveals that no pre-existing DTs, models,

TABLE 1. Outputs of the planning stage.

Stage	Stage outputs	
1. Planning	Manufacturing need identification	The goal is to provide the system with the ability to determine the condition of the equipment, understand what might be wrong, and predict future failure with sufficient accuracy to avoid unnecessary downtime or failures and provide net financial benefit in terms of increased uptime, reduced unscheduled downtime due to failure, and minimal unnecessary downtime due to falsely predicting failure
	DT solution alternative(s) proposition	A PdM strategy for monitoring and prediction of the equipment condition is well-suited to inform the management of the optimum time to schedule maintenance *For the sake of brevity, only one DT solution alternative will be explored, and then compared to the alternative of doing nothing*
	DT contribution identification	- DT output: Equipment health state estimate that will be used to determine the condition of the pump system and predict future failure as part of a PdM strategy - DT quality-of-output: a confidence interval assigned to all potential health state values given current and historical observations from the physical system
	Go/No-Go decision	The need is determined to be addressable using a DT solution. Reducing unscheduled downtime resulting from equipment failure can be addressed by a PdM strategy *A "Go" decision is recommended at this stage*

TABLE 2. Data quality assessment of the data resources.

Data quality metric	Measurement	Quality metric assessment
Volume	The datasets of the historical vibration signals consist of 7588 files. Each file consists of 20480 data points. Data archives contain three complete run-to-failure events.	The amount of accessed data is sufficient.
Velocity	For the studied system, data is generated and collected every 10 min and is made available to the virtual plane at a negligible latency.	The speed at which data is accessed is sufficient.
Variety	The collected input datasets include only one structured data type namely, vibration signals recorded using similar accelerometers. The output data is the indication and context associated with fail events. Organizing and analyzing the data in a meaningful way is not an issue.	This data is organized, has a defined length, and defined format.
Veracity/Validity	The collected input datasets consist of three run-to-failure tests, capture complete degradation histories of the system, and are synchronized to the output data (failure indications with context). The data is credible enough from which insights can be gained.	The credibility of the collected data is sufficient.
Value	Visualization of the vibration signals reveals that we can gain useful insights from the collected data (e.g., see Figure 7).	Early insights reveal that using these data in a DT solution will add value to the decision-making.

or analytics algorithms have been identified for the system. Only vibration data that have been collected over the course of three run-to-failure tests, from accelerometers mounted to each of the four bearings that support the pump, are available. The bearing vibration data are accessible and no barriers to data acquisition have been identified. The available data are made up of a series of signal snapshots, taken every 10 minutes. Each snapshot consists of an accelerometer output signal recorded at a sampling frequency of 20 kHz for 1 second. A qualitative analysis of datasets of historical vibration signals from the system is performed to evaluate the data quality requirements in terms of the five Vs of big data, namely: volume, velocity, variety, veracity, and value as shown in Table 2. This analysis reveals that the existing data resources can be used to provide the DT contribution with sufficient quality.

Based on these considerations, the SME recommends the design and development of a completely new solution using the available vibration data. Next, the SME uses the O-O constructs of generalization and aggregation (as necessary) to define how DT resources (existing and envisioned) are reused, combined, and shared. A high-level hierarchy for the DT solution that defines how DT resources for the studied system are reused, combined, and shared is shown in Figure 7. SME analysis reveals that the envisioned DT solution will consist of the following (see Figure 7): A single bearing PdM DT class will be developed based on the existing vibration data along with SME input, and will deliver the bearing state metric with a confidence interval as identified in the planning stage. The bearing PdM DT class will be instantiated for the four bearings in a given pump and the information then aggregated to provide a pump health state and confidence interval at the pump level. At the system-level, the pump PdM DT class will be instantiated for the four pumps constituting the overall pump system with the information then aggregated at the top level to provide a pump system overall health state and confidence interval.

Step 2.2. Quantitative assessment: This step of the Requirements and Analysis stage is concerned with assessing and quantifying the quality of data that can be collected from the "real system".

The historical datasets collected from the system consist of three run-to-failure tests, which means a complete degradation history is captured in each test. Mathematical calculations and statistical analysis of these historical data are

