Towards a Model Factory Experimentation Environment for Cyber-Physical Twins

Andreas Emrich, Oliver Gutermuth, Michael Frey, Peter Fettke, Peter Loos
German Research Center for Artificial Intelligence (DFKI) & Saarland University
(andreas.emrich / oliver.gutermuth / michael.frey / peter.fettke / peter.loos) @dfki.de

Abstract

Industry 4.0 has brought about tremendous changes in equipping machinery and factory setups with sensors and bridging the gap between the digital and the physical world. Process mining has proven to be a valuable tool for analyzing industrial workflows, gathering models, and checking the conformance of executions. However, faults that occur seldom in industrial processes cannot be easily learned by applying machine learning methods. Explicit nominal models can help to close this gap. The given approach shows how nominal product, resource, and process models can be used in a physical twin environment to enhance process mining tasks and related error root cause analysis. In this scenario a model factory serves as physical twin of a real-life factory. The paper concludes with a depiction of a potential proof-of-concept.

Keywords: Industry 4.0, Hybrid AI, Process Mining.

1. Introduction

Industry 4.0 has brought about radical changes in equipping machinery and factory setups with sensors and bridging the gap between the digital and the physical world (cf. Lasi et al., 2014). Process mining has proven to be a valuable tool for analyzing industrial workflows, gathering models and checking the conformance of executions. However, faults that occur seldom in industrial processes cannot be easily learned by applying machine learning methods.

Hence, we opt for an approach that seeks to combine machine learning with nominal models to overcome the learning problem while trying to address the learning of rare cases with a physical twin model of an Industry 4.0 factory to implement a rapid test environment for provoking rare faults and training the underlying machine learning models much faster.

The **goal** of this paper is to develop and demonstrate a novel approach for dealing with rare class learning phenomena in cyber-physical process mining. We will outline a conceptual framework that concentrates on evaluating the outcome of cyber-physical processes along with their constituent root causes that may reside in the process data itself, or machine and product data accordingly.

With an exemplary use case for series production, we highlight some problems that make a pure simulation approach on synthesized data difficult and provide a small-scale model factory setup that allows for rigorous experimentation on rare cases. This is especially important when the root causes and effects of such faults are not known and prevent the creation of an accurate simulation model. We describe this setting from the algorithmic and physical setup and conclude the paper with an outlook on its evaluation.

This paper follows Design Science Research (DSR) **methodology**. According to Hevner's design guidelines, we design an artifact (cf. Guideline 1 of Hevner et al., 2004) to explore the problem space. We will propose possible avenues for evaluation (cf. Guideline 3 of Hevner et al., 2004). As a fully automated approach is not feasible in the cyber-physical environment, we aim for rigorous experimentation in which we vary sensor and algorithmic configurations (cf. Guideline 6 ”Design as a search process” of Hevner et al., 2004).

According to Peffers' DSR process model (cf. Peffers et al., 2007), we are in an early stage of our research, where we designed the artifact (cf. sections 5 and 6. We motivate the relevance in section 3, and

URI: https://hdl.handle.net/10125/107283
978-0-9981331-7-1
(CC BY-NC-ND 4.0)

we give an outlook on how the evaluation could be performed based on our prototype, which in this stage should be seen as a proof-of-concept (cf. Nunamaker et al., 2015) in Section 7. We conclude the paper with a summary of the results, a discussion of current limitations, and an outlook on future work.

2. Related Work

2.1. Predictive Process Monitoring in the Internet of Things

Predictive Process Monitoring aims at monitoring a current process instance and predict its current or next steps - or the outcome of the process (cf. Di Francescomarino et al., 2018; Evermann et al., 2017). As a sub-discipline of process mining it has the goal to predict the future of an ongoing process execution. Many approaches focus on atomic process logs and look at the mere sequence of process steps. For the Internet of Things, we do not have such logs available unless we specifically enact operations by calling actuators, but we have to derive those logs from sensor data with event abstraction techniques. Hence, the prediction problem comes into play even before assessing logs.

At the same time, detecting activities in the Internet of Things is a complex task. Different sensors and continuous data can be used for the detection and fusion of these methods. However, uncertainty remains a big problem in this context, where process knowledge in models could help remove uncertainty and provide improved quality results.

There has been considerable work on process mining leveraging IoT data. Knoch et al. (Knoch et al., 2018.) have demonstrated that sensor data could be used to monitor manual work processes. Rebmann et al. even demonstrated that the fusion of different sensors could improve process mining capabilities (cf. Rebmann et al., 2019). De Leoni and Pellattiero showed in recent work, that certain aggregation steps might help to obtain better results in the process mining process (cf. de Leoni and Pellattiero, 2022). First attempts have already been investigated in prior settings of our LEGO® factory (cf. Rehse et al., 2018; Rehse et al., 2019).

