A Unified Digital Twin Framework for Real-time Monitoring and Evaluation of Smart Manufacturing Systems

Yassine Qamsane[1], Chien-Ying Chen[2], Efe C. Balta[1], Bin-Chou Kao[2], Sibin Mohan[2], James Moyne[1], Dawn Tilbury[1] and Kira Barton[1]

Abstract— **Digital Twin (DT) is one of the key enabling technologies for realizing the promise of Smart Manufacturing (SM) and Industry 4.0 to improve production systems operation. Driven by the generation and analysis of high volume data coming from interconnected cyber and physical spaces, DTs are real-time digital images of physical systems, processes or products that help evaluate and improve business performance. This paper proposes a novel DT architecture for the real-time monitoring and evaluation of large-scale SM systems. An application to a manufacturing flow-shop is presented to illustrate the usefulness of the proposed methodology.**

I. Introduction

Current trends in information and communication technologies gave rise to Cyber-Physical Systems (CPS) which manage interconnected physical assets and computational capabilities of a system. These transformative technologies enable the possibilities promised by Smart Manufacturing (SM) and industry 4.0, to track and use process data up and down the supply chain. More effective autonomous smart factories, with the ability for self-management and self-optimization, can be achieved through the large interconnection of CPS. This class of systems are able to communicate, perceive their environment, interpret information, and act on the physical world [1]. Despite these prospects, modeling and real-time control methods of CPS still face two main challenges: (i) lack of models that can accurately replicate the dynamics of physical systems while incorporating real-time manufacturing data; and (ii) lack of real-time optimization and control algorithms to generate effective on-line production control actions based on real-time manufacturing data and performance prediction [2]. We use Digital Twin (DT) technology which bridges the physical and digital worlds to handle the first challenge and provide a means to address the second one. DT refers to a digital equivalent of physical products, assets, processes or systems. It is used for describing and modeling the corresponding physical counterpart in a digital manner [3]. It combines modeling, simulation, and emulation technologies with other analytics to better understand aspects of current manufacturing operations (*e.g.,* health monitoring) or to predict aspects of future behaviors of the manufacturing system (*e.g.,* predictive maintenance) [4], [5]. The adoption of the Industrial Internet of Things (IIoT) technologies has played an important role in making DTs cost-effective to implement. IIoTs enable ubiquitous connectivity that allows systems to report their status, working conditions, and ambient environments to the DTs so that the latter can remain in lock-step with their physical counterparts. This capability allows the DTs to provide an up-to-date representation of the SM system.

The main contribution of this paper is to introduce a unified DT modeling framework for SM systems. The role of the DT framework is to provide a real-time extensible global view of a manufacturing system by deploying multiple DTs at multiple levels of the automation pyramid of the International Society of Automation ISA–95 [6]. The DT framework is used within the Software-Defined Control (Fig. 1), where it operates with a set of applications and a decision maker to monitor, control, predict, and re-configure (as necessary) complex production processes. The DTs within the framework are organized in a class structure to provide capabilities that are important to the optimization of the manufacturing environment; instances are stored in a DT pool with a DT manager handling the communication between DT instances and with the decision maker. Four classes that are important to most manufacturing environments are defined.

The rest of this article is structured as follows. Section II provides background on the research related to this work. Section III introduces the DT platform providing details of its components and how they are related. Section IV describes four DT classes that are typically needed to build a real-time global view of a given SM system. Section V demonstrates the usefulness of the proposed approach through a case study using a manufacturing flow-shop example. Section VI summarizes the contributions of this paper and presents some challenges related to the development of DTs.

II. Background

A. Related Work

Digital modeling and simulation technologies have become widely used in many engineering domains thanks to the ubiquitous connectivity of devices and the amount of data being moved between these devices or through the cloud. DT-related methodology and technology are being applied in different industrial fields and are showing great potential. Industrial applications of DTs mainly focus on the areas of product design, production, Prognostic and Health Management (PHM), and human-machine interaction, where DTs have shown superiority over traditional solutions [4],

*This work was supported by NSF 1544678.

[1]Y. Qamsane, E. Balta, J. Moyne, D. Tilbury and K. Barton are with the Department of Mechanical Engineering, University of Michigan, Ann Arbor, MI 48109, USA. {yqamsane,baltaefe,moyne,tilbury,bartonkl}@umich.edu

[2]C.Y. Chen, B.C. Kao, and S. Mohan are with the Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA. {cchen140,bkao2,sibin}@illinois.edu

[7], [8], [9], [10], [11], [12]. Tao et al. presented a DT-based method for product design connecting the physical and virtual products to improve product customization [8]. Schleich et al. highlighted a DT reference model that enables quality evaluation to ensure that the required geometrical features of the product are satisfied regardless of the presence of geometrical part deviations [9]. The DT concept has been introduced into the production floor to make manufacturing systems more dependable and flexible. A DT conceptual framework was developed in [10] for monitoring and optimizing physical manufacturing workshops based on context data. Bottani et al. developed a cyber-physical automated guided vehicle DT to improve material handling operation in Job-Shop manufacturing systems [11]. In the field of PHM, DTs have been used to predict the time at which a system will no longer operate as envisioned and meet the desired performance. Tao et al. proposed a DT method that depicts geometry, physics, and behavior of an equipment to detect environment disturbances, potential faults in the equipment and defects in the models [13]. A structural modeling concept, the airframe DT, was proposed in [12] to design, maintain, reduce uncertainty, and improve robustness of airframes. Some studies have also investigated the connections between humans and DTs in production area. A DT approach that enables the communication and coordination of operators with the production system was proposed in [14]. Such DTs facilitate the integration of humans in the decision-making process for self-controlling systems.

