Proceedings of the 2024 Winter Simulation Conference
H. Lam, E. Azar, D. Batur, S. Gao, W. Xie, S. R. Hunter, and M. D. Rossetti, eds.

LEARNING SIMULATION-BASED DIGITAL TWINS FOR DISCRETE MATERIAL FLOW SYSTEMS: A REVIEW

Christian Schwede[1,2], and Daniel Fischer[1]

[1]University of Applied Science and Arts Bielefeld, Bielefeld, NRW, GERMANY
[2]Fraunhofer Institute for Software and System Engineering (ISST), Dortmund, NRW, GERMANY

ABSTRACT

Digital Twins (DTs) play a crucial role in the fourth industrial revolution. In the context of discrete material flow systems, companies under constant competitive pressure seek solutions to minimize costs and maximize performance. Simulation-based DTs can help make optimal decisions in the design, planning, and control of these systems. Such DTs are until now created and updated by domain experts producing costs and are often not considering the advances made in machine learning to improve prediction quality. Learning DTs out of data could be the solution for a broader application. A lot of work has already been done that contributes to this endeavor, yet relevant building blocks originate from different scientific areas resulting in the use of different terminology. Thus we present a holistic review of relevant work and analyze the state of the art based on a new classification scheme deriving relevant building blocks and gaps for future research.

1 INTRODUCTION

Digital Twins are one of the key enablers of digital transformation (Kritzinger et al. 2018). All DTs have in common that they can process a constant data flow from the corresponding physical system to a virtual copy of this system. They use the virtual system to analyze and evaluate possible changes to the system and respond with the best change suitable for a given scenario (Grieves and Vickers 2017). Following this definition, a DT consists of three parts: A physical system in real space, a virtual system in virtual space, and the connection that ensures steady bi-directional data flow between these two systems. DTs are often distinguished from digital models (DMs) and digital shadows (DSs) with respect to the degree of automation of this data flow (Kritzinger et al. 2018). While DMs have a bidirectional manual data flow, DSs rely on an automated data flow into the virtual system and DTs have an automated flow in both directions.

DTs are used in a lot of industries, ranging from classical manufacturing to defense, automotive, and recycling. The paper at hand focuses on the manufacturing industry, more generally discrete material flow systems (DMFS). They can be characterized as systems processing discrete objects (parts) that move at regular or irregular intervals along transportation routes or conveyor lines, comprising production and logistic systems (Arnold and Furmans 2006). DMs have been used to design, plan, and control these DMFS for decades, e.g., in the context of material flow simulations, logistic assistance systems (Zajac and Schwede 2016), and the digital factory (Thiede et al. 2013). With DSs and DTs the focus from models for single use cases shifted to a holistic approach to have a virtual copy during the whole life cycle of the DMFS (Abdoune et al. 2023). The aspect of an automated data flow from the model to the physical system is not all that relevant for the management of DMFS, since human decision-makers are in the loop and decisions are not time-critical within seconds. Therefore, we will from here on refer to DSs and DTs as digital twins. Aside from the aspect of representing the current state as well as managing the historical data of the physical system, a core task of DTs is predicting behavior and evaluating changes. Especially in the area of DMFS,

979-8-3315-3420-2/24/$31.00 ©2024

the use of discrete-event simulation (DES) within DTs is widespread (Korth et al. 2018), which is why we will focus our work on Simulation-based digital twins (SBDTs) (Lugaresi and Matta 2021).

Based on the rise of attention in Artificial Intelligence (AI) and Machine Learning (ML) and supported by the aspect that DTs basically are a huge storage of Big Data, the topic of auto-generating or learning DTs from data has recently become an interesting topic in research and application for three main reasons: The broad application for SBDTs has been hindered by the difficulty to create them. On the one hand, creating (simulation) models of DMFS is still a labor-intensive task for experts (Charpentier and Véjar 2014; Lugaresi and Matta 2021). On the other hand, a DT once created has to be kept up to date when the underlying system changes (Friederich et al. 2022; Lugaresi and Matta 2021), which again can only be done manually by experts (Denno et al. 2018). Thus, *reducing the manual expert labor to create DT models* (1) and *automatically updating the DT models* (2) are two of the reasons for learning DTs from data. Updating must be differentiated from synchronizing. While synchronization based on transaction data is the standard behavior of every DT, updating refers to structural changes of the model based on changes in the master data of the system (e.g., new processes, resources, or changed behavior). Since the accuracy of the prediction is an important aspect of decision-making (Grieves 2022), *improving the prediction accuracy of the DT model* (3) e.g., by using ML is the third reason.

Getting an overview of the state-of-the-art in this area is not trivial since contributions are made from a wide range of different scientific areas. From the DT community, *data-driven DTs* are investigated aiming at using data to create and update DTs for production environments (Friederich et al. 2022). Besides that, coming from the area of software automation, low code approaches are carried out to create process-aware digital twin cockpits (Bano et al. 2022). From the simulation community, work has been done in *automated model generation* (Goodall et al. 2019) and *data-driven simulation* (Biesinger et al. 2019). From the business process engineering field, *process mining* deals with generating predictive models from event log data (van der Aalst 2012) mainly focusing on business processes in general rather than manufacturing. Production system identification aims at bridging this gap focusing on information about infrequent events (Denno et al. 2018). Lastly, from the data science community partial models are learned from big data, that could be relevant for an overall SBDT for DMFS: *Process time prediction* (Müller and Grumbach 2023), *product quality prediction* (Czimmermann et al. 2020), *predictive maintenance* (Nunes et al. 2023).