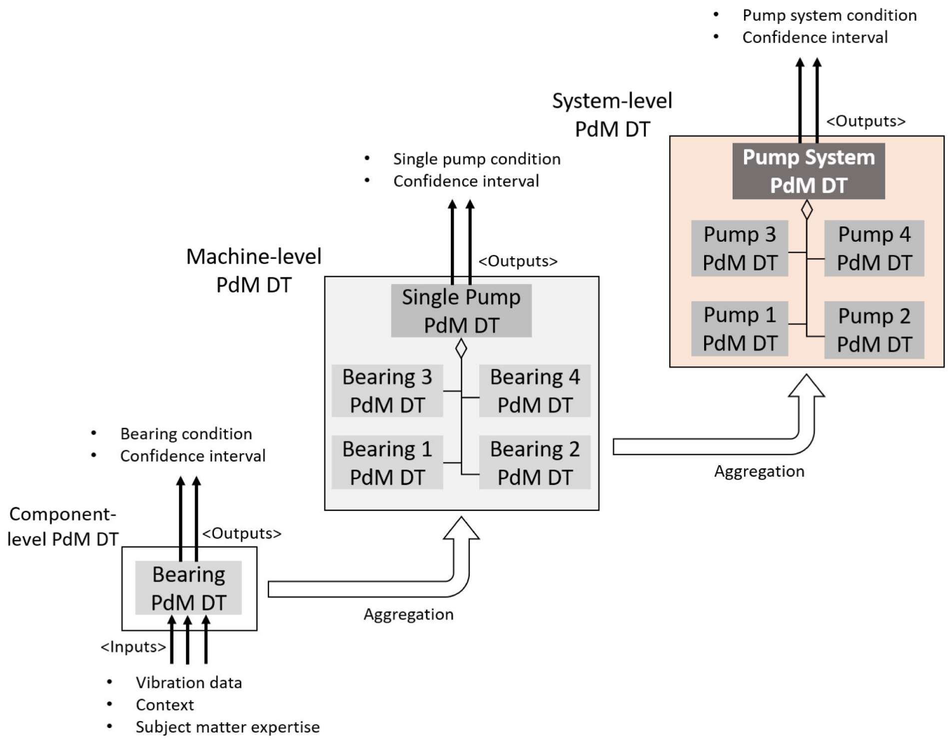

FIGURE 7. High-level hierarchy of the DT solution for the case study example.

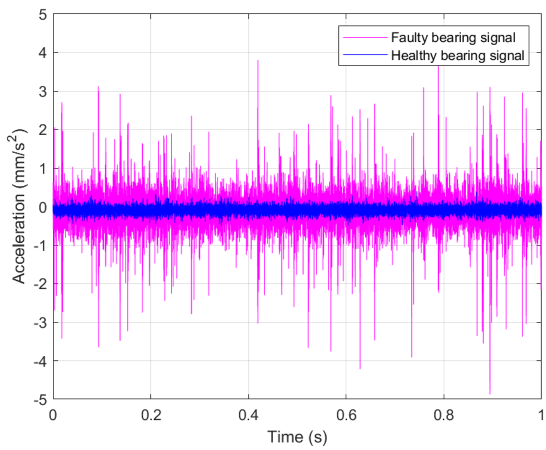

FIGURE 8. A plot of healthy and faulty data of a single bearing.

performed by the SME to help improve model quality of the DT solution. Features that are best suited for health state estimation will exhibit behavioral differences between periods of healthy and degrading operation. Statistical analysis is used to quantify the suitability of these features, however SME input is also crucial because statistical analysis alone does not provide a complete picture of the degradation behavior needed to predict failure. The SME helps determine the features that best discriminate between normal and abnormal states of the system. For example, the SME might select causal features over reactive features. The PdM DT solution development process will ultimately strive to use the most effective features as condition indicators. The SME would also provide input into the potential repeatability and re-usability of the solution, given that the statistical analysis comprises only three run-to-failure tests.

An example feature that displays these types of differences, vibration kurtosis, is shown in Figure 9. This plot shows kurtosis values for two bearings over the course of a single run-to-failure test. One of the bearings remained healthy for the duration of the test, while the other showed severe signs of degradation after the test concluded. A comparison between the failed and healthy bearings confirms that increased magnitudes in the kurtosis value can be attributed to bearing degradation. This difference in magnitude between healthy and degrading bearings indicates that vibration kurtosis is a useful feature for detecting degradation onset. Similar behavior is present for the skewness, peak frequency, and peak-to-peak distance features.

Outputs of the requirements and analysis stage for the case study are shown in Table 3.

3) STAGE 3: DESIGN

A design of the intended PdM DT solution is established in this stage based on the requirements and assessments realized in the previous two stages. These requirements are mapped into a global DT O-O hierarchy model that defines the DT classes, their behaviors, attributes, methods, and interactions. As described in the design stage in Section 3.2, the key

TABLE 3. Outputs of the requirements and analysis stage.

