Proceedings of the 2022 Winter Simulation Conference
B. Feng, G. Pedrielli, Y. Peng, S. Shashaani, E. Song, C.G. Corlu, L.H. Lee, E.P. Chew, T. Roeder, and P. Lendermann, eds.

DATA-DRIVEN RELIABILITY MODELING OF SMART MANUFACTURING SYSTEMS USING PROCESS MINING

Jonas Friederich

Mærsk Mc-Kinney Møller Institute
University of Southern Denmark
Campusvej 55
Odense, 5230, DENMARK

Sanja Lazarova-Molnar

Inst. of Appl. Informatics and Formal Descr. Methods
Karlsruhe Institute of Technology
Kaiserstr. 89
Karlsruhe, 76133, GERMANY

ABSTRACT

Accurate reliability modeling and assessment of manufacturing systems leads to lower maintenance costs and higher profits. However, the complexity of modern Smart Manufacturing Systems poses a challenge to traditional expert-driven reliability modeling techniques. The growing research field of data-driven reliability modeling seeks to harness the abundance of data from such systems to improve and automate the reliability modeling processes. In this paper, we propose the use of Process Mining techniques to support the extraction of reliability models from event data generated in Smart Manufacturing Systems. More specifically, we extract a stochastic Petri net which can be used to analyze the overall system reliability as well as to test new system configurations. We demonstrate our approach with an illustrative case study of a flow shop manufacturing system with parallel operations. The results indicate, that using Process Mining techniques to extract accurate reliability models is feasible.

1 INTRODUCTION

The industry is currently undergoing a transformation process towards a new level of value chain organization and control, often referred to as Industry 4.0. New technologies, such as the Internet of Things, Cloud Computing, Big Data and Artificial Intelligence, have emerged and are making manufacturing systems ever smarter (Zheng et al. 2018). Smart Manufacturing Systems (SMSs) enable companies to meet current challenges, including customer demands for high product quality, short lead times, low costs and a high degree of customization in a globalized market with demand fluctuations (Qu et al. 2019). In addition, large amounts of data are collected from sensors and production equipment, that can be used to support data-driven decisions.

On the flip side of the progress of SMSs is their increasing complexity which makes it more difficult to maintain the systems and to identify possible vulnerabilities that affect their reliabilities. To this end, reliability modeling includes a number of techniques for planning and monitoring reliable manufacturing systems and for detecting such vulnerabilities (Blischke and Murthy 2011). However, conventional reliability modeling relies on expert knowledge of the system under study, which can become a bottleneck as systems become more complex (Friederich and Lazarova-Molnar 2021a). Moreover, SMSs are subject to frequent modifications that can quickly make such static reliability models obsolete when the system topology changes (Lugaresi and Matta 2021). Thus, there is a need to dynamically generate accurate reliability models for manufacturing systems based on real-time data streams to ensure optimal exploitation in the shop-floors. The emerging research field of data-driven reliability modeling aims to utilize data stemming from SMSs to automate or at least support the development of accurate reliability models for SMSs (Friederich and Lazarova-Molnar 2021b).

978-1-6654-7661-4/22/$31.00 ©2022 IEEE

Process mining (PM) is a process management technique that enables business processes to be reconstructed and evaluated on the basis of digital traces in information systems. PM thus facilitates to model the implicit and otherwise hidden process knowledge contained in data to make it tangible and transportable (van der Aalst 2016). We can use PM to compensate for the aforementioned disadvantages of SMSs in terms of reliability modeling. Since PM is a data-driven technique, less expert knowledge of the system under study is required, and the extracted models are updated as the system topology changes.

In this article, we present a method for data-driven reliability modeling of SMSs using PM. More specifically, we extract a stochastic Petri net, a modeling formalism commonly used for reliability modeling, from event data generated in manufacturing systems. We consider the path that material takes through a system, the frequencies of manufacturing activities, activity durations, resource capacities as well as failure and repair times of production resources. The extracted reliability model can be simulated to analyze the system and to compute key performance indicators, such as the system and resource reliability. In addition, the extracted model can be used to evaluate new configurations of buffer and resource capacities or the impact of additional production resources. We test our proposed method using an illustrative case study of a flow shop manufacturing system with parallel operations.

The remainder of the paper is structured as follows: in Section 2, we review related work on reliability modeling of manufacturing systems. Section 3 covers our proposed approach for data-driven reliability modeling of SMS. We provide a case study of our proposed approach in Section 4. Finally, Section 5 concludes the paper by providing a summary and an outlook.

2 RELATED WORK

Reliability modeling and assessment of manufacturing systems has a long history (Chlebus and Werbińska-Wojciechowska 2016). Many modeling formalisms, such as reliability block diagrams (RBDs), fault tree analysis (FTA) and Petri nets (PNs), have been applied to the manufacturing domain. (Tont et al. 2008) present a method for availability and reliability assessment of complex manufacturing systems using RBDs and Monte-Carlo simulation. A method for mission reliability modeling of discrete manufacturing processes and systems for the domestic weapons industry using RBDs is proposed by (Liu et al. 2013). (Fazlollahtabar and Niaki 2018) use both FTA and RBDs to evaluate the reliability of a manufacturing system consisting of multiple industrial robots. An application of FTA to a printed circuit board assembly system is provided by (Shu et al. 2006). (Yan et al. 2017) compare FTA and PNs for mission reliability modeling of an automated guided vehicle. (Adamyan and He 2004) perform failure and reliability analysis of manufacturing systems using counters in PNs.