The aim of this paper is to provide an overview of the different contributions, identifying useful approaches, building blocks as well as existing gaps. Aside from looking at the scientific field, the application area, the methods used, and the data required, we will have a look at the reasons for learning the DT in the specific case. Furthermore, we propose a novel classification scheme based on the single model components needed to create a SBDT for DMFS. Relevant literature contributions are evaluated based on the proposed framework. Our three main contributions in the area of learning SBDTs for DMFS are:

1. Identification of relevant scientific areas contributing to the topic
2. A novel classification scheme to identify relevant building blocks and gaps of a given solution
3. Overview of research gasps and existing building blocks in the current state-of-the-art

The rest of the paper is organized as follows: In Section 2 we provide a brief overview of related work in the different areas mentioned above. Section 3 describes the research methodology used for literature gathering and the novel classification scheme is presented. Section 4 summarizes the main findings of the literature review, followed by Section 5 which discusses the findings and gaps. We conclude the paper with Section 6 where we give an outlook on future research possibilities.

2 RELATED WORK

In this section, we review the relevant literature beginning with literature reviews that deal with learning SBDTs for DMFS. Then we will summarize the work done in the four relevant research areas dealing with learning SBDTs that we introduced in the last part of section 1: Process mining, data-driven simulation, data-driven DTs, and ML-based component behavior prediction that consists of process time prediction, product quality prediction and predictive maintenance.

2.1 Learning Digital Twins

Even though there are several literature reviews concerning DTs in general (Liu et al. 2021; Melesse et al. 2020) the subject of how SBDTs especially in the area of DMFS can be learned out of data has not been addressed holistically so far to the best of the authors knowledge. We believe this is the case because of relevant work being spread over the four following scientific areas. Adamenko et al. differentiate two types of DT creation methods: data-driven and system-driven (Adamenko et al. 2020). While data-driven approaches lead to model-free DTs used as input for ML, system-driven approaches are models created by experts and mainly used in the area of SBDTs. Adamenko et al. state that a combination of both approaches would be optimal without going into further detail. Learning DTs would be such a combination.

2.2 Process Mining

Process Mining consisting of several sub-tasks, roots from business process modeling and aims at automatically exploiting knowledge from event log data that is available in IT systems about business processes. *Process discovery* is used to generate a process model based on Petri-nets that can be transformed into a simulation model. *Process conformance checking* compares real and discovered process models. *Process enhancement techniques* aim to extract performance information of the process (Pourbafrani et al. 2020), while *process prediction* tries to predict relevant performance measures from the event logs (Tax et al. 2016). There has been a lot of work on process mining in recent years. Even though business processes in general are far less complex than material flow systems, we identified a couple of contributions that could be beneficial for the generation of SBDTs for DMFS. We will present the publications clustered by the components that they learn and not in temporal order.

Rozinat et al. dive into subfields of process mining, namely decision mining and organizational mining, and create a simulation model from the logs deriving different aspects or sub-models from the data. Typical for process mining they learn part-type-based process paths for a 1:1 relationship, not considering supply parts or divergence of the part flow, as well as used resources and process time models. Additionally, they use decision trees to learn part type-based routing and priority rules (Rozinat et al. 2009).

Pourbafrani et al. learn a system dynamics simulation model from the event log which gives insight into parameter influences on business performance indicators. With respect to SBDTs, they learn the standard information of process path for 1:1 part transformation and used resources (Pourbafrani et al. 2020).

Not only focusing on parts, more detailed information about resources is also crucial. Event logs contain valuable insights about resources and their scarcity. Martin et al. are able to predict resource availability by considering information about rare events and digging deeper than just modeling standard availability. They extend their work including process structures and batch processing knowledge in the model, thus raising the accuracy of process execution (Martin et al. 2017).

In the area of process prediction, a focus is set on predicting the next process that will be executed. Thus, dealing with learning situation-specific routing rules (Moon et al. 2021; Evermann et al. 2017) and additionally complex process time models (Tax et al. 2016). All three publications use deep learning for prediction (large language models, recurrent neural networks, and long short-term memories).

Martin et al. focus on the prediction of interarrival times of parts at the entry points of a system in the context of business processes. This approach offers the possibility to integrate queuing behavior, making it valuable for practical applications (Martin et al. 2016).

Bano et al. is the only approach dealing with generating DTs directly. They use a low-code approach to create a “process-aware digital twin cockpit”. Their DT is generated iteratively by log data derived from sensors. The discovered models serve as a foundation for a code generator to create DTs as well as for the user interface to display individual performance indicators and the status of the system. Since the approach generates data classes to capture the data received, it is the closest approach to auto-generating a complete DT by combining the scientific areas of process mining and software engineering. Nevertheless generating simulation models is not part of the approach (Bano et al. 2022).

A different approach aims to predict the remaining cycle times of products which can be seen as an alternative or supplement to simulation-based cycle time prediction. Also utilizing process mining, Choueiri et al. use multi-linear regression and a transition system to predict the cycle times. Being one of the few approaches applied in a manufacturing environment they achieve to learn simple process time models, a process path based on part-types for a job shop problem without considering supply parts or routings.

The last work goes beyond the field of process mining, called *product system identification* (Denno et al. 2018). They focus on manufacturing systems in a flow job setting and use SCADA events including resource failures to generate the model. In addition to process mining, they use genetic programming and probabilistic neural networks to generate a validated information model and create a mechanism to update the model based on new events. They argue that process mining is not complex enough to model all behaviors of DMFS and achieve auto-generation advances, especially for resource models predicting capacities for parallel processing, standard and exceptional availability as well as simple priority rules.

