Proceedings of the 2021 Winter Simulation Conference
S. Kim, B. Feng, K. Smith, S. Masoud, Z. Zheng, C. Szabo, and M. Loper, eds.

REQUIREMENTS FOR DATA-DRIVEN RELIABILITY MODELING AND SIMULATION OF SMART MANUFACTURING SYSTEMS

Jonas Friederich
Sune Chung Jepsen
Sanja Lazarova-Molnar
Torben Worm

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

ABSTRACT

Planning and deploying reliable Smart Manufacturing Systems (SMSs) is of increasing interest to both scholars and practitioners. High system reliability goes hand in hand with reduced maintenance costs and enables optimized repairs and replacements. To leverage the full potential of SMSs and enable data-driven reliability assessment, data needs should be precisely defined. System integration is a key concept of the Industry 4.0 initiative and it can aid the extraction of the needed data. In this paper, we study the data requirements for a novel middleware for SMSs to enable and support data-driven reliability assessment. We present this middleware architecture and demonstrate its application through a case study, which is used to generate exemplary data that corresponds to the derived requirements. The data requirements and the middleware architecture can support researchers in developing novel data-driven reliability assessment methods, as well as assist practitioners in designing and deploying SMSs in companies.

1 INTRODUCTION

Smart Manufacturing (SM) is a technological concept that aims to turn some of the principles of the Industry 4.0 (I4.0) vision into reality (Kusiak 2018). The implementation of SM involves integration of sensors in production assets to collect data on their operational status and performance. An asset in this context is defined as the interface to the machine software, but in general it can be anything that adds value to an I4.0 solution such as machines or other software (Plattform Industrie 4.0 2021a). Thus, Smart Manufacturing Systems (SMSs) focus on the connection of such assets through networks to monitor all aspects of the production processes.

Engineers have long made the design and construction of reliable manufacturing systems a high priority (Chakraborty and Ankiah 1989; Miriyala and Viswanadham 1989). Besides reducing maintenance cost (Kuo and Kim 1999), proper reliability assessment of manufacturing systems can also be used to optimize maintenance schedules (Liu et al. 2018). With the advent of new technologies (e.g., system integration, Internet of Things, simulation, autonomous robots) as part of SM, there is the potential for enhancing, improving and even developing new methods to enable data-driven assessment of reliability (Lazarova-Molnar and Mohamed 2019). These new methods must be able to cope with the increase in potential operational risks that come hand in hand with the increasing complexity of SMSs (Han et al. 2019).

In Friederich and Lazarova-Molnar (2021), we proposed a framework for data-driven reliability assessment. Data, such as condition monitoring data and event data can be used to automatically generate reliability models (e.g., Fault Trees (Ruijters and Stoelinga 2015) and Reliability Block Diagrams (Čepin 2011)).

978-1-6654-3311-2/21/$31.00 ©2021 IEEE

The generated reliability models can be used to run simulations and to calculate reliability measurements of systems under study. Data requirements of reliability modeling methods vary a lot. While classical Reliability Block Diagrams rely solely on information about states of individual assets and manufacturing lines, Fault Trees require detailed semantic information about faults that happened and potentially can happen. Satisfying these data requirements calls for novel approaches of system design. Asset integration plays a vital role and enables communication and data gathering across a manufacturing line.

There are few research efforts in data-driven simulation and reliability assessment that focus on manufacturing systems (Lugaresi and Matta 2020; Khorshidi et al. 2016) as well as other systems (Lazarova-Molnar et al. 2020). Lugaresi and Matta (2020) propose a method for generating and tuning discrete-event simulation models for manufacturing applications. A novel data-driven system reliability and failure behavior modeling method is introduced by Khorshidi et al. (2016). The authors verify their method using a case study of a manufacturing line and compare it to conventional approaches. In Lazarova-Molnar et al. (2020), we present an approach for data-driven learning and analysis of Fault Trees of systems with multi-state components. Based on the extracted Fault Trees, reliability measures of the system under study can be estimated.