Most of these approaches focus on using DTs to solve a particular problem (*e.g.,* equipment health monitoring, product design, system design, quality evaluation). However, a unified DT framework to be used in multiple coordinated applications (PHM, scheduling/dispatching, rerouting, self-organization and optimization, *etc.*) is still needed. In this paper, we propose a unified DT platform that operates within a Software-Defined Control (SDC) framework for flexible control reconfiguration of smart manufacturing systems. The proposed DT platform uses historical and real-time data to provide the SDC controller with a centralized view that is used to provide comprehensive DT capabilities such as to predict and detect anomalies, monitor equipment health, monitor production in real-time, optimize scheduling/dispatching, improve the system self-organizing and learning, and propose novel control plans.

There exist some commercial frameworks that use plant simulation software to understand the impact of performed actions or boundary conditions in production systems. Ultimately these frameworks allow for more informed decision making based on operations visualization. For the sake of brevity, we consider Tecnomatix from Siemens, AutoMod from Applied Materials, and Emulate3D from Rockwell automation. These software solutions focus on simulating the physical system and use that simulation to predict behavior, whereas our approach subsumes their capabilities and addresses the more general problem of simulation/emulation of a physical system or a process. Our solution does not require the physical simulation of the system. Thus, solutions

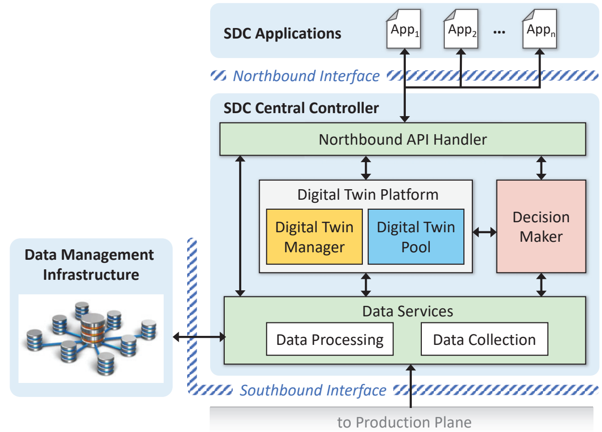

Fig. 1: Overview of the SDC framework.

provided by plant simulation software could be incorporated into our DT framework, but cannot represent the entire scope of DTs within this framework.

B. Software-Defined Control

The SDC is a framework that enables flexible control of smart manufacturing systems [15], [16]. SDC employs a global view of the SM system, including physical components and cyber components to help improve manufacturing productivity, efficiency, quality, and security. In building this global view, the SDC consolidates data at the control, operations, and business levels to support operations management with control reconfiguration recommendations. The SDC consists of a set of centralized data management infrastructures, a central controller, and a set of applications (Fig. 1). The data management infrastructures are used to store data consolidated from the plant floor, operations management, and business levels. The central controller, the key piece of SDC, uses the consolidated data to generate a real-time global view of the manufacturing system. Applications such as anomaly detection, rerouting, planning, *etc.*, use this global view to support the central controller in its tasks. The separation of the applications plane from the central controller allows easier incorporation of third-party applications and algorithms without affecting the system design. The information flow within the central controller is supported by the SDC interfaces. A southbound interface is used for the collection and transformation of plant floor data by means of unified and standardized protocols (*e.g.,* MTConnect and OPC UA) prior to its use by the DTs and data analytics. A northbound interface enables communication between SDC applications and the central controller. An eastbound interface is used for communications with the MES and the ERP, and allows system administrators to configure and manage the central controller. The central controller consists of a decision maker, data services and a DT platform. The decision maker determines the recommendations to transmit to the MES. Data services manage the information flow to and within the central controller and offer two services: (i) the data collection service used for requesting specific data to be collected (from the plant floor) by the southbound interface, and (ii) the data processing service used for accessing data