2.3 Automated Model Generation and Data-Driven Simulation

For the area of material flow simulations, two streams become apparent: automated model generation (Milde and Reinhart 2019) or automatic model generation (Huang et al. 2011) and data-driven simulation (Charpentier and Véjar 2014). Both approaches aim at the use of data to reduce the manual efforts to perform a simulation study. 35% of the effort can be assigned to the model development and 28% to the data analysis (Milde and Reinhart 2019). Even though the focus is not set to DTs, the simulation model aspect is highly relevant for SBDTs. In both areas, there are approaches reaching from automated parametrization of existing models and automated reconfiguration of existing model elements to auto-generation of complete models. Milde and Reinhart auto-generated the simulation model of a job shop manufacturing system based on artificially generated event logs and resource error data. They can generate process paths, needed resources, process and setup times as well as priority rules and routing rules. The custom-tailored algorithm generates a core manufacturing simulation data model to be imported for simulation into a standard simulation software (Milde and Reinhart 2019).

Charpentier and Véjar use spatial-temporal data from a smart manufacturing environment. With individual programming, they can generate process paths of part types and part type-based routing rules as well as a process time model and a localization of the resources within the shop floor. They also learn M:N part transformations and design a strategy for a delayed update of the model when the online data changes over time (Charpentier and Véjar 2014).

Another approach uses shop floor operational data to update the process time model of the manually modeled simulation of a remanufacturing scenario (Goodall et al. 2019). Lugaresi and Matta entail process mining in their approach to learn a process structure out of event log data. They can generate a model for the main part flow including resources, process times, and quota-based routing rules. They also derive part-specific frequencies of part arrival in the source of the system. Furthermore, they stress that model tuning is an important part of simulation since data from real DMFS can be noisy and the right level of abstraction has to be found (Lugaresi and Matta 2021).

Finally, Smith also generates the model directly from spatial-temporal data based on an individual programming approach. Generating locations of resources, process time models, and the capacity of the resources to process parts in parallel as well as a quota-based routing rule (Smith 2015).

2.4 Data-Driven Digital Twin

Whereas the areas so far focused mainly on generating petri-nets or discrete-event simulation, this area focuses on generating DTs for DMFS directly (while rarely considering simulation). Biesinger et al. show how important data about resources, products, and process information is to build a data-driven DT

(Biesinger et al. 2019). Kumbhar et al. use data in combination with process mining to generate knowledge about process structures. They use SCADA events and argue that these event log data sources provide vital information for such DTs. As one of the few works, they manage to learn N:1 relationships, including supply parts into their transformation model (Kumbhar et al. 2023). Friederich et al. have the most holistic approach to generating a SBDT for DMFS from our point of view. They use state, event data, and condition monitoring data to combine methods from process mining and ML to generate and update a SBDT (Friederich et al. 2022). Liu et al. in contrast use knowledge graphs as a different structure to project data knowledge for a DT. Their insights about updating DT models are valuable (Liu et al. 2023). The last two contributions coin the term "digital twin shop floor" when developing a data-driven DT. Zhang et al. use ML models to predict time and quality models (Zhang et al. 2024). Zhuang et al. on the other hand mention the importance of big data for prediction purposes and develop a framework for comparing simulation and data prediction accuracy (Zhuang et al. 2018).

2.5 ML-Based Component Behavior Prediction

The last field we consider originates from the data science community and is focused on using Ml-based regression to predict certain partial aspects system. We consider process time model prediction, product quality prediction, and predictive maintenance relevant to DMFS. Some work has recently been performed on learning complex situation-specific process time models using ML (Bender et al. 2022; Müller and Grumbach 2023; Rizzuto et al. 2021). Besides predicting process time, several researchers use ML to evaluate the quality of products (Muhr et al. 2020; Lehr et al. 2020; Czimmermann et al. 2020). Often summarized under the term anomaly detection as well, visual information of cameras or other sensor information is used to assess the quality of a product. The last cornerstone of ML-based component behavior prediction is predictive maintenance. The topic is well-studied and results can be used to predict resource unavailability (Nunes et al. 2023).

3 METHOD

For our classification of relevant literature, we conducted a systematic literature review (SLR) according to a proposed framework (Kitchenham et al. 2009). We approached the following research questions:

1. What are the reasons to learn parts of the SBDT?
2. Which parts of SBDTs are learned in the context of DMFS?
3. What are the current gaps in the literature?

3.1 Keyword Search and Criteria

To cover as much ground as possible, we used Google Scholar to search for relevant publications. Scholar searches a wide variety of databases and sort them according to their relevance for the provided search strings. The quality of the search is highly determined by the words and synonyms used. As stated, the contributions of DT learning come from different scientific areas and use a lot of different wordings for similar topics (Jones et al. 2020). Thus, we tried to make up for that with a high semantic variance. Combined with logical conjunction and disjunction we used the strings in Table 1. The statement in the first column addresses the target model (DT or simulation) and was combined with all statements in the second column at the same time. The first row addresses the aspect of data as the source, the second is dedicated to the reason for learning (e.g., generation or update), the third is dedicated to the application domain of DMFS and the fourth is dedicated to specific terms from the four scientific areas presented in section 2.

Table 1: Search strings used (The asterisk tells the search engine to accept anything after the symbol).

