A conceptual architecture and model for smart manufacturing relying on service-based digital twins

Tiziana Catarci*, Donatella Firmani†, Francesco Leotta*, Federica Mandreoli‡, Massimo Mecella*, Francesco Sapio*
*Sapienza Università di Roma, Italy. Email: {lastname}@diag.uniroma1.it
†Università Roma Tre, Italy. Email: donatella.firmani@uniroma3.it
‡Università di Modena e Reggio Emilia, Italy. Email: federica.mandreoli@unimore.it

***Abstract*—The technological foundation of smart manufacturing consists of cyber-physical systems and the Internet-of-Things (IoT). Each IoT device in a smart factory can be coupled with a *digital twin*, that is, a dynamic virtual representation of the physical system across its life-cycle using real-time sensor data. Currently, the manufacturing process itself, the involved devices, and how they interact, is designed by human experts in a traditional way. We envision an architecture where humans can instead specify a *goal* and take advantage of technologies such as digital twins to automatically compose the corresponding physical processes, sharing some analogies with the notion of *Web service composition*.**

I. Introduction

Last years witnessed a continuous evolution of technologies in the fields of communication, networking, storage and computing, that found their way in the traditional world of industrial automation. This trend, functional to increase productivity and quality, to ease workers' lives, and to define new business opportunities, goes under the name of *smart manufacturing* or *Industry 4.0*.

Digital factory is a key concept. It aims at using digital technologies to promote the integration of product design processes, manufacturing processes, and general collaborative business processes across factories. An important aspect of this integration is to ensure interoperability between machines, products, processes, and services. A digital factory consists of a multi-layered integration of the information related to various activities along the factory and related resources. Actors can fall in different categories, being humans (i.e., final users or participants in the production process), information systems or industrial machines. These physical entities must have a faithful representation in the digital world, usually referred to as *digital twins*.[1]

A digital twin (DT) exposes a set of *services* allowing to execute certain operations and produce data describing its activity. We can imagine these data stored in a factory *data space* together with other information, e.g., data available from the company and production history, worker suggestions and preferences. Such services are typically used to query or manipulate the state of the system, and associated data are leveraged for diagnostics and prognostics. The availability of DT services and data can have a huge impact on the design of manufacturing processes, by allowing automatic recovery and optimization, and even automatic composition of the intermediate steps for achieving a production goal.

Inspired by the research about automatic orchestration and composition of *software artefacts*, such as Web services, we argue that: *(i)* an important step towards the development of new automation techniques in smart manufacturing is the modeling of DT services and data as software artifacts; *(ii)* the principles and techniques for composition of artifacts in the digital world can be leveraged to improve automation in the physical one.

A crucial difference between traditional software artifacts used in composition techniques and DTs is that DTs may not share the same view of the world. Modern information systems and industrial machines may natively come out indeed with their digital twin. In other cases, especially when the approach is applied to already established factories and production processes, digital twins are obtained by wrapping actors that are already in place. In such scenarios, data management techniques (including integration and exchange) are a key ingredient for DTs interoperability.

Properties of DTs. DTs are commonly known as a key enabler for the digital transformation in manufacturing, however, in the literature, there is no common understanding concerning this term. Different definitions agree on features such as *(i)* connectivity, i.e., the ability to communicate with other entities and digital twins, *(ii)* autonomy, i.e., the possibility for the DT to live independently from other entities, *(iii)* homogeneity, i.e., the capability, strictly connected to the autonomy, that allows to use the same DT regardless of the specific production environment, *(iv)* easiness of customization, i.e., the possibility to modify the behavior of a physical entity by using the functionalities exposed by its DT, and *(v)* traceability, i.e., the fact that a DT leaves traces of the activity of the corresponding physical entity.

Our contribution. In this context, the contribution of this paper is twofold: *(i)* initially, we aim at reviewing recent approaches for handling DTs, in order to fully understand the relationship between DTs and software artefacts such

[1] See, for instance, http://www.forbes.com/sites/bernardmarr/2017/03/06/what-is-digital-twin-technology-and-why-is-it-so-important

as Web services (WSs); and then *(ii)* we propose a complex architecture for digital factories, which exploits digital twins and aim at exploiting automatic composition techniques of WSs, based on the seminal results of [5]. Our proposal captures analogies and differences between DTs and WSs, and enables integration and composition of DTs through offered services and data available in the data space. We believe that the similarity between DTs and WSs envisioned in our approach will foster further research on smart manufacturing.