2.2. Mining of Rare Cases

Deep learning has been widely adopted for process mining tasks in recent years (cf. Neu et al., 2022). The majority of process-mining-related contributions do not focus on rare cases. From a methodological standpoint, process mining approaches allow identifying such cases by measuring the frequencies of process variants occurring in analyzed data. A rare case contains a rather seldom sequence of events caused by included rare events, an uncommon order of events, or even both. An advanced definition of a rare case might also consider relevant metrics. Therefore, a case containing common events occurring in an usual order might still be classified as rare when metrics such as duration times deviate significantly from comparative data. However, basic process mining literature does not highlight rare case phenomena as a focal point. Instead, they sometimes neglect uncommon cases since the respective methods often try to display high-frequency process patterns showing usual, widespread procedures while also allowing to analyze rare ones.

The application of process mining becomes more specific when it's put into context. Hence, organizations might focus on rare cases as soon as their occurrence critically impacts essential goals. Such circumstances require appropriate analysis to identify, predict, and affect their appearances. A typical field for those applications is the need for early fault detection in complex systems such as machines or industrial environments where complex sensor data makes machine learning techniques feasible. Dangut et al. describe a comprehensive approach to focus on rare cases in the context of predictive maintenance for aircraft components (Dangut et al., 2021). They present a log-based machine learning model to prevent extremely rare failures and therefore have to deal with imbalanced data sets. This challenge is also fundamentally addressed by Ali et al. (Ali et al., 2019). The authors perform a literature review and systematically identify challenges in handling imbalanced class problems using machine learning algorithms during classification procedures. As a result, they present approaches to deal with this problem and list particular advantages and disadvantages. Park et al. offer a hybrid solution to handle unbalanced data from industrial processes (Park et al., 2019). Their approach separates the detection problem to determine rare events and the diagnosis problem to identify rare fault events, resulting in a significant improvement in accuracy. Overall, it can be stated, that rare case mining is a known problem, yet there are not many applications in process mining, and none in IoT-based process mining.

3. Motivating Use Case

In order to demonstrate our approach, we opted for a use case that fulfills the following requirements:

R1 - High degree of automation: On the one hand, like in most production-line settings, we strive for an environment that is fully automated.

R2 - Non-deterministic errors and failures: However, many factors can contribute to deviations in execution that cause execution errors and production failures.

R3 - Observable behaviour can be learned: For the automatable and often occurring parts of the use case process, machine learning approaches can help traditional predictive process monitoring approaches.

R4 - Rare or non-occurring behaviour can be described by nominal (process) models: The overall goal is to have error flows that describe how occurring failures can be communicated or mitigated. Nominal product models can observe product features and decide whether a failure is happening or not.

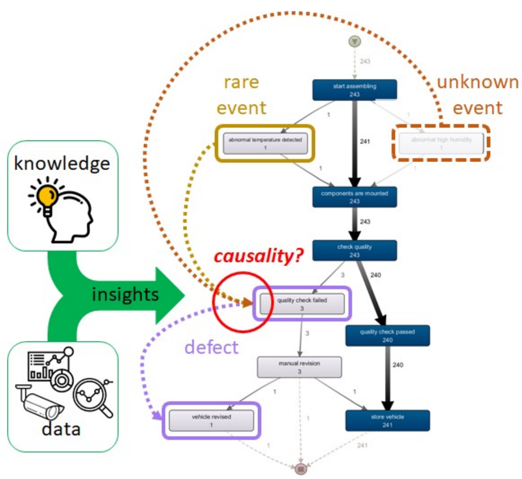

Figure 1. Rare and unknown events in workflow networks of processes

Figure 1 shows a process map generated from exemplary data. It illustrates the addressed problem and our hybrid solution approach. The presented graph describes occurring events in chronological order. The frequency of occurring events and the respective paths are annotated and highlighted. Branches breaking away from the "happy path" demonstrate that certain circumstances during a process can lead to results of particular interest. In this example, insufficient product quality was determined in three individual process instances. Here, the claim is now mainly on the causes to allow predicting and avoiding such defects. Indications of possible causes of defects can be found above all in recorded sensor data of cyber-physical systems (CPS). The example shows one case where the flaw's reason is an excessively high temperature. The significant deviation of the measured value is already classified as relevant and is therefore considered as a discrete event. However, the figure also indicates that a significant humidity deviation may have led to the defect in another case. Regardless of whether sensor technology for this influence is missing or specific measured values were disregarded, circumstances of this unknown event remain hidden. Compared to the simplified example of Figure 1, the major problem of respective real-world-environments is complexity. A CPS often depends on a large number of sensors and measures amounts of data accordingly. In this context, defining and observing relevant events from complex sensor data constellations are challenging due to uncertainties regarding the completeness of considered influences and causal dependencies leading to a specific constellations' relatively rare occurrence.