	AND (("data-driven") **OR** ("data*"))

((“digital twin”) **OR** (“simulation))	**AND** ((“adaptive”) **OR** (“update*”) **OR** (“synchro*”) **OR** (“online”) **OR** (“self*”) **OR** (“generat*”))
	AND ((“manufacturing”) **OR** (“production*”) **OR** (“scheduling”) **OR** (“iot”)” **OR** (“shop floor”) **OR** (“cps”) **OR** (“cyber-physical*”) **OR** (“supply chain”) **OR** (“warehouse”) **OR** (“assembly”) **OR** (“material flow”) **OR** (“shop floor”))
	AND ((“process mining”) **OR** (“automatic model generation”) **OR** (“production system identification”) **OR** (“predictive maintenance”) **OR** **((**“prediction”) **AND** (“product quality”) **OR** (“process time”)) **OR** (“process”) **OR** (“cockpit”) **OR** (“”))

This way, we researched a total amount of 245 papers. To distill the relevant ones for the classification, we applied the following criteria:

1. The paper was written in English.
2. The paper was published in a journal, conference proceedings, magazine, and book.
3. Duplicated papers were only considered once.
4. The paper specifically focuses on the application of DMs in the context of DMFS.
5. In the context of the above applications, all sectors are feasible.
6. The paper provides a building block to learn parts of a SBDT from data in the given context.
7. The paper mentions creation objectives, model components, and the methods used.

Following these rules, we identified a total of twenty-two articles to sort into our classification scheme. Lots of papers were excluded because they did not match the learning criteria (e.g., only synchronization) or had no DMFS as an application domain.

3.2 Classification Scheme

Our classification scheme is based on the basic model components needed to describe DMFS (Figure 1). Since there has been no scheme focusing on learnable components of DMFS the classes were derived from existing literature (Arnold and Furmans 2006) and standards (Simulation Interoperability Standards Organization 2012) on DMFS. The features were derived from the contributions analyzed for the review,

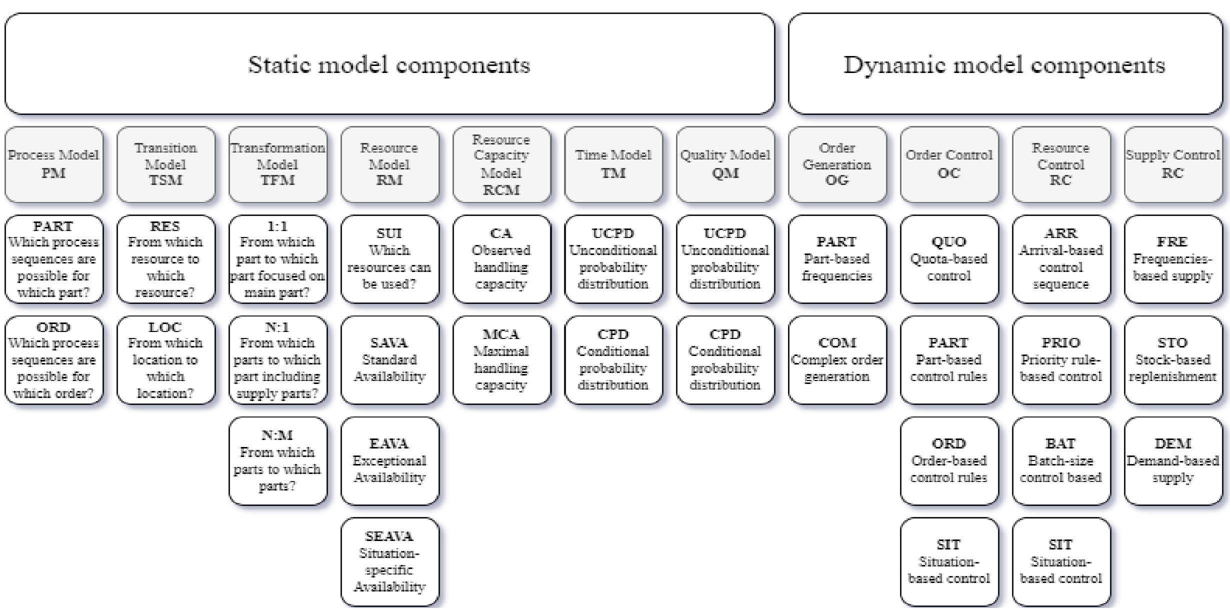

Figure 1: Classification scheme.

with one exception: since there were no contributions focusing on "supply control", the features were created based on existing supply strategies categorizations (Hellingrath et al. 2004).

DMFS can be described by a set of static and a set of dynamic components (Arnold and Furmans 2006). *Static components* describe the possibility space of the system (e.g., processes that can be performed, resources that can be used), while *dynamic components* define the concrete material flow for a certain part or order (e.g., routing rules). Static components are parts, resources, and processes (Simulation Interoperability Standards Organization 2012). Parts are transformed by processes using resources, sometimes based on orders. Processes have a predecessor-successors relationship (*process model*). The transformation can have an impact on spatial position (*transition model*) and physical properties of the parts (*transformation model*). It needs resources to be executed (*resource model*), resources have the capacity to handle parts in parallel (*resource capacity model*) and the process takes time (*time model*). It also affects the quality of the parts involved (*quality model*).