Outline. The paper is organized as it follows. Section II surveys relevant literature about DTs, and outlines available technologies. Section III describes the proposed architecture, and finally Section IV concludes the paper.

II. Surveying digital twins

We perform a survey on relevant work, by adopting methods typical of a Systematic Literature Review (SLR) [12, 13]. Rather than strictly performing a SLR, we adopt some SLR techniques in order to summarize the current status of the technology, highlight the benefits and limitations of the proposed methods, and suggest areas for further investigation. Specifically, we define a set of queries over Google Scholar, and focus on the recent works that are returned for those queries, as summarized in Table I. Intuitively, our queries correspond to the following research questions: *(R1)* which is the actual overlapping between DTs and cyber-physical systems? *(R2)* how much is the DT concept strictly related to smart manufacturing? *(R3,R4)* What current approaches for DTs propose as far as semantics and data exchange? *(R5)* How can I predict the behaviour of a DT without its physical counterpart?

Finally, we use papers in Table I to identify other related works. The resulting body of papers is discussed in the following, together with the answers to the above questions.

(R1) DTs and cyber-physical systems. DTs monitor and control physical entities, where physical entities send data to update what are commonly referred to as the *virtual models* [14, 3].

Authors in [1] introduce a DT reference model for a cloud-based cyber-physical system, such as a smart city. The model is based on a smart interaction controller using a Bayesian belief network. They divide the system into three operational modes, namely *(i)* physical level sensors-fusion mode, *(ii)* cyber-level digital twin services-fusion mode and *(iii)* a deep integration of sensor-services fusion mode. They provide a context-based control decision scheme that uses Bayesian networks and fuzzy logic based rules to select any of these system modes for inter-system interactions.

Authors in [27] introduce a novel architecture for large-scale DT platforms including a distributed DT cooperation framework, flexible data-centric communication middleware, and the platform-based DT application to develop a reliable advanced driver assistance system.

(R2) DTs and smart manufacturing. In the the traditional manufacturing process, the production plan is generated based on the new and historical orders. Then the preparation for production is carried out, such as equipment maintenance and material collection. Finally, formal production is executed according to the plan. If some conflicts appear, the plan is modified to adapt to the actual situations. After production, finished products are inspected to ensure whether they meet requirements, and consequently transported into the warehouse or repaired. Information generated during production such as process documents and fault records are kept in files for the next round. The work of [23] provides a conceptual model and specific operation mechanisms for a DT *shop-floor*, that is, a basic unit of manufacturing, where data from both physical and virtual sides as well as the fused data are provided to drive all the steps of the production process.

Authors in [24] introduce a DT-based system for the waste electrical and electronic equipment (WEEE) recovery to support the manufacturing/re-manufacturing operations throughout the product's life cycle. The system is based on a Service Oriented Architecture (SOA), in which both the electronic device and its DT are considered as the services. The DT plays as the bridge between the physical world and the cyber world. As the product moves from the beginning-of-life to middle-of-life and end-of-life, the knowledge and information maintained inside its DT becomes bigger and richer. The digital world is supported and hosted by a cloud computing environment. Computer-aided engineering modules help to simulate the performance to validate and improve the product design. During manufacturing processes, the operations are controlled by the manufacturing execution system. With the help of multiple enablers, e.g., smart monitors, wireless sensors and other IoT devices, the DT is strengthened by the related operation and product performance data collected from the factory. After an end user purchases the product, she can interact with the DT via multiple enablers, e.g., a RFID reader and near-field communication, which are affordable devices for general usage.

Authors in [25] provide an overview of Industry 4.0 features in multiple reference architectures and develops a maturity model for IT architectures for data-driven manufacturing. The group around Reference Architecture Model Industry 4.0 – RAMI – developed a detailed conceptual architecture and a first implementation of the digital twin, the open Asset Administrative Shell. It consists of an operating system and can be integrated into a SOA. RAMI is explicitly defined as a SOA, relying on functionalities encapsulated into services. As an example, RAMI provides access to an administrative shell through a service interface and a common data format. More in details, the digital twin in the Industrial Internet Reference Architecture is located in the entity abstraction layer of the control domain. It directly

Table I
MOST RELEVANT RESULTS TO OUR GOOGLE SCHOLAR QUERIES ("DIGITAL TWIN"+ "X") AND RELATED WORKS, POSSIBLY FOCUSING ON PUBLICATION YEAR $\geq Y$.

	query	results	related
R1	"cyber physical system"	[14, 3]	[1, 27]
R2	"shop floor" / "industry 4.0" ($\geq$ 2018)	[23, 24]	[25]
R3	"semantic" / "model operations" ($\geq$ 2018)	[28, 4]	[2, 20]
R4	"data exchange"	[20]	[22]
R5	"simulation" ($\geq$ 2018)	[19]	[10, 7, 26]

builds upon of the communication layer and provides an abstraction of sensor values and actuators.