All of the aforementioned contributions manually model the system under study, which requires extensive knowledge of the system. In addition, such models become obsolete as soon as the configuration of the system changes and must therefore be adapted to the new configuration. Data-driven reliability assessment of SMS seeks to utilize the abundance of data provided by such systems to automate the processes of reliability modeling and analysis either completely or at least partially (Lazarova-Molnar and Mohamed 2019). To the best of our knowledge, there exist only a handful of contributions addressing this novel and promising research field. (Rodseth et al. 2018) present a framework for a reliability-based cyber-physical system and discuss how it could be implemented in the manufacturing industry. In particular, the authors examine and demonstrate the balance between reliability engineering and machine learning in big data analytics. (Lazarova-Molnar et al. 2020) propose a concrete method for data-driven FTA that extracts fault trees from time series data of a system. Furthermore, the authors simulated the extracted fault trees to estimate reliability measures of systems under study. A method to extract RBDs from data generated in a manufacturing system is proposed by (Friederich and Lazarova-Molnar 2021a).

Data-driven reliability assessment of SMS requires a comprehensive understanding of the underlying system structure and processes. (Friederich et al. 2022) present a framework for data-driven digital twins of SMS in which the use of PM and machine learning to obtain such a comprehensive understanding is proposed. (Lugaresi and Matta 2021) propose a method that generates digital twins based on event logs generated in manufacturing systems. Based on the generated digital twin, system performance indicators, such as resource throughput and production time, can be estimated.

We have identified a lack of data-driven methods to extract accurate reliability models of manufacturing systems. Therefore, in this paper, we propose a method to extract such models based on data generated in SMS using PM techniques. The main novelty of our method is an end-to-end pipeline for extracting reliability models of SMS using stochastic Petri nets as modeling formalism.

3 PROPOSED APPROACH FOR DATA-DRIVEN RELIABILITY MODELING

In this section, we describe our proposed approach for data-driven reliability modeling of SMS, and Figure 1 outlines the corresponding framework. Starting from a manufacturing system, the system's data is collected and distributed to the company's information systems. Several information systems in the company may collect information about the physical system, such as Supervisory Control and Data Acquisition (SCADA), Programmable Logic Controller (PLC), Manufacturing Execution System (MES), or Enterprise Resource Planning (ERP) (Friederich et al. 2022). The data captured by such systems is then aggregated and synthesized in event logs and state logs.

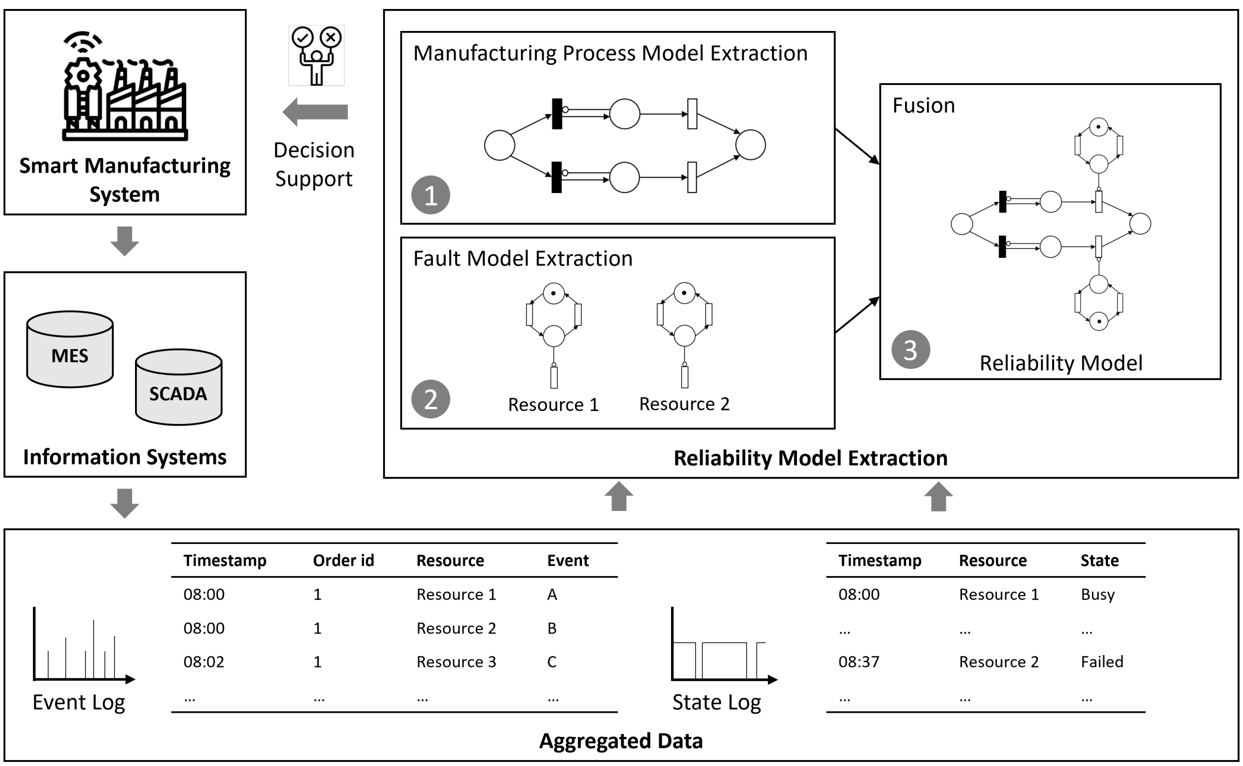

Figure 1: Data-driven reliability modeling using process mining and Petri nets.