Stage	Stage outputs		
2. Requirements and Analysis	Step 2.1. Qualitative analysis	Set of resources to realize the solution	- No pre-existing DTs, models, or analytics algorithms have been identified - Solution will use vibration data collected over the course of three run-to-failure tests from pump bearings. • No barriers to accessing the data have been identified • SMEs indicate that vibration data could be a good indicator of bearing health and predictor of bearing failure. *Based on these resource requirements, the PdM DT solution requires the design and development of a completely new solution using vibration data*
		Data quality assessment (qualitative)	- Available data consist of a series of accelerometer signal snapshots, taken every 10 minutes at a sampling frequency of 20 kHz for 1 second - The assessment of the volume, velocity, variety, veracity, and value of the data reveals that the quality of data is sufficient for providing the DT contribution (see Table 2) - Each run-to-failure test was conducted in a controlled, experimental environment, so missing data values are rare and the signal-to-noise ratio for the measurements is high enough to disregard external impacts on the signals *The data is of sufficient quality to move forward without adjusting the data collection strategy.*
		High-level O-O model describing the reuse of existing DT resources	A high-level O-O model facilitating the reuse and extension of the existing DT resources through the aggregation construct is established (see Figure 6)
		Go/No-Go decision	The qualitative analysis of the available resources reveals that the PdM DT solution can be achieved *A "Go" decision is recommended at this step*
	Step 2.2. Quantitative analysis	Data quality assessment (quantitative)	The assessment and quantification of data quality through high level statistical and SME input reveals that the quality of data is sufficient for providing the DT contribution
		Analysis of metrics of success of the DT solution based on statistical analysis of historical data	Extraction and analysis of statistical diagnostic features from the data along with SME confirmation indicate that vibration kurtosis, skewness, peak frequency, and peak-to-peak distance features are useful for detecting degradation onset and that a multivariate detection approach could be implemented
		Go/No-Go decision	The quantitative analysis of the available resources reveals that the PdM DT solution can be achieved *A "Go" decision is recommended at this step*

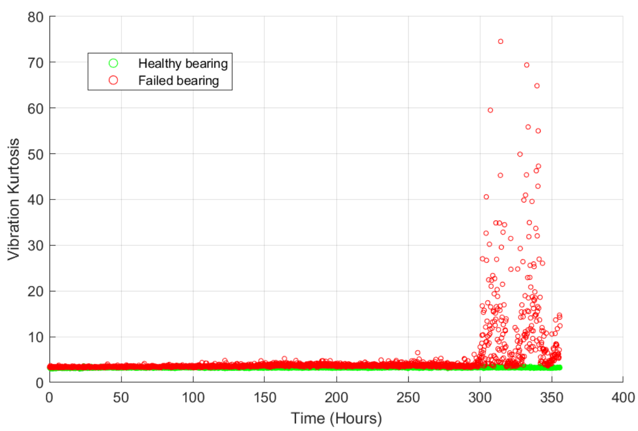

FIGURE 9. Kurtosis variation for two bearings over the course of a single run-to-failure test.

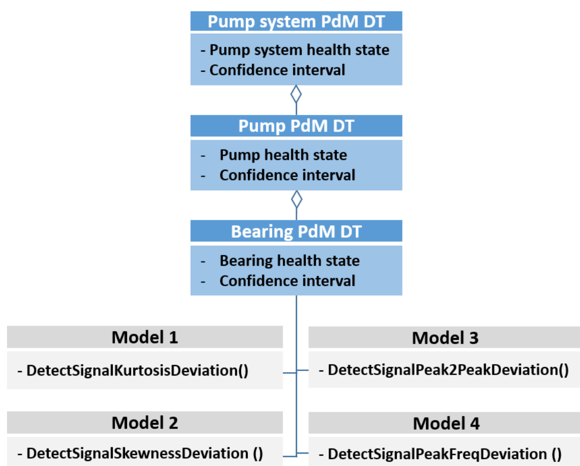

FIGURE 10. The proposed DT O-O hierarchy for the case study,, using individual models for each signal analysis (alternatively these analyses could be combined).

to constructing the global DT O-O hierarchy model is to define the DT classes for the different DT objects and their interactions.

The high-level DT O-O model that was established in the qualitative assessment of stage 2 (see Figure 7) shows that the system can be broken down from the bottom up into a bearing level, pump level, and pump system level. These components / sub-components represent DT objects for which DT classes will be developed. Additionally, the DT O-O hierarchy model at this stage includes the actual model that the DT solution will use to compute the intended contribution. As described in the quantitative assessment in Stage 2, the bearing PdM DT class will take as input four features namely, kurtosis, skewness, peak frequency, and peak-to-peak distance to detect the degradation onset at the bearing level. The information will be aggregated from the bottom up to the pump level, then to the pump system level using aggregation functions.