(R3) Models of and in digital twins. Authors in [28] introduce a semantic schema representation that categorises industrial data streams in DTs, allowing subscribers to consume similar data from multiple assets within a single data analytics pipeline, and paving the way for a more intuitive management of digital twin representations from industrial assets.

In order to execute operations between products, processes and resources, authors in [4] propose the usage of the popular Automation Markup Language (AutomationML), for modelling a structural parts machining cell. AutomationML stores engineering information following an object-oriented paradigm and allows to model physical and logical components as data objects. An object may consist of other sub-objects, and may itself be a part of a larger composition or aggregation.

Authors in [2] model digital factories as the DTs of physical facilities, replicating the facility in terms of installed machinery, material handling equipment, and layout. The digital factory is supported by a formal ontology and a mathematical model for quantifying the processing capabilities. Manufacturing Service Description Language (MSDL) is a descriptive ontology developed for representation of capabilities of manufacturing services. MSDL decomposes the manufacturing capabilities into five levels of abstraction, namely, supplier-level, shop-level, machine-level, device-level, and process-level. A unique feature of MSDL is that it is built around a service-oriented paradigm, therefore, it can be used for representing a manufacturing system as a collection of manufacturing services. MSDL was initially designed to enable automated supplier discovery in distributed environments with focus on mechanical machining services. However, to address a wider range of services offered by small and medium-sized suppliers in contract manufacturing industry, the service ontology can always be extended systematically through active involvement of a community of ontology users.

All the above works aim at building models *of* some aspects of DTs. Clearly, a DT itself consists of an information model to describe its physical twin [11]. Specifically, a DT can consist of different models and data, that take into account the different phases of product life cycle. As discussed in [20], such models include: system models, functional models, 3D geometric models, multiphysics models, manufacturing models, and usage models.

(R4) Data exchange. Some problems affecting product data management and application in Product Lifecycle Management (PLM) exist as follows: *(1)* due to the different purposes and tasks, the data generated in various phases of the entire product lifecycle may form information islands between different phases of the product lifecycle; *(2)* there is a lot of duplicate data in different phases of product lifecycle. These duplicate data may cause wasting of resources and data sharing problem; *(3)* the interaction and iteration between the so-called *big data analysis* and various activities in the entire product lifecycle are nowadays relatively absent; *(4)* the current applications of big data prefer to put emphasis on the analysis of physical product data rather than the data from virtual models.

Authors in [20] use AutomationML (analogously to the already mentioned work of [4]) to model attributes of a DT, and discuss its application towards the data exchange between different systems that are connected with the DT itself.

Authors in [22] discuss more in general how to generate and use converged DT data to better serve product lifecycle, so as to drive product design and manufacturing.

(R5) Simulation in DTs. Simulation is widely used to predict how the physical twin of a DT can be expected to perform in the real world, thus it is crucial in a hypothetical DTs composition task. By incorporating data from the physical twin into the DT, engineers can improve system models, and subsequently use the results of the analysis with the DT to improve the operation of the physical system in the real-world.

Authors in [19] introduce simulation techniques to make digital twins experimentable, in order to have Experimentable Digital Twins (EDTs). The resulting networks of interacting EDTs is simulated in virtual testbeds, providing new foundations for comprehensive simulation-based systems engineering. The networking of EDTs with real assets leads to hybrid application scenarios in which EDTs are used in combination with real hardware, thus realizing complex control algorithms, innovative user interfaces, or mental models for intelligent systems.

Authors in [10] present an approach to automatically generate the virtual environment from specifications. The architecture of their framework is composed of two main modules, the generator and the virtual environment. The generator module takes engineer- and domain-specific knowledge as input to create the virtual environment. Once the DTs and the network topology have been generated, the virtual environment can operate in two modes. First, the virtual environment provides a simulation mode, in which the DTs run independently from the physical environment. Second, the replication mode records events such as network traffic from the physical environment and replicates them in the virtual setup. The framework also includes multiple modules that can be activated on demand, such as monitoring, security analysis and intrusion detection.

Authors in [7] focus on the efficacy of modelling and simulating the as-manufactured geometry of each individual component towards the realization of a DT for non-standardized material test specimen.