The above-mentioned research efforts propose novel data-driven methods for discrete-event simulation or specific reliability assessment methods, which rely to a great extent on data that is generated by the systems under study. However, we have no knowledge of a contribution aiming to define general data requirements for SMSs in support of these data-driven methods, also relevant for defining the underlying software architectures. Thus, the research goal of this article is the identification and specification of data requirements for a novel middleware, the I4.0 Information Backbone (I4.0-IB), so that it can support data-driven simulation and reliability assessment of SMSs. In short, our main contributions towards this goal are

- Specification of data requirements for data-driven reliability assessment and simulation of SMSs
- Presentation of a novel middleware architecture for SMSs in support of the defined data requirements
- Illustrative simulation model of a manufacturing line that generates exemplary reliability-relevant data

The defined data requirements are further categorized based on the structural granularity of the reliability-relevant data. Based on these categories, we provide a mapping to common reliability assessment methods.

The remainder of this article is structured as follows: We begin by providing background on Industry 4.0 and common approaches for reliability assessment of manufacturing systems in Section 2. In Section 3, we define the data requirements for reliability assessment of Smart Manufacturing Systems. We present the case study and the novel information backbone in Section 4. In Section 5, we discuss implications of this work, followed by a summary and outlook in Section 6

2 BACKGROUND

2.1 Industry 4.0

The industry is currently entering the fourth industrial revolution, opening new possibilities through data-related technologies such as the Internet of Things (IoT), cloud, big data and simulation, as well as operational technologies such as autonomous robots, augmented reality and additive manufacturing (Lazarova-Molnar and Mohamed 2019). This plethora of new technologies is continuously driving the development of new applications aiming to improve manufacturing performance (Dalenogare et al. 2018).

I4.0 is formally defined as *the intelligent networking of machines and processes for industry with the help of information and communication technology* (Plattform Industrie 4.0 2021a). Based on this definition, it can be inferred that software plays a vital role in connecting and supporting information flows towards an intelligent network. Figure 1 shows the shift from traditional hierarchical manufacturing to an network-oriented I4.0 approach (Jepsen et al. 2020). The figure supports the I4.0 vision of an intelligent network by symbolizing information flows between assets regardless of layers. This opens up new possibilities for applications to obtain information more effectively that was more cumbersome in traditional manufacturing.

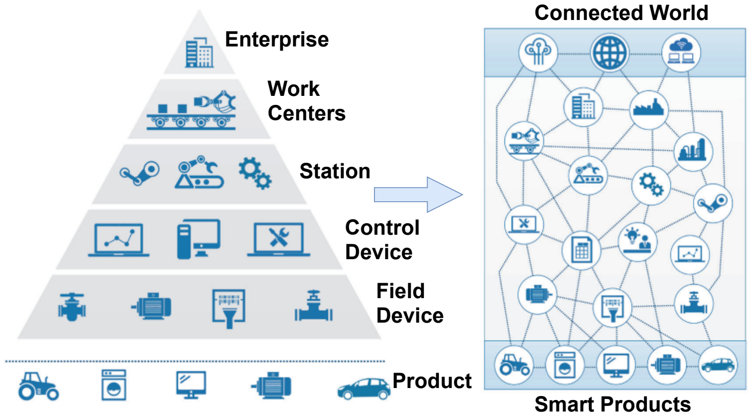

Figure 1: Shift from hierarchical manufacturing to an network oriented I4.0 approach (Jepsen et al. 2020).

Based on an intelligent network, new applications in the field of simulation and reliability assessment of manufacturing systems can be developed. These applications specifically include data-driven solutions that take advantage of easy access to data. While traditionally developed simulation and reliability models can quite accurately represent a system at specific points in time, data-driven solutions can more accurately reflect potential changes to a system that occur as time unfolds (Lugaresi and Matta 2020).

2.2 Reliability Assessment of Manufacturing Systems

Reliability is an important performance metric for any product, system or service and thus also plays a vital role in planning, deploying and operating contemporary production lines. In relation to manufacturing systems, reliability refers to the likelihood that the intended function will be performed for a specified period of time (Lazarova-Molnar et al. 2017). The reliability metric $R(t)$ is mathematically defined as $R(t) = \int_t^\infty f(x)dx$ where t is the period of time and $f(x)$ is the failure probability density function of a given distribution. Related metrics are *mean time to failure* (MTTF) for non-repairable systems, *mean time between failures* (MTBF) for repairable systems, *mean time to repair* (MTTR) and operational availability A. MTTF is defined by the mean of $R(t)$ as $MTTF = \int_0^\infty R(t)dt$ and MTBF as $MTBF = \frac{T(t)}{F}$ where $T(t)$ is the total operating time and F is the number of failures. MTTR is the average time it takes to restore assets. The operational availability A can, for instance, either be calculated as $A = \frac{uptime}{uptime+downtime}$ or $A = \frac{MTBF}{MTBF+MTTR}$.