The DT O-O (aggregation) hierarchy model for this use case is established as follows:

1) DT classes are identified for each DT object in the system. Each of the system components / subcomponents (i.e., bearing, pump, and overall pump system) represents a DT object and a DT class is identified for each of these objects based on the DT contribution. At the bearing level, a bearing PdM DT class is identified to estimate the health state of a bearing using the signal kurtosis, skewness, peak frequency, and peak-to-peak distance features. A model uses four inputs and computes one output metric related to the DT contribution (Figure 10). At the pump level, a pump PdM DT class is identified to estimate the health state of a pump by leveraging instances of the bearing PdM DT class. At the pump system level, a pump-system PdM DT class is identified to estimate the health state of the overall pump system by leveraging instances of the pump PdM DT class. Figure 10 shows the established aggregation DT O-O hierarchy for the case study based on the identified DT classes.
2) After the DT classes are identified and the aggregation DT O-O hierarchy model is established, since the system consists of four pumps, the pump PdM DT class is instantiated for each of them. Each of the pump PdM DT class instances will consist of four bearing PdM DT class instances as each pump comprises four monitored bearings. Moreover, each bearing PdM DT class instance will detect degradation onset based on the signal kurtosis, skewness, peak frequency, and peak-to-peak distance values. This could be done using a model for each approach (as shown in Figure 10) or a more complex model that combines the outputs of these four analyses. Figure 11 shows the instantiation of the aggregation DT O-O hierarchy model for the studied pump system. This design model facilitates developing a PdM DT for a single pump then extending it to the three other pumps in the system. Note that some details may slightly differ for the different pumps, which might require model tuning when extending and reusing the DT resouces. This could be done using a model for each approach (as shown in Figure 9) or a more complex model that combines the outputs of these four analyses. Sharing of the O-O hierarchy for DT classes improves the design process greatly and promotes reusability across similar physical components on the plant floor. Outputs of the design stage for the case study are shown in Table 5.

4) STAGE 4: DEVELOPMENT

In the development stage, the outputs of the requirements & analysis, and design stages are transformed into a complete working DT solution capable of addressing the manufacturing need established in the planning stage. A bottom up development approach is followed using the DT O-O aggregation hierarchy model of Figure 11. At the bearing level, the bearing PdM DT class is developed. Since the objective of the bearing PdM DT class is to detect degradation onset in a single bearing, a discrete event system (DES) model with two health states: healthy and degrading, and a single state transition: degradation onset (denoted by α) is used. The initial state in the DES model is the healthy state, which is derived from the assumption that pump bearings exhibit an extended period of uniform operation at the beginning of their life.

TABLE 4. Performance metrics from validation testing of the DT solution.

Test #	Bearing number	Predicted final health state	True final health state	Test length	Estimated degradation onset time
1	1	Healthy	Healthy	359 hours 10 mins	N/A
	2	Healthy	Healthy		N/A
	3	Degrading	Degrading		301 hours 30 min
	4	Degrading	Degrading		266 hours 30 mins
2	1	Degrading	Degrading	163 hours 50 mins	118 hours 20 mins
	2	Healthy	Healthy		N/A
	3	Healthy	Healthy		N/A
	4	Healthy	Healthy		N/A
3	1	Healthy	Healthy	1053 hours 50 mins	N/A
	2	Healthy	Healthy		N/A
	3	Degrading	Degrading		1039 hours 20 mins
	4	Healthy	Healthy		N/A

The primary capability of the bearing PdM DT class is to detect when degradation onset has occurred. Criteria for detecting the α state transition will be defined in terms of features extracted from periodic vibration samples, as described in Stage 2.

The modeling approach described above is developed for the bearing PdM DT class to detect degradation onset in a single bearing. This DT class is instantiated for each of the four bearings in a pump. Information from the four bearing PdM DT instances is aggregated at the pump level DT class using a "worst-case" aggregation function where the health state of the pump is equal to the worst health state of the four bearings. The pump PdM DT class is instantiated for each of the four pumps constituting the system. Information from the four pump PdM DT instances is aggregated at the pump system level DT class. At the pump system level, three out of the four pumps must be available all time to handle maximum capacity. Thus, the pump system health is in a degrading state if at least two pumps are in a degrading state.

As mentioned earlier, a complete PdM solution provides outputs of (1) a prediction of a future event, (2) a time horizon for that prediction, and (3) quality metrics delineating the confidence in the prediction event (e.g., probability) as well as event time horizon (e.g., confidence limits). For the sake of brevity, we only described the health state estimator portion of the full PdM strategy. The time horizon for the prediction, quality metrics delineating the confidence in the prediction event, and event time horizon are yet to be developed. These capabilities are left for future work.

