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

FUSING EXPERT KNOWLEDGE AND DATA FOR SIMULATION MODEL DISCOVERY IN DIGITAL TWINS: A CASE STUDY FROM RELIABILITY MODELING

Michelle Jungmann[1] and Sanja Lazarova-Molnar[1,2]

[1]AIFB Institute, Karlsruhe Institute of Technology, Karlsruhe, GERMANY
[2]Mærsk Mc-Kinney Møller Institute, University of Southern Denmark, Odense, DENMARK

ABSTRACT

Integrating expert knowledge in data-driven Digital Twins can lead to better-informed underlying models. Achieving systematic integration, however, remains a complex challenge. In this study, we propose an initial approach for hybrid model extraction by systematically fusing expert knowledge statements with Internet of Things data from manufacturing systems, such as event and state logs. We outline two main strategies to facilitate the fusion of data and expert knowledge in a systematic way. We, furthermore, present a case study in reliability assessment of manufacturing systems showcasing our methodology within this specific domain. Using our four fusion algorithms, we automatically extract reliability models from both data and expert knowledge statements. Finally, we conduct a comprehensive analysis of the results and draw conclusions regarding the efficacy of the fusion algorithms for Digital Twin model extractions.

1 INTRODUCTION

Digital Twins (DTs) have emerged over the last few years from the Industry 4.0 revolution and are now applied in various areas such as smart cities, healthcare or manufacturing (Fuller et al. 2020), e.g., within Cyber Physical Production Systems (Uhlemann et al. 2017). DTs provide capabilities such as analysis, simulation and optimization (Vogel-Heuser et al. 2021) that can be utilized, e.g., for reliability assessments to achieve lower maintenance costs and downtimes in production systems (Friederich et al. 2021).

Data-driven DT model extractions have gained attention in literature (Fuller et al. 2020) and enable up-to-date digital representatives of the corresponding manufacturing systems in continuously changing production environments. Data-driven DTs utilize (real-time) data from emerging technologies like Internet of Things (IoT) devices or sensors. However, since data-driven DT model extraction approaches try to extract and update DT models without much required human effort (Friederich et al. 2022), data-driven DT model extractions are often unable to detect certain events of interest, e.g., due to a lack of relevant IoT data (Niloofar and Lazarova-Molnar 2021) or a high interdependency of events. However, even with a vast amount of available information some events cannot be detected efficiently and effectively (Hu et al. 2023).

The integration of expert knowledge (EK) in data-driven DT model extractions can mitigate these data-driven challenges. EK has the potential to complement and support data-driven DT model extractions with valuable information and, thus, enable better-informed DT models. In contrast to data-driven extractions, experts, such as engineers, have the capability to set the system in context and use their experience to understand and interpret certain events of interest (Bender and Fish 2000). Therefore, an expert can simply know the missing information about relevant events and behaviors of a given system.

However, hybrid extractions that fuse EK with IoT data for DT model extractions present a research gap and a highly complex task (Jungmann and Lazarova-Molnar 2024). Difficulties arise as EK is usually presented in ambiguous, unstructured and context-dependent natural language, which is still challenging for machines to understand (Chowdhary 2020). Furthermore, fusing EK and data results in challenges that stem from combining two independent data sources, which contain different information levels, follow different processing characteristics and can contradict each other (Jungmann and Lazarova-Molnar 2024).

In this paper, we address the gap of hybrid DT model extraction approaches and contribute towards achieving better-informed DTs. We propose a hybrid extraction approach that utilizes EK for DT model extractions by fusing EK and IoT data for the extraction of Petri net models. Our approach fuses EK and IoT data through two main strategies, one during and one after data-driven model extraction. Further, our approach involves analyzing expert knowledge statements (EKSs), introducing EKS evaluation dimensions and proposing formalization approaches for reliability information. Within the two main strategies we propose four fusion algorithms that utilize formalized EKSs (FEKSs). To demonstrate and validate our approach, we conduct a case study as a Proof of Concept (POC) in the field of reliability assessment.

The remainder of the paper is structured as follows: In Section 2, we provide background information and related work for DT model extractions and hybrid model development approaches. We further provide an overview about reliability modeling in manufacturing. In Section 3, we propose our approach with introducing different aspects of EKSs and designing two strategies and four fusion algorithms. We present and validate our case study in Section 4. In Section 5, we close our paper with a summary and outlook.

2 BACKGROUND AND RELATED WORK

In this section, we provide background information and related work for this paper. We first summarize available DT model extraction approaches outlined in literature. Next, we highlight the current gap in systematic hybrid model extractions that fuse EK and IoT data by providing an overview of available hybrid model development approaches. Finally, we introduce reliability modeling as foundation for our case study.

2.1 Model Extraction in Digital Twins

In our work, we define DTs as digital representatives of their corresponding real-world physical entities. Both components are connected over an intra-twin communication link so that data and decisions can be exchanged automatically (Luan et al. 2021). In literature, there are three different approaches for developing the underlying simulation models: 1) data-driven extraction, featuring an automated model extraction from IoT data; 2) knowledge-driven modeling, featuring manual modeling from experts; and 3) hybrid extraction, featuring a combination of data and knowledge for the extraction of models (Wunderlich et al. 2021).

As knowledge-driven modeling requires high manual efforts from experts, it has the tendency to be highly specific and not easily adaptable to system changes (Friederich et al. 2022). Due to, e.g., emerging technologies such as IoT, knowledge-driven modeling approaches were replaced over the last few years by upswinging data-driven DT model extractions. Data-driven DT model extractions are designed and developed by authors such as Resman et al. (2021), Friederich et al. (2022) or Stojanovic and Milenovic (2018). A common characteristic for data-driven extractions is the aim to keep required manual human interventions to a minimum and utilize widely available IoT and sensor data instead (Friederich et al. 2022). Therefore, data-driven extractions can respond to continuously changing production environments in an automated way, however, they usually require a vast amount of data and sometimes struggle to detect events even if enough data is available (Wunderlich et al. 2021). On top of that, data-driven extractions lack valuable information of available EK with enriched context and experience from experts (Niloofar and Lazarova-Molnar 2021) that could complement the missing information. In the next subsection, we provide an overview of available hybrid extractions that fuse EK and data for the development of models.