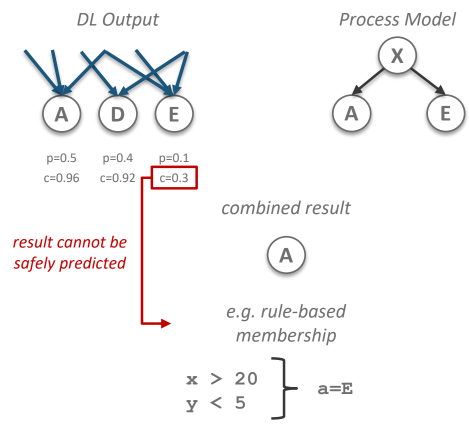

Figure 2. Example for combining nominal process models and ML-based process mining models

Figure 2 shows the probabilities for the next process step and the corresponding process model. As it can be clearly seen, the deep learning prediction algorithm favors activity types A and D according to their high probabilities (variable p). It predicts E only with very small confidence (variable c). When looking at the corresponding process model, we know that the model only allows for activities A and E. Depending on whether we have a normative or conformative execution, D would be discarded then. So, for now, we could conclude that A is the expected result. However, including a further nominal model could hint that E is the correct answer. As it can be seen, including process model knowledge in the mining process can resolve some but not all problems. Further nominal models we have e.g. about machine and product states could help to further improve those predictions. Moreover, such models are invaluable, when the ground truth data is not

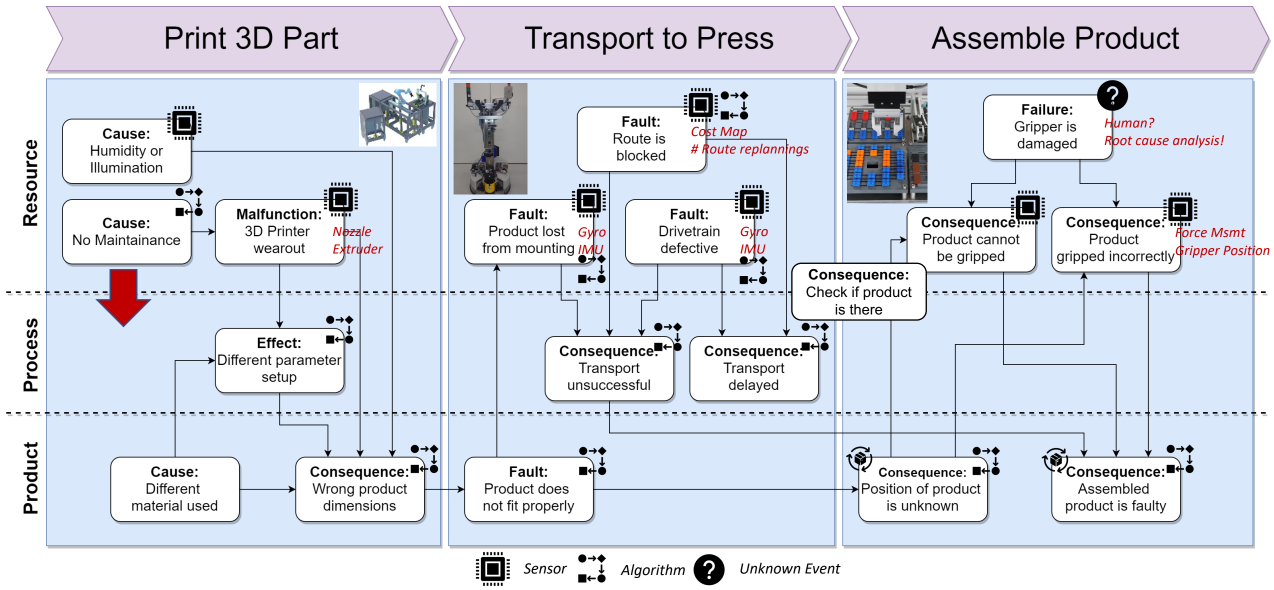

Figure 3. Motivating use case

sufficient to safely predict the different activities.

Figure 3 depicts our sample use case we have chosen to meet these requirements: In our case, a USB stick is being assembled from regular and custom printed LEGO® bricks. We have three major process phases that our production process follows: First, a custom 3D brick is printed, then it is transported with an autonomous guided vehicle (AGV) towards the assembly plant, where the parts are being picked and assembled. Throughout this process, various rare exterior events and conditions could arise and consequently cause faults.

This use case enables the construction of a physical twin of the depicted production process to have a similar technical setup with similar sensors on a much smaller scale. Within this physical twin, we try to introduce rare events by experimenting with exterior conditions like humidity and lighting conditions, configuring different wear-out stages in parts of the production line, and confronting the AGV with unknown obstacles. By doing so, we can **induce such events and still learn from their occurrence**, especially in this LEGO® model setting, as we assume that this can be transferred to larger plants to derive the required learnings. As we can see that some of the occurring events are rather on the **process** side, whereas others relate to the production **resource** itself (e.g., the press) or the **product** itself. Having nominal models in place that describe the product behaviour can hence help to **handle unknown and rare events even if the case base for machine learning is insufficient**.