The process model describes the possible process flows for different part types or even individual orders. The transition model describes the possible movement of the parts from one resource to another or from one spatial position to another. The transformation model describes which parts can be transformed into which parts (e.g., assembly, packing, unpacking, forming) and can be based on a 1:1, N:1, or N:M cardinality. The resource model describes the resource options that are needed to execute the process (e.g., transport by automated guided vehicle or forklift and driver) and states the standard, exceptional, and situation-based availability. The latter allows the prediction of unavailability in a certain situation (e.g., in the case of predictive maintenance). The resource capacity model describes the observed or maximum capacity of a resource. The time model describes the process time that passes while the quality model describes the possible impact on the parts' qualities. Both models can be unconditional probability distributions (e.g., uniform or normal distribution) or conditional probability distributions keeping specific characteristics of the situation into account (e.g., deep neural networks).

Dynamic components are used to define the concrete dynamic material flow within the DMFS. There are four components *order generation*, *order control*, *resource control, and supply control*. Order generation defines the load the system must process. We differentiate between part-based frequencies which are defined at the sources of a system and complex order generation, which generates high-variant orders based on historical customer demand data. Order control defines how parts are processed, sometimes referred to as routing rules (Milde and Reinhart 2019). There can be simple quota-based mechanisms (e.g., 30 % of the parts go to process 1 and 70% to process 2), and there can be rules based on part types for parts with low variants or order-specific routings for high-variant parts. The most complex routing would be situation-based since they take e.g., resource and part availability as well as delivery dates into account. Resource control defines how resources decide to handle processing requests, also sometimes referred to as priority rules (Milde and Reinhart 2019). The control can be based only on the arrival sequence of the request (e.g., FIFO, LIFO). It can be based on priority rules based on attributes of the part or order (e.g., shortest-job-first, due date) including combinations of these simple rules. It can describe a batch-based processing defining batch sizes for different part types or can be situation-specific trying to optimize the processing based on defined objectives. Supply control describes how supply parts are provided. It can be based on frequencies (e.g., 500/h), based on stock monitoring (e.g., Kanban), or demand-based (e.g., JIT).

The specific realization of the respective model component can be read as a complexity degree from top to bottom (e.g., a model is less complex that assigns an unconditional probability distribution as a time model to a process than an unconditional one, gaining the possibility to predict the process lead time for a specific situation).

4 RESULTS

The proposed classification scheme was applied to a collection of texts identified with the mentioned literature technique (Kitchenham et al. 2009). In Table 2 we present the main findings. The relevant papers are classified first with respect to the reason why model learning was used (reduce creation effort (R1), update the DT (R2), improve prediction accuracy (R3)) and second according to the static and dynamic components that are learned from data and which degree of complexity the learned model component has.

Table 2: Summary of the literature review.

Publication	R1	R2	R3	PM	TSM	TFM	RM	RCM	TM	QM	OG	OC	RC	SC
Bano et al. 2022	✓			PART	RES	1:1	SUI		UCPD					
Biesinger et al. 2019	✓	✓				1:1	SUI		UCPD					
Charpentier and Vejár 2014	✓	✓		PART	LOC	M:N			UCPD			PART		
Choueiri et al. 2020			✓	PART	RES				UCPD					
Denno et al. 2018	✓	✓		PART	RES	1:1	EAVA	CA	UCPD			PART	ARR	
Evermann et al. 2016	✓											SIT		
Friederich et al. 2022	✓	✓	✓	PART	RES	1:1	SEAVA		UCPD					
Goodall et al. 2019		✓							UCPD					
Kumbhar et al. 2023	✓			PART	RES	N:1	SUI		UCPD					
Liu et al. 2023		✓		PART	RES	N:1	SUI							
Lugaresi and Matta 2021	✓			PART	RES	1:1	SUI		UCPD		PART	QUO		
Martin et al. 2015	✓										PART			
Martin et al. 2016	✓						EAVA							
Martin et al. 2017	✓			PART									BAT	
Milde and Reinhart 2019	✓			PART	RES	1:1	SUI		UCPD			PART	BAT	
Moon et al. 2021	✓											SIT		
Pourbafrani et al. 2020	✓					1:1	SUI		UCPD					
Rozinat et al. 2009	✓			PART		1:1	SUI		UCPD		PART	PART		
Smith 2015	✓			PART	LOC			CA	UCPD			QUO		
Tax et al. 2016	✓								CPD			SIT		
Zhang, J. et al. 2023			✓						CPD	CPD				
Zhuang et al. 2018			✓						CPD	CPD				
SUM	**17**	**6**	**4**	**12**	**10**	**11**	**11**	**2**	**16**	**2**	**3**	**9**	**3**	**0**

To answer our first research question as we can see from the summary, almost all publications learn components of the DT from data to reduce manual modeling effort. Only six papers focus on updating the model components and just four papers deal with improving prediction quality. The most holistic update approach is performed on a semantic web-based DT (Liu et al. 2023). A memorizing-forgetting model is used to classify new data points over time in long-term and short-term knowledge. Another approach is based on constant updates of a genetic programming population considering SCADA events as individuals (Denno et al. 2018). The other approaches either focus on updating probability distributions with new data (Goodall et al. 2019; Friederich et al. 2022) or integrate or delete components of the model based on a delayed reaction to changing event logs (Biesinger et al. 2019; Charpentier and Véjar 2014). All but one paper that has a focus on improving the prediction qualities chooses this as the only reason to learn the model component from data. The reason here is surely the complexity of integrating ML into a DT framework without increasing manual work that must be done for data preparation, feature selection, and meta-parameter optimization. The only approach that integrates automated ML into a DT framework conceptually (Friederich et al. 2022), only validates it on predicting resource unavailability due to disruptions.