2.2 Fusion of Data and Expert Knowledge for Model Development

Only limited resources in literature address tailored hybrid extractions for simulations and model developments, with even fewer discussing the systematic and explicit fusion of EK and data within DTs. Thus, we discovered a notable gap in systematic and seamless hybrid extractions that fuse natural language EK and IoT data for DT model extractions in our literature review (Jungmann and Lazarova-Molnar 2024).

Vogel-Heuser et al. (2021) highlight the importance of integrating EK and data within DTs, yet their implementation remains open. Whereas Eirinakis et al. (2020), Deng et al. (2021) and Ladj et al. (2021) mention and propose approaches combining data-driven techniques and EK, they do not systematically aim for fusion. While Todorovski and Džeroski (2006) and Hu et al. (2023) aim for a systematic fusion of EK and data, their approaches are not within the context of DTs. Dixit et al. (2017) propose a fusion of domain knowledge with Petri nets, focused in constraint template declarations, however, also outside of a DT context. Wunderlich et al. (2021) describe hybrid approaches as effective modeling for DTs. They propose a hybrid approach with a fitting algorithm for one converter that, however, encounters shortcomings and further does not utilize Petri nets and natural language EK. In our previous work (Jungmann and Lazarova-Molnar 2024), we proposed a Fusion DT Framework, striving for a systematic and seamless combination of natural language EK and IoT data for DT model extractions, while identifying key challenges and opportunities to address the gap in systematic hybrid extractions.

In reliability analysis, which is our case study application domain, there is limited literature for data-driven DT reliability model extractions. However, there are to the best of our knowledge no hybrid extraction approaches or methods available in literature for fusing EK and IoT data to extract DT reliability models. In this paper, we continue our previous effort by further defining and demonstrating our approach to fuse EKSs and IoT data for data-driven reliability model extraction.

2.3 Reliability Centered Modeling in Manufacturing

In this subsection, we introduce reliability within the manufacturing context to support our case study. Lazarova-Molnar et al. (2017) define reliability as the probability of a system functioning as expected over a specified time period. Key reliability metrics include dependability, availability, performability (Lazarova-Molnar et al. 2017), mean time to failure, mean time between failures and mean time to repair (Friederich et al. 2021). Various methods, such as Reliability Block Diagrams, Petri nets, Markov Models or Discrete Event Simulations (Friederich et al. 2021) can be used to assess reliability. In our approach, we use Petri nets as modeling formalism due to the possibility for their direct extraction using process mining techniques and their widespread use in reliability assessments (Kabir and Papadopoulos 2019).

In the context of manufacturing, reliability of Smart Manufacturing Systems (SMSs) is increasingly important, as faults in manufacturing systems can lead to outages in production and, therefore, to high costs (Webert et al. 2022). With a well-informed DT model, upcoming failures can be predicted and optimally prevented through required maintenance (Zonta et al. 2020). In this paper, we adhere to common reliability terminology, where a *fault* denotes a trigger or causing event for a failure and *failure* denotes when a system or component can no longer fulfill its intended function (Lazarova-Molnar et al. 2017). In this paper, we process EKSs regarding manufacturing systems' reliability, typically concerning faults and failures.

The shift from knowledge-driven to data-driven modeling approaches, as previously discussed, is also evident in reliability modeling for manufacturing (Friederich and Lazarova-Molnar 2022). Friederich and Lazarova-Molnar (2022) propose an approach for a data-driven reliability model extraction for SMSs. To enable data-driven reliability modeling and simulation, Friederich et al. (2021) specify data requirements. According to their work, a SMS contains different data levels. The data levels are defined as state data, event data and condition monitoring data. The different data levels each contain a different level of model detail, with state data as least detailed and condition monitoring data as most detailed. State data contains information of the states of each SMS's component, event data contains discrete events of the components and condition monitoring data contains health data of the components, e.g., sensor values.

3 PROPOSED APPROACH TO FUSE EXPERT KNOWLEDGE STATEMENTS WITH INTERNET OF THINGS DATA FOR DIGITAL TWIN MODEL EXTRACTION

In this section, we outline our approach to integrate EKS and IoT data for hybrid DT model extractions. First, we describe EKSs and their relation to IoT data levels within the reliability domain. Next, we define three evaluation dimensions for EKSs and propose approaches to formalize reliability relevant EKSs for a later fusion. Finally, we propose two main strategies for the fusion of FEKSs with IoT data during and after the initial model extraction. For each strategy, we propose two fusion algorithms to fuse both FEKS fault and event sequence information with a data-driven extraction for Petri net models.

3.1 Expert Knowledge Statements

In this subsection, we propose our mapping of EKSs to IoT data levels and three evaluation dimensions of EKSs. Next, we propose formalization approaches for two types of reliability information in EKSs.

3.1.1 Expert Knowledge Statements in Relation to Levels of Internet of Things Data

In Subsection 2.3, we presented three distinct IoT data levels derived from manufacturing systems. Here, we introduce a fourth level and explain the correspondence of EKSs to these data levels. EKSs can often contain information similar in detail to IoT data levels. Thus, EKSs and IoT data levels can be effectively matched. In Figure 1, we illustrate IoT data levels and examples for corresponding EKSs with a focus on the domain of reliability. We underlined the faults stated in each example EKS, in Figure 1, and sorted the EKSs into their corresponding IoT data levels based on the data level of the underlined fault.