4. Framework Design

4.1. Errors, Failures and Faults

The technical concept describes data generation and analysis to investigate procedures for handling rare classes. The interaction of sensors and actuators shapes a production process where machines convert particular material into a product. Products have certain specifications and properties that can vary depending on a separate order. We assume a regular production process whenever a finalized product matches the specifications of its order and all quality requirements. If the characteristics deviate and issues are evaluated as a product defect, we conclude a failure caused by at least one fault. If the causal chain of fault, failure, and defect occurs seldom or is partially unknown, various problems arise in dealing with such cases. Our contribution presents approaches and solutions for such challenges. To mimic industry setups from reality where rare cases of high interest occur, it's crucial to include several requirements for fault-failure-relations within our physical twin:(i) Occurring faults still allow process completion. (ii) Failures that have an impact on the product quality become evident within data. (iii) Faults to failures and failures to defects are $n:1$ related. (iv) Several faults can happen in sequence.

Meeting those requirements guarantee the need for analysis since root causes and dependencies are not obvious. Figure 4 shows a slice of respective fault-failure-defect-relations presenting a sample case that is covered within our model-factory. The discovered

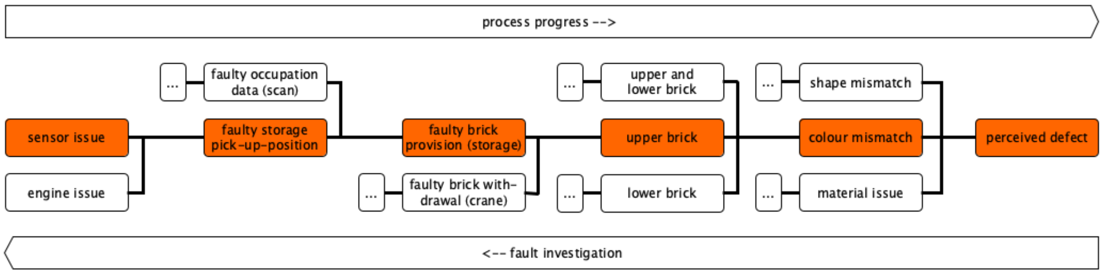

Figure 4. Fault-failure-defect-relation

relation chains will be modeled with knowledge graphs to enable quick querying of these aspects as run-time (cf. Steinmetz et al., 2022 for a similar approach for representing digital twin models in knowledge graphs).

Allowing this wide range of issues within our processes is essential to have an appropriate combination of physical environment generating data accordingly to evaluate our solutions. However, the focus for our research is finding appropriate solutions to deal with rare and unknown fault classes which are the minority of cases in industry and the twin we are presenting.

4.2. Physical twins

Contrary to a mere digital twin, which mirrors the functionality and state of a real physical entity in a digital format, the model factory we developed also mirrors the capabilities of the larger factory in the physical sense. On the abstraction level of components and their respective Skills the two factories are nearly identical. The specific physical implementation differs of course, but the input and output materials as well as executed process steps are interchange able. The machine, their machine capabilities and sensor parameters are specified in an OPC-UA format (cf. Sidorenko et al., 2021).

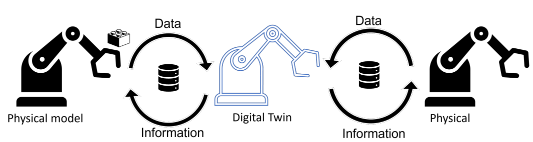

Figure 5. Physical and Digital Twins

4.3. Common Analytic Tasks for both Physical Twins

Abstracting event logs for process mining from sensor data: Generated data of addressed industrial environments shows two major levels of abstraction. First, the product specification determines individual work steps and materials usage. Thus, for each product type, a system controls the execution of tasks for production. Crucial information can be recorded at this level by logging relevant events and accurate timings. At the more detailed level of abstraction, the sensor data of integrated machines exist related to the task's execution. They control specific functionalities of machines and are used for individual activities that are components of the tasks above. In contrast to discrete documentation of events of the rough process, sensor data basically provide information permanently as a stream. Essential data usage is the common consideration of recorded events under consideration of the sensor data streams for reconstructing individual production processes. In particular, process sequences and associated sensor data patterns are used for this purpose. By abstracting the event logs using the Skill format, we abstract from the specifics of the physical twins and create a common signature that is comparable among the two physical twins.

Analyzing the error root causes and cause-effect chains: In the context of fault classes, the primary interest is identifying a fault's root cause(s). Here, a major challenge of rare or unknown classes results from the lack of reference data leading to limited feasible machine learning methods. For these cases, hybrid approaches will be introduced that incorporate nominal models. This approach should lead to the fact that irrelevant information can be excluded to redirect the focus of machine learning procedures on crucial aspects and thus enable better results in classification. In addition, we will investigate whether a deliberate provocation of faults can lead to additional reference data for learning rare classes or the occurrence of previously unknown classes.

4.4. Algorithms for IoT Process Mining

Based on the data that will be captured, the processes should be analyzed in order to determine the activities in which specific errors and faults occur and how likely it is that a certain error will occur in a certain part of the process. A majority of algorithms in process mining are rather sequence-oriented and hence just focus on the execution sequence of process activities. In our scenario, however, other workflow data or environmental conditions may have a significant effect on the process outcome and hence cannot be entirely covered by a pure analysis of the process flow.