A manufacturing system event log captures data from processes inherent to a manufacturing system. Data in event logs can be acquired by MES or ERP systems. We adopt the general assumptions about a process as stated by van der Aalst (van der Aalst 2016). However, we adjust them slightly to fit the manufacturing domain:

- A manufacturing process is triggered by production *orders*.
- An *order* is characterized by a sequence of activities which are begun and ended by *events*.

- Each *event* has a corresponding *timestamp* and *resource* that is seized and released.

Considering the above mentioned assumptions, we define an event log EL as a set of event entries in the following way:

$$EL = \{E_0, E_i, ..., E_m\},\ i = 1, ..., m.$$

Each event log entry is defined as a tuple $E_i = (t, o, r, e)$, where t is the timestamp, o is the order identifier, r is the resource identifier and e is the event identifier. Order identifiers are unique numbers and represent individual production orders, resource identifiers are unique strings and represent production resources such as Automated Guided Vehicles (AGVs) or assembly cells. Event identifiers are unique strings and mark the beginning of activities such as transport or assembly operations.

A resource state log captures state changes of the resources of a manufacturing system. Data for such logs can be acquired by PLCs or SCADA systems. For our approach, we define a state log SL as a set of state change entries as follows:

$$SL = \{S_0, S_i, ..., S_n\},\ i = 1, ..., n$$

Each state log entry is defined as a tuple $S_i = (t, r, s)$, where t is the timestamp, r is the identifier for the resource that changes its state and $s \in \{busy, idle, failed\}$ is the new state the resource transitioned to. Figure 2 displays a possible operational state changes of resources, as captured by the state log.

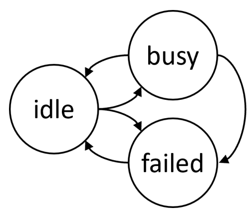

Figure 2: Possible operational state changes.

The manufacturing process model is extracted from an event log and a resource state log (Subsection 3.1). Subsequently, the manufacturing process model is enriched with resource fault models that capture probability distributions describing failures and repairs of resources, estimated from the resource state log (Subsection 3.2). We use stochastic Petri nets (SPNs) for describing manufacturing process models. Formally, the class of SPNs that is considered in this paper can be described in the following way:

$$SPN = (P, T, A, m_0)$$

where:

- $P = \{P_1, P_2, .., P_p\}$ is the set of places, drawn as circles
- $T = \{T_1, T_2, .., T_q\}$ is the set of transitions along with their distribution functions or weights, drawn as bars
- $A = A^I \cup A^O \cup A^H$ is the set of arcs, where A^O is the set of output arcs, A^I is the set of input arcs and A^H is the set of inhibitor arcs and each of the arcs has a multiplicity assigned to it,
- m_0 is the initial marking of the Petri net.

Each transition is defined as $T_i = (e, r, n, f, type)$ and corresponds to an event $\{E.e\}$, $\forall E \in EL$ in the event log. Thus, we use the name of the event e to label the transition T_i. r is the resource, n is the frequency (number of times the transition fired) and f is a probability distribution function if the corresponding transition is timed and a firing weight if it is immediate. $type$ is the type of the transition where $type \in \{timed, immediate\}$. The set of arcs are defined such that

$$A^O = \{a_1^o, a_2^o, ..., a_k^o\},\ A^I = \{a_1^i, a_2^i, ..., a_j^i\} \text{ and } A^H = \{a_1^h, a_2^h, ..., a_i^h\}$$

where:

$$A^H, A^O \subseteq P \times (T \cup I) \times \mathbb{N},\ A^I \subseteq (T \cup I) \times P \times \mathbb{N}.$$

3.1 Extraction of the Manufacturing Process Model

We extract the manufacturing process model in six steps, with each step increasing the model detail (Figure 3). The material flow process represents the path that production orders follow through the system. From a SPN modeling perspective, such a process might consist of several timed and immediate transitions. Timed transitions may represent activities such as the transport of material or the operation of an assembly cell. Immediate transitions may represent routing decisions in case of parallel operations or other events without activities. Each transition may be annotated with the frequency of the corresponding events. Furthermore, the activity durations of timed transitions can be described by probability distribution functions. The manufacturing process might contain resources with capacities such as buffers or assembly cells. Such characteristics are commonly modeled using inhibitor arcs preventing a transition from firing. Figure 4 provides an overview of the mentioned concepts and how we will model them using data provided by a manufacturing system.

In the following sections (Sections 3.1.1 - 3.1.5), we describe each of the mentioned steps in detail.

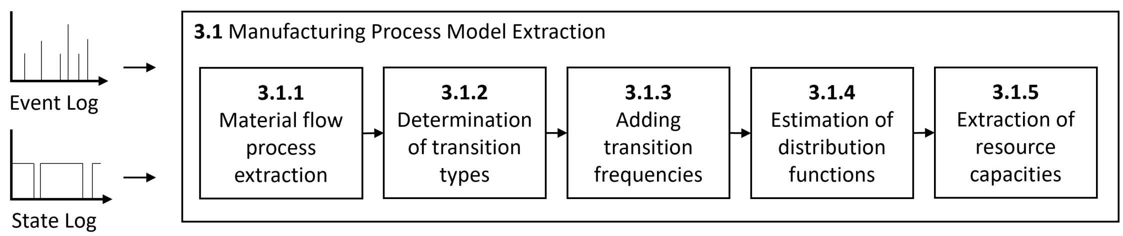

Figure 3: Extraction of the manufacturing process model.