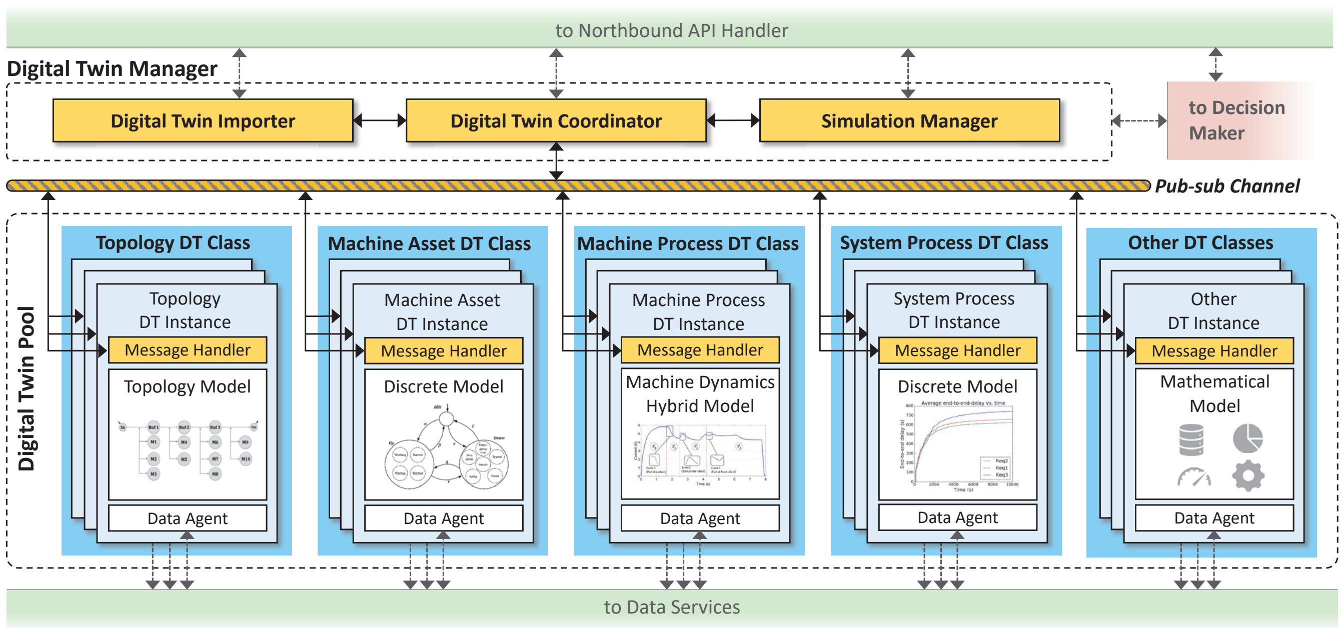

Fig. 2: Overview of the DT platform inside the SDC. A DT class can have multiple DT instances deployed in the DT pool.

through the southbound interface and the database. The DT platform hosts DTs of the system that are used to identify the changing conditions and predict the effect of suggested configurations before their deployment.

III. The Digital Twin Platform

The proposed DT platform provides the SDC framework with the capability to construct aspects of a global view of the SM system that could be used for multiple purposes (*e.g.,* real-time monitoring, anomaly detection/prediction, real-time optimization, *etc.*). The DT platform provides real-time modeling of the manufacturing system at multiple levels of the ISA-95 standard. Different model types are incorporated into the DT platform. Models are built at the machine and system levels considering the continuous, discrete, and hybrid behaviors at each level. They also merge physics-based and data-driven knowledge to improve analysis.

The architecture of the DT platform is shown in Fig. 2. It mainly consists of a DT pool and a DT manager. The DT pool hosts instances of different classes of DTs interconnected via a publish-subscribe (pub-sub) infrastructure[1]. The DT manager manages the DT mission by handling initialization of the DT pool, coordinating the pub-sub communication across DT instances, handling the application requests from the *Northbound API Handler*, and coordinating the interaction with the decision maker.

A. Digital Twin Pool

As mentioned earlier, DTs are digital equivalents of physical products, assets, processes, and systems, which we refer to as *DT classes*. A DT class can be instantiated multiple times and be deployed in the DT platform as *DT instances*. For example, the machine asset DT class can have multiple machine asset DT instances that are associated with each machine in a SM system. In the context of SDC, the DT instances are building blocks that enable an SDC user to customize the functionality of the DT platform based on a particular need for the SM system. We refer to the set of DT instances deployed in the DT platform as the DT pool.

Each DT instance is composed of a mathematical model, a data agent and a message handler, as shown in Figs. 2-3.

1) Mathematical Model: the mathematical model is the core of a DT instance. It is an abstract representation (e.g., physics-based) that describes the behavior of the physical asset or a process. Experimental data are used to validate and continuously update the model for maximizing the applicability and value of the digital representation. These data include data generated by the plant (through the data agent) or the other DT instances (through the message handler).

2) Data Agent: each DT instance has a data agent that connects it to the *data services*. It allows a DT instance to acquire real-time (as well as historical) data via a standardized interface. This interface also enables the flexibility to include any plant data source from the *data services* if needed.

3) Message Handler: DT instances and the DT manager are interconnected via a pub-sub infrastructure in the DT pool. Depending on the intended goal, a DT instance can work either individually or cooperatively (*i.e.,* the functionality of a DT could depend on data generated by another DT) to deliver the intended functionality. Each DT instance is embedded with a message handler to process the messages publishing to and/or subscribing from other DT instances. Details regarding the pub-sub are discussed in Section III-C.

B. Digital Twin Manager

The DT manager coordinates the required interactions between the DT instances and other SDC components, *i.e.,* the SDC applications and the decision maker. Tasks handled by the DT manager include responding to requests

[1]Publish-subscribe pattern is an effective communication paradigm where publisher's messages are distributed to the corresponding subscribers.

of information exchange between the DT instances (via the pub-sub infrastructure) and with the other SDC components, querying devices real-time status, and initiating simulation tasks. The DT manager comprises *(a)* a DT importer, *(b)* a DT coordinator and *(c)* a simulation manager.

1) Digital Twin Importer: a user of the DT platform can customize the system by defining and/or initializing a set of DT instances that provide a set of desired functions for the SM system. Users can dynamically import/remove a DT instance into/from the DT pool via the DT importer. In addition, the DT class, provided functionality, and data dependencies (topics to be published and subscribed) are registered with the DT coordinator when a DT instance is imported and initialized.

2) Digital Twin Coordinator: the DT coordinator is the access point to the DTs at runtime. It handles data sharing between the DTs and the requests sent by the SDC applications and decision maker. Depending on the type of requests, the DT coordinator routes the corresponding tasks and data to the corresponding DTs. It acts as a message broker (*e.g.,* a ActiveMQ broker) that handles the pub-sub data flow across the DT instances. A DT publishes data whenever it determines that an update of the data is needed. A DT can also register with the message broker to listen to a set of data published by other DTs. By doing so, the subscriber DT is guaranteed to be notified and obtain the latest data whenever there is an update from a publisher DT. In the case a new DT is imported, the latest data for its subscribed topics are provided by the DT coordinator once the new DT is verified and started.

3) Simulation Manager: in addition to the real-time modeling capabilities, some DT classes also have simulation capabilities that allow the corresponding DT instances to predict the machine and system future states. By using a technology such as Functional Mock-up Interface (FMI) [17], it is possible to run simulations over multiple DT instances. This capability is useful for validating a manufacturing plan change, predicting manufacturing performance, evaluating across different plant configurations, *etc.* A simulation manger is used in the DT platform to manage the simulation tasks requested by SDC applications or the decision maker. On receiving a simulation request, the simulation manager parses the request and distributes it to the corresponding DT instances. Simulation results are collected by the simulation manager afterwards and returned to the requester.