Table 1 shows the different algorithms that are envisioned to be tested in our model factory. One crucial task is the **activity detection** of the next process step. Conceptually, this comes down to a multi-class classification of the next process label. As a baseline many approaches in literature favor the use of long short-term memory (LSTM) networks that are especially a good fit for sequential data and which are usually activity with a softmax function, i.e. the individual label probabilities sum up to 1. As loss function here, a sparse categorical cross entropy loss is planned. A more advanced approach is the **sequence prediction** which seeks to predict a whole execution sequence of process activities. The sequence length hence can be encoded by multi-step encoder-decoder LSTMs. Here rectified linear units (ReLu) or sigmoid are candidates for the activitation function, and per-class binary cross entropy loss for the loss function. For simple **Error prediction**s we use similar methods as for the activity recognition. Here, we predict the respective errors based on the same input data. However, we use a sigmoid function as an activation function here, as we perform a multi-label classification as errors could occur simultaneously.

These methods do not take into account the severity of errors. Therefore we try to apply an approach translated from risk management to this domain in the following way:

$$\text{score}_{\text{Fault}} = p_{\text{Fault}} * \text{severity}_{\text{Fault}} \tag{1}$$

We can use this as a basic cost function for a Deep Reinforcement Learning approach leveraging LSTMs with policy gradients or deep Q-learning networks (DQNs) (cf. Osband et al., 2016). This ensures that nominal knowledge and the risk score can be considered.

5. Monitoring, Actuation and Analysis in the Model Factory

This section describes how our model factory gathers, monitors, and analyzes the respective sensor data. On a low level, the LEGO® sensors and controllers have to be integrated on a local Raspberry Pi 4.0 device via Message Queueing Telemetry Transport (MQTT) over a Mosquitto server (cf. subsection 5.1). Furthermore, we provide an outlook in which software and analytics components are already implemented and envisioned in the future for our model factory (cf. subsection 5.2).

5.1. Technical Integration of the Model Factory

Considering the setup of the LEGO® factory's relevant hardware (cf. Figure 6), a server side instance of a Process Engine manages the manually designed business process. This instance acts as the control authority for the factory. The next physical component is a RaspberryPi 4 which is the central hub for a number of LEGO® specific EV3 devices and non LEGO® sensors, such as inertial measurement units (IMU) and environmental sensors. The EV3 devices are physically connected to the RaspberryPi and have their own local network for communication. The EV3 devices are able to run a limited amount of Python code and are capable of supporting up to four sensors and up to four actuators. They provide both signal transmission and power supply to the connected devices. Communication between the server side control authority, the RaspberryPi and the EV3 devices is implemented using the MQTT protocol whith the MQTT broker hosted on the RaspberryPi so it is capable to access both the internal network of the EV3 devices but also the network connecting it to the server side. Furthermore, the Apache Kafka event streaming platform is utilized to provide a second means of communication to multiple decoupled analytics components on the backend side. This allows for targeted data transfer via a publish and subscribe pipeline.

5.2. Software and Analytics Architecture

The software and analytics architecture as depicted in Figure 6 shows which functionality is implemented or

Table 1. Algorithms for IoT Process Mining

Process Problem	Activity detection / prediction (1 step)	Sequence prediction	Process Outcome Prediction	Error Prediction
Base problem	Multi-class classification	Sequence of multi-class classifications	Minimize risk score (cf. above)	Multi-label classification
DL Type	LSTM	Multi-step encoder-decoder LSTM	Deep Reinforcement Learning: LSTM with policy gradients / DDQN	LSTM
Activation Function	Softmax	Rectified linear units (ReLU) / Sigmoid	ReLU or variants	Sigmoid
Loss function	Sparse categorical cross entropy loss	Per-class binary cross entropy loss	Per-class binary cross entropy loss	Per-class binary cross entropy loss