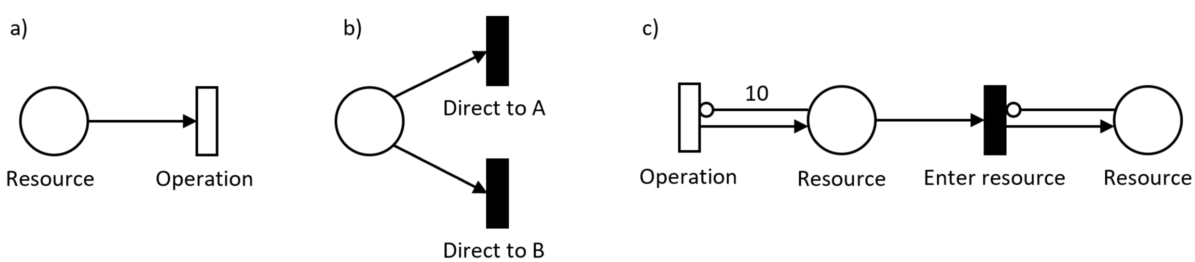

Figure 4: SPN modeling concepts for manufacturing processes. a) Timed transition to represent an operation activity, b) immediate transitions to represent routing decisions, c) use of inhibitor arcs to represent resource capacities.

3.1.1 Extraction of the Material Flow Process

The material flow process represents the path that production orders follow through the manufacturing system. Based on the event log, we apply the α-miner (van der Aalst et al. 2004) to extract the SPN that is representing the material flow process of the system. The α-miner is a popular process discovery algorithm capable of extracting Petri nets consisting of an initial marking describing the initial state of a model, the actual process model, and a final marker describing the final state of a model. However, the algorithm is not able to detect loops and to distinguish between implicit and required places which might result in additional non-required places in a discovered Petri net.

The α-miner algorithm consists essentially of two steps: (1) identify ordering relations in the log and (2) convert those relations to a Petri net. There are four types of ordering relations that the α-miner can detect:

- Directly-follows ($a > b$): if activity a is directly followed by activity b.
- Sequence ($a \rightarrow b$): if $a > b$ and not $b > a$.
- Parallel ($a || b$): if both $a > b$ and $b > a$.
- Choice ($a \# b$): if neither $a > b$ nor $b > a$.

Figure 5 displays, how the identified relations are then converted into a Petri net. The left Petri net corresponds to a sequence pattern ($a \rightarrow b$), the middle Petri net to a XOR-split pattern ($a > b$, $a > c$ and $b \# c$) and the right Petri net to an AND-split pattern ($a > b$, $a > c$ and $b||c$). A detailed description of the algorithm is provided in (van der Aalst et al. 2004).

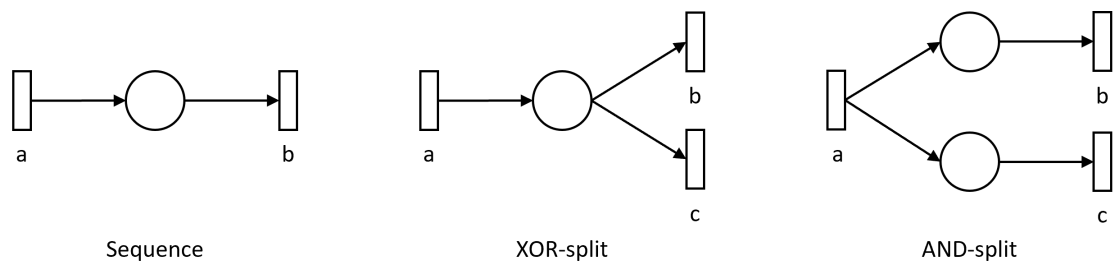

Figure 5: Petri net patterns that can be identified by the α-miner.

3.1.2 Determination of Transition Types

Current process discovery algorithms are not developed for the extraction of SPNs. Thus algorithms such as the α-miner (van der Aalst et al. 2004) or the inductive miner (Leemans et al. 2013) extract Petri nets with only one transition type (i.e., timed transitions). Therefore, in cases where it applies, we need to set some of the extracted timed transitions to immediate transitions. We consider a transition as timed if the activity times of the corresponding resource is known. Activity times of resources can be derived from the operational state changes in the state log (i.e., idle, working). Thus, if the resources of transitions $T_i.r$ correspond to resources that appear in the state log $\{S.r\}$, $\forall S \in SL$, transitions are considered as timed transitions. All remaining transitions are set to immediate transitions (Algorithm 1).

3.1.3 Extraction of Immediate Transition Weights

Transition weights $T_i.n$ are the number of times each transition fired based on the information available in the EL. Such weights provide useful quantitative information to the extracted model and can be used to assess the systems performance and to calculate firing probabilities of immediate transitions. We can use the previously described directly-follows relationships of activities in the EL (e.g., frequency activity a was followed by activity b) to estimate the transition frequencies and add them to the extracted SPN. For cases where one activity is followed by more than one other activity (e.g., activity a was followed by activity b and c) we simply add all frequencies of the following activities (Algorithm 1).