To answer the second research question looking at the static model components, for the process model all contributions focus on part type-based process graphs, which can only work for low variance products. The transition model is normally described by possible resource-to-resource connections (usually workstations). Only two contributions use spatial locations since they use not only event logs but spatial data from autonomous moving vehicles. For the transformation model, the vast majority only focuses on simple 1:1 part transformation. Two papers focus on more complex transformation with supply parts and only one paper considers the general M:N relationship, where individual programming is used to retrieve the information from the data. For the resource model normally the suitability of resources for a certain process is learned. Two papers focus also on standard and exceptional availability, adding data about resource status changes into their algorithms. Only one paper focuses on using ML-based availability prediction using condition monitoring data and combines methods from process mining and ML. The Resource capacity model is considered by only two contributions focusing on the capability of parallel processing as observed. No paper addresses to learn the maximal capacity of resources. The time model is the most common component, but the focus is nearly always on simple unconditional distributions. Only two papers focus on ML-based process time prediction. The same papers are the only ones that integrate product quality prediction.

The dynamic components are far less considered in the contributions. Order Generation is done based on learning part-wise frequencies. Complex order generation is not considered. Order control is learned on nearly all levels: Quota-based, Part-wise, and also based on the current situation in the field of process prediction using Deep Learning (large language models, recurrent neural networks, or long short-term memories). Resource control is done only by three contributions: one based on arrival sequences learning FIFO and LIFO behavior and two based on batch control, learning the batch size used for a process by a resource. We found no work for learning supply control behavior.

5 DISCUSSION

To summarize the findings: there is still work to do until all components of SBDTs for DMFS are generated automatically and additionally take advantage of the maximal prediction quality. One big challenge is surely the data availability and quality. Approaches based on standardized event logs (like process mining) or smart factory environments have demonstrated the power of good data quality for the learning of model components. It would be interesting to switch the task of process mining from “What can I get from as little data as possible?” to “Which data do companies need to capture to be able to learn the whole SBDT?”. Even though the results show that for a lot of model components building blocks are already available, there is the need to integrate the building blocks into one framework – especially dealing with the issue of automating the ML process completely to not only improve prediction quality but reduce manual expert labor as well. Furthermore, more complex smart manufacturing (e.g., job shop, matrix-production) environments with high-variant products are still underrepresented. Most work is based on standard parts with a single process sequence, ignoring supply parts and complex product structures. Especially the business process focus of the process mining area lacks the complexity of DMFS (Denno et al. 2018). Finally, there is a need for more work on the dynamic components. While order control is covered quite well, order generation, resource control, and especially supply control needs to be focused in the future.

To finally answer the third research question about the existing gap in the literature, we summarize the points discussed above:

1. Since a lot of building blocks for the learning model components exist, the task at hand is to create frameworks that integrate all the building blocks, focusing on the automation of the ML process.
2. More work should be dedicated to the learning of dynamic system behavior. Especially order generation, resource control, and most of all supply control are still blank spaces.
3. With a lot of work being available on flow job-based MFS with standardized low-variant products, more work should be dedicated to complex job shop environments with high-variant products.
4. As data standardization (as with event logs) boosts the advances in the field of learning models, future analyses should be focused on what kind of data would be necessary to fully automate the generation and update of SBDTs.

6 CONCLUSION AND OUTLOOK

This paper focuses on learning SBDS for DMFS. After introducing the numerous relevant scientific areas that contribute to this overall objective, we introduced a novel classification scheme based on breaking down DMFS into static and dynamic components and their possible realizations. These realizations served as a reference to what could possibly be learned in the context of these systems. Conducting a systematic review of respective literature, we analyzed why learning was applied, which components were learned to which degree and had a look at data requirements and the methods applied. The three main contributions were (1) "identification of relevant scientific areas contributing to the topic" detailed in section 2, (2) "a novel classification scheme to identify relevant building blocks and gaps of a given solution" detailed in section 3.2 and (3) "an overview of research gasps and existing building blocks in the current state of the art" detailed in section 4 and summarized in section 5. Some of the details of the analysis, such as the detailed methods used in the publications could not be addressed within the limitations of this paper.

For future work, it would be interesting to broaden the view to comparable application domains. Especially in the domain of continuous material flow, there have already been some notable advances in auto-generating SBDTs (Martinez et al. 2018). It would also be of interest to think about a standard data format (such as event logs or even based on event logs) to be able to completely learn SBDTs. This standard could be used as a blueprint for the industry and their IT service provider to answer the question of what data should be captured in their material flows. Furthermore, there is a need for more publically available industry data sets for the research community to advance faster in this area. Finally, an interesting area only recently arisen is the use of large language models to generate models out of process descriptions and application domain expert knowledge (Kourani et al. 2024).

REFERENCES

Abdoune, F., M. Nouiri, O. Cardin, and P. Castagna. 2023. "Digital Twin Lifecycle: Core Challenges and Open Issues." In *Service Oriented, Holonic and Multi-Agent Manufacturing Systems for Industry of the Future,* edited by T. Borangiu, et al., 157–167. Cham: Springer International Publishing.

Adamenko, D., S. Kunnen, R. Pluhnau, A. Loibl, and A. Nagarajah. 2020. "Review and Comparison of the Methods of Designing the Digital Twin." *Procedia CIRP* 91:27–32.

Arnold, D. and K. Furmans. 2006. *Materialfluss in Logistiksystemen*. 5th ed. Berlin: Springer.

Bano, D., J. Michael, B. Rumpe, S. Varga, and M. Weske. 2022. "Process-Aware Digital Twin Cockpit Synthesis From Event Logs." *Journal of Computer Languages* 70:101121.