Authors in [26] discuss upcoming challenges to exploit the full potential of modelling and simulation within smart factories; to address these challenges, they present a framework for modelling and simulation in CPS-based factories.

From the discussed literature, we observe that DTs are used in a variety of smart manufacturing scenarios, possibly confused with the more general cyber-physical systems. Models of DTs mainly focus on specific interoperability aspects rather than capturing the whole operation semantic. We note that data exchange techniques for DTs are available in literature, although they do not take fully into account the heterogeneity of a digital factory data space. No work to the best of our knowledge deals with automatic DT composition.

Finally, state-of-the-art simulation techniques show promising results.

A. Technologies for realizing digital twins

Several technologies do exist allowing to implement DTs. In general, current solutions provided by some of the key IoT platforms are pretty basic. There is a lot of work needed for digital twins to become a reality for developers.

The exceptions appears to be the Eclipse Ditto and Bosch IoT Things solutions; they appear to have the most advanced technology for DTs, basic building blocks seem to be there, nevertheless there are still missing features, like simulation and data integration.

Open source solutions. Eclipse Ditto[2] is a project created by the Eclipse Foundation and contains the technology used for IoT solutions. The Eclipse Ditto website contains detailed documentation and a sandbox server.

Commercial solutions. GE Predix[3] is a sophisticated solution dealing with asset-centric digital twins. GE Predix provides a detailed tutorial on how to create a digital twin for analytics. It also has an Asset Service that allows to model its own assets, which can be essential to any DT solution. Bosch IoT Things[4] is the digital twin solution by Bosch. Bosch provides detailed technical documentation, including developer guides, demo applications, and hosted dashboard. The Bosch DT solution appears to be the most advanced and, more importantly, the most accessible to developers.

Cloud-based solutions. Microsoft Azure IoT[5] includes the concept of *device twin* as part of their device management solution. A device twin is automatically created when a device is connected to the MS IoT Hub. The device twin is represented by a JSON file that stores the device state information that can be used to synchronize device information with back-end processes. Amazon AWS IoT[6] refers to a digital twin through the introduction of the notion of a *device shadow*. A device shadow is a JSON file that contains the state information, meta-data, timestamp, unique client token, and version of a device connected to the device shadow service. There are three basic REST APIs that can be used to interact with the device shadow: GET, UPDATE, DELETE. You can also interact with device shadows using MQTT messages. IBM Watson IoT[7] introduces three different products (IBM Rational Rhapsody, Rational Engineering Lifecycle Manager, IBM Rational Lifecycle Integration Adapters) to deal with digital twins. Apparently, IBM is the vendor who is investing the most in promoting the concept of digital twin (see also https://www.youtube.com/watch?v=RaOejcczPas).

III. Architectural Model

Our approach, inspired by the Roman model for service composition [6, 5], considers smart manufacturing scenarios where DTs of physical systems – or, simply, *twins* – provide stateful services wrapping the functionalities of machines and tasks of human operators.

In such contexts, data are usually available through several sources not sharing a common schema and vocabulary, as DTs come from different vendors. It is reasonable to consider an heterogeneous data space where a mediator is present and it also takes the role of translator between different formalisms and access methods.

We consider learning as a fundamental feature of DT. Learned functions include the automatic generation of alarms, but also automatic triggering of actions and status changes. Additionally, twins can be queried on learned functions and, as a result, the data space is far more dynamic than in more traditional scenarios.

[2] See https://www.eclipse.org/ditto

[3] See https://www.ge.com/digital/iiot-platform

[4] See https://www.bosch-iot-suite.com/service/things

[5] See https://azure.microsoft.com/en-us/services/iot-hub

[6] See http://aws.amazon.com/iot

[7] See https://ww.ibm.com/internet-of-things

As in modern micro-services architectures, notifications based on publish&subscribe is a common architectural pattern, and therefore we provide for subscriptions to events generated by other DTs.

We propose an architecture for a smart manufacturing process based on DTs as depicted in Figure 1, where the main components are the DTs, the data space, human supervisors and a mediator.

DTs wrap physical entities involved in the process. These physical entities can be manufacturing machines or human operators. A DT exposes a Web API consisting, in general, of three parts: the synchronous one, the query interface and the asynchronous one. The synchronous interface allows to give instructions to the physical entity. These instructions may, for example, produce a state change in a manufacturing machine (in case the twin is over a machine) or ask a human operator to perform a manual task (in case the twin is over a manufacturing worker). The query interface allows for asking information to the physical entity about its state and related information; noteworthy, these latter can be obtained by applying diagnostic and prognostic functions results of machine learning. The asynchronous interface generates events available to subscribers.