3.1.4 Estimation of Timed Transition Distribution Functions

As noted in Section 3.1.2, an extracted transition where the resource associated with the transition, $T_i.r$, occurs in the SL is considered a timed transition (i.e. $T_i.type = timed$). Each timed transition has a probability distribution function $T_i.f$ that determines its firing times. We utilize the operational state changes of resources in the SL to determine these functions.

The beginning of a resource activity is marked by the state change of the resource to "busy" ($S_i.s = busy$) and the end of the activity is marked by the subsequent state change to "idle" ($S_i.s = idle$). The time difference between these two state changes is the duration of the resource activity. Therefore, cases where resources change state to "failed" ($S_i.s = failed$) during their "busy" or "idle" time are not considered when estimating transition firing time distributions. Activity durations for each transition are stored in a list.

Extraction of activities durations for resources that are utilized by only one timed transition in the SPN is trivial. However, when two or more transitions utilize the same resource, we can no longer tell which operational state changes belong to which transition just by considering the SL. For such scenarios, we join both the SL and the EL based on the common attribute timestamp (ts) to determine which event in the EL relates to which resource state changes in the SL (recall that transitions correspond to an event $\{E.e\}$, $\forall E \in EL$ in the event log).

To estimate the theoretical probability distributions that extracted activity durations follow, we use the the Maximum Likelihood Estimation (MLE) method (Myung 2003). MLE estimates the parameters of a given probability distribution by maximizing a likelihood function, such that the assumed theoretical distribution best describes the observed data. We fit the normal and lognormal distributions to the extracted activity durations, since these two distributions are commonly used in manufacturing to describe activity durations. We assess the goodness of fit by using the sum of squared errors (SSE) between the data and the fitted distributions. The probability distribution with the lowest SSE is set to be the probability distribution function of the corresponding transition $T_i.f$. Algorithm 1 shows the process of determining transitions types (Section 3.1.2), adding firing frequencies (Section 3.1.3) and the estimation of timed transition distribution functions.

Algorithm 1: Extraction of transition types, firing frequencies and timed transition probability distributions

```
Input: event log EL, state log SL, material flow Petri net SPN
Output: SPN with identified transition types, firing frequencies and distributions
R ← {S.r}, ∀S ∈ SL ;                         // complete set of resources from the SL
for T_i in SPN do
    T_i.n ← getDirectlyFollowsFrequency(T_i, T_following)
    if T_i.r ∉ R then
        T_i.type ← immediate
    else
        T_i.type ← timed
        activityDurations ← getActivityDurations(T_i, EL, SL)
        distributions ← fitDistributions(activityDurations)
        T_i.f ← bestDistribution(distributions)
    end
end
```

3.1.5 Extraction of Resource Capacities

We use inhibitor arcs to model capacities of resources in the SPN (Figure 4). To determine locations of inhibitor arcs, we consider transitions where the resource associated with the transition, $T_i.r$, is, for example, a buffer or an assembly cell as resources with capacities.

Algorithm 2 shows the calculation of resource capacities. For each entry E_i in the EL, *currentLoad* stores the current load of a resource as time passes. Whenever the event of an event log entry $E_i.e$ indicates that a production order is entering the resource, the current load is increased by one and added to the time series array $loadTS$. When an event indicates that a production order is exiting the resource, the current load is reduced by one. The resource capacity is the maximum load over time recorded in $loadTS$. Finally, the extracted resource capacity is set as the cardinality of the corresponding inhibitor arc in the SPN.

Algorithm 2: Extraction of resource capacities

```
Input: SPN, EL
Output: SPN with resource capacities
R ← {T.r}, ∀T ∈ SPN, r = buffer ∨ assembly cell ;          // resources with capacity
for resource ∈ R do
    currentLoad ← 0
    loadTS ← [] ;                                  // time series of resource capacities
    for E_i ∈ EL do
        if "enter resource" ∈ E_i.e then
            currentLoad ← currentLoad + 1
            loadTS.add(currentLoad)
        end
        if "exit resource" ∈ E_i.e then
            currentLoad ← currentLoad − 1
        end
    end
    resourceCapacity ← max(loadTS)
    SPN.addResourceCapacity(resourceCapacity) ;              // add resurce capactiy to
     corresponding inhibitor arc in SPN
end
```

3.2 Extraction of Resource Fault Models

Based on the state log, we extract fault models for each resource. Figure 6 shows an exemplary Petri net for such a model. The initial marking in the *Resource OK* place represents a fully operational resource. A timed transition represents a failure after a random amount of time sampled from a probability distribution. Once a token is created in the *Resource failed* place, the resource is defect and needs to be repaired. By using an inhibitor arc to block the operation of the corresponding resource during repair, the model can be integrated into a manufacturing process model. In case the same resource is executing more than one operation, we add inhibitor arcs from the failed place to the corresponding operation transitions. The repair is represented by another timed transition which also has a probability distribution function that describes the repair duration.

We extract the fault models in two steps (Figure 7, Algorithm 3). First, the two necessary places ("Resource OK" and "Resource failed") and transitions ("Fail" and "Repair") are generated for each resource in the state log $\{S.r\}$, $\forall S \in SL$. The fault models are then integrated in the manufacturing process model. This is done by connecting them with transitions that utilize the corresponding resource $T_i.r$ using inhibitor arcs.

Second, similarly to the estimation of timed transition probability distribution functions of the manufacturing process model (Section 3.1.4), we estimate the probability distribution functions in the fault models using the operational state changes in the SL. The failure of a resource is marked by the transition of the resource to the "failed" state ($S_i.s = failed$) and the repair by the subsequent transition to the "idle" state ($S_i.s = idle$). The time difference between the change from a busy or idle to the failed state is the time to failure and the time difference between the failed and the subsequent idle state is the time to repair.