C. Publish-Subscribe Infrastructure

In the pub-sub infrastructure, a topic represents a type of information for which data is generated by instances of one or more DT classes. As in a conventional topic-based pub-sub infrastructure, the data published for a topic (in the form of messages) is distributed to the DT instances that subscribe to it. A topic can be in any form depending on the design and purpose of the publisher DT instance. It can be a single value (*e.g.,* a variable that represents system throughput), a vector (*e.g.,* a set of variables that models the dynamics of a machine), or a complex structure (*e.g.,* a series of nodes and edges modeling the plant's layout). A DT instance can publish to multiple topics and there can also be multiple DT instances publishing to the same topic.

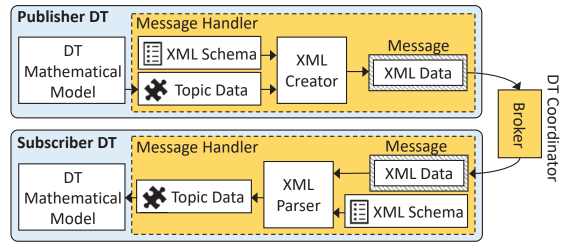

Fig. 3: Data flow between the publisher and subscriber DTs.

In the DT pool, the Extensible Markup Language (XML) format is used to encapsulate data in a message published to a topic. XML supports a validation scheme in which a predefined XML schema that describes the structure of an XML format can be used to validate the correctness of the given XML data against such a format (*i.e.,* whether the XML data consists of correct type and the number of elements and attributes) as depicted in Fig. 3. The XML schema for a topic is provided to the DT coordinator by the publisher DT instance that first successfully creates the topic. The coordinator uses the schema to validate that a DT instance is correctly registering for publishing to an existing topic. It is also used when subscribing to the topic to check if the schema matches its expectation.

D. Synchronization

1) Time Synchronization: time synchronization is vital to having a precise time reference in the SDC framework. It can be achieved by using a time synchronization protocol such as Precision Time Protocol (PTP). PTP enables a time accuracy in the sub-microsecond range in the DT manager, DT instances and data services. Data shared in the SDC framework is time-stamped so that it can be referred based on the precise synchronized time.

2) Data Synchronization: in SDC, two aspects of data synchronization are considered: *(i)* the synchronization between the DT models and the physical assets or processes and *(ii)* the synchronization between DT instances.

In SDC, *data services* in the central controller allow data collection and storage through a data management infrastructure. A DT instance can utilize its data agent to obtain the collected data from the data services, allowing a DT instance to synchronize the implemented model with its associated physical asset or process. Synchronization between DT instances is ensured by the topic-based data sharing mechanism in the pub-sub architecture. When new data is generated and published by a DT instance, the coordinator distributes it (wrapped in a topic-based message) to the corresponding subscriber DT instances promptly. As a result, the information for a topic across the associated DT instances is consistent and the states of the DT models are synchronized.

IV. Digital Twin for Global View

As noted in Section III, different classes in the DT platform can be supported depending on the needs of the manufacturing environment. In this section we introduce four classes of DTs that would often be used to construct a real-time global view of a SM system in order to address common manufacturing issues such as throughput and quality optimization, and reduction of cost and variability. Note that a manufacturing system might include instances of other DT classes depending on particular needs. The proposed DT platform is flexible enough to allow including other DTs.

A. Topology Digital Twin

The topology DT is a representation of the physical layout of the system in real-time[2]. Such a representation allows to accurately and timely track the availability of machines and transport systems and their connectivity. This can be used by apps such as flexible rerouting to find the best routing paths according to current system workload. The topology DT is a directed graph where each node (vertex) is a component that processes parts and each link (edge) is a physical path on which parts can be transported from one node to another. For instance, a conveyor that carries parts from a cell to another is a path (edge) that links the two cells (nodes).

We divide the manufacturing equipment into nodes and links. Each work station (*e.g.,* milling/turning machine, assembly/welding robot, quality inspection, buffer, *etc.*) in the manufacturing system is a single node, whereas a possible material flow between two nodes (*e.g.,* conveyor, AGV, gantry) is represented by a directed link. Links could be unidirectional or bidirectional.

Formally, we define the topology DT as a tuple $T = (N, L, In, Out, \Delta)$ where:

- N is a finite set of nodes;
- L is a set of links that connect some of these nodes;
- In is the set of material flow inputs;
- Out is the set of material flow outputs;
  The system may consist of unique or multiple material flow inputs/outputs. In and Out are defined as nodes, *i.e.,* $\{In, Out\} \in N$.
- $\Delta : N \times L^* \rightarrow N$ is a function that defines the flow transitions, where L^* denotes the set of all finite link concatenations in L. An element $l \in L^*$ is a sequence of links. The length of a sequence l is given by the number of its involved links. A sequence $l \in L^*$ consists of at least 1 link.

The construction of the topology DT consists of first enumerating the equipment of the system to verify the set of nodes and links. Second, the interconnections between the nodes and links are defined as flow transitions to depict the relationships between the machines/stations and transport systems. Eventually, corresponding plant floor data are synchronized with the DT through the data agent to provide a real-time replica of the physical layout.

[2]Real-time is used here to indicate that the DT representation is updated with sufficient promptness so that appropriate decisions can be made based on the assumption that the DT is an up-to-date representation of the system.

