Simulation-ready digital twin for realtime management of logistics systems

1st Benjamin Korth
Information Logistics and Decision Support Systems
Fraunhofer Institute of Material Flow and Logistics IML
Dortmund, Germany
benjamin.korth@iml.fraunhofer.de

2nd Christian Schwede
Information Logistics and Decision Support Systems
Fraunhofer Institute of Material Flow and Logistics IML
Dortmund, Germany
christian.schwede@iml.fraunhofer.de

3rd Markus Zajac
Information Logistics and Decision Support Systems
Fraunhofer Institute of Material Flow and Logistics IML
Dortmund, Germany
markus.zajac@iml.fraunhofer.de

***Abstract*—Event-discrete simulation is a key method for decision support systems in planning and control of logistics systems. The ability to start a simulation based on the current situation of the system in real-time is central for these systems. In this paper, we present our system architecture that combines a real-time digital twin of logistics systems with simulation logic in a single (modularized) model. This combination not only reduces offline work creating and maintaining such a model and decision support system but also reduces runtime in this critical real-time use-case. The approach is demonstrated in two industrial use cases.**

***Index Terms*—digital twin, simulation, logistics, management**

I. Introduction

Decision Support is an important use-case for digital twins in planning and control of logistics systems. Logistics systems such as warehouses, distribution centers, production environments or supply networks can be considered as complex systems with respect to the interaction of numerous entities in a highly dynamic environment [1], [2]. In such systems, business objects are transformed in processes by resources to fulfill (customer) orders in an efficient way. These processes create an enormous amount of data (Big Data). To achieve efficiency (with respect to costs and performance) the usage of resources in logistics systems has not only to be terminated optimally in advance (resource planning) but also to be steered optimally in real-time in case of unforeseen deviations (controlling). To make good decisions in planning and control decision makers in logistics need to (a) know that there is a problem with the current plan, (b) generate decision scenarios that make sense and (c) evaluate these scenarios with respect to efficiency. To support these needs, a decision support system must have:

1) Knowledge about current state of the system including orders, resources, entities and processes;
2) Knowledge about the future situation of the system including future orders and processes as well as availability of resources and entities;
3) Functions of identifying current and upcoming disruptions;
4) Functions to generate decision scenarios that are feasible;
5) Functions that evaluate these decision scenarios.

The first two requirements ((1), (2)) are basic properties of digital twins being digital representation of real-world objects and processes. In this paper, we want to present an approach that not only considers the present status of a system as focus of the digital twin but also includes the dimension of time. Hence, past and possible future states are elemental parts of our digital model thus covering also requirement (3) and (4) as we will show. Other than common approaches, we split the twin in a dynamic (events in the past, present and future) and static model (process structure, resources, orders, business objects). This allows us to store past, present and future of the system in the same format. It also makes it possible to run a simulation directly based on the static model and one chosen dynamic status. This has the advantage that there is no warm up phase needed and the online-simulation actually has real-time abilities. Furthermore, there is no need for a separate simulation model (that normally would have a rather huge overlap with the digital twin model). Only the logic of passing from one state to another (dynamics) has to be modeled additionally. This reduces adaptation times in case of structural changes in the real world. Analyses from former projects have shown that in the past import into the simulation model and warm-up phases needed 90% of the runtime in logistics use cases. This time can now be saved with our approach.

FIGURE 1 illustrates our vision of a digital twin-based decision support in logistics as a virtual experiment field for planner and controller [3]. The real physical system is virtually represented by the processes resources, business objects and orders in the static process model of the twin. Real-time events produced by scanners or sensors are transmitted to the twin directly or via existing systems such as Warehouse Management Systems (WMS). Thus, the dynamic part of the