Bender, J., M. Trat, and J. Ovtcharova. 2022. "Benchmarking AutoML-Supported Lead Time Prediction." *Procedia Computer Science* 200:482–494.

Biesinger, F., D. Meike, B. Kraß, and M. Weyrich. 2019. "A Digital Twin for Production Planning Based on Cyber-Physical Systems: A Case Study for a Cyber-Physical System-Based Creation of a Digital Twin." *Procedia CIRP* 79:355–360.

Charpentier, P. and A. Véjar. 2014. "From Spatio-Temporal Data to Manufacturing System Model." *Journal of Control, Automation and Electrical Systems* 25:557–565.

Choueiri, A. C., D. M. V. Sato, E. E. Scalabrin, and E. A. P. Santos. 2020. "An Extended Model for Remaining Time Prediction in Manufacturing Systems Using Process Mining." *Journal of Manufacturing Systems* 56:188–201.

Czimmermann, T., G. Ciuti, M. Milazzo, M. Chiurazzi, S. Roccella, C. M. Oddo, and P. Dario. 2020. "Visual-Based Defect Detection and Classification Approaches for Industrial Applications - A Survey." *Sensors* 20:1459.

Denno, P., C. Dickerson, and J. A. Harding. 2018. "Dynamic Production System Identification for Smart Manufacturing Systems." *Journal of Manufacturing Systems* 48:192–203.

Evermann, J., J.-R. Rehse, and P. Fettke. 2016. "Predicting Process Behaviour using Deep Learning." *arXiv preprint arXiv: 1709.04608*.

Friederich, J., D. P. Francis, S. Lazarova-Molnar, and N. Mohamed. 2022. "A Framework for Data-Driven Digital Twins of Smart Manufacturing Systems." *Computers in Industry* 136:103586.

Goodall, P., R. Sharpe, and A. West. 2019. "A Data-Driven Simulation To Support Remanufacturing Operations." *Computers in Industry* 105:48–60.

Grieves, M. 2022. "Intelligent Digital Twins and the Development and Management of Complex Systems." *Digital Twin* 2:8.

Grieves, M. and J. Vickers. 2017. "Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems." In *Transdisciplinary Perspectives on Complex Systems*, edited by F.-J. Kahlen, S. Flumerfelt, and A. Alves, 85–113. Cham: Springer.

Hellingrath, B., M. Keller, and M. Witthaut. 2004. "Klassifizierung von Dispositionsstrategien großer Logistiknetze." Technische Universität Dortmund.

Huang, Y., M. D. Seck, and A. Verbraeck. 2011. "From Data to Simulation Models: Component-Based Model Generation With a Data-Driven Approach." In *2011* Winter Simulation Conference (WSC), 3719–3729 doi: 10.1109/WSC.2011.6148065.

Jones, D., C. Snider, A. Nassehi, J. Yon, and B. Hicks. 2020. "Characterising the Digital Twin: A Systematic Literature Review." *CIRP Journal of Manufacturing Science and Technology* 29:36–52.

Kitchenham, B., O. Pearl Brereton, D. Budgen, M. Turner, J. Bailey, and S. Linkman. 2009. "Systematic Literature Reviews in Software Engineering – a Systematic Literature Review." *Information and Software Technology* 51:7–15.

Korth, B., C. Schwede, and M. Zajac. 2018. "Simulation-Ready Digital Twin for Realtime Management of Logistics Systems." In *2018 IEEE International Conference on Big Data (Big Data)*, December 10–13, 2018, Seattle, WA, 4194–4201.

Kourani, H., A. Berti, D. Schuster, and W. M. P. van der Aalst. 2024. "Process Modeling with Large Language Models." *arXiv preprint arXiv: 2403.07541*.

Kritzinger, W., M. Karner, G. Traar, J. Henjes, and W. Sihn. 2018. "Digital Twin in Manufacturing: A Categorical Literature Review and Classification." *IFAC-PapersOnLine* 51:1016–1022.

Kumbhar, M., A. H. Ng, and S. Bandaru. 2023. "A Digital Twin Based Framework for Detection, Diagnosis, and Improvement of Throughput Bottlenecks." *Journal of Manufacturing Systems* 66:92–106.

Lehr, J., A. Sargsyan, M. Pape, J. Philipps, and J. Krüger. 2020. "Automated Optical Inspection Using Anomaly Detection and Unsupervised Defect Clustering." In *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 8–11 September 2020, Vienna, Austria, 1235–1238.

Liu, M., S. Fang, H. Dong, and C. Xu. 2021. "Review of Digital Twin About Concepts, Technologies, and Industrial Applications." *Journal of Manufacturing Systems* 58:346–361.

Liu, S., P. Zheng, L. Xia, and J. Bao. 2023. "A Dynamic Updating Method of Digital Twin Knowledge Model Based on Fused Memorizing-Forgetting Model." *Advanced Engineering Informatics* 57:102115.

Lugaresi, G. and A. Matta. 2021. "Automated Manufacturing System Discovery and Digital Twin Generation." *Journal of Manufacturing Systems* 59:51–66.

Martin, N., B. Depaire, and A. Caris. 2015. "Using Process Mining to Model Interarrival Times: Investigating the Sensitivity of the ARPRA Framework." In *2015* Winter Simulation Conference (WSC), 868–879 doi: 10.1109/WSC.2015.7408214.

Martin, N., M. Swennen, B. Depaire, M. Jans, A. Caris, and K. Vanhoof. 2017. "Retrieving Batch Organisation of Work Insights From Event Logs." *Decision Support Systems* 100:119–128.