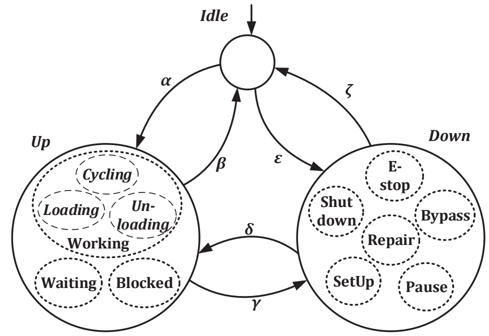

Fig. 4: The discrete model in the machine asset DT.

B. Machine Asset Digital Twin

The machine asset DT is a generic discrete model that provides access to the structure, behavior and working conditions of an individual manufacturing unit. We define a machine asset DT as a Finite State Machine (FSM) with 3 global states: *Idle*, *Up*, and *Down*. A machine could have multiple *Up* and *Down* sub-states as shown in Fig. 4. Transitions between these states could be event-driven or time-driven. Annotating edges, *i.e.,* $\{\alpha, \beta, \gamma, \delta, \varepsilon, \zeta\}$, refer to the occurrence of an event or the elapse of some time. It is also possible to have transitions between states inside the *Up* and *Down* superstates. More details could be found in [18].

To build a machine asset DT, the Subject Matter Expert (SME) starts with verifying, from the states consolidated in the generic model of Fig. 4, the set of states that a machine has. Then, the transitions between these states are defined and depicted in the model. Ultimately, the model is synchronized with the machine data through the data agent to provide a real-time replica of the physical machine.

C. Machine Process Digital Twin

The processing environment provided by the machine (*e.g.,* when operating on a part) is captured by the machine process DT. For instance, Ordinary Differential Equations (ODEs) are used for modeling the continuous variables of interest, namely states, of the manufacturing unit in closed-loop. The generic model of non-linear ODEs are given by

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u}, t) \tag{1}$$

where, $\boldsymbol{x} \in \mathbb{R}^n$ is the state vector and $f(\boldsymbol{x}, \boldsymbol{u}, t)$ is the flow dynamics of the state according to $\boldsymbol{u} \in \mathbb{R}^m$, and time variable $t \in [0, \infty)$. Initial conditions of the dynamics are given as $\boldsymbol{x}(0) = \bar{\boldsymbol{x}}$, which is updated using the data agent at the time of initialization. Various dynamics can be lumped into a system of ODEs and represented as Eq. 1 without loss of generality. The DT is updated with data at a predefined frequency, which results in difference equations given by

$$\boldsymbol{x}_{k+1} = f(\boldsymbol{x}_k, \boldsymbol{u}_k, k) \tag{2}$$

where $\boldsymbol{x}_k \in \mathbb{R}^n$ is the discrete state variable at time step $k \in \mathbb{N}$, and $\boldsymbol{u}_k \in \mathbb{R}^m$ is the discrete control input at time step k. The models of the manufacturing unit may be predefined by an SME, or can be learned from the data streams. While learning the model from data is a non-trivial task, parameters

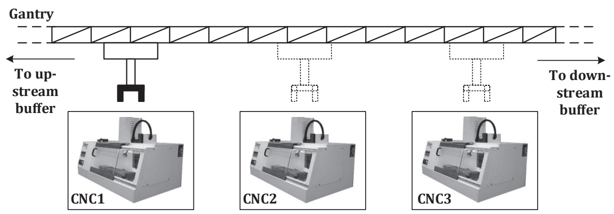

Fig. 5: The flow shop system example.

of a certain class of model can be identified using system identification and machine learning techniques [19].

While continuous dynamics are often utilized in machine process digital twins, discrete state-based models are often used to model the processing capability as a unit of work and then provide analysis so as to control or optimize the process with discrete actions such as changing process parameters each time a new part enters the machine. Model-based Process Control (MPC), virtual metrology, virtual sensing, sensor fusion, and model-based predictive maintenance are common techniques used in this domain [20], [21], [22].

D. System Process Digital Twin

The system process DT models the material moving in the SM system and provides insight on high-level process information. The outcomes of the system process DT are the system-level performance metrics such as system/cell cycle time, Work-In-Process (WIP), quality, and throughput.

Formally, the system process DT is defined as a tuple $P = (T, W, loc, pos)$ where:

- T is the topology model defined in the topology DT.
- W is a set of unique identifiers for all materials that will ever be seen in the system, *i.e.,* $W = \{w_i | i \in \mathbb{N}\}$.
- loc is a function, $loc : W \rightarrow N \cup L$, that gives the location of a material in the physical topology model.
- pos is a function, $pos : W \rightarrow \mathbb{N}$, that gives the actual position of a material in the location indicated by the function loc. For example, $loc(w_i) = B_1, pos(w_i) = 2$ indicates that the material w_i is at the second slot in the buffer B_1.

Using run-time data from the factory floor over a period of time via the data services, a series of discrete states and their transitions can be constructed. By analyzing the part positions across multiple states, the system-level performance metrics (*e.g.,* throughput, WIP) in a given time span can be estimated. Also, quality data for individual manufacturing units are attributed to the yield of the individual units, although additional considerations may be added.

V. Implementation and Evaluation

A. System description and DTs development

To illustrate the usefulness of the proposed DT framework, we use the conceptual flow shop system example of Fig. 5 with real CNC data. The system is comprised of 3 CNC machines (cnc_1, cnc_2, and cnc_3). Each CNC can mill different part features (f_1, f_2, and f_3). Parts are transported between the CNC machines and the downstream/upstream buffers through a gantry system.

$$\Delta(In, gantry) = cnc_1; \quad \Delta(cnc_1, gantry) = In$$
$$\Delta(cnc_1, gantry) = cnc_2; \quad \Delta(cnc_2, gantry) = cnc_1$$
$$\Delta(cnc_2, gantry) = cnc_3; \quad \Delta(cnc_3, gantry) = cnc_2$$
$$\Delta(cnc_3, gantry) = Out; \quad \Delta(Out, gantry) = cnc_3$$

The initial production plan is defined by the SME such that parts are moved to cnc_1, cnc_2, and cnc_3 to realize features f_1, f_2, and f_3, respectively. The following DTs are build for the real-time monitoring and reconfiguration of the system.