978-1-5386-5035-6/18/$31.00 ©2018 IEEE

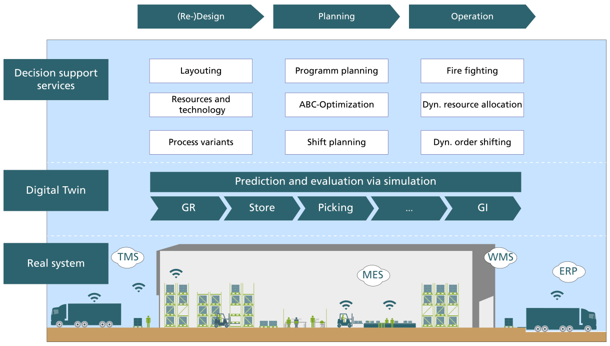

Fig. 1. Application field of digital twin

model is generated enabling present and past situations to be analyzed. Simulation of the material flow is an inherent part of the digital twin allowing prediction of the future of the system. On top of the digital twin lies the layer of decision support services that can for instance be used within so called Logistics Assistance Systems [4]. These services assist the planner and controller in generating feasible decision scenarios based on the digital twin. This can be as simple as supporting the planner with a workflow engine to create scenarios manually or as complex as optimization algorithms that automatically generate the scenarios. On top the picture are the three different phases along the lifecycle of a logistics system: In the design phase the system is created or updated. The digital twin exists and is changed before its physical sibling is implemented or adapted. Decision services include dimensioning, layouting, process design, resource and technology selection. In the planning phase, the digital twin and physical system are already connected and synchronized. Decision support comprises program and resource planning, stock optimization, time-window planning etc. The third phase is the operation phase. Decision support is about prediction of disruptions using online-simulation and firefighting measures such us real-time resource reallocation and order shifting.

In the remainder of this paper, we will have a look at the academic literature (SECTION II) to point out the research gap, present our approach (SECTION III) in detail and explain why we meet the five requirements for decision support systems in planning and controlling of logistics systems. In SECTION IV we present two demonstration scenarios from German industry and conclude with a summary and an outlook (SECTION V).

II. State of Research

The concept of the digital twin originated in 2002 at the University of Michigan [5]. With its core elements of real space and virtual space, as well as the idea that these spaces are interconnected and exchange information over the entire life cycle, this concept contains all the elements of a digital twin. The definition at that time had a strong product focus, so the digital twin was defined as a set of virtual information constructs that fully describes a potential or actual physical manufactured product from the micro atomic level to the macro geometrical level [5]. In the recent past, the strong product reference was loosened, so that the concept of the digital twin was adapted also for the consideration of data-intensive production systems (in particular cyber-physical production systems) [6], [7]. Due to the large amounts of real-time data, the typical characteristics of Big Data becomes noticeable in such systems: Volume (large amount of data), Variety (various types and forms of data), and Velocity (high data generation speed) [6], [8]. However, as both data analysis and the use of simulation to support the development and continuous validation of such systems are becoming increasingly important, digital twins must be able to handle such large amounts of data. Suitable IT systems should form the basis for the realization of a digital twin for such production systems [6], [7], [9].

This adaptation leads to a stronger concretization of the task areas and characteristics of a digital twin, but also to new circumstances, which make an efficient implementation more difficult. These aspects are considered below. In contrast to the above definition, more recent definition approaches and descriptions additionally refer to systems and the information associated with them in the various life-cycle phases. Digital twin is now described as ”a comprehensive physical functional description of a component, product or system that contains more or less all the information that could be useful in all current and subsequent life cycle phases” [10]. A further field of activity in this context is that a digital twin should not only collect all relevant data, but also, based on this data, manage tailor-made models (in particular information and simulation models for the life cycle phase under consideration) and provide them to applications. This means that a digital twin must be able to provide actual (partial) models for different purposes, but not a large model with information of all life cycle phases [10], [11]. Digital twins can therefore be characterized in the following way [10]:

- A digital twin is a linked collection of different types of data (like operation data) as well as different models
- A digital twin evolves with the real system along its life cycle
- A digital twin is able to derive solutions relevant for the real systems (e.g. optimize operation and service)

One of the most important aspects that complicates the implementation of a phase-overlapping digital twin is the fact that information of the different phases of the life cycle is produced by different tools in different formats, so that the information flow between the phases is usually not fluent [9]. Information must therefore be converted laboriously between different formats. Thus, much time is expended in finding the relevant information at all. An approach according to this scheme is described by BOSCHERT and ROSEN. The digital twin aggregates all information from different tools, arranges them phase-specifically and converts them to share them with other phases. In [6] a digital twin and Big Data driven prediction framework for assembly shop-floor is described, which uses suitable big data storage systems to manage large amounts of data.