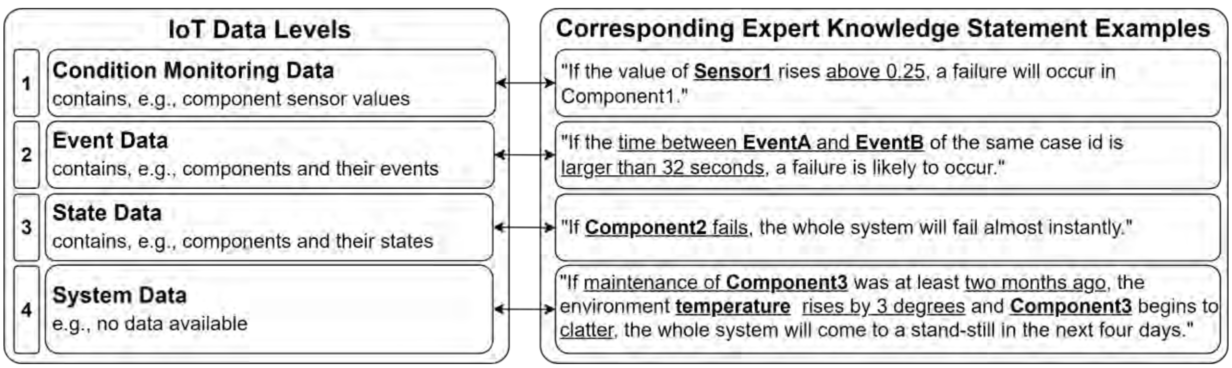

Figure 1: IoT data levels and examples of corresponding EKSs containing similar information.

Figure 1 shows exemplarily that information in EKSs is not limited to one but rather spanning multiple IoT data levels. Thus, one EKS statement can contain, e.g., information about involved components in the failure (based on information at state data level) and faults (based on information at condition monitoring data level). An example for this is the EKS corresponding with condition monitoring data in Figure 1. As experts often know "what if", "when" and "how" something happens, derived from experience and context, contextual information and interpretations are embedded in EKSs. The EKS corresponding to the event data level in Figure 1, e.g., states that a time range larger x serves as fault for the system component. This information can enrich pure data-driven analysis with valuable complementing information, e.g., where to find the fault. A fusion of EKS with data-driven analysis is especially valuable for information extractions in complex cases where the information cannot efficiently be derived from a data analysis involving one or multiple IoT data levels without a pinpoint from EKS. In such cases, information from EKS can add or guide where and what to look for in an efficient way, as experts simple can know the relevant information. Further, EKSs on the fourth level can contain information beyond typical IoT data levels, as stated in the example EKS with, e.g., maintenance intervals, temperatures and their combinations. As IoT data might not be available on this level, information from EKS is essential for, e.g., a fault analysis.

3.1.2 Evaluation of Expert Knowledge Statements

Experts typically have valuable knowledge of systems they work with. This knowledge also applies to the domain of reliability. However, every expert is an individual with different experience levels, context considerations, judgements and trustworthiness. Thus, EKSs can vary widely in content, level of detail and relevance. On top of that, natural language EKSs can be ambiguous (Chowdhary 2020).

In this section, we identified three dimensions to evaluate EKSs for our case study. Based on the designed dimensions we drafted four different EKSs in our case study containing reliability information that is realistically phrased. The dimensions are described as follows:

1.) *degree of belief* refers to the inaccuracy in EKSs. EKSs contain uncertainties (Liu et al. 2017) expressed through terms such as "believe", "could" or "probably". Processing methods, such as fuzzy-logics or fuzzy Petri nets (Liu et al. 2017), can take those uncertainties into account and transform them, e.g., into probabilities. EKSs, additionally, often feature approximate values or ranges rather than exact figures, which further increases the inaccuracy of content in EKSs that needs to be covered.

2.) *correspondence to IoT data level* defines the difficulty of detecting a failure solely based on IoT data. Thus, difficulty is influenced by the number of IoT data logs required for a detection of the fault/event and its correlation with the failure. If the fault is only detectable when, e.g., a state and a condition monitoring log are connected, the difficulty is higher than when only an event log is needed. Additionally, the difficulty increases if the fault is infrequent or only indirectly correlated or IoT data is lacking.

3.) *time distance between fault and failure* refers to the time distance between fault occurrence and failure manifestation. If a fault directly causes a failure, the distance is lower. However, if detecting a fault requires analyzing historical data from before the failure (e.g., data from an hour prior), the EKS is deemed more challenging due to the temporal distance.

3.1.3 Formalization of Expert Knowledge Statements

Given the complexity of natural language (Chowdhary 2020), formalizing EKSs is a pre-requirement for a subsequent processing and integration with IoT data. While structuring knowledge can be challenging, it can be supported by techniques such as ontologies and knowledge graphs (Jurasky et al. 2021). Due to the limited scope of our POC, we consider these advanced approaches as a topic for future exploration.

In our case study, we formalize EKSs into FEKSs with our two proposed POC approaches, outlined in Table 1. Both approaches assume all EKS information as true, not conflicting and independent of the degree of belief. The first approach formalizes fault information using an if-then-in logic. For more complex EKSs containing fault details, the if-then-in logic also allows an incorporation of, e.g., multiple logical conditions and time ranges. Failure event sequences are formalized using the first-next-last logic under the assumption that most events can be explicitly identified with three known events, especially if the event of interest is stated in the middle. We also assume that longer event sequences can be divided into packages of three.

For our formalization approaches, accurate and consistent naming conventions in the formalized FEKSs are vital to enable a further (semi-)automated fusion with IoT data. This allows a straightforward extraction of values from FEKSs using, e.g., basic pattern recognition. For this POC, we restrict variable values to specific predefined options in the FEKSs, such as the state value to "failure". Our future work will include enhancing formalization approaches to accommodate misspellings and enable free-text expressions.

Table 1: Formalization approaches for EKSs with fault and failure event sequence reliability information.

Reliability Information	**Formalization Approach**
fault	IF + *condition* + THEN + *state* + IN + *component/system*
event sequence	FIRST + *event1* + NEXT + *event2* + LAST + *event3*

3.2 Main Strategies and Algorithms for Fusing Expert Knowledge Statements with Data

The fusion of EK with IoT data for Petri net model extractions, can occur at different stages of the model extraction process (Dixit et al. 2017). For our case study, we propose the integration of EKSs into Petri nets in two main strategies: 1) *during* the Petri net model extraction (*a priori*), and 2) *after* the Petri net model extraction (*a posteriori*). The *a priori* and *a posteriori* strategies are both executed in two ways to fuse failure event sequence and fault reliability information into Petri nets. Thus, we introduce four fusion algorithms, which are displayed in Table 2 with their respective strategy, formalization logic, required input and results. In the following, we detail synthetic data requirements as input for the execution of the *a priori* strategy fusion algorithms as well as introduce the four different fusion algorithms and their validation.