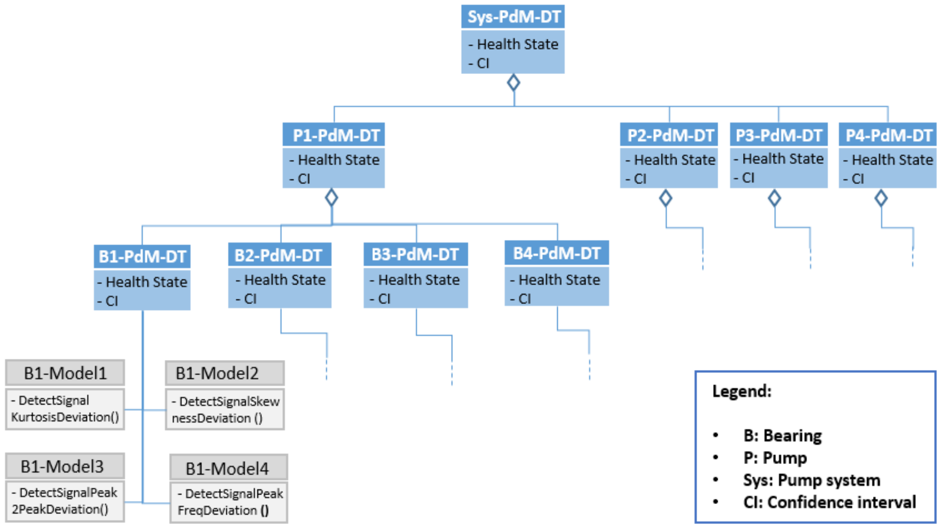

FIGURE 11. Instantiation of the DT O-O hierarchy model for the studied system.

TABLE 5. Outputs of the design, development and testing stages.

Stage	Stage outputs
3. Design	- A detailed DT O-O aggregation hierarchy model that provides the structure of the DT solution, the services it provides, and the behavior it exhibits is established (Figures 9 and 10) • The established DT O-O aggregation hierarchy model shows how the DT objects information will be leveraged and aggregated from the bottom up to the system level • The DT O-O hierarchy model provides a blueprint of how the DTs within the DT solution can be reused, combined, shared, and aggregated, which facilitates the task in the development stage
4. Development	- A bearing PdM DT class is developed to estimate degradation onset for bearings. • A four-model approach using four statistical features is developed to detect degradation onset in each bearing • Following the guidelines of the DT O-O hierarchy model, the bearing PdM DT class is instantiated for the 4 bearings in each pump - Information is aggregated from the bottom up to the pump system level using aggregation functions
5. Testing	- The DT solution has been evaluated based on historical data. - Failure tests are simulated by sequentially inputting vibration measurements to the DT and recording the health state output after each measurement - The DT solution makes an accurate final health state classification in all eight experimental validation cases. Using the off-line data and analysis, the DT solution is validated to meet the manufacturing need identified in the planning stage

Outputs of the design stage for the case study are shown in Table 5.

5) STAGE 5: TESTING

The testing stage of this case study requires evaluating the proposed DT solution based on the historical data collected from the pump system. Historical data from three pump run-to-failure tests are available for analysis and validation of the DT solution by sequentially inputting vibration measurements to the DT and recording the health state output after each measurement. Performance of the DT solution can be assessed based on the final health state classification as well as the time at which degradation onset is estimated to have occurred (if applicable). The DT final health state classification can be compared with the true health state of the pump bearing, which was recorded at the conclusion of each run-to-failure test. Table 4 summarizes the performance metrics from validation testing.

When compared to the true final health state of each bearing, the DT solution makes an accurate final health state classification in all experimental validation cases. In each test, the PdM DT and accompanying models correctly identified the bearings that were experiencing degradation within the system. It is also notable that degradation onset was identified well before the system failure that prompted the end of each test.

After testing and validating the DT solution, it can be deployed on the real system. Compared against the do-nothing alternative, the developed DT solution increases the availability of the system by avoiding unplanned downtime. Early detection of degradation onset through the developed DT solution helps the management to determine the condition of the equipment and understand what might be wrong. Degradation onset was detected 92 hours and 40 mins before the end of test 1, 45 hours and 30 mins before the end of test 2, and 14 hours and 30 min before the end of test 3. These detections give the management sufficient time to repair the defective parts of the system before complete degradation and thus increase the availability of the system.

Outputs of the design stage for the case study are shown in Table 5.

C. OUTLOOK

The SDLC-based methodology presented in this paper details the off-line portions of the DT lifecycle for successful handoff to the on-line deployment and maintenance phase. In the on-line deployment and maintenance phase, the qualified off-line solution is deployed, used, continuously evaluated, and maintained [8]. The deployment includes the integration of the DT solution into the system where it uses run-time data from its operation environment to assess the system and make recommendations. Continuous maintenance and evaluation are required for continued effective use of the DT solution. Maintenance includes determining if and when the DT solution should be updated. Updates could be implemented on-line (e.g., model tuning) or might require the DT solution to be taken off-line (e.g., model rebuilding). DT solution maintenance is an often difficult and underappreciated area in the DT lifecycle [8].

The case study presented in this paper uses only vibration data in a univariate model that is less comprehensive than a multivariate model, and thus has limitations for a conditioning monitoring DT. There is often more than just one factor at play and a univariate model is unable to take other factors into account. Although using only vibration data may introduce limitations within a condition monitoring DT, we chose this example for the sake of simplicity in illustrating the steps of the SDLC-based DT development methodology, which is the focus of this paper. A more detailed case study is left for future work.