As already described in the introduction (see section I), decision support in logistics is an important use-case for digital twins. Although concepts for real-time decision support system architectures already exist (e.g. [12]), certain aspects, especially those concerning simulation-based decision support in logistics, are not sufficiently examined and validated. These are aspects that concern the creation of simulation models, but also runtime aspects in the execution of simulations. Today, simulation models are usually created (or generated) for specific simulation tools and represent additional models in the digital twin, which have to be managed and, above all, kept up-to-date. At this point, the question arises whether a model (consisting of static and dynamic parts, see section I), which is always up-to-date, would be sufficient for simulation runs. In this case, it would mean that an additional simulation model does not have to be created (or generated). Furthermore, conventional simulation models often entail the necessity for preliminary simulations in order to put a modeled system into an initial state necessary for the real simulation (so called warm-up phase). This procedure leads to runtime losses. In contrast, a digital twin would in principle be able to create a model which corresponds to an initial state for the simulation, so that preliminary simulations would not be necessary. The work cited does not investigate such questions and approaches.

III. Architecture of a Simulation Ready Digital Twin

Based on the requirements of a decision support system in SECTION I, this section will now present the approach of how a digital twin can be extended with simulation capabilities and how an architecture can be structured.

At first, there has to be a model that represents the real world system. In order to synchronize the model with the real world system a connection must be established. This is a main challenge, because the ability to measure the real state is limited for bigger and complex systems or in high level of detail.

To get a future view of the system or to evaluate decision scenarios, the behavior of the system given any possible state is required. Because a system can have a complex behavior where analytical methods are difficult to apply, the event discrete simulation is a suitable method to transfer a given state to future states. Here it is to be noted that a simulation should not hinder further updates to the digital twin.

Independently of the usage of simulation capabilities, a digital twin is intended to be a knowledge store and provider. So the architecture must allow to store the current state and historical data (that may be generated by simulations) and support users deriving usable informations of the data.

Before the structure of the architecture is explained, first the intentions on how it should work will be presented. In the bottom of FIGURE 2 the digital representation of the current system state is displayed. By this is meant that all relevant properties are present with assigned values. To become a digital twin, a connection to the real world is needed. Since only the changes of the state are necessary to be recognized events are used to update the digital twin (orange hexagons). Virtual scenarios, like in simulations, can start with a given state and use own events to update their own digital twins (green, yellow and red). Changes of a digital twin represents the development from states in the past to the future. This data could be a valuable treasure since every state of the system can be restored. But the data amount can be huge so it should not be stored in the digital twin model, but in a separate store like a "logbook" (upper part of FIGURE 2). There could also be multiple logbooks to store different scenarios.

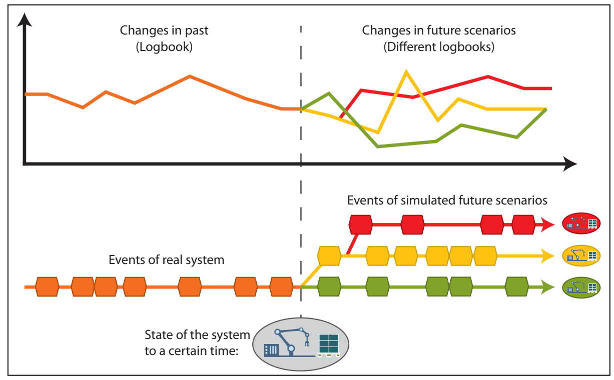

Fig. 2. Approach of how a digital twin is updated in the proposed architecture and how both the state and the changes are stored.

Now that the intentions of a digital twin architecture with simulation capabilities were described the following part explains the approach to create the architecture. First the concept is divided into components with different tasks. After that a description of each component is shown.

To break down the approach into components the following questions could be asked:

- how to represent a system in a digital model?
- how can the model be updated?
- what kind of data occur?
- when is the data needed?
- which functions are needed?