Table 2: Fusion algorithms within the a priori and a posteriori strategies.

Algorithm	**Strategy**	**Reliability Infor.**	**Formalization**	**Input Data**	**Result**
PRE-SEQ	a priori	event sequence	first-next-last	synthetic event log	modified transitions
PRE-FA	a priori	fault	if-then-in	synthetic state condition log	guard functions
POST-SEQ	a posteriori	event sequence	first-next-last	FEKSs	modified transitions
POST-FA	a posteriori	fault	if-then-in	FEKSs	guard functions

3.2.1 Synthetic Data as Input for the a Priori Fusion Algorithms

Both fusion algorithms of the *a priori* strategy require synthetic data as input in addition to IoT data. For the algorithms, we differ between two types of synthetic data, aligned with the IoT data levels of the EKS.

Synthetic event data is used as the input for *PRE-SEQ* and generated manually in our case study. Synthetic event data from EK provides reliability information about event sequences for which, e.g., no IoT data is available. As the model extraction in our case study is based on an α-miner (Van Der Aalst et al. 2004), the log only requires a small volume since one appearance of the event of interest is enough.

Synthetic state condition data is a log that contains both condition monitoring and state data entries. This synthetic data is used as the input for *PRE-FA* and generated semi-automated. The mapping of existing IoT state with condition monitoring log entries is done automatically based on the timestamps (TSs). Next, key sensor values are either highlighted manually by experts or automatically extracted from FEKSs information. With this, the *PRE-FA* algorithm is guided on where to look for the extraction of the faults.

3.2.2 Fusion Algorithms for Petri net Model Extractions and Approach Validation

In the following, we detail and explain our four developed fusion algorithms and our validation approach. In Algorithms 1 - 4, we detail every fusion algorithm with a pseudocode extract, starting with the algorithms that focus on failure event sequence information and continuing with fault information.

Algorithm 1: PRE-SEQ pseudocode extract.

```
e.g. FIRST = event after term "FIRST"; e.g.
  FIRST-NEXT = place from FIRST to NEXT;
merge synthetic and IoT event log;
while Petri net ddra extraction do
    block place from FIRST to LAST;
    block input arc FIRST-LAST to LAST;
    add input arc NEXT-LAST to LAST;
    block output arc FIRST to FIRST-LAST;
    add output arc FIRST to FIRST-NEXT;
end
```

Algorithm 2: POST-SEQ pseudocode extract.

```
e.g. FIRST = event after term "FIRST"; e.g.
  FIRST-NEXT = place from FIRST to NEXT;
execute ddra library on IOT state and event log;
add place from FIRST to NEXT;
add place from NEXT to LAST;
add immediate transition of NEXT;
add input arc FIRST-NEXT to NEXT;
add input arc NEXT-LAST to LAST;
add output arc FIRST to FIRST-NEXT;
add output arc NEXT to NEXT-LAST;
```

Algorithm 3: PRE-FA pseudocode extract.

```
syndata = synthetic state condition log;
sv = sensor value; lsv = lowest sv;
filter syndata for resource equal to component;
for entry in syndata do
    get sv based on EKS condition (e.g. sv from
      [row-1] where row status equal to "failure")
    add sv to sv list
end
find lsv out of sv list;
set guard function to equal or larger than lsv;
modify assigned transition to type immediate;
add guard function to assigned transition;
```

Algorithm 4: POST-FA pseudocode extract.

```
c = condition; t = transition; s = statement;
for statement in list of FEKS do
    set c to s between "IF" and "THEN";
    add c to c list;
end
for t in transitions of model do
    if t is assigned to one c in c list then
        modify t to type immediate;
        set guard function of t to assigned c;
    end
end
```

The *PRE-SEQ* algorithm merges the synthetic and IoT event log for extraction and hinders/adds parts of the Petri net during the generation. The *POST-SEQ* algorithm extracts the transitions and their sequence directly from FEKS and adds new transitions to the model. The *PRE-FA* algorithm calculates the threshold of the fault value from and guided by synthetic data that leads to a component failure. The *POST-FA* algorithm extracts the stated fault directly from FEKSs and adds the fault as guard function to the model.

We perform our validation with three extracted Petri nets from the following extraction strategies: 1) solely data-driven, 2) *a priori* (*PRE-SEQ* and *PRE-FA*) and 3) *a posteriori* (*POST-SEQ* and *POST-FA*). For the validation, we use a face validity and manual comparison. We define the validation goal in three key performance indicators (KPIs) to determine the number of correct timed and immediate transitions as well as guard functions. Based on the KPIs and extracted Petri nets, we explain, discuss and interpret the results.

4 CASE STUDY: FUSION OF EXPERT KNOWLEDGE STATEMENTS WITH DATA FOR RELIABILITY CENTERED DIGITAL TWIN MODEL EXTRACTION

In this section, we demonstrate our approach from Section 3 with a POC for the fusion of IoT data and EKSs, focused on reliability DT models. With our approach we were able to extract more precise DT models, as a step towards systematically utilizing EK for automated model extractions within DTs. With both our fusion strategies we were able to extract Petri nets including more detected fault sequences and faults than in a data-driven extracted Petri net. Based on our validation, the a posteriori strategy performs slightly better than the a priori strategy in this case study. In the next subsection, we outline the case study setup for fusing EKSs in the *a priori* and *a posteriori* strategies with our four designed fusion algorithms.

4.1 Case Study Setup

We base our case study on the model presented by Friederich and Lazarova-Molnar (2022) with slight modifications, so that the model consists of two manufacturing cells - Cell1 (C1) and Cell2 (C2) and one Automated Guided Vehicle (AGV) that delivers material to C2. To demonstrate the fusion of EK, we additionally modified the model by adding four EK related behaviors. The resulting modified model containing the EK information is what we refer to as the *ground-truth model*. We use this model to generate data from which we then re-discover the ground-truth model as Petri nets with our fusion algorithms.