Topology DT: an SME constructs the initial topology DT by verifying the set of nodes and links in the system and their interconnection. The SME verifies that $L = \{gantry\}$ is the only bidirectional link to move material between the nodes $N = \{In, cnc_1, cnc_2, cnc_3, Out\}$. The flow transitions are defined by:

The corresponding plant floor data are synchronized with the DT through the data agent to provide a real-time replica of the physical layout as shown in Fig. 6.

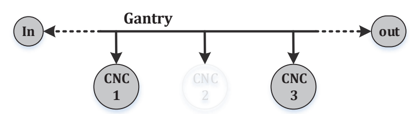

Fig. 6: The topology DT with cnc_2 omitted.

Machine Asset DTs: machine asset DTs are built to provide insight on the context and behavior of individual manufacturing units in real-time. For instance, the machine asset DT for cnc_1 is represented in Fig. 7. The cnc_1 has the capability of processing part features f_1 (state "$Cycle\_f_1$"), f_2 (state "$Cycle\_f_2$"), and f_3 (state "$Cycle\_f_3$"). a part that requires feature f_1 is detected and $tool_1$ is installed ($pf_1 \wedge tl_1$), the machine starts its "$Cycle\_f_1$" which takes the cycle time $\tau(C_1)$. After the elapse of $\tau(C_1)$, the machine ends its cycle (event "ec_1") and transitions to "$Idle$". In a similar way, the parts that require features f_2 and f_3 go to the states "$Cycle\_f_2$" with the cycle time $\tau(C_2)$ and "$Cycle\_f_3$" with $\tau(C_3)$, respectively. If the corresponding tool is not outfitted, a tool change is necessary to switch between processing f_1, f_2, and f_3. The setup times $\tau(S_1)$, $\tau(S_2)$, and $\tau(S_3)$ are associated to the setup states "$Setup_1$", "$Setup_2$", and "$Setup_3$", respectively. If a fault (ft) is detected, the machine goes to "$Down$" state. If the fault is cleared and the reset button is pressed ($\neg ft \wedge reset$), the machine transitions to the "$Idle$" state. Cycle times and setup times are estimated using historical data.

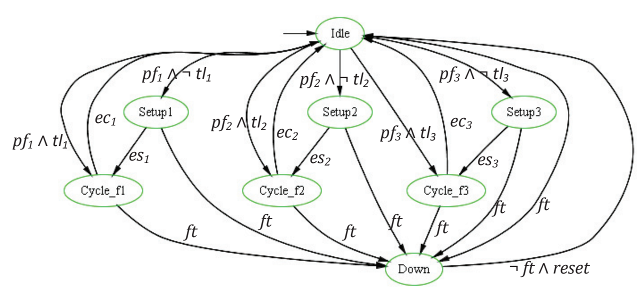

Fig. 7: Machine asset DT for cnc_1.

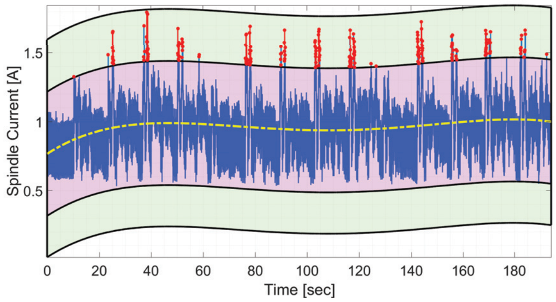

Fig. 8: Anomaly detection on the cnc_2 using the machine process DT. Dashed center line is a fitted curve on the measured spindle current. Anomalous peak spindle current measurement are shown with red marked points.

Machine Process DT: the machine process DT in this case study is tasked with modeling the spindle current of a CNC. Spindle current provides insight into the equivalent forces acting on the spindle. If there is excessive loading, the tool may break and damage the part in process. To identify the peaks in the spindle current and classify the ones that could cause tool breakage, an adaptive limit based anomaly detection scheme is implemented. Fig. 8 illustrates the adaptive limit based anomaly detection for tool breakage in cnc_2. An SME defines safety limits for the spindle current peaks, using heuristics and historical spindle current measurements available through the machine process DT. An anomaly detection app uses the spindle current data from the machine process DT and fits a fourth order polynomial using weighted regularized least squares (dashed center-line in Fig. 8). The inner limit signifies tool-wear for the spindle tool and the outer limit is the safety limit for tool breakage.

B. Anomaly Detection and Control Reconfiguration

An SDC anomaly detection app identifies the spindle current measurements that breach the inner limit (red markers in Fig. 8), and signals an anomaly prior to an actual tool breakage. Thus, a fault ("ft") event is triggered. Consequently, the cnc_2 is omitted in the topology DT of Fig. 6 indicating that the machine is not available. In the machine asset DT of cnc_2, the cnc_2 transitions to the "$Down$" state. This example illustrates the potential use of machine process DT to monitor process parameters and machine health for anomaly detection and performance monitoring.