The chosen subdivision is shown in FIGURE 3. First a digital model is needed that could represent the current state of system with all relevant objects, attributes and relationships. In order to create a digital twin, an update mechanism is needed. For this purpose the component called "Event-Controller" is in charge. It contains the knowledge on how an event occurring on the shop floor logistics system can be transferred into changes in the model. Examples for events are barcode or rfid scans, bookings in a warehouse management system or sensor values. Analogous to that the simulation component has the ability to create synthetic events for future model changes in scenarios. These events may be different and numerous compared to real world events because simulation controllers have no limitation in what level of detail the events are generated. The changes of the model and therefore the history can be written to a logbook. It enables restoring every state of

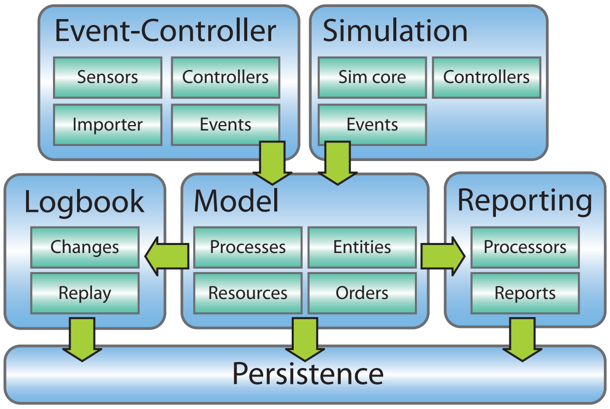

Fig. 3. Architecture of a digital twin.

the model that had occurred. The reporting component is used to create useful information for users like key performance indicators or time series values that could be visualized. Finally a persistence component is required to store a model, the reports of the reporting component and logbooks. In the following the different components of the architecture were explained.

Digital model

The center of our digital twin architecture forms a digital model of real world system that contains all relevant objects with their attributes and relationships to represent a single state of the real world objects at a point in time. To keep the model lean and easy to update, it is important to exclude historical informations and data points for statistical calculations from the model. The model is also capable of informing other components about model changes. Since every model is domain specific, here are some abstract examples of the contained objects:

- real world objects like business objects or resources
- data objects like orders
- processes that transforms business objects

Event-Controller

The leading requirement of a digital twin is, that it is always up-to-date. To achieve this, there must be an link to the real world that transforms a model state to the next state if necessary. In the simplest case, there are sensors that directly measures attribute values. A measured change of value at a point in time can be considered as an event. Since there might not be a sensor for every single relevant attribute, more complex changes could be necessary at an event occurrence. For the logistic domain, we differentiate between logistic events and atomic events. Logistic events are events measured by sensors or reported by resources (e.g. a scan in goods receipt). Atomic events are changes of attributes in the data model of the digital twin. An important task of the event-controller is to translate logistic to atomic events and to carry out the changes in the model. Because they may need more semantic knowledge about the monitored system to update the model, they could be use-case specific.

Reporting

To benefit from the digital twin it is necessary to gain decision-relevant informations from it. Often the current state is not enough for that purpose. In addition, aggregated key performance indicators or time series values are needed to measure the costs and performance like in simulation studies. To derive this data from a digital twin without inflating it by various statistical or historical data, we use reporting processors that monitors the current state (the model) and changes of it. A reporting processor can therefore get values from the twin in a fixed periodic manner or it can subscribe for certain changes in the twin. If such a change occurs, the processor can update its specific and separate data model.

Simulation-Controller

In planning processes, especially if the system in question does not exist yet, experiments could be carried out by simulations. To extend the digital twin by simulation capabilities, it is necessary to depict the behavior of the real world object in controllers. The controllers have access to the current state of the controlled element of the digital twin and can create events for the future. If the controllers interact with each other, they build an agent based simulation. It is important to point out that in this approach data from the twin is not imported into another simulation model of an existing simulation software. Here the simulation controllers only describe how the elements of the digital twin changes over time producing atomic events on the model. Thus the basic static model of the twin is unique and information doesn't need to be modeled and stored redundantly.

Logbook

The logbook of a digital twin logs every atomic event of the model state. It enables restoring every occurred state of a digital twin from an initial state and observing the model development without a repeated simulation. With this feature, it is possible to create a planning or simulation scenario out of a certain state that occurred in the real world. This means that there is no need for a warm-up phase as has to be performed by other state-of-the-art discrete-event material-flow simulators. Also it is possible to derive new reporting data in retrospect by a replaying function of the logbook. The structure of a logbook entry is quite simple and consists of the virtual timestamps, an object reference, the attribute and the new value. If it should be possible to rewind a model state to a state in the past, also the old value of the object must be stored. Since the logbook grow over time, it is possible to save complete snapshots of the model and to cut of logbook entries with timestamps before the snapshot time.