It is important to note how manufacturing machines and human actors can be considered identical from the point of view of the offered API, e.g., human actors produce asynchronous events as well, for example generating alarms.

The data space contains all the data available to the process. These data are heterogeneous in their nature from the access technology point of view, the employed schema (or its absence) and the employed vocabulary. It is important to note how the DTs contribute to the data space with both the query API and the asynchronous one. Other sources for the data space may include relational and no-SQL databases or unstructured sources such as spurious files, which constitute the factory information system.

The human supervisor is the one defining the goals of the process in terms of both final outcomes and key performance indicators to be obtained.

In order to reach the goal defined by the human supervisor, available twins and data must be integrated. This task is fulfilled by the mediator. The mediator acts in two phases: the *synthesis phase* and the *execution phase*. During the synthesis phase, the specifications of the APIs exposed by digital twins and the meta-data (e.g. data source schemas) available in the data space, are composed in order to construct a *mediator process*. During the execution phase, the mediator runs its program by preparing the input messages for the single twins involved in the proper sequencing/interleaving. Indeed, as each twin may potentially adopt a different language and vocabulary, in order to compose required input/output messages, the mediator translates and integrates the data available in the data space to comply with the format requested by the specific called service.

An important aspect of the proposed architecture is that multiple companies can participate in the process (typically those ones involved in the value chain). Again, it is not reasonable to have twins directly communicating with one another. Once again, the role of the mediator is fundamental, being the component that can access the services offered by the twins available in the different companies.

Example. In the following, we will give an intuition of the proposed approach by considering a cardboard boxes real manufacturing scenario involving three companies: the cardboard manufacturer, the die cutter manufacturer and a delivery service. For the sake of simplicity, we will consider three twins: the twin corresponding to the die cutter manufacturer, the twins corresponding to the delivery services (potentially they might be many) and a twin corresponding to a single die cutter installed at the cardboard manufacturer factory. □

A. The data space as a polystore

In digital factories, it is of strategic importance to provide effective mechanisms for searching information along diverse and distributed data sources. Indeed, when dealing with data management, organizations are becoming polyglot [15]: they tend to adopt the data management systems that are most suitable to the kind of data, that can significantly vary. Polystores [21], together with its first reference implementation BigDAWG [8], have been proposed recently as a valuable solution for this scenario. A polystore system provides a loosely coupled integration over multiple, disparate data models and query languages. In this system, queries are posed over *islands of information*, i.e. collections of data sources, each accessed with a single query language, and the same data can be accessed from multiple islands. Data transformation and migration in polystores have been considered in [9].

In our approach, the data space can be modelled as a polystore. We inherit the data modelling approach proposed in [15] where a polystore is made of a set of databases stored in a variety of data management systems, each one potentially offered by a twin through the query interface.

A collection of operators act on the polystore with the aim of supporting the data access needs due to digital twin composition.

Example (cont). The polystore will contain the APIs of the twins and the historical production and shipping data from the die cutter manufacturer and the delivery services. □

B. Modelling the DTs

We model the behaviour of DTs as guarded automaton (by extending [5]): a DT wraps a physical entity which, in a production process, follows specific stages/steps. The