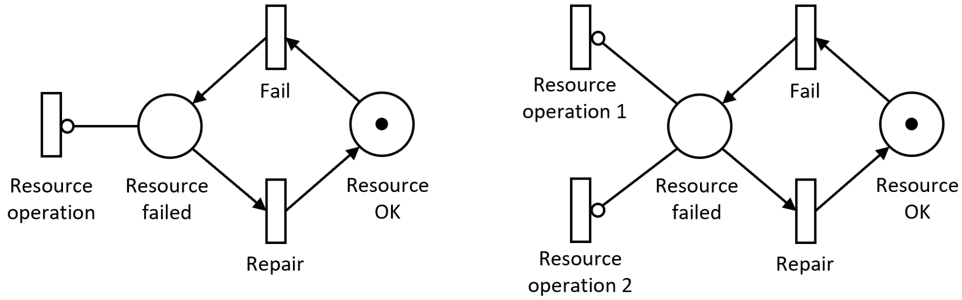

Figure 6: Exemplary fault models for a resource conducting one (left) and two (right) operations.

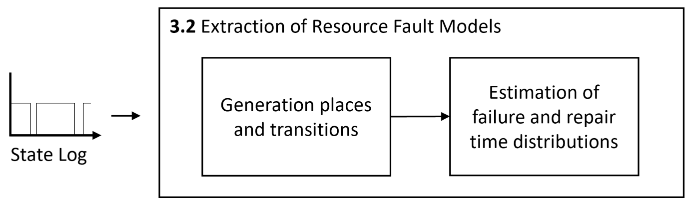

Figure 7: Extraction of resource fault models.

To estimate the theoretical probability distributions of the extracted time to failures and time to repairs, we again use the MLE method. For the failures, we fit distributions such as lognormal, Weibull and exponential and for the repairs distributions such as lognormal, normal and exponential. We evaluate the goodness of fit using SSE between the empirical data and the fitted probability distributions.

Algorithm 3: Extraction of resource fault models

Input: SPN, SL
Output: SPN with fault models
$R \leftarrow \{S.r\}$, $\forall S \in SL$
for $resource\ in\ R$ **do**
  $faultModel \leftarrow generateFaultModels(resource)$
  $integrateToSPN(SPN, faultModel)$
  $estimateFailureAndRepairDistributions(resource, SL, SPN)$
end

4 CASE STUDY

The case study that we use to demonstrate our approach to data-driven reliability modeling is a simple flow line found in many factory floors. It consists of four physical resources (i.e., Warehouse, AGV, Cell 1, Cell 2) and a Manufacturing Execution System (MES) that controls the production process. The warehouse stores raw material and finished products. Both Cell 1 and Cell 2 conduct the same assembly operation and operate in parallel. Thus, the flow line can still operate even if one of the two cells fails. The AGV transports raw material for a new order to one of the assembly cells. After the assembly operation, the finished products are automatically stored in the warehouse. Both, Cell 1 and Cell 2 have a buffer. In case the AGV fails, Cell 1 and Cell 2 use raw material from their buffers until they drain. The production sequence can be summarized as follows:

1. A new production order arrives - MES initiates the production process
2. AGV transports the raw material to one of the assembly cells

3. Assembly cell conducts assembly operation
4. Finished product is automatically stored in the warehouse
5. The MES is informed about the end of the production process

The arrival times of new orders and the operating times of the AGV and both assembly cells are stochastic. Furthermore, the maintenance policy is reactive, i.e., resources are repaired when they fail. Failure and repair times are only recorded for the AGV, Cell 1 and Cell 2. For this case study, the warehouse, the MES and other auxiliary systems do not fail. Figure 8 displays the block diagram of the described case study used in this paper.

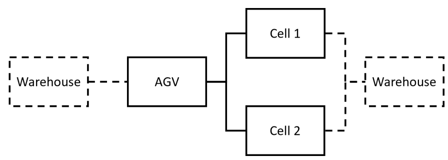

Figure 8: Block diagram of the case study.

We generated data using a simulation model of the described case study. We ran the simulation for 150 hours which resulted in 2338 completed orders. Table 1 and 2 show an excerpt of the recorded event and resource state log.

We applied our proposed approach to the generated data (Friederich 2022). To do so, we implemented the approach in Python using mainly the process mining library *pm4py* (Berti et al. 2019) and the library *fitter* (Cokelaer 2021) for distribution fitting. *pm4py* provides implementations for several process discovery algorithms such as the α-miner and a comprehensive Petri net implementation. *fitter* provides an implementation of the MLE algorithms for fitting of several distributions to empirical data. To export the model, we can use the Petri Net Markup Language (PNML) defined by the standard ISO/IEC 15909. Figure 4 displays the extracted SPN and an excerpt of the corresponding PNML file.

Table 1: Generated event log.

timestamp	order ID	resource	event
...	...	...	
00:24:41	454	MES	new order
00:24:41	454	MES	direct to line 2
00:24:41	454	AGV	transport to cell 2 buffer
00:25:31	455	MES	new order
00:25:31	455	MES	direct to line 1
00:26:01	451	MES	order completed
00:27:11	453	Cell 1 buffer	enter cell 1
00:27:11	453	Cell 1	operation
...	...	...	...