Martinez, G. S., S. Sierla, T. Karhela, and V. Vyatkin. 2018. "Automatic Generation of a Simulation-Based Digital Twin of an Industrial Process Plant." In *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*, October 21–23, 2018, Washington, DC, 3084–3089.

Melesse, T. Y., V. Di Pasquale, and S. Riemma. 2020. "Digital Twin Models in Industrial Operations: A Systematic Literature Review." *Procedia Manufacturing* 42:267–272.

Milde, M. and G. Reinhart. 2019. "Automated Model Development and Parametrization of Material Flow Simulations." In *2019* Winter Simulation Conference (WSC), 2166–2177 doi: 10.1109/WSC40007.2019.9004702.

Moon, J., G. Park, and J. Jeong. 2021. "POP-ON: Prediction of Process Using One-Way Language Model Based on NLP Approach." *Applied Sciences* 11:864.

Muhr, D., S. Tripathi, and H. Jodlbauer. 2020. "Divide and Conquer Anomaly Detection: A Case Study Predicting Defective Engines." *Procedia Manufacturing* 42:57–61.

Müller, A. and F. Grumbach. 2023. "Predicting Processing Times in High Mix Low Volume Job Shops." In *16th International Doctoral Students Workshop on Logistics, Supply Chain and Production Management*, June 20, 2023, Magdeburg, Germany.

Nunes, P., J. Santos, and E. Rocha. 2023. "Challenges in Predictive Maintenance – a Review." *CIRP Journal of Manufacturing Science and Technology* 40:53–67.

Pourbafrani, M., S. J. van Zelst, and W. M. P. van der Aalst. 2020. "Supporting Automatic System Dynamics Model Generation for Simulation in the Context of Process Mining." In *Business Information Systems: 23rd International Conference, BIS 2020, Colorado Springs, CO, USA, June 8–10, 2020, Proceedings*, edited by W. Abramowicz and G. Klein, 249–263. Cham, Switzerland: Springer International Publishing.

Rizzuto, A., D. Govi, F. Schipani, and A. Lazzeri. 2021. "Lead Time Estimation of a Drilling Factory with Machine and Deep Learning Algorithms: A Case Study." In *2nd International Conference on Innovative Intelligent Industrial Production and Logistics (IN4PL 2021)*, October 28–30, 2021, Valletta, Malta, 84–92.

Rozinat, A., R. S. Mans, M. Song, and W. van der Aalst. 2009. "Discovering Simulation Models." *Information Systems* 34:305–327.

Simulation Interoperability Standards Organization. 2012. Standard for Core Manufacturing Simulation Data – XML Representation. https://www.nist.gov/publications/core-manufacturing-simulation-data-uml-model-standard, accessed 22nd September.

Smith, C. R. 2015. "The Programmatic Generation of Discrete-Event Simulation Models from Production Tracking Data." Master's thesis, Ira A. Fulton College of Engineering and Technology, Brigham Young University, Provo, Utah. http://scholarsarchive.byu.edu/etd/5829, accessed 25th September 2024.

Tax, N., I. Verenich, M. La Rosa, and M. Dumas. 2016. "Predictive Business Process Monitoring with LSTM Neural Networks." *arXiv preprint arXiv: 1612.02130.*

Thiede, S., Y. Seow, J. Andersson, and B. Johansson. 2013. "Environmental Aspects in Manufacturing System Modelling and Simulation—State of the Art and Research Perspectives." *CIRP Journal of Manufacturing Science and Technology* 6:78–87.

Van der Aalst, W. 2012. "Process Mining." *Communications of the ACM* 55:76–83.

Zajac, M. and C. Schwede. 2016. "Cross-Process Production Control by Camera-Based Quality Management Inside a Logistic Assistance System." In *Logistics and Supply Chain Innovation,* edited by H. Zijm, et al., 353–362. Cham: Springer International Publishing.

Zhang, J., J. Liu, C. Zhuang, H. Guo, and H. Ma. 2024. "A Data-Driven Smart Management and Control Framework for a Digital Twin Shop Floor With Multi-Variety Multi-Batch Production." *The International Journal of Advanced Manufacturing Technology* 131:5553–5569.

Zhuang, C., J. Liu and H. Xiong. 2018. "Digital Twin-Based Smart Production Management and Control Framework for the Complex Product Assembly Shop-Floor." *The International Journal of Advanced Manufacturing Technology* 96:1149–1163.

AUTHOR BIOGRAPHIES

CHRISTIAN SCHWEDE is Professor of Big Data Analysis at the University of Applied Science and Arts Bielefeld. He is a member of the Center for Applied Data Science and a Board member of the Institute for Data Science Solutions. Additionally, he is a senior scientist at Fraunhofer ISST in Dortmund responsible for AI in Logistics. His research area is the application of data- and model-driven methods to optimize logistics and production. He specializes in automotive logistics and agent-based decentralized production systems. His email address is christian.schwede@hsbi.de. His webseite is https://www.hsbi.de/personenverzeichnis/christian-schwede.

DANIEL FISCHER is a research assistant at the University of Applied Science and Arts Bielefeld (HSBI). He is currently enrolled in the Data Science major at HSBI. His work involves the application of explainable AI (xAI) methods on deep reinforcement learning agents in flexible job shop production settings. His research interest includes digital twins, simulation, and AI. His email address is daniel.fischer@hsbi.de. His website is https://www.linkedin.com/in/daniel-fischer96.