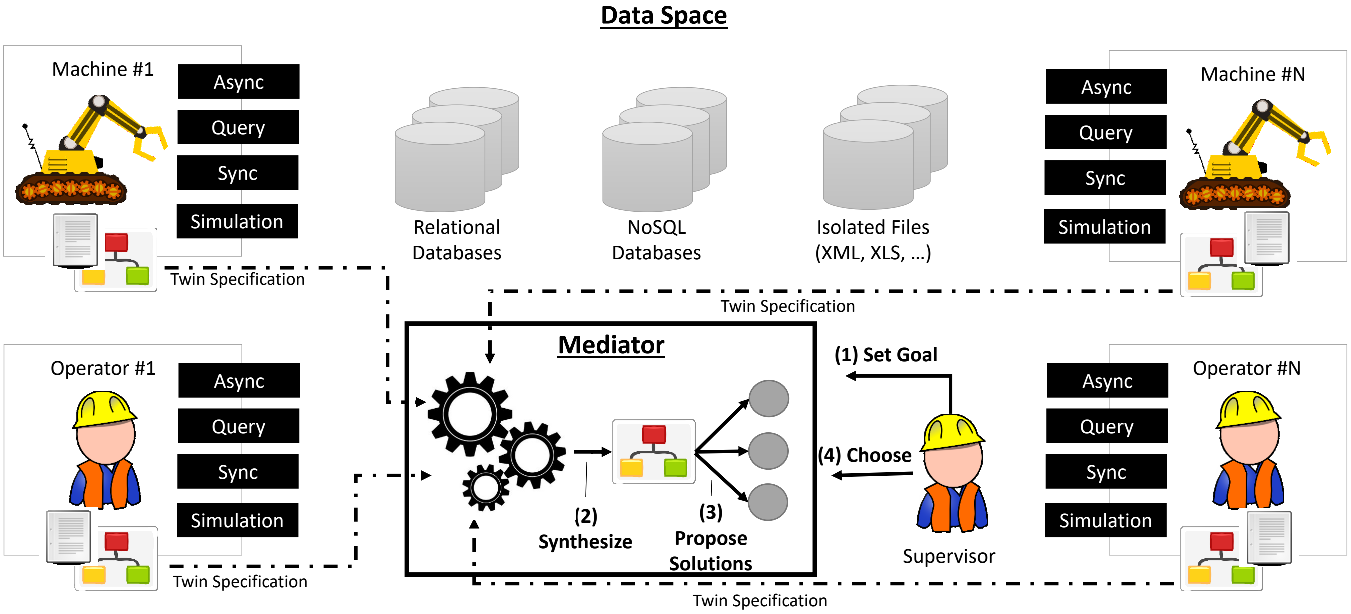

Figure 1. The proposed architecture.

synchronous API of the twin corresponds then to input messages of the guarded automaton. The main difference here consists in what the local storage contains and how this influence automaton transitions and atomic processes.

In [5], the local storage of a Web service contains variables instantiated and modified by the automaton during the execution, and by messages sent by the mediator. Here we extend to have autonomous threads enriching the local storage with measurements coming from sensors and outcome of machine learning predictions. In this sense, data provided by sensors and machine learning are somewhat similar. We add a further $UStore$ (where the U stands for *uncertain*) containing triples (s, m, c) where s is a variable name corresponding to a sensor or a prediction task, m is the information measured or predicted, and c is the confidence in this value. As a consequence, we can define the transition conditions of the guarded automaton in terms of $UStore$ as well. The result is the possibility to include in the model automatic transitions that are not the result of explicit message exchange.

The data contained in the local storage is part of the polystore defining the process. In order to query this data, each digital twin provides the query API.

The DT can additionally provide asynchronous data to other twins and to the mediator itself through its asynchronous API. In particular, asynchronous events can be generated in response to a change of the $UStore$ or to the transition of the twin automaton. Interested recipients subscribe to this events following a publish/subscribe architectural pattern.

Example (cont). The digital twin corresponding to the die cutter contains at least two states: `mounted` and `unmounted`. At any time, the twin provides information about the number of rotations performed and the residual life expectancy. The twin corresponding to the die cutter manufacturer may provide information about the time needed to produce a new die cutter, whereas the twin of a delivery service may provide information about the expected shipping time from the die cutter manufacturer factory to the cardboard manufacturer factory. □

C. Actual and simulation perspectives

Machine learning represents a fundamental feature of DTs, especially for simulation purposes. Most real systems that are confronted with multiple data streams benefit from machine learning and analysis to make sense of the data. For example, machine learning can automate complex analytical tasks, evaluate data in real time, regulate behavior with the minimum need for supervision, and increase the likelihood of desired results [16]. The uses of machine learning within a DT include: supervised learning (for example, using neural networks) of the preferences and priorities of the user in an experimental test bed based on simulation [18]; learning without supervision of objects and models using, for example, clustering techniques in virtual and real environments [17]; and strengthening the learning of system and environment states in uncertain and partially observable operating environments.

Another fundamental feature of a digital twin is simulation. We have already introduced the role of machine learning in the population of the $UStore$, but it has a fundamental role in simulation too. In particular, when proposing possible solutions, the mediator may require to simulate the result of operations on the twins following the human-in-the-loop philosophy. When the supervisor takes a decision, the actions in the proposed plan are executed in the real world and the actual results can be used to improve the simulation feature in a reinforcement learning pattern.

In order to do this, both twins and data space have two perspectives. The *actual* perspective reflects the current state of the physical world, whereas the *simulation* perspective allows the mediator to perform simulations useful to produce alternatives for the supervisor.

As a result, each DT has two instances of the corresponding automaton, in different states and with different values in the local storage. The same holds for the other data sources composing the polystore. To access the functionalities of the simulation perspective, the DT provides a further simulation API.