V. CONCLUSION

DT technology has gained wide attention from both academia and industry as a key game changer in advancing Smart Manufacturing and Industry 4.0 initiatives. A significant advantage of DT technology is the use of data gathered from different dimensions in the manufacturing value chain to derive time-sensitive decisions. Example applications of DT technology include evaluating the current condition or health state of assets and processes, predicting future behaviors, refining or reconfiguring control, and optimizing operations. Key challenges that impede the application of DT technology in manufacturing include data quality issues such as the integration and organization of data from different sources, as well as the problem of assessing the DT design and development throughout its lifecycle. Most of today's DT development approaches are ad-hoc where DTs are expected to be used once or a few times within a limited timeframe associated with a particular project. Consequently, there is a need for a systematic and unified DT methodology where DTs could be used/reused over time in multiple coordinated applications.

Motivated by this need, this paper introduces an SDLC-based DT design and development methodology that details the off-line activities of requirements analysis, design, development, and testing of DT solutions for successful handoff to on-line deployment. The methodology builds on our novel baseline framework that includes a DT definition and O-O specification to address important aspects of requirements of DT solutions in operation today, in the near future, and over the long-term [8]. Based on this baseline framework, the proposed methodology derives implementable DT solutions while taking into account: (1) the specificity of DT characteristics and requirements, (2) understanding the manufacturing context in which the DTs will operate, and (3) the object-oriented aspects required to achieve DT capabilities of scalability, reusability, interoperability, interchangeability, and extensibility.

The main benefit of the proposed DT design and development methodology is that it serves as a procedure for good practice in DT development as it provides practitioners with guidelines across the DT lifecycle to envision, design, develop, and test extensible DT frameworks.

Planned future work includes the development of a universal data communication infrastructure that supports DT frameworks throughout their lifecycle while promoting DT capabilities of scalability, reusability, interoperability, interchangeability, extensibility verification and validation, and maintainability.

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] H. Lasi, P. Fettke, H. G. Kemper, T. Feld, and M. Hoffmann, "Industry 4.0," *Bus. Inf. Syst. Eng.*, vol. 6, no. 4, pp. 239–242, 2014.

[2] 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.

[3] A. Parrott and L. Warshaw, *Industry 4.0 and the Digital Twin*. Deloitte Univ. Press, 2017, pp. 1–17. [Online]. Available: https://scholar.google.com/scholar?hl=en&as_sdt=0,23&q=industry+4.0+and+the+Digital+Twin.+Deloitte+University.+Press

[4] M. Grieves and J. Vickers, "Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems," in *Transdisciplinary Perspectives on Complex Systems*, F. J. Kahlen, S. Flumerfelt, and A. Alves, Eds. Cham, Switzerland: Springer, 2017, doi: 10.1007/978-3-319-38756-7_4.

[5] 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.

[6] Q. Qi, F. Tao, T. Hu, N. Anwer, A. Liu, Y. Wei, L. Wang, and A. Y. C. Nee, "Enabling technologies and tools for digital twin," *J. Manuf. Syst.*, vol. 58, pp. 3–21, Jan. 2021.

[7] 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.

[8] J. Moyne, Y. Qamsane, E. C. Balta, I. Kovalenko, J. Faris, K. Barton, and D. M. Tilbury, "A requirements driven digital twin framework: Specification and opportunities," *IEEE Access*, vol. 8, pp. 107781–107801, 2020.

[9] M. Gorrod, "The software development lifecycle," in *Risk Management Systems* (Finance and Capital Markets Series). London, U.K.: Palgrave Macmillan, 2004, doi: 10.1057/9780230510296_4.

[10] J. Broad, "System development life cycle (SDLC)," in *Risk Management Framework: A Lab-Based Approach to Securing Information Systems*, J. Broad, Ed. Boston, MA, USA: Syngress, 2013, pp. 39–45, doi: 10.1016/B978-1-59749-995-8.00005-3.

[11] *ISO/IEC/IEEE International Standard-Systems and Software Engineering–Vocabulary*, ISO/IEC/IEEE Standard 24765, 2017.

[12] A. R. Otero, *Information Technology Control and Audit*, 5th ed. Boca Raton, FL, USA: CRC Press, 2018, doi: 10.1201/9780429465000.

[13] D. Cearley, N. Jones, D. Smith, B. Burke, A. Chandrasekaran, C. Lu, and K. Panetta, "Gartner top 10 strategic technology trends for 2020–smarter with gartner," *Gartner*, Oct. 2019. [Online]. Available: https://www.gartner.com/smarterwithgartner/gartner-top-10-strategic-technology-trends-for-2020/

[14] B. Marr. (Nov. 2019). *The 5 Biggest Technology Trends Disrupting Engineering and Design in 2020*. [Online]. Available: https://www.forbes.com/sites/forbes-personal-shopper/2020/06/08/best-gaming-monitors-for-2020/#2488beb93288

[15] E. H. Glaessgen and D. S. Stargel, "The digital twin paradigm for future NASA and US Air Force vehicles," in *Proc. Collection Tech. Papers-AIAA/ASME/ASCE/AHS/ASC Struct., Struct. Dyn. Mater. Conf.*, 2012, p. 1818.