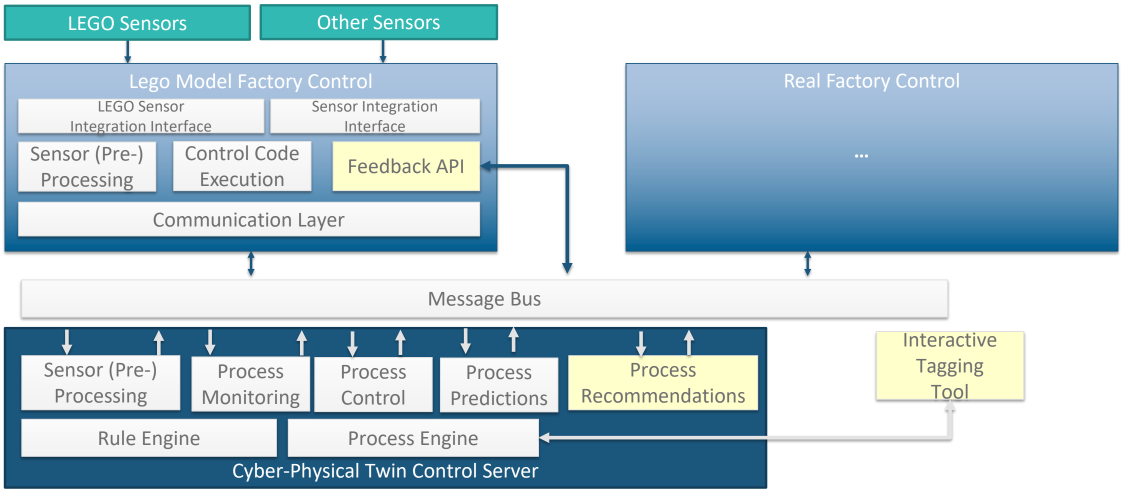

Figure 6. Software Architecture of the LEGO® model factory

planned in future iterations (yellow) in a server backend or dedicated frontends.

For the **backend server**, a core part is the Process Engine, which serves as a central entity to monitor, control, and enact the different functionalities in the depicted factory process. Hence, every operation and thereby every state change of a component that is physically executed is being reported back to the Process for keeping track of the current execution state (*Process Monitoring*). Furthermore the BPE enacts functionalities by triggering Skills of said components in the physical twin (*Process Control*). Skills represent a standardised means of describing the capabilities of a component, including possible input and output parameters as well as error scenarios (cf. the used specification format of Sidorenko et al., 2021). Specific Skill-relevant knowledge can be evaluated in the *Rule Engine* which is based on JBoss Drools and the graph database neo4j for more complex reasoning tasks. The signals in both directions are being exchanged via Apache Kafka (*Message Bus*) and MQTT (*LEGO® and Sensor Integration Interfaces*) respectively. As BPE we use the widely available open source software implementation jBPM. *Process Predictions* serve as entry-point to provide process-related risk estimations for faults, failures and defects. Future iterations plan to involve *Process Recommendations* to provide feedback to workers to handle or even prevent such events.

The *LEGO® Model Factory Control* and the *Real Factory Control* both comprise interfaces to the underlying, different sensor infrastructures via MQTT protocols. Both controls have in common that they pre-process and aggregate the sensor information into dataframes and capture them via the Skill dataformat and push them to the Message Bus.

As possible means for interaction with workers in a later development stage it is planned to leverage the capabilities of a *head-up display (HUD) frontend* or a *mixed reality interface*. The HUD display is rather directed to the operators controlling the machinery and checking in from to time to get summarized depictions of the factory state, whereas the Mixed Reality Interface is envisioned for interactive fault analysis. Hence it should also visualize errors and enable feedback mechanisms of the worker in the future as well as display the current execution state of the process. An *Interactive Tagging Tool* serves as labelling source for observations made on the Model Factory in order to support supervised machine learning on the observed phenomena.

6. Prototype Design

The physical prototype is a model factory that mimics an industry environment. It is intended to provide a source for data generation and can be used to demonstrate several use cases in the context of rare cases within IoT scenarios. The model factory's current setup contains 21 sensors and 18 actors that allow the assembling of LEGO® parts according to a particular product specification. The technical architecture which is necessary to apply the control code and collect data can be found in section 5. Using LEGO® components for this purpose empowers our research to design a custom factory setup where rare cases basically can occur. Such occurrences within a model factory become visible in the physical process and data, while real-world environments usually rely on data analysis only.

The factory consists of certain components that are physically separated and fulfill different tasks. This environment includes two material storages providing several LEGO® bricks for assembling, a transportation crane, a press combining two bricks, a quality check for product evaluation, and a product storage for final placement. Whenever an assembly is initiated, the storages will provide bricks according to the respective order. Next, the crane picks them up and puts them into the press while the press prepares a unique frame to precisely support the placement to achieve the intended shape of the product. When both bricks are in position, the press applies pressure to combine them and releases the resulting product for transportation by the crane, which takes it to quality assurance. Here, the product is assessed according to its specification and visually checked for physical anomalies. As a last step, the crane picks up the product and puts it into the product storage. This architecture is described in Figure 7 by using HERAKLIT (cf. Fettke and Reisig, 2021).

A crucial feature of the factory is its ability to create many product variants. While the lower brick remains the same size, it can just have different colors. The upper brick can differ in colors but also in three possible sizes. Additionally, each upper brick type can be assembled at three different positions on top of the lower brick. Considering the shape variations and four different possible colors for each brick, a total number of 144 product variants result.

Whenever a product is ordered, the factory executes the process automatically, allowing mass assembling due to the CPS setup without manual tasks. Therefore, the built-in sensors are essential to the infrastructure since they control the execution of tasks. During assembling, faults can occur, leading to product failures that will be identified at the quality assurance. However, the failure detection can only classify failure types, where one type can result from different faults or respective fault combinations. Hence, the recorded data has to be investigated to clarify the failure's root cause. An example of such a scenario is a deviation in the color of at least one brick. This flaw can easily become identified by either a sensor or picture evaluation within quality assurance. However, the root cause could result from a faulty classification of bricks in storage locations, a faulty provision of a material storage location to the crane, a fault in reaching the correct stop location of the crane resulting in picking a brick from a wrong storage line, or any combination of those issues.