Example (cont). The twin corresponding to the die cutter may expose a simulation API that can help to simulate what happens if the die cutter undergoes a given setting of the rotation speed throughout a 24h period. □

D. Semantics

The principal differences between the mediator proposed in [5] and the evolution we propose to meet the requirements of smart manufacturing, are *(i)* the way the goal is defined by the supervisor, and *(ii)* the need for it to operate a translation over the data available in the data space in order to perform its coordination task between the digital twins required to reach the goal.

In [5], the goal of the mediator was a guarded automaton to be synthesized combining the automatons of the single services. This is not appropriate for our scenario, as it requires an expensive human work and the vocabulary of the automaton to be compliant with that of twins. We propose instead to formulate goals in terms of key performance indicators in a declarative manner. In order to do that, the goal declaration must be translated in terms of the input/output messages of twins' automatons and the content of the data space. Additionally, the mediator must be able to discover twins and their capabilities in terms of offered services.

Example (cont). The production goal may be to avoid interruptions in the production process. Currently, for economic reasons, manufacturers prefer not to store in the warehouse replacement production machines and tools. As a consequence, the supervisor may instruct the mediator to order a new die cutter by predicting when the current one is going to break, taking into account production time on the die cutter manufacturing site and shipping time. In order to satisfy the goal, the mediator will discover the different services and automatically understand how to combine them. □

IV. Concluding remarks

In this paper, we provide an initial scouting of the possibilities offered by composition techniques in smart manufacturing. The intuition here is that, likewise a Web service, a DT, which is a fundamental concept in smart manufacturing, can be described as a stateful automaton and, as a consequence, DTs can be combined following approaches that have been proposed to combine Web services aiming at a specific goal. This paper first provides a literature review concerning DTs and the different ways to model and implement them, and then outline an extension of the service composition modeling framework provided in [5] to meet the challenging requirements imposed by smart manufacturing.

The proposed approach requires extensive future work to be fully technically realized and then validated. In order to do that, we are conducting a set of real world studies in manufacturing companies (cf. the example provided in the text).

As a first step, we are developing frameworks for real DTs of manufacturing machines and company services.

Finally, a point not addressed by the current paper, is how goals, expressed in terms of key performance indicators, can be translated in a composition of twins. In general, several possible solutions will be provided to the supervisor, who will decide which solution to apply according to a confidence estimation. How this confidence must be computed, according to the quality of machine learning tasks involved in DTs and in the mediator for entity resolution, is a challenging ongoing research task.

Acknowledgements. This work has been partly supported by the European Commission through the H2020 project *FIRST – virtual Factories: Interoperation suppoRting buSiness innovaTion* (grant agreement # 734599). The work of Tiziana Catarci, Francesco Leotta, Massimo Mecella and Francesco Sapio has been also partly supported by the research contract with Rota Laser Dies.

References

[1] 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.

[2] F. Ameri and R. Sabbagh, "Digital factories for capability modeling and visualization," in *IFIP International Conference on Advances in Production Management Systems*. Springer, 2016, pp. 69–78.

[3] R. Baheti and H. Gill, "Cyber-physical systems. the impact of control technology, t. samad and am annaswamy," *IEEE Control Systems Society*, vol. 1, 2011.

[4] J. Bao, D. Guo, J. Li, and J. Zhang, "The modelling and operations for the digital twin in the context of manufacturing," *Enterprise Information Systems*, pp. 1–23, 2018.

[5] D. Berardi, D. Calvanese, G. De Giacomo, R. Hull, and M. Mecella, "Automatic composition of transition-based semantic web services with messaging," in *Proceedings of the 31st international conference on Very large data bases*. VLDB Endowment, 2005, pp. 613–624.

[6] D. Berardi, D. Calvanese, G. De Giacomo, M. Lenzerini, and M. Mecella, "Automatic composition of e-services that export their behavior," in *Service-Oriented Computing - ICSOC 2003, First International Conference, Trento, Italy, December 15-18, 2003, Proceedings*, ser. Lecture Notes in Computer Science, M. E. Orlowska, S. Weerawarana, M. P. Papazoglou, and J. Yang, Eds., vol. 2910. Springer, 2003, pp. 43–58.

[7] A. Cerrone, J. Hochhalter, G. Heber, and A. Ingraffea, "On the effects of modeling as-manufactured geometry: toward digital twin," *International Journal of Aerospace Engineering*, vol. 2014, 2014.