Table 2: Generated resource state log.

timestamp	resource	state
...	...	...
00:22:59	AGV	idle
00:24:41	AGV	busy
00:25:55	Cell 2	failure
00:27:11	Cell 1	idle
00:27:11	Cell 1	busy
00:27:52	AGV	idle
00:27:52	AGV	busy
00:31:16	Cell 1	idle
...	...	...

5 SUMMARY, CONCLUSION AND OUTLOOK

In this article, we proposed a method for data-driven reliability modeling of Smart Manufacturing Systems. The data requirements for the proposed method are an event log that captures the flow of material through a system and a state log that captures state changes of resources. The method consists of two steps, i.e., the extraction of the manufacturing process model and the extraction of resource fault models. For the former, a Petri net representing the material flow using a process discovery algorithm is extracted. Furthermore, transition types are determined, transition weights are added, transition distribution functions are estimated and resource capacities are extracted. For the latter, resource fault models are generated and the corresponding failure and repair distributions are estimated. We applied our method to an illustrative

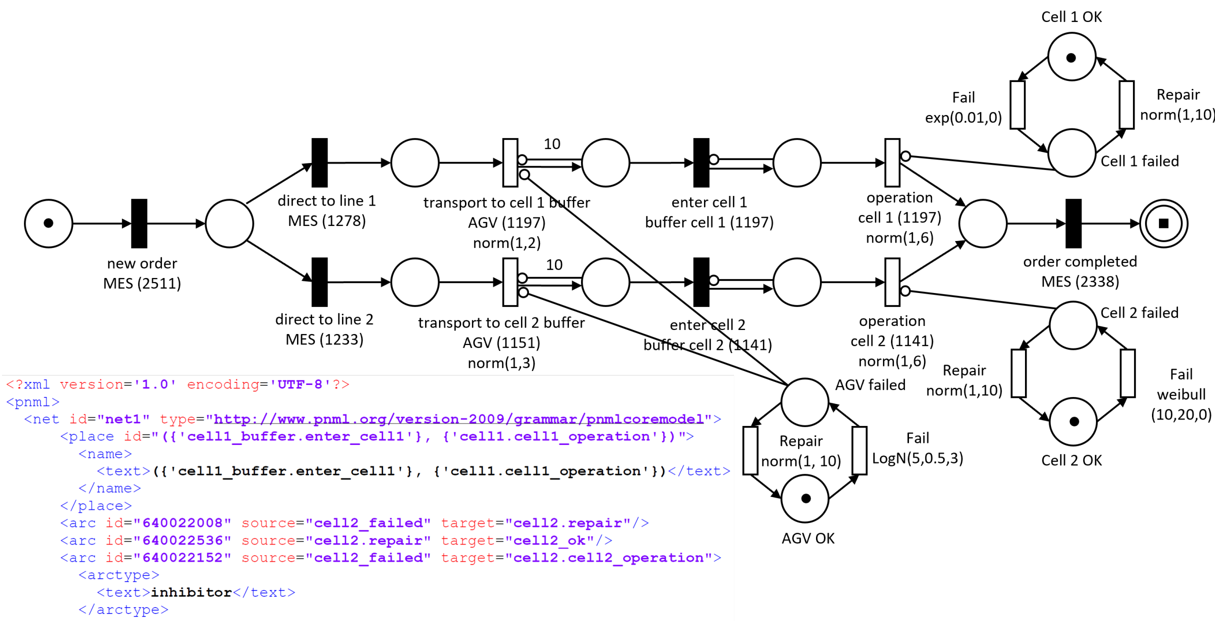

Figure 9: Extracted reliability model.

case study of a flow shop manufacturing system. We were able to extract the reliability model from data generated in the system.

Besides providing detailed insights to the current state of a manufacturing system, the extracted model can be used for decision support in several ways: (1) test new configurations (e.g, add another cell/AGV), (2) support purchase decisions (e.g., investing in new resources) or (3) test new maintenance strategies (e.g., corrective vs. preventive maintenance). To inform decisions about the previously described changes to the system, the changes are first applied to the model and then the model is simulated to estimate the reliability gain or loss.

In future work, we aim to extend our proposed method to be able to detect reworks and other common manufacturing patterns. Furthermore, we want to define methods to properly validate an extracted model. We also want to look into data-driven identification of race age or enable policies for timed transitions.

REFERENCES

Adamyan, A., and D. He. 2004. "System Failure Analysis Through Counters of Petri Net Models". *Quality and Reliability Engineering International* 20(4):317–335.

Berti, A., S. J. van Zelst, and W. van der Aalst. 2019, May. "Process Mining for Python (PM4Py): Bridging the Gap Between Process- and Data Science". In *Proceedings of the ICPM Demo Track 2019*, 13–16. Aachen, Germany.

Blischke, W. R., and D. N. P. Murthy. 2011, September. *Reliability: Modeling, Prediction, and Optimization*. John Wiley & Sons.

Chlebus, M., and S. Werbińska-Wojciechowska. 2016. "Issues on Production Process Reliability Assessment – Review". *Research in Logistics & Production* 6(6):481–497.

Cokelaer, Thomas 2021. "Fitter: Fit Data to Many Distributions". https://github.com/cokelaer/fitter, accessed 4th March 2022.

Fazlollahtabar, H., and S. T. A. Niaki. 2018, March. "Fault Tree Analysis for Reliability Evaluation of an Advanced Complex Manufacturing System". *Journal of Advanced Manufacturing Systems* 17(1):107–118.