As cnc_2 is no longer available, the SDC decision maker evaluates a cost function and feasibility constraints to reconfigure the manufacturing cell. An optimization module inside the decision maker uses the data from DTs to formulate and compute an optimization problem for optimal reconfiguration decisions. Since all CNCs in the system are capable of milling all three features, the possible number of combinations of feature assignments for the remaining two CNCs is $2^3 = 8$. Let $\boldsymbol{\phi}_j = (\mu(f_1), \mu(f_2), \mu(f_3))$ denote a possible assignment combination where $\mu(f_i)$ denotes which CNC is assigned for the milling of a particular feature f_i. Additionally, let $\lambda_i(\boldsymbol{\phi}) \in \mathbb{N}$ denote the number of cnc_i assignments in the combination $\boldsymbol{\phi}_j$, and $\Lambda_i = \{f_i | i \in \{1, 2, 3\}\}$ denote the corresponding features. Then a cost function for the manufacturing time of each assignment in the case study is given as:

$$J_T(\boldsymbol{\phi}_j) = \sum_{i=1}^{3} \tau_{\mu(f_i)}(C_i) + \begin{cases} \sum_{\forall k \in \Lambda_i} \tau_i(S_k), & \textbf{if } \lambda_i(\boldsymbol{\phi}) > 1 \\ 0 & otherwise \end{cases}$$

Additionally, define the quality function as the product of the yields of the machines in a given combination $\boldsymbol{\phi}_j$.

$$J_Q(\boldsymbol{\phi}_j) = \prod_{i=1}^{3} q_{\mu(f_i)},$$

where $q_{\mu(f_i)} \in (0, 1]$ denotes the yield of the CNC assigned for the feature f_i. Combining the two objectives, an optimization problem is formed as

$$\min_{\boldsymbol{\phi}} \; J(\boldsymbol{\phi}_j) = \alpha_0 J_T(\boldsymbol{\phi}_j) + \alpha_1 (1 - J_Q(\boldsymbol{\phi}_j)) \tag{3a}$$

$$\text{s.t.}: \; \boldsymbol{\phi}_j \in Traces(T) \tag{3b}$$

$$J_T(\boldsymbol{\phi}_j) \leq \tau_{max} \tag{3c}$$

$$J_Q(\boldsymbol{\phi}_j) \geq q_{min} \tag{3d}$$

where, α_0, α_1 are normalized weights on the time and quality, respectively, and are determined by the decision maker, τ_{max} denotes the maximum allowable processing time for the manufacturing cell, q_{min} denotes the minimum allowable yield for the manufacturing cell, and the constraint in Eq. 3b denotes that each $\boldsymbol{\phi}_j$ is a feasible assignment with respect to the topology DT. The minimum quality and maximum processing time constraints are added to ensure efficient solutions, but based on the feasibility of the optimization these constraints may be relaxed. Numerical values for the cycle and setup times of cnc_1 and cnc_3 are given in Table I.

TABLE I. Cycle and setup times of cnc_1 and cnc_3 for the reconfiguration example. All units are in seconds.

Machine	$\tau(S_1)$	$\tau(S_2)$	$\tau(S_3)$	$\tau(C_1)$	$\tau(C_2)$	$\tau(C_3)$
cnc_1	7	10	14	320	307	410
cnc_2	7	10	14	332	310	384

The yield of cnc_1 is $0.95, 0.94, 0.96$ for features f_1, f_2, f_3 respectively. The yield of cnc_2 is $0.99, 0.98, 0.92$ for features f_1, f_2, f_3 respectively. For the optimization in (3), the 8 solutions that satisfy the condition in constraint (3b) are evaluated. The constraints are chosen as $\tau_{max} = 1070$ sec and $q_{min} = 0.8$. The weights α_i are normalized by the maximum values of the functions $J_T(\boldsymbol{\phi}_j)$ and $J_Q(\boldsymbol{\phi}_j)$ in the implementation of the cost function. The problem is encoded as a mixed integer linear program and solved using $Matlab$.

TABLE II. Results for the reconfiguration in the case study.

Type	ϕ^*	$J_T(\boldsymbol{\phi}_j^*)$	$J_Q(\boldsymbol{\phi}_j^*)$
Best Yield	cnc_1,cnc_3,cnc_1	1061 sec	0.89
Best Time	cnc_1,cnc_1,cnc_3	1028 sec	0.82
Mixed	cnc_1,cnc_3,cnc_3	1038 sec	0.85

The type of solution, optimizer assignment ϕ^*, associated time, and associated quality results evaluated by the optimization are shown in Table II. The best yield solution uses ($\alpha_0 = 0, \alpha_1 = 1$), the best time solution uses ($\alpha_0 = 1, \alpha_1 =$

0), and the mixed solution uses ($\alpha_0 = 0.8, \alpha_1 = 0.2$). Based on the type of solution evaluated by the decision maker, a reconfiguration is implemented in the system.

This case study shows how the DT platform uses multiple DTs for machine failure prediction, rescheduling, and recommendation of new control reconfiguration actions. Using a single DT would not be sufficient to address all these purposes. A machine process DT, herein uses the spindle current signature in real-time to monitor the tool health. An SDC app uses the image provided by this DT to predict tool breakage. The defective machine is then avoided and a reconfiguration action is evaluated by the SDC decision maker, which uses the feasible routes, machine availability, and machine capability, to provide an optimal reconfiguration of the system with respect to throughput and quality. The machine process DT alone is incapable of providing all this information, which requires cooperation with other DTs. In summary, cooperation of multiple DTs within the DT platform is needed to address multiple purposes, mainly real-time monitoring, anomaly detection/prediction, and real-time reconfiguration and optimization.

VI. Discussion and Conclusion

The proposed DT framework offers SDC users the flexibility and agility to design, build, extend, and maintain DT systems at a faster pace and to coordinate these systems around factory-wide objectives to accommodate SM systems with a variety of requirements. It helps improve quality and throughput of the production while reducing waste in time and resource. The end result is advancement in SM, DT technology, and the manufacturing industry.