Persistence

The persistence component is needed to save and restore a state of the digital twin. It also is responsible to save logbook entries. Because of the different types of data, the persistence component has an abstract interface to store a digital twin model or logbook entries. The former is an object-relational model where document based databases are suitable. For the large number of logbook entries on the other hand different variants are suitable. Due to performance or space issues, a compressed file format with a high performance database is supported.

IV. Use cases

To demonstrate the concept of a simulation-ready digital twin for management of logistics systems we will describe two industrial projects where the concept was applied in real-world uses cases. In the first one the digital twin of a warehouse was used for shift planning of blue-collar workers and time window planning. The aims of the second project targeted at real-time transparency and short-term control of a production environment of an medium sized metal-processing enterprise.

Decision support system for warehouse optimization

In the highly dynamic business of food logistics, dispatchers have to handle permanent changes in the order volume and composition. There are also external influences like weather or traffic conditions. For a warehouse near Hanover (Germany) the main challenge is the optimization of shift planning and the creation of an optimized time window plan under the mentioned conditions. As shown in FIGURE 4, the site contains an automated high-bay storage area (partly cooled), a picking area (also partly cooled), an inbound and an outbound area with truck ramps. Provision and loading at the ramps are handled by fork lifts. All areas are connected by a multi-level conveyor system. In the outbound direction, the conveyor system leads through a labeling and a pallet wrapper station. Despite velocity and capacities of the different systems being known, there can be unfavorable conditions in which the conveyor system is a bottleneck, which causes a long queue that reaches back to the high storage system. Because of the site structure and the business conditions, a lot of non-linear effects must be considered during the daily planning. A suitable method to get transparency of these effects is the discrete-event simulation, especially if the system in question is too complex for analytical methods. To use the simulation method in a daily business, there are two challenges to solve. First, a dispatcher is rarely a simulation expert. So it is necessary to provide a tool that transfers the decisions to the simulation, runs it and helps the dispatcher to interpret the results. Secondly, the current warehouse situation must be mirrored in the simulation model. If this is done permanently, the model becomes a digital twin of the warehouse.

To solve the first challenge, a web based Logistic Assistance System based on the described framework was developed. The decision support services allow providing task-specific input masks for the dispatchers to create decision scenarios. At first, the current time-window plan is shown. Already at this point, a dispatcher can make adjustments based on his experience. The next step is to create or adjust the shift plan.

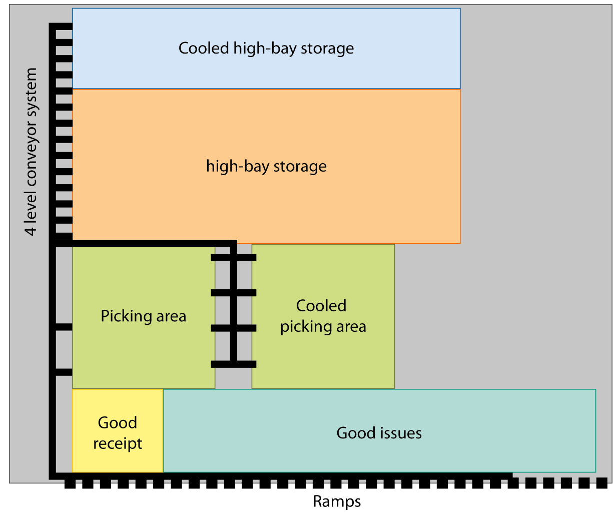

Fig. 4. Schematic layout of the warehouse

When a decision scenario is complete, it can be evaluated by simulation. Therefore a digital twin with simulation capability is needed in the current state of the warehouse. Through the permanently update, no warm-up phase of the simulation is necessary. The simulation plays through the scenario and creates measurements that are aggregated and shown to the dispatcher by the decision support system. He can now see the consequences of his decisions. That is primarily the predictions of the order-finishing times, but also the workload of employees and the conveyor system. With this detailed information, the dispatcher can rework his plan until he gets a satisfiable result. Especially he now knows exactly how many employees are needed in the different areas of the site over time.