Friederich, Jonas 2022, April. "Implementation of the Presented Approach for Data-Driven Reliability Modeling". https://github.com/jo-chr/data-driven-reliability-modeling, accessed 14th September 2022.

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

Friederich, J., and S. Lazarova-Molnar. 2021a. "Process Mining for Reliability Modeling of Manufacturing Systems with Limited Data Availability". In *2021 8th International Conference on Internet of Things: Systems, Management and Security*, 1–7. Gandia, Spain.

Friederich, J., and S. Lazarova-Molnar. 2021b. "Towards Data-Driven Reliability Modeling for Cyber-Physical Production Systems". *Procedia Computer Science* 184C:589–596.

Friederich, J., G. Lugaresi, S. Lazarova-Molnar, and A. Matta. 2022. "Process Mining for Dynamic Modeling of Smart Manufacturing Systems: Data Requirements". In *International Conference on Manufacturing Systems 2022*, 546–551. Lugano, Switzerland.

Lazarova-Molnar, S., and N. Mohamed. 2019. "Reliability Assessment in the Context of Industry 4.0: Data as a Game Changer". *Procedia Computer Science* 151:691–698.

Lazarova-Molnar, S., P. Niloofar, and G. K. Barta. 2020. "Data-Driven Fault Tree Modeling For Reliability Assessment Of Cyber-Physical Systems". In *Proceedings of the 2020 Winter Simulation Conference*, edited by K.–H. Bae, B. Feng, S. Kim, S. Lazarova–Molnar, Z. Zheng, T. Roeder, and R. Thiesing. 2719–2730. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Leemans, S. J. J., D. Fahland, and W. M. P. van der Aalst. 2013. "Discovering Block-Structured Process Models from Event Logs - A Constructive Approach". In *Application and Theory of Petri Nets and Concurrency*, edited by J.-M. Colom and J. Desel, Lecture Notes in Computer Science, 311–329. Berlin, Heidelberg: Springer.

Liu, Y. C., W. J. Zhang, G. C. Fu, and N. Li. 2013. "Mission Reliability Modeling of Manufacturing Processes and System". *Applied Mechanics and Materials* 248:450–455.

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

Myung, I. J. 2003, February. "Tutorial on Maximum Likelihood Estimation". *Journal of Mathematical Psychology* 47(1):90–100.

Qu, Y. J., X. G. Ming, Z. W. Liu, X. Y. Zhang, and Z. T. Hou. 2019, August. "Smart Manufacturing Systems: State of the Art and Future Trends". *The International Journal of Advanced Manufacturing Technology* 103(9):3751–3768.

Rodseth, H., P. Schjolberg, R. Eleftheriadis, and O. Myklebust. 2018. "Reliability-Based Cyber Plant". In *Safety and Reliability – Safe Societies in a Changing World*, Proceedings of ESREL 2018. Leiden: CRC Press.

Shu, M.-H., C.-H. Cheng, and J.-R. Chang. 2006, December. "Using Intuitionistic Fuzzy Sets for Fault-Tree Analysis on Printed Circuit Board Assembly". *Microelectronics Reliability* 46(12):2139–2148.

Tont, G., M. Iliescu, and D. George. 2008, June. "A Methodology of Availability Assessment for Complex Manufacturing Systems". *WSEAS Transactions on Systems* 7:822–832.

van der Aalst, W. 2016. "Data Science in Action". In *Process Mining: Data Science in Action*, edited by W. van der Aalst, 3–23. Berlin, Heidelberg: Springer.

van der Aalst, W., T. Weijters, and L. Maruster. 2004, September. "Workflow Mining: Discovering Process Models from Event Logs". *IEEE Transactions on Knowledge and Data Engineering* 16(9):1128–1142.

Yan, R., L. M. Jackson, and S. J. Dunnett. 2017, September. "Automated Guided Vehicle Mission Reliability Modelling Using a Combined Fault Tree and Petri Net Approach". *The International Journal of Advanced Manufacturing Technology* 92(5):1825–1837.

Zheng, P., H. wang, Z. Sang, R. Y. Zhong, Y. Liu, C. Liu, K. Mubarok, S. Yu, and X. Xu. 2018, June. "Smart Manufacturing Systems for Industry 4.0: Conceptual Framework, Scenarios, and Future Perspectives". *Frontiers of Mechanical Engineering* 13(2):137–150.

AUTHOR BIOGRAPHIES

JONAS FRIEDERICH is a PhD student at the Mærsk Mc-Kinney Møller Institute, University of Southern Denmark. In 2020, he obtained his Diplom in Information Systems from the Technical University of Dresden, Germany. The aim of Jonas' doctoral project is to design and develop tools and methods that can be used to learn reliability models from data generated in cyber-physical production systems. His research interests cover modeling and simulation, process mining, machine learning and computer vision. His email address is jofr@mmmi.sdu.dk.

SANJA LAZAROVA-MOLNAR is a Professor at the Institute of Applied Informatics and Formal Description Methods, Karlsruhe Institute of Technology. She is also a Professor at the University of Southern Denmark, where she leads the research group Modelling, Simulation and Data Analytics. She is a Senior Member of The Institute of Electrical and Electronics Engineers (IEEE), and currently serving as Director-at-Large on the Board of Directors of The Society for Modeling & Simulation International (SCS). Furthermore, she is Chair of IEEE Denmark and Vice-Chair of IEEE Denmark Women in Engineering Affinity Group. Her email address is sanja.lazarova-molnar@kit.edu.