Our case study consists of the following five steps: 1) Design of four EKSs; 2) Formalization of the EKSs into FEKSs; 3) Design of the ground-truth model and extraction of a Petri net on a data-driven basis from the generated logs; 4) Generation of synthetic log data and execution of the *a priori* and *a posteriori* fusion algorithms to re-discover the ground-truth model as Petri nets capturing the EKSs; and 5) Validation of the effectiveness of our approach by comparing Petri net models extracted in a purely data-driven manner with the Petri net models extracted by fusing data and EK via the two strategies.

4.2 Step 1: Case Study Expert Knowledge Statements

We designed four EKSs, each characterized across the three evaluation dimensions from Subsection 3.1. For the POC, we assumed the degree of belief to be 100%, fault information to correspond to sensor values and no EKS contradictions. The created EKSs are listed in Table 3 and ordered with increasing complexity.

Table 3: Case study's EKSs containing different evaluation dimension features.

	Expert Knowledge Statements
EKS1	After Cell1 fails, the repair needs between 20 minutes and 2 hours before the operation of Cell1 can be continued and the order can be completed.
EKS2	If the sensor of manufacturing Cell1 reaches a value above approximately 4, the operation capacity of Cell1 drops and Cell1 needs repair.
EKS3	Cell2 fails most of the time within the next hour as soon as I have heard a sharp loud noise from the machine in Cell2. I guess the noise is somewhere above 90 decibels.
EKS4	The AGV fails in rare cases during the unloading and can't proceed in its operation unless twisted Material2 is removed from the AGV. It could be because a human worker jostled the forks minimally when loading the AGV manually. The AGV loading happens after a new order arrived and before the AGV transports Material2 to Cell2.

EKS1 uses information on the event data level as it contains information about the sequence of events and needed time distance. EKS2 and EKS3 use information from condition monitoring and corresponding state data levels. EKS4 contains information corresponding to state and system data levels. For EKS2, sensor data are available for the fault from condition monitoring logs. EKS3 is similar to EKS2 but more complex, as historic data from the last hour needs to be considered from the available IoT data. Furthermore, noise is not a fault in itself but can be an indicator of a fault. For this POC, we assume a direct correlation between noise and the C2 fault and, thus, we handle noise as a fault. EKS4 is the most complex and difficult to detect from data, as no IoT data is available on material twists and manual loading processes.

4.3 Step 2: Formalization of Expert Knowledge Statements

We used our two formalization approaches for fault and event sequence reliability information from Subsection 3.1.2 to formalize the EKSs manually. The resulting FEKSs are shown in Table 4.

Table 4: FEKSs, formalized from EKSs in Table 3.

	Formalized Expert Knowledge Statements
FEKS1	FIRST repair_cell1 NEXT cell1_operation LAST order_completed
FEKS2	IF cell1_sensor >= 4 THEN failure IN cell1
FEKS3	IF cell2_sensor [in range – 60min] >= 90 THEN failure IN cell2
FEKS4a	IF material2 == twisted THEN failure IN AGV
FEKS4b	FIRST new_order NEXT manual_loading LAST agv_transport_and_unload

FEKS1 contains failure event sequence information for C1. Hence, it is formalized with the first-next-last logic. FEKS2 adopts an if-then-in logic as it contains fault information. FEKS3 also follows the if-then-in logic and incorporates a time range in the condition. FEKS4 comprises various statements, beginning with fault and followed by failure event sequence information. However, since specific values are absent in the EKS, its formalization is ambiguous. For FEKS4, possible faults could be a twist in Material2 or an error in the manual loading. Considering the expert's suggestion that a faulty loading process could lead to material twisting, twisted Material2 is chosen as the primary fault and listed in FEKS4a. The information about the loading process event sequence is specified using the event sequence formalization in FEKS4b.

4.4 Step 3: Ground-Truth Model and Petri Net Generation via a Data-Driven Extraction

Next, we created the ground-truth model by adding reliability information about AGV, C1 and C2 from EKS1-4 in Table 3. We simulated the ground-truth model to generate IoT condition monitoring, event and state logs that contain the same reliability information as in EKS1-4. As the EKS includes condition-based triggers for failures, the ground-truth model is adapted to simulate this behavior. From the event and state logs, we extracted a Petri net using the data-driven model extraction library ddra (Friederich 2023), based on an α-miner (Van Der Aalst et al. 2004). The Petri net extracted on a data-driven basis with the initial ddra library is displayed in Figure 2, where ddra annotates the Petri net model with time distribution functions for each transition. For a better comprehension, we redrew the Petri nets in this paper to enhance only the graphical representation and marked the included FEKSs information.

Figure 2 shows, e.g., the failure event sequence of FEKS1 (purple boxes), where C1 needs to be repaired before the operation can be continued and the order is completed. For FEKS4b (orange boxes), the “manual_loading” event is missing in the Petri net. For FEKS2 (green box), the fault is not correctly captured in this Petri net, as the ground-truth model actually simulates a failure if the triangular distributed sensor value exceeds 4.02. However, even if ddra would be able to detect that it is a condition-based fault, it would not be able to detect the correct fault efficiently as the data-driven ddra does not know where to look for in its search for the correct fault without EKS. Faults can happen in many different variants.

In the next section, we extended the ddra library with our four fusion algorithms so that EKSs is utilized to include the incorrect or missing information from the ground-truth model in the extracted Petri nets.

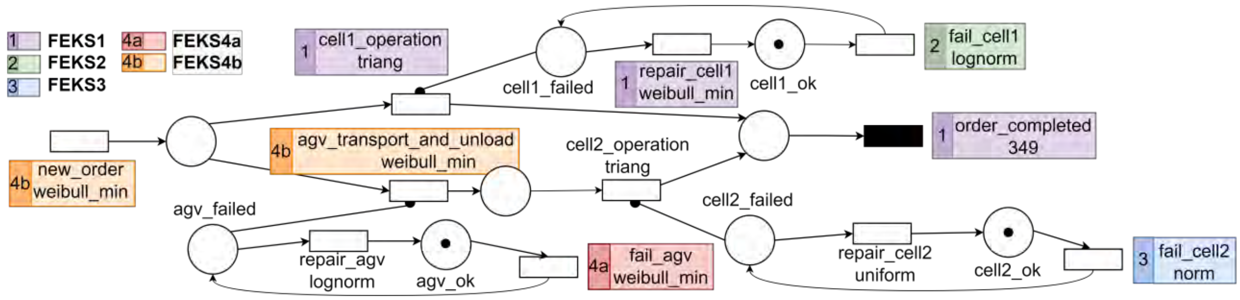

Figure 2: Petri net, extracted on a data-driven basis using the ddra library.