[8] J. Duggan, A. J. Elmore, M. Stonebraker, M. Balazinska, B. Howe, J. Kepner, S. Madden, D. Maier, T. Mattson, and S. Zdonik, "The bigdawg polystore system," *ACM Sigmod Record*, vol. 44, no. 2, pp. 11–16, 2015.

[9] A. Dziedzic, A. J. Elmore, and M. Stonebraker, "Data transformation and migration in polystores," in *Proc. of the 2016 IEEE High Performance Extreme Computing Conference (HPEC)*, 2016, pp. 1–6.

[10] M. Eckhart and A. Ekelhart, "Towards security-aware virtual environments for digital twins," in *Proceedings of the 4th ACM Workshop on Cyber-Physical System Security*. ACM, 2018, pp. 61–72.

[11] L. Hu, N.-T. Nguyen, W. Tao, M. C. Leu, X. F. Liu, M. R. Shahriar, and S. N. Al Sunny, "Modeling of cloud-based digital twins for smart manufacturing with mt connect," *Procedia Manufacturing*, vol. 26, pp. 1193–1203, 2018.

[12] B. Kitchenham, "Procedures for performing systematic reviews," Department of Computer Science, Keele University, UK, Keele University. Technical Report TR/SE-0401, 2004.

[13] B. Kitchenham and S. Charters, "Guidelines for performing systematic literature reviews in software engineering," Technical report, EBSE Technical Report EBSE-2007-01, 2007.

[14] E. A. Lee, "Cyber physical systems: Design challenges," in *2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC)*. IEEE, 2008, pp. 363–369.

[15] A. Maccioni and R. Torlone, "Augmented access for querying and exploring a polystore," in *Proc. of the 34th IEEE International Conference on Data Engineering, ICDE*, 2018, pp. 77–88.

[16] A. M. Madni, C. C. Madni, and S. D. Lucero, "Leveraging digital twin technology in model-based systems engineering," *Systems*, vol. 7, no. 1, p. 7, 2019.

[17] A. M. Madni and M. Sievers, "Model-based systems engineering: Motivation, current status, and research opportunities," *Systems Engineering*, vol. 21, no. 3, pp. 172–190, 2018.

[18] M. Schluse, L. Atorf, and J. Rossmann, "Experimentable digital twins for model-based systems engineering and simulation-based development," in *2017 Annual IEEE International Systems Conference (SysCon)*. IEEE, 2017, pp. 1–8.

[19] M. Schluse, M. Priggemeyer, L. Atorf, and J. Rossmann, "Experimentable digital twinsstreamlining simulation-based systems engineering for industry 4.0," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 4, pp. 1722–1731, 2018.

[20] G. N. Schroeder, C. Steinmetz, C. E. Pereira, and D. B. Espindola, "Digital twin data modeling with automationml and a communication methodology for data exchange," *IFAC-PapersOnLine*, vol. 49, no. 30, pp. 12–17, 2016.

[21] M. Stonebraker, "The case for polystores," 2015. [Online]. Available: https://wp.sigmod.org/?p=1629

[22] F. Tao, J. Cheng, Q. Qi, M. Zhang, H. Zhang, and F. Sui, "Digital twin-driven product design, manufacturing and service with big data," *The International Journal of Advanced Manufacturing Technology*, vol. 94, no. 9-12, pp. 3563–3576, 2018.

[23] F. Tao and M. Zhang, "Digital twin shop-floor: a new shop-floor paradigm towards smart manufacturing," *Ieee Access*, vol. 5, pp. 20 418–20 427, 2017.

[24] X. V. Wang and L. Wang, "Digital twin-based WEEE recycling, recovery and remanufacturing in the background of industry 4.0," *International Journal of Production Research*, pp. 1–11, 2018.

[25] C. Weber, J. Königsberger, L. Kassner, and B. Mitschang, "M2DDM – a maturity model for data-driven manufacturing," *Procedia CIRP*, vol. 63, pp. 173–178, 2017.

[26] S. Weyer, T. Meyer, M. Ohmer, D. Gorecky, and D. Zühlke, "Future modeling and simulation of cps-based factories: an example from the automotive industry," *IFAC-PapersOnLine*, vol. 49, no. 31, pp. 97–102, 2016.

[27] S. Yun, J.-H. Park, and W.-T. Kim, "Data-centric middleware based digital twin platform for dependable cyber-physical systems," in *2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN)*. IEEE, 2017, pp. 922–926.

[28] P. Zehnder and D. Riemer, "Representing industrial data streams in digital twins using semantic labeling," in *2018 IEEE International Conference on Big Data (Big Data)*. IEEE, 2018, pp. 4223–4226.