7. Evaluation Concept

Over the next few months, test runs are planned with our factory setting in which we try to induce rare events and evaluate different configurations of sensors and algorithms. With many physical components, complete automation is not possible, and the respective evaluations have to be supervised by human operators.

Divide and conquer : Divide the process into different chunks of execution frequency. They can be run separately and hence could provide the necessary data input variations for the other execution chunks.

Build nominal models on baseline data : Record a fair amount of baseline reference data, that can be considered as ”good” instances, in order to build a nominal model.

Identify unknown faults through outlier detection : Deviation models can help to detect outliers from the norm and can be used for detail inspections of the products, in order to verify whether these outliers caused any observable product faults.

Synthesize and augment test data : The recorded chunks can be synthesized, improved and simulated using Monte Carlo methods.

Simulate with rare portions of the process : Use the above mentioned augmented data to simulate the process and automate the test execution for other parts. By doing so, you ensure that more variety is in there as in the standard probability distribution of process executions. Here especially executions of the Deep Reinforcement Learning algorithm as described in Section 4.4 should be used to predict the end-to-end probabilities of certain fault risks.

For our given use case, it makes sense to divide the process for the parts of the 3D printer and the AGV as these offer a wide range of variety and the execution needs eager supervision and possibly intervention by operators. Once those logs of the respective process parts of the 3D printer and AGV can be synthesized, the latter process part of the assembly can be tested in a much quicker and automated manner.

The following conditions could be changed and parametrized throughout execution in order to provoke rare events:

lighting : The lighting conditions can have an effect on the optical recognition of the color sensors. Light intensity and color might have interference with certain colors of bricks.

humidity : The room humidity has an impact on the reliability of the 3D printer.

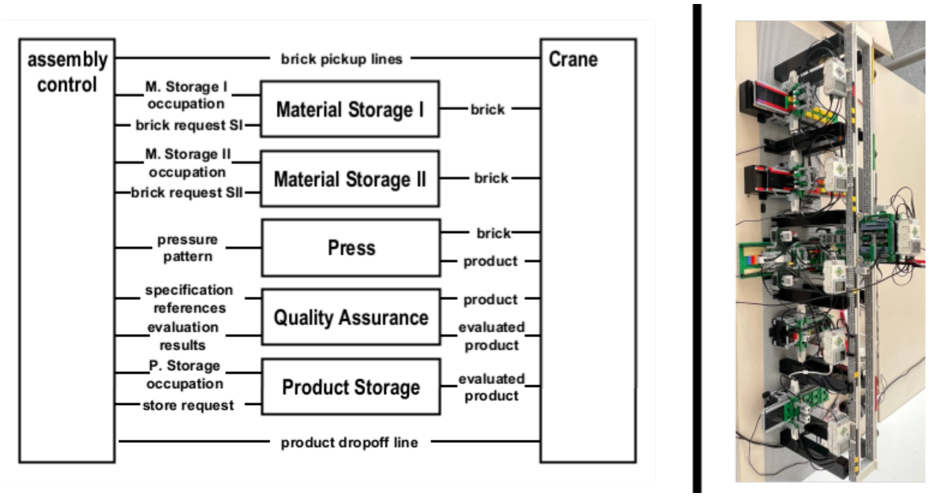

Figure 7. Conceptual and physical architecture of assembly stage

room temperature : The room temperature has an impact on the reliability of the 3D printer and the quality of the resulting printed brick.

product variants : It is not efficient to permutate all 144 product variants, but permutating the form variations for the "press" component and the color variations for the optical sensors is interesting to provoke possible errors and faults.

In terms of evaluation metrics, not the overall mining accuracy is paramount: As each process activity is associated with a certain risk weight, the overall system should optimize all activities' weighted per-class accuracy.

8. Conclusion

Our approach has shown a concept for experimenting physical processes for rare case process mining. We developed a framework for series production based on fault and failures that proposes to evaluate process execution success based on the failure rates actually achieved throughout execution. Hence, rarely occurring process events that have a high impact on this, have to be considered and weighted accordingly. We explained our approach, how to use process mining algorithms in combination with synthesized data and manipulations in execution experiments in order to provoke rare events. In a LEGO® factory's exemplary production setting, we demonstrate how such data can be obtained and integrated to perform real-time analysis of these events.

The approach is in the stage of implementation at the moment, and we cannot offer a fully integrated evaluation. However, we have achieved significant steps in this development process. The choice of sensors is neither exhaustive nor representative for industrial settings, whereas other essential aspects, such as the explainability of the results presentations to the user and the assistance functionality towards them, are not described in this paper.

Overall, actual large-scale experimentation is the natural next step to evaluate the feasibility and value of the presented approach. We are confident that it could be applied and tailored to other scenarios in sensor-based production and could inspire future work in that direction. It has the potential to build cost-effective models of factory setups that could be used as a base to pre-train machine learning models in a much faster and cost-efficient way as on the target factory settings. Also, applications for worker-focused systems could be conceivable (cf. Raso et al., 2018; Yigitbas et al., 2021), which could prove to be quality-of-life improvements for the day-to-day work in quality inspections.