4.5 Step 4: Fusion of Expert Knowledge Statements with Data for Model Extractions

We implemented the fusion of EKSs and IoT data from Subsection 3.2, using two strategies for executing the fusion algorithms on FEKSs. We directly compared the results of the algorithms *PRE-SEQ* and *POST-SEQ* as well as *PRE-FA* and *POST-FA,* as they focus on the same type of reliability information.

4.5.1 Generating Synthetic Data for the A Priori Strategy PRE-SEQ and POST-SEQ Algorithms

We created the synthetic data types, described in Subsection 3.2.1, for both fusion algorithms of the *a priori* strategy. Table 5 shows a simplified excerpt of synthetic event data containing a failure event sequence of FEKS4b that served as input for *PRE-SEQ*. Table 6 shows a simplified excerpt of synthetic state condition data containing merged sensor values of FEKS2 that served as input for *PRE-FA*. For FEKS2, the sensor value one step prior to the failure state is marked as fault based on the information in the EKS.

Table 5: Synthetic event log.

TS	ID	Resource	Event	Type
1	602	mes	new_order	NA
2	602	human	manual_loading	start
3	602	human	manual_loading	end
4	602	agv	agv_transport_and_unload	start
5	602	agv	agv_transport_and_unload	end

Table 6: Synthetic state condition log.

TS	Resource	State	Sensor	Value
1	cell1	busy	cell1_sensor	4.02
2	cell1	failure	cell1_sensor	0
21	cell1	busy	cell1_sensor	3.47
22	cell1	busy	cell1_sensor	4.81
23	cell1	failure	cell1_sensor	0

4.5.2 Execution of PRE-SEQ Algorithm (A Priori) and POST-SEQ Algorithm (A Posteriori)

Next, we executed the *PRE-SEQ* and *POST-SEQ* fusion algorithms that focus on integrating event sequence information. For implementation, we created a new component for each in the ddra library. The result of both algorithms for FEKS4b is displayed in Figure 3 with an extract for the *PRE-SEQ* and the complete Petri net for the *POST-SEQ* result. The relevant novel parts are marked in orange. The results of both fusion algorithms show the newly added loading transition. However, the results differ, as the *POST-SEQ* algorithm adds the new transition as an immediate one, since there is no information about the distribution in the FEKS, and keeps the old place from “new_order” to “agv_transport_and_unload”.

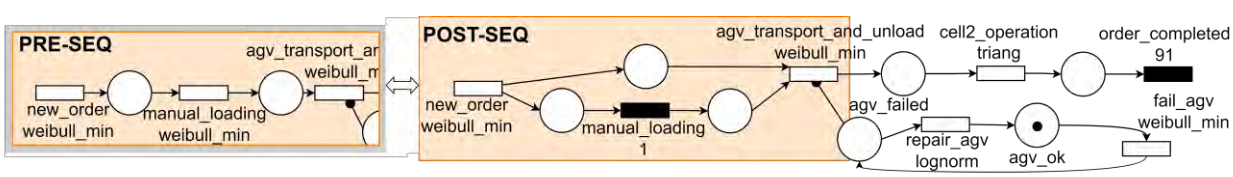

Figure 3: PRE-SEQ (left) and POST-SEQ (right) Petri net extract for FEKS4b featuring transitions.

4.5.3 Execution of PRE-FA Algorithm (A Priori) and POST-FA Algorithm (A Posteriori)

Subsequently, we executed the *PRE-FA* and *POST-FA* algorithm on FEKS2 - FEKS4a, as they contain fault information. The Petri net extraction results for FEKS2 of both algorithms are shown in Figure 4 with a green marking of the novel parts. Both algorithms change the timed failure transition into an immediate transition and discover the fault sensor correctly for the guard function. *PRE-FA* calculates a threshold of 4.02 as fault value for the guard function, *POST-FA* extracts the value 4, which leads to different results.

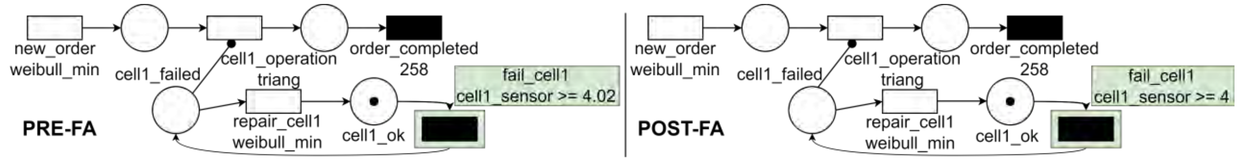

Figure 4: PRE-FA (left) and POST-FA (right) Petri net extract for FEKS2 featuring guard functions.

4.6 Step 5: Analysis and Validation of the Fusion Algorithms

We calculated the validation KPIs, shown in Table 7, as described in Subsection 3.2.2, from the Petri nets in Figures 2 and 5. The Petri net in Figure 5 contains the information of both the *a priori* and *a posteriori* Petri nets, as for each extracted FEKSs the results of both *PRE-* and *POST-* fusion algorithms are visible. Therefore, one result is included and labeled in the Petri net and the other is stated below in parenthesis.

Table 7: KPI results of the data-driven, a priori and a posteriori extractions.

KPI	Data-driven Extraction	A Priori Strategy	A Posteriori Strategy
Correct timed transitions	7	8	8
Correct immediate transitions	1	3	4
Correct guard functions	0	2	3

From the KPIs, the Petri net from the data-driven extraction (Figure 2) includes the least correct extracted parts. This is the case, as the data-driven extraction can correctly detect event sequences only if respective IoT data is available. Furthermore, the data-driven extraction fails to capture faults, as the extraction only uses event and state logs. However, even if the data-driven extraction were to incorporate the neglected condition monitoring log, it may not be able to detect FEKS3 and FEKS4a/b as they are more complex due to their time distance between fault and failure and non-existent available IoT data.