The synchronization of a simulation model with the current warehouse state was the much bigger challenge. Due to missing or unknown interfaces of existing IT systems like the ERP-system (enterprise resource planning system), WMS (warehouse management system) or TMS (transport management system) some workarounds have to be made to collect data from these systems. For example the current order states couldn't be queried directly at the ERP-system. Instead, a few manual steps are necessary to export the data to a file and then import it to the model. Since the time window and personal deployment planning is only carried out a few times in each shift, the additional effort is justifiable. Further updates, like blocked docks or disturbed lifts have to be set manually.

To discover the benefit of the decision support system several tests were performed with different subjects. There were experienced dispatchers and inexperienced people with only little knowledge of the site. The latter knew only the average personal team size in each area and the typical workload profile of a day. The test setup was as follows. First every candidate has to make a plan without any support. To do that, they get a realistic time window plan that contains

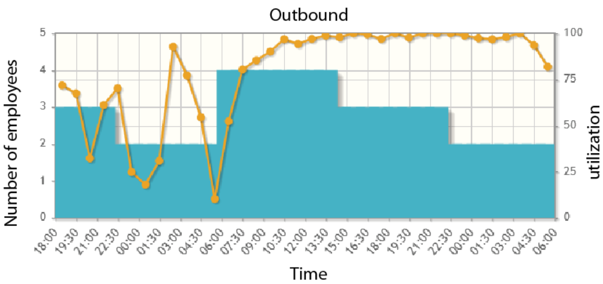

Fig. 5. Utilization of employees in the outbound area in a unsupported scenario of an unexperienced user.

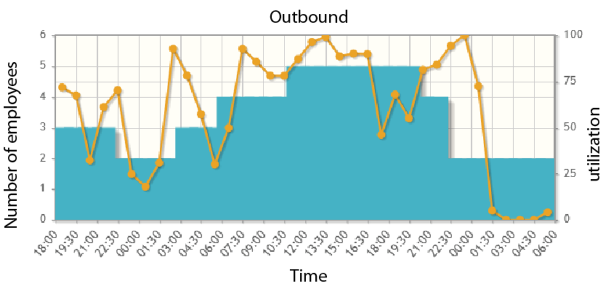

Fig. 6. Utilization of employees in the outbound area in a supported scenario of an unexperienced user.

truck arrival and departure times and the number of pallets that have to be handled. With this information, the candidates create shift plans and make adjustments to the time window plan. For the next step, the use of the decision support system was allowed. The candidate starts with the same setup and the just created shift plan and runs a simulation. As an example result the utilization of employees in the outbound area are shown by the yellow line in FIGURE 5. The blue bars show the number of employees in the area at each hour of the simulation scope. The overload of the employees in the day is visible as the utilization is near 100 percent. Further results are the utilizations of employees in other areas, the punctuality of orders and the utilization of the conveyor system.

In multiple iterations, the plan can be further adjusted. For this the dispatcher has to interpret the results and choose an appropriate measure because the overload in the outbound area could have multiple causes. In the simplest case, the number of deployed employees is too low and must be increased. For this test case, the dispatcher chooses a redistribution of the employees to be aligned with the actual demand. The simulation result of the final solution is shown in FIGURE 6. By the earlier increase in the number of workers, the workload was reduced far enough that all outbound orders are finished four hours earlier.

The main findings of this evaluation were that each candidate planned a larger number of man-hours in the unsupported scenario. The average number of man-hours used in the different scenarios are shown in TABLE I. In the supported scenarios every candidate had reduced the sum of needed man-hours by roughly 10% without orders being delayed. Curiously the inexperienced dispatchers had the better results in the supported scenario regarding the man-hours. This could be explained by the missing knowledge about the limited personal flexibility. While the inexperienced users had used the minimal number of employees for each hour and work area, the experienced users tried to keep the total number of all employees constant for each shift.

TABLE I
AVERAGE MAN-HOURS IN THE UNSUPPORTED AND SUPPORTED SOLUTIONS FOR INEXPERIENCED AND EXPERIENCED USERS

	Unsupported		**Supported**	
	Inexp.	**Exp.**	**Inexp.**	**Exp.**
Outbound	127	144.5	131	132
Inbound	36	38	29.5	38
Picking	112	141	89	109
Sum	275	323.5	249.5	279

In summary, the permanent readiness to prove decisions by simulations generates certainty and enables the users to further reduce buffer capacities with lower risk. The presented digital twin architecture is an enabler to increase the performance and to externalize the knowledge about interrelationships in complex systems. If it is embedded in a decision support system, even inexperienced users can make useful decisions and learn faster.