[16] Q. Liu, H. Zhang, J. Leng, and X. Chen, "Digital twin-driven rapid individualised designing of automated flow-shop manufacturing system," *Int. J. Prod. Res.*, vol. 57, no. 12, pp. 3903–3919, Jun. 2019.

[17] J. Leng, Q. Liu, S. Ye, J. Jing, Y. Wang, C. Zhang, D. Zhang, and X. Chen, "Digital twin-driven rapid reconfiguration of the automated manufacturing system via an open architecture model," *Robot. Comput.-Integr. Manuf.*, vol. 63, Jun. 2020, Art. no. 101895.

[18] W. Luo, T. Hu, Y. Ye, C. Zhang, and Y. Wei, "A hybrid predictive maintenance approach for CNC machine tool driven by digital twin," *Robot. Comput.-Integr. Manuf.*, vol. 65, Oct. 2020, Art. no. 101974.

[19] D. Jones, C. Snider, A. Nassehi, J. Yon, and B. Hicks, "Characterising the digital twin: A systematic literature review," *CIRP J. Manuf. Sci. Technol.*, vol. 29, pp. 36–52, May 2020.

[20] A. Fuller, Z. Fan, C. Day, and C. Barlow, "Digital twin: Enabling technologies, challenges and open research," *IEEE Access*, vol. 8, pp. 108952–108971, 2020.

[21] 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," *Journal of Intelligent Manufacturing*, vol. 31, pp. 1313–1337, Nov. 2019.

[22] *Automation Systems and Integration–Digital Twin Framework for Manufacturing*, International Organization for Standardization, Geneva, Switzerland, Standard ISO/DIS-23247, 2020. [Online]. Available: https://www.https://www.iso.org/standard/75066.html

[23] G. Shao and M. Helu, "Framework for a digital twin in manufacturing: Scope and requirements," *Manuf. Lett.*, vol. 24, pp. 105–107, Apr. 2020.

[24] 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/PI40/Redaktion/EN/Downloads/Publikation/Details_of_the_Asset_Administration_Shell_Part1_V1.html

[25] M. Platenius-Mohr, S. Malakuti, S. Grüner, and T. Goldschmidt, "Interoperable digital twins in IIoT systems by transformation of information models: A case study with asset administration shell," in *Proc. 9th Int. Conf. Internet Things*, 2019, pp. 1–8.

[26] 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.

[27] J. Cheng, H. Zhang, F. Tao, and C.-F. Juang, "DT-II: Digital twin enhanced industrial Internet reference framework towards smart manufacturing," *Robot. Comput.-Integr. Manuf.*, vol. 62, Apr. 2020, Art. no. 101881.

[28] *About the Unified Modeling Language Specification Version 2.5.1*, Object Management Group (OMG), Needham, MA, USA, 2017.

[29] A. Dennis, B. Wixom, and D. Tegarden, *Systems Analysis and Design: An Object-Oriented Approach With UML*. Hoboken, NJ, USA: Wiley, 2015.

[30] S. Balaji and M. S. Murugaiyan, "Waterfall vs. V-Model vs. Agile: A comparative study on SDLC," *Int. J. Inf. Technol. Bus. Manage.*, vol. 2, no. 1, pp. 26–30, 2012.

[31] P. Isaias and T. Issa, *High Level Models and Methodologies for Information Systems*. New York, NY, USA: Springer, 2015, doi: 10.1007/978-1-4614-9254-2.

[32] *Common Automation Device. Profile Guideline*, International Electrotechnical Commission, document IEC TR 62390, 2005.

[33] Y. Demchenko, C. de Laat, and P. Membrey, "Defining architecture components of the big data ecosystem," in *Proc. Int. Conf. Collaboration Technol. Syst. (CTS)*, May 2014, pp. 104–112.

[34] C. Gehrmann and M. Gunnarsson, "A digital twin based industrial automation and control system security architecture," *IEEE Trans. Ind. Informat.*, vol. 16, no. 1, pp. 669–680, Jan. 2020.

[35] J. Leng, S. Ye, M. Zhou, J. L. Zhao, Q. Liu, W. Guo, W. Cao, and L. Fu, "Blockchain-secured smart manufacturing in industry 4.0: A survey," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 51, no. 1, pp. 237–252, Jan. 2021.

[36] H.-N. Dai, H. Wang, G. Xu, J. Wan, and M. Imran, "Big data analytics for manufacturing Internet of Things: Opportunities, challenges and enabling technologies," *Enterprise Inf. Syst.*, vol. 14, nos. 9–10, pp. 1279–1303, Nov. 2020.