The fusion strategies overcome these shortcomings by either integrating FEKSs directly or synthetic data from FEKSs to correctly detect more elements in the extracted Petri nets. From the KPIs, the *a posteriori* strategy (with *POST-SEQ* and *POST-FA*) leads to the best results on FEKS1-4. The *a priori* strategy (with *PRE-SEQ* and *PRE-FA*) misses one immediate transition and responding guard function for FEKS4a, as without available IoT data no synthetic data can be generated semi-automatically.

However, the results of the validation ultimately depend on the underlying EKSs and if the EKSs are more suitable for an *a priori* or *a posteriori* strategy. In this case study, one possible resulting Petri net is shown in Figure 5 using three out of four fusion algorithms. For the resulting Petri net, we manually selected fusion algorithms for each FEKS and labeled the algorithm that extracted the information. Thus, the fault information of FEKS2 and FEKS3 are extracted via *PRE-FA*, as the synthetic state condition data contain more detailed sensor values than the FEKSs. FEKS4a is extracted with *POST-FA*, as there are no IoT data available. FEKS4b is extracted from *PRE-SEQ* to not include the additional place shown in Figure 3.

Furthermore, with a more thorough comparison of the *a priori* and *a posteriori* strategies for the *PRE-FA* and *POST-FA* results, e.g., of FEKS2 and FEKS3, we conclude that the *PRE-FA* algorithm could be used to validate EKSs. *PRE-FA* is able to find precise fault values from IoT sensor data but relies on the FEKS as guidance on where to look for the values, whereas *POST-FA* just uses the information from the

FEKSs. Therefore, the fault values from *PRE-FA* and *POST-FA* could differ significantly in their values, e.g., if experts make a mistake, state a vague guess about reliability information or IoT data is incorrect.

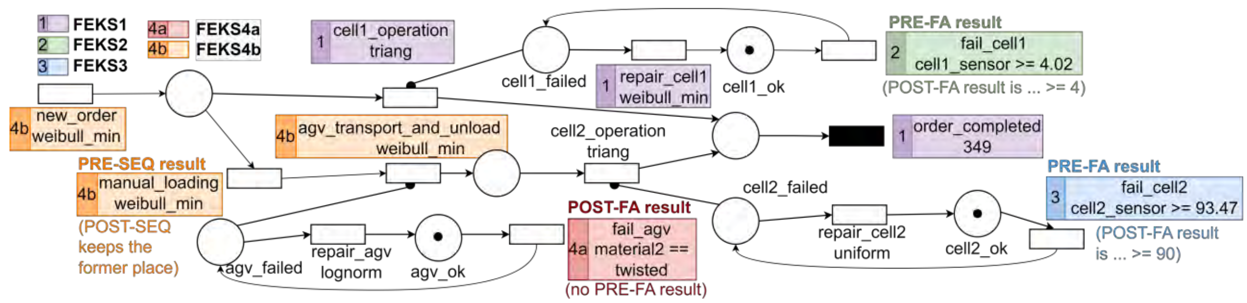

Figure 5: Petri net extracted by fusing IoT data and FEKSs using PRE-FA, POST-FA and PRE-SEQ.

5 SUMMARY AND OUTLOOK

In this paper, we continued our earlier work on the framework for fusing data and expert knowledge for Digital Twins, by proposing an approach to fuse expert knowledge with data for Petri net model extractions. Our two main strategies, *a priori* and *a posteriori*, consider including expert knowledge during and after the model extraction, correspondingly. We designed and implemented four fusion algorithms (two for each strategy) as a Proof of Concept for our case study in reliability modeling. For this, we devised and formalized four expert knowledge statements. We, furthermore, generated synthetic data as input for our *a priori* fusion algorithms. Next, we executed and explained the results of each fusion algorithm and extracted corresponding Petri nets. Our case study results demonstrate the feasibility of explicitly combining expert knowledge and data for different reliability information. In our future work we will refine our approach and extend its scope by, e.g., adding functionality to handle conflicts between and within expert statements and IoT data, as well as automating our formalization and algorithm decisions. Further, we aim to validate the correctness of IoT data and expert knowledge statements and integrate uncertainty of expert knowledge.

REFERENCES

Bender, S., and A. Fish. 2000. "The Transfer of Knowledge and the Retention of Expertise: The Continuing Need for Global Assignments". *Journal of Knowledge Management* 4 (2):125-137.

Chowdhary, K. R. 2020. "Natural Language Processing". In *Fundamentals of Artificial Intelligence,* edited by K. R. Chowdhary, 603-649, New Delhi, India: Springer India.

Deng, J., Q. Zheng, G. Liu, J. Bai, K. Tian, C. Sun, Y. Yan, and Y. Liu. 2021. "A Digital Twin Approach for Self-Optimization of Mobile Networks". *2021 IEEE Wireless Communications and Networking Conference Workshops (WCNCW),* March 29th-April 1st, 1-6.

Dixit, P. M., J. C. A. M. Buijs, W. M. P. van der Aalst, B. F. A. Hompes, and J. Buurman. 2017. "Using Domain Knowledge to Enhance Process Mining Results". *International Symposium on Data-Driven Process Discovery and Analysis (SIMPDA),* December 6th-8th, Vienna, Austria, 76-104.

Eirinakis, P., K. Kalaboukas, S. Lounis, I. Mourtos, J. M. Rožanec, N. Stojanovic, and G. Zois. 2020. "Enhancing Cognition for Digital Twins". *2020 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC),* June 15th-17th, 1-7.

Friederich, J. 2023. Data-Driven Reliability Assessment of Cyber-Physical Production Systems. https://github.com/jo-chr/ddra, accessed 22nd February.