The overall DT platform is partitioned into individual DT classes that allow to model the major components such as physical topology, machine assets, machine processes, and system processes in order to respond to several SM problems. DT instances of these classes are ultimately combined under the coordination of the DT manager. The coordination requires the DT platform to operate at a performance level high enough so that reconfiguration decisions can be made without the decision process time impacting the throughput or quality of the system. On the other hand, the functionality of some DTs rely on other neighboring DTs. Therefore, a failure in a DT (*e.g.,* a DT unexpectedly disconnected from the pub-sub infrastructure) can result in a chain reaction and lead to malfunction in the entire DT platform. To reduce the likelihood of interruption, the SDC user may deploy redundant DTs as backups or DTs and the SDC may employ strategies to provide alternative solutions if a DT anywhere in the decision chain is not available or does not provide information. This allows the DT platform to guarantee the availability and maintain the functionality at small or no cost.

In current design, simulations are performed independently by individual DTs that support a simulation functionality. However, this limits the capability of predicting the states of the global view. In future work, we will improve the simulation manager with a more comprehensive simulation mechanism in which simulation can be carried out in a group of DTs. This would also enhance the flexibility in the DT platform to support more complex applications.

References

[1] S. Jeschke, C. Brecher, T. Meisen, D. Özdemir, and T. Eschert, "Industrial internet of things and cyber manufacturing systems," in *Industrial Internet of Things*. Springer, 2017, pp. 3–19.

[2] G. Pedrielli and F. Ju, "Simulation-predictive control for manufacturing systems," in *International Conference on Automation Science and Engineering (CASE)*. IEEE, 2018, pp. 1310–1315.

[3] M. Grieves, "Digital twin: Manufacturing excellence through virtual factory replication," *White paper*, pp. 1–7, 2014.

[4] F. Tao, H. Zhang, A. Liu, and A. Nee, "Digital twin in industry: State-of-the-art," *IEEE Transactions on Industrial Informatics*, 2018.

[5] J. Moyne, E. Balta, I. Kovalenko, Y. Qamsane, and K. Barton, "The digital twin in the manufacturing ecosystem of the future," in *1st International Workshop on Smart Manufacturing Modeling and Analysis (SM^{2}A)*. IEEE, 2019.

[6] B. Scholten, *The road to integration: A guide to applying the ISA-95 standard in manufacturing*. Isa, 2007.

[7] K. M. Alam and A. El Saddik, "C2ps: A digital twin architecture reference model for the cloud-based cyber-physical systems," *IEEE Access*, vol. 5, pp. 2050–2062, 2017.

[8] F. Tao, F. Sui, A. Liu, Q. Qi, M. Zhang, B. Song, Z. Guo, S. C.-Y. Lu, and A. Nee, "Digital twin-driven product design framework," *International Journal of Production Research*, pp. 1–19, 2018.

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

[10] J. Leng, H. Zhang, D. Yan, Q. Liu, X. Chen, and D. Zhang, "Digital twin-driven manufacturing cyber-physical system for parallel controlling of smart workshop," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–12, 2018.

[11] E. Bottani, A. Cammardella, T. Murino, and S. Vespoli, "From the cyber-physical system to the digital twin: the process development for behaviour modelling of a cyber guided vehicle in m2m logic," *XXII Summer School Francesco TurcoIndustrial Systems Engineering*, pp. 1–7, 2017.

[12] E. Tuegel, "The airframe digital twin: some challenges to realization," in *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference 20th AIAA/ASME/AHS Adaptive Structures Conference 14th AIAA*, 2012, p. 1812.

[13] F. Tao, M. Zhang, Y. Liu, and A. Nee, "Digital twin driven prognostics and health management for complex equipment," *CIRP Annals*, vol. 67, no. 1, pp. 169–172, 2018.

[14] I. Graessler and A. Poehler, "Integration of a digital twin as human representation in a scheduling procedure of a cyber-physical production system," in *International Conference on Industrial Engineering and Engineering Management (IEEM)*. IEEE, 2017, pp. 289–293.

[15] F. Lopez, Y. Shao, Z. M. Mao, J. Moyne, K. Barton, and D. Tilbury, "A software-defined framework for the integrated management of smart manufacturing systems," *MFG Letters*, vol. 15, pp. 18–21, 2018.

[16] E. C. Balta, D. M. Tilbury, and K. Barton, "A centralized framework for system-level control and management of additive manufacturing fleets," in *International Conference on Automation Science and Engineering (CASE)*. IEEE, 2018.

[17] Modelica Association. (2013) Functional mock-up interface. Accessed: 14 Jun. 2019. [Online]. Available: https://fmi-standard.org/

[18] Y. Qamsane, E. C. Balta, J. Moyne, D. Tilbury, and K. Barton, "Dynamic rerouting of cyber-physical production systems in response to disruptions based on sdc framework," in *2019 American Control Conference (ACC)*. IEEE, 2019.

[19] A. P. Estrada-Vargas, E. López-Mellado, and J.-J. Lesage, "Input–output identification of controlled discrete manufacturing systems," *Int. Journal of Systems Science*, vol. 45, no. 3, pp. 456–471, 2014.

[20] A. A. Khan, J. R. Moyne, and D. M. Tilbury, "An approach for factory-wide control utilizing virtual metrology," *IEEE Transactions on Semiconductor Manufacturing*, vol. 20, no. 4, pp. 364–375, 2007.

[21] J. Iskandar, J. Moyne, K. Subrahmanyam, P. Hawkins, and M. Armacost, "Predictive maintenance in semiconductor manufacturing," in *2015 26th Annual SEMI Advanced Semiconductor Manufacturing Conference (ASMC)*. IEEE, 2015, pp. 384–389.

[22] Q. Qi and F. Tao, "Digital twin and big data towards smart manufacturing and industry 4.0: 360 degree comparison," *Ieee Access*, vol. 6, pp. 3585–3593, 2018.