Digital twin of an metal processing enterprise

The aim of the second use case is the support of production management of a medium-sized metal-working enterprise with about 35 employees. In production, mainly semi-automated machines like lathes, drills and mills are used, but there are also some fully automatic machines that can execute different work steps without manual interventions. The production management was done by operation cards, which were printed for each customer order.

Through increased market requirements, like customers demanding shorter delivery times, fixed delivery dates and a rising number of orders, the enterprise is forced to reorganize the production. At first an analysis was conducted to identify the main challenges. The findings show that customer orders had long throughput times and that buffer spaces in each work area are occupied most of the time. As the main cause for these problems it was discovered that often too many orders are active at the same time and they are processed without concerning their respective urgency.

To approach these problems it was decided to reduce the number of simultaneously active orders in production. On one hand this will relieve the buffers capacities. On the other hand, it will shorten the throughput times by a closer focus. To fulfill the customer demand of fixed delivery dates as well, releasing and prioritizing orders become more important tasks. For this, the production manager needs a detailed real-time overview of the current order states and shop-floor condition. Furthermore he needs a preview of finishing dates and resource utilizations in the future.

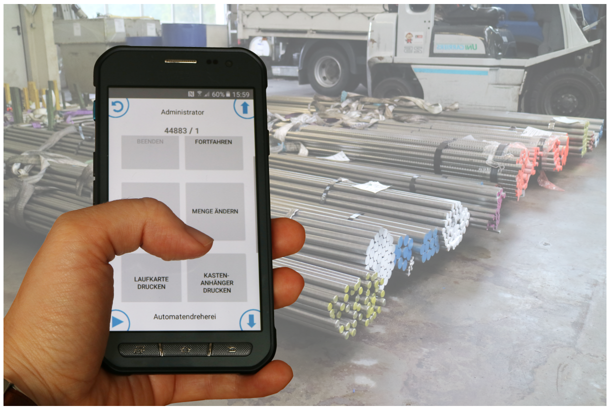

Fig. 7. Update of order state via smart device application.

The mentioned requirements were addressed by our simulation-ready digital twin approach. To create the digital twin, all processes, resources, products were recorded and transferred in a digital model (FIGURE 8). Since the processes are mostly analog the synchronization of the model with the current state turned out to be difficult. To avoid a high investment to equip all machines with different sensors or bar-code scanners, cheaper cellphones were purchased and an application (see FIGURE 7) was developed that enables updating the respective order state. The procedure to perform a production step for an order and to collect the necessary data is as follows. At first a worker registers his device at a work area and gets a list of open orders that must be processed there. This list is sorted by priorities so he can choose the one with the highest urgency. After that the App displays the next production step and the worker can assign it to an available machine. If a setup process is necessary, it can also be started. After finishing a production step, the worker also confirms it by the application. To depict the material flow in the digital twin, additional knowledge about the shop-floor structure and part demands for each process is needed. With this knowledge, additional bookings (e.g. of stocks) by the workers can be avoided but presuppose a strict observance of the work plans.

With the collected data, the production planner has an overview of the current state of the production. It is also possible to provide him reports about the actual processing and waiting times for each order in the past. But to provide the required short-term preview that helps him to prioritize or release orders purposefully, the digital twin must be extended with simulation capabilities. In order to prevent the simulation from changing the state of the digital twin that represent the current state of the shop floor an independently fork of it must be created first. Future changes to this fork are also made by events. Since there are no events of the real world to update the simulation state they have to be generated synthetically by the simulation controllers. For example to create a process start event for an order, all starting conditions have to be checked, like needed materials in the respective process buffers and available resources. If no disturbances are allowed when a

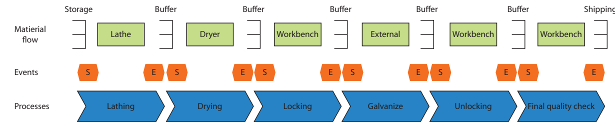

Fig. 8. Typical work plan in the production with material flow and basic events. In a simulation scenario, the events must be generated synthetically.

process runs, plan values or historical values can be used as processing times. For processes that were already started at the beginning of the simulation, the assumed process time must be reduced by the elapsed time. Furthermore, strategies for resource allocations are necessary to complete the simulation control.