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

Friederich, J., S. C. Jepsen, S. Lazarova-Molnar, and T. Worm. 2021. "Requirements for Data-Driven Reliability Modeling and Simulation of Smart Manufacturing Systems". In *2021 Winter Simulation Conference (WSC),* 1-12 https://doi.org/10.1109/WSC52266.2021.9715410.

Friederich, J., and S. Lazarova-Molnar. 2022. "Data-Driven Reliability Modeling of Smart Manufacturing Systems Using Process Mining". In *2022 Winter Simulation Conference (WSC),* 2534-2545 https://doi.org/10.1109/WSC57314.2022.10015301.

Fuller, A., Z. Fan, C. Day, and C. Barlow. 2020. "Digital Twin: Enabling Technologies, Challenges and Open Research". *IEEE Access* 8:108952-108971.

Hu, F., W. Zhong, L. Ye, D. Duan, and Q. Zhang. 2023. "A General Paradigm of Knowledge-Driven and Data-Driven Fusion". *2023 15th International Conference on Advanced Computational Intelligence (ICACI),* May 6th-9th, Seoul, South Korea, 1-7.

Jungmann, M., and S. Lazarova-Molnar. 2024. "Towards Fusing Data and Expert Knowledge for Better-Informed Digital Twins: An Initial Framework". *The 7th International Conference on Emerging Data and Industry (EDI40),* April 23rd-25th, Hasselt, Belgium.

Jurasky, W., P. Moder, M. Milde, H. Ehm, and G. Reinhart. 2021. "Transformation of Semantic Knowledge into Simulation-Based Decision Support". *Robotics and Computer-Integrated Manufacturing* 71:102174.

Kabir, S., and Y. Papadopoulos. 2019. "Applications of Bayesian Networks and Petri Nets in Safety, Reliability, and Risk Assessments: A Review". *Safety Science* 115:154-175.

Ladj, A., Z. Wang, O. Meski, F. Belkadi, M. Ritou, and C. Da Cunha. 2021. "A Knowledge-Based Digital Shadow for Machining Industry in a Digital Twin Perspective". *Journal of Manufacturing Systems* 58:168-179.

Lazarova-Molnar, S., N. Mohamed, and H. R. Shaker. 2017. "Reliability Modeling of Cyber-Physical Systems: A Holistic Overview and Challenges". *2017 Workshop on Modeling and Simulation of Cyber-Physical Energy Systems (MSCPES),* April 21st, Pittsburgh, Pennsylvania, USA, 1-6.

Liu, H.-C., J.-X. You, Z. Li, and G. Tian. 2017. "Fuzzy Petri Nets for Knowledge Representation and Reasoning: A Literature Review". *Engineering Applications of Artificial Intelligence* 60:45-56.

Luan, T. H., R. Liu, L. Gao, R. Li, and H. Zhou. 2021. "The Paradigm of Digital Twin Communications". *arXiv preprint arXiv: 2105.07182.*

Niloofar, P., and S. Lazarova-Molnar. 2021. "Fusion of Data and Expert Knowledge for Fault Tree Reliability Analysis of Cyber-Physical Systems". *2021 5th International Conference on System Reliability and Safety (ICSRS),* November 24th-26th, Palermo, Italy, 92-97.

Resman, M., J. Protner, M. Simic, and N. Herakovic. 2021. "A Five-Step Approach to Planning Data-Driven Digital Twins for Discrete Manufacturing Systems". *Applied Sciences* 11 (8):3639.

Stojanovic, N., and D. Milenovic. 2018. "Data-Driven Digital Twin Approach for Process Optimization: An Industry Use Case". *2018 IEEE International Conference on Big Data (Big Data),* December 10th-13th, Seattle, Washington, USA, 4202-4211.

Todorovski, L., and S. Džeroski. 2006. "Integrating Knowledge-Driven and Data-Driven Approaches to Modeling". *Ecological Modelling* 194 (1-3):3-13.

Uhlemann, T. H. J., C. Lehmann, and R. Steinhilper. 2017. "The Digital Twin: Realizing the Cyber-Physical Production System for Industry 4.0". *Procedia CIRP* 61:335-340.

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

Vogel-Heuser, B., F. Ocker, I. Weiß, R. Mieth, and F. Mann. 2021. "Potential for Combining Semantics and Data Analysis in the Context of Digital Twins". *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 379 (2207):20200368.

Webert, H., T. Döß, L. Kaupp, and S. Simons. 2022. "Fault Handling in Industry 4.0: Definition, Process and Applications". *Sensors* 22 (6):2205.

Wunderlich, A., K. Booth, and E. Santi. 2021. "Hybrid Analytical and Data-Driven Modeling Techniques for Digital Twin Applications". *2021 IEEE Electric Ship Technologies Symposium (ESTS),* August 4th-6th, Arlington, Virginia, USA, 1-7.

Zonta, T., C. A. da Costa, R. da Rosa Righi, M. J. de Lima, E. S. da Trindade, and G. P. Li. 2020. "Predictive Maintenance in the Industry 4.0: A Systematic Literature Review". *Computers & Industrial Engineering* 150:106889.

AUTHOR BIOGRAPHIES

MICHELLE JUNGMANN is a Ph.D. student at the Institute of Applied Informatics and Formal Description Methods at the Karlsruhe Institute of Technology. Her research is focused on the fusion of expert knowledge and data for hybrid model extraction approaches in the context of Digital Twins. Her email address is michelle.jungmann@kit.edu.

SANJA LAZAROVA-MOLNAR holds two full professorships, at the Karlsruhe Institute of Technology, and the University of Southern Denmark. Her research focuses on data-driven simulation, digital twins, and cyber-physical systems modeling for reliability and energy efficiency enhancement. Actively engaged in developing advanced methodologies, she leverages her expertise to optimize complex systems through data-driven simulation. She leads activities focused on digital twins and data-driven simulation in several European and national projects. Furthermore, Professor Lazarova-Molnar assumes leadership roles in IEEE and The Society for Modeling & Simulation International (SCS), contributing significantly to these professional organizations. She was also one of the Proceedings Editors for the Winter Simulation Conference in 2019 and 2020 and an associate editor of SIMULATION: Transactions of The Society for Modeling and Simulation International. Her email address is sanja.lazarova-molnar@kit.edu.