Acknowledgments

This paper has been written in the context of the project "RACKET" funded under the ID "01IW20009" by the German Government (BMBF). We want to thank our partner departments Innovative Factory Systems (IFS) and Augmented Vision (AV) and our student researchers Patrick Eiden and Ankit Agrawal for their support in implementing the presented setup.

References

Ali, H., Salleh, M., Saedudin, R., Hussain, K., & Mushtaq, M. (2019). Imbalance class problems in data mining: A review. *Indonesian Journal*

of Electrical Engineering and Computer Science, *14*.
Dangut, M. D., Skaf, Z., & Jennions, I. K. (2021). An integrated machine learning model for aircraft components rare failure prognostics with log-based dataset. *ISA Transactions*, *113*, 127–139.
de Leoni, M., & Pellattiero, L. (2022). The Benefits of Sensor-Measurement Aggregation in Discovering IoT Process Models: A Smart-House Case Study. In A. Marrella & B. Weber (Eds.), *Business process management workshops* (pp. 403–415).
Di Francescomarino, C., Ghidini, C., Maggi, F. M., & Milani, F. (2018). Predictive Process Monitoring Methods: Which One Suits Me Best? In M. Weske, M. Montali, I. Weber, & J. vom Brocke (Eds.), *Business process management* (pp. 462–479).
Evermann, J., Rehse, J., & Fettke, P. (2017). Predicting process behaviour using deep learning. *Decis. Support Syst.*, *100*, 129–140.
Fettke, P., & Reisig, W. (2021). Modelling service-oriented systems and cloud services with heraklit. *Advances in Service-Oriented and Cloud Computing: International Workshops of ESOCC 2020, Heraklion, Crete, Greece, September 28–30, 2020, Revised Selected Papers 8*, 77–89.
Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly*, *28*, 75–105.
Knoch, S., Ponpathirkoottam, S., Fettke, P., & Loos, P. (2018). Technology-enhanced process elicitation of worker activities in manufacturing. In E. Teniente & M. Weidlich (Eds.), *Business process management workshops* (pp. 273–284).
Lasi, H., Fettke, P., Kemper, H., Feld, T., & Hoffmann, M. (2014). Industry 4.0. *Bus. Inf. Syst. Eng.*, *6*(4), 239–242.
Neu, D. A., Lahann, J., & Fettke, P. (2022). A systematic literature review on state-of-the-art deep learning methods for process prediction. *Artificial Intelligence Review*, 1–27.
Nunamaker, J. F., Briggs, R. O., Derrick, D. C., & Schwabe, G. (2015). The last research mile: Achieving both rigor and relevance in information systems research. *Journal of Management Information Systems*, *32*, 10–47.
Osband, I., Blundell, C., Pritzel, A., & Van Roy, B. (2016). Deep exploration via bootstrapped dqn. *Advances in neural information processing systems*, *29*.
Park, P., Marco, P. D., Shin, H., & Bang, J. (2019). Fault detection and diagnosis using combined autoencoder and long short-term memory network. *Sensors*, *19*(21).
Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of Management Information Systems*, *24*, 45–77.
Raso, R., Emrich, A., Burghardt, T., Schlenker, M., Sträter, O., Fettke, P., & Loos, P. (2018). Activity monitoring using wearable sensors in manual production processes - an application of cps for automated ergonomic assessments. *Multikonferenz Wirtschaftsinformatik 2018*, 231–242.
Rebmann, A., Emrich, A., & Fettke, P. (2019). Enabling the Discovery of Manual Processes Using a Multi-modal Activity Recognition Approach. In C. Di Francescomarino, R. Dijkman, & U. Zdun (Eds.), *Business process management workshops* (pp. 130–141).
Rehse, J., Dadashnia, S., & Fettke, P. (2018). Business process management for industry 4.0 - three application cases in the dfki-smart-lego-factory. *it Inf. Technol.*, *60*(3), 133–141.
Rehse, J.-R., Mehdiyev, N., & Fettke, P. (2019). Towards Explainable Process Predictions for Industry 4.0 in the DFKI-Smart-Lego-Factory. *KI - Künstliche Intelligenz*, *33*(2), 181–187.
Sidorenko, A., Volkmann, M., Motsch, W., Wagner, A., & Ruskowski, M. (2021). An OPC UA model of the skill execution interaction protocol for the active asset administration Shell. *Procedia Manufacturing*, *55*, 191–199.
Steinmetz, C., Schroeder, G. N., Sulak, A., Tuna, K., Binotto, A., Rettberg, A., & Pereira, C. E. (2022). A methodology for creating semantic digital twin models supported by knowledge graphs. *2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA)*, 1–7.
Yigitbas, E., Karakaya, K., Jovanovikj, I., & Engels, G. (2021). Enhancing human-in-the-loop adaptive systems through digital twins and vr interfaces. *2021 International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)*, 30–40.