While a simulation is carried out, the same reports can be collected as for the real production with the same report processors because the only runs on the digital twin model. If more than one simulation has run with different decision scenarios (e.g. various order schedules) a master has a well-founded prediction for the respective consequences and can choose the best one for realization.

V. Conclusion and Outlook

In this paper, we argued the need of an efficient integration of event-discrete simulation into a digital twin architecture in real-time logistics use cases. We presented our architecture and framework and demonstrated the application in two industrial use-cases. The key principle of the architecture is the split of static and dynamic parts of the model making past, present and future of the logistics system efficiently available for decision support in the same semantic and syntactic data model.

Our current work is focused on the creation of the digital twin (static model as well as behavior modeled in the simulation controller). We believe that it could be possible to learn the digital twin or certain elements out of the real-time data based on machine learning. The empty twin would be connected via interfaces to the real world. Data would be gathered for adequate time span. And finally the data can be used to train artificial neural networks that can replace the controller in the simulation.

References

[1] C. M. Meyer, W. Kersten, and C. Nede, Integration des Komplexitts-managements in den strategischen Fhrungsprozess der Logistik. Zugl.: Hamburg-Harburg, Techn. Univ., Institut fr Logistik und Unternehmens-fhrung, Diss., 2006 u.d.T.: Meyer, Christian Martin: Komplexittsmanagement in der Logistik, 1st ed. Bern: Haupt, 2007.

[2] Y. Bar-Yam, General features of complex systems, Encyclopedia of Life Support Systems (EOLSS), UNESCO, EOLSS Publishers, Oxford, UK, p. 1, 2002.

[3] Deiseroth J.; Klennert M.; Thissen, S.; Schwede, C.; Toth, M. (2013): Virtual Experiment Fields for Logistical Problem Solving. In: Proceedings of the 17th Cambridge International Manufacturing Symposium - September 2013.

[4] M. Zajac and C. Schwede, Cross-Process Production Control by Camera-Based Quality Management Inside a Logistic Assistance System, in Lecture Notes in Logistics, Logistics and Supply Chain Innovation, H. Zijm, M. Klumpp, U. Clausen, and M. t. Hompel, Eds., Cham: Springer International Publishing, 2016, pp. 353362.

[5] Michael Grieves, Origins of the Digital Twin Concept, Florida Institute of Technology / NASA, Working Paper, 2016.

[6] C. Zhuang, J. Liu, and H. Xiong, Digital twin-based smart production management and control framework for the complex product assembly shop-floor, Int J Adv Manuf Technol, vol. 19, no. 2, p. 201, 2018.
[7] T. H.-J. Uhlemann, C. Schock, C. Lehmann, S. Freiberger, and R. Steinhilper, The Digital Twin: Demonstrating the Potential of Real Time Data Acquisition in Production Systems, Procedia Manufacturing, vol. 9, pp. 113120, 2017.
[8] M. Chen, S. Mao, and Y. Liu, Big Data: A Survey, Mobile Netw Appl, vol. 19, no. 2, pp. 171209, 2014.
[9] S. Malakuti and S. Grner, Architectural aspects of digital twins in IIoT systems, in Proceedings of the 12th European Conference on Software Architecture Companion Proceedings - ECSA '18, Madrid, Spain, 2018, pp. 12.
[10] S. Boschert and R. Rosen, Digital TwinThe Simulation Aspect, in Mechatronic Futures: Challenges and Solutions for Mechatronic Systems and their Designers, P. Hehenberger and D. Bradley, Eds., Cham, s.l.: Springer International Publishing, 2016, pp. 5974.
[11] 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. 1217, 2016.
[12] K. A. Delic, L. Douillet, and U. Dayal, Towards an architecture for real-time decision support systems: challenges and solutions, in Proceedings 2001 International Database Engineering and Applications Symposium, Grenoble, France, Jul. 2001, pp. 303311.