[37] S. Kadre and V. R. Konasani, *Practical Business Analytics Using SAS: A Hands-on Guidele*. Berkeley, CA, USA: Apress, 2015.

[38] J. R. Moyne, "Method and apparatus for optimizing profit in predictive systems," U.S Patent 9 582 828, Feb. 28, 2017.

[39] J. Moyne, "Bringing subject matter expertise and analytics together in solutions for microelectronics smart manufacturing: Consensus, domains, and approaches," in *Proc. 31st Adv. Process Conf. (APC)*, San Antonio, TX, USA, 2019, pp. 1–29.

[40] A. Egyed, "UML/analyzer: A tool for the instant consistency checking of UML models," in *Proc. 29th Int. Conf. Softw. Eng. (ICSE)*, May 2007, pp. 793–796.

[41] J. Iskandar, J. Moyne, K. Subrahmanyam, P. Hawkins, and M. Armacost, "Predictive maintenance in semiconductor manufacturing," in *Proc. 26th Annu. SEMI Adv. Semiconductor Manuf. Conf. (ASMC)*, May 2015, pp. 384–389.

[42] *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/

[43] Y. Qamsane, A. Tajer, and A. Philippot, "Towards an approach of synthesis, validation and implementation of distributed control for AMS by using events ordering relations," *Int. J. Prod. Res.*, vol. 55, no. 21, pp. 6235–6253, Nov. 2017.

[44] R. Drechsler, *Formal System Verification. State-of the-Art and Future Trends*. Cham, Switzerland: Springer, 2018, doi: 10.1007/978-3-319-57685-5.

[45] M. Borg, C. Englund, K. Wnuk, B. Duran, C. Levandowski, S. Gao, Y. Tan, H. Kaijser, H. Lönn, and J. Törnqvist, "Safely entering the deep: A review of verification and validation for machine learning and a challenge elicitation in the automotive industry," 2018, *arXiv:1812.05389*. [Online]. Available: http://arxiv.org/abs/1812.05389

[46] J. Lee, "Bearing data set," IMS, NASA Ames Prognostics Data Repository, Rexnord Tech. Services, Univ. Cincinnati, Cincinnati, OH, USA, Tech. Rep., 2007. [Online]. Available: https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/?fbclid=IwAR14VMB3ZlIbAhkfdZIkZv5EL-fCYtuq1GQyKuQOZEI1TivOBmWdTJcCXyQ

[47] H. Qiu, J. Lee, J. Lin, and G. Yu, "Wavelet filter-based weak signature detection method and its application on rolling element bearing prognostics," *J. Sound Vib.*, vol. 289, nos. 4–5, pp. 1066–1090, Feb. 2006.

[48] J. Wang, L. Ye, R. X. Gao, C. Li, and L. Zhang, "Digital twin for rotating machinery fault diagnosis in smart manufacturing," *Int. J. Prod. Res.*, vol. 57, no. 12, pp. 3920–3934, Jun. 2019.

[49] Y. Lei, *Intelligent Fault Diagnosis and Remaining Useful Life Prediction of Rotating Machinery*. Oxford, U.K.: Butterworth-Heinemann, 2016.

[50] S. Ferguson, E. Bennett, and A. Ivashchenko, "Digital twin tackles design challenges," *World Pumps*, vol. 2017, no. 4, pp. 26–28, Apr. 2017.

[51] D. Buono, D. Siano, E. Frosina, and A. Senatore, "Gerotor pump cavitation monitoring and fault diagnosis using vibration analysis through the employment of auto-regressive-moving-average technique," *Simul. Model. Pract. Theory*, vol. 71, pp. 61–82, Feb. 2017.

YASSINE QAMSANE (Member, IEEE) received the M.Sc. 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.

JAMES MOYNE (Member, IEEE) received the B.S.E.E. and B.S.E. degrees in mathematics, and the M.S.E.E. and Ph.D. degrees from the University of Michigan, Ann Arbor, in 1984, 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. He is also 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 the CTO of the Integrated Measurement Association.

MAXWELL TOOTHMAN received the B.S. degree in mechanical engineering from the Georgia Institute of Technology, Atlanta, GA, USA, in 2018. He is currently pursuing the Ph.D. degree in mechanical engineering from the University of Michigan, Ann Arbor, MI, USA.

His current research interests include smart manufacturing, data science, signal processing, and predictive maintenance for continuous manufacturing systems.

ILYA KOVALENKO (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. and Ph.D. degrees in mechanical engineering from the University of Michigan, Ann Arbor, MI, USA, in 2018 and 2020, respectively. He is currently a Research Fellow with the Department of Mechanical Engineering, University of Michigan. His current research interests include control theory and application, with a focus on multi-agent systems, advanced manufacturing, cyber-physical systems, and robotics.

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.

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.

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.

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, 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 Achievement Award, and the 2017 ASME Dynamic Systems and Control Young Investigator Award.