Over the past few decades, scientists and practitioners have developed many methods to assess reliability of systems. Typically these methods have a qualitative and a quantitative aspect. Qualitative aspects include the analysis of the interaction of the components of a system and the derivation of possible events that can lead to failures. Quantitative aspects comprise the analysis of the failure rates, repair times and, eventually, calculation of reliability metrics. In the following, we describe in no particular order some of the most commonly used methods of reliability assessment and highlight their qualitative and quantitative aspects.

Fault Tree Analysis (FTA) is a deductive method for modeling failures at component level that ultimately lead to a system failure (Ruijters and Stoelinga 2015). With respect to manufacturing systems, FTA can be used to model failure dependencies of the manufacturing system itself or of specific assets that make up the system (e.g. automated guided vehicles, assembly tracks, collaborative robots). The qualitative aspect of FTA includes the derivation of the failure events that will lead to the system failure. In

FTA terminology these events are referred to as minimal cut sets. The quantitative aspect of FTA includes calculation of reliability metrics such as MTTF, MTTR and operational availability.

Reliability Block Diagram (RBD) is an inductive method to provide a schematic representation of a system to analyze and assess its reliability (Čepin 2011). Each block in the diagram represents a component and the corresponding reliability metric. Blocks can be connected in either sequential or parallel configuration. RBDs help engineers to understand architectures of systems and narrow them down to discover systems' weakest links. The qualitative aspect of the RBD method refers to the identification of the blocks (e.g. assets of a production line) and their dependencies. Calculations of reliabilities of each block, as well as the reliability of the entire system, make up the quantitative aspect.

Petri Nets (PNs) are a common tool for assessing reliability and safety of complex systems or networks of systems. PNs that fire transitions after a random delay which is determined by a random variable are referred to as stochastic PNs (Adamyan and He 2002). By constructing PNs, engineers can assess the impact of failures and their sequence on reliability as well as analyze combined failure modes and estimate their probability of occurrence. PNs can also be used for FTA by replacing logic gate functions (Liu and Chiou 1997). The qualitative aspect of constructing PNs includes determining the different components of a system (i.e., places, transitions) and their relationships (i.e., arcs). Similar to FTA, calculating reliability metrics such as MTTF, MTTR and the reliability itself are part of the quantitative aspect of PNs.

Markov Models (MMs) for reliability assessment are based on Markov chains and are commonly used to represent a system at various stages (i.e., states) at any given time. Transitions connect states and represent rates or probabilities that a system will change from one state to another. The key feature of MMs is that they are *memoryless* (Ali 2019). *Memoryless* means, that the future state of the system depends only on information from the current state and not on any information from any states prior to the current state. Memorylessness imposes an oversimplifying assumption on the reliability models that MMs can handle. The qualitative aspect of MMs is the derivation of a transition diagram including possible states and transition probabilities of a system. The quantitative aspect encompasses the calculation of reliability metrics.

Discrete Event Simulation (DES) is a popular modeling and simulation paradigm. With DES, simulation progresses in time by processing a sequence of events. Each event is scheduled at a specific point in time and changes the state of the system when it is executed (Banks et al. 2009). For reliability assessment of manufacturing systems, DES can be used for the simulation and calculation of reliability metrics (Kampa et al. 2017). The qualitative aspect of DES includes the identification of possible events and parameters that influence the state of the system. The quantitative aspect comprises calculation of reliability metrics such as MTTF, MTTR and reliability.

3 RELIABILITY-RELATED DATA REQUIREMENTS FOR SMART MANUFACTURING SYSTEMS

In this Section, we define data requirements for supporting data-driven reliability modeling and assessment for SMSs. The presented work is within the context of developing a novel information backbone for the smart manufacturing facility in the Industry 4.0 Lab (Farahani 2021) at the University of Southern Denmark.

In Friederich and Lazarova-Molnar (2021), we proposed a framework for data-driven reliability assessment and identified relevant data sources in contemporary manufacturing systems. Building up on this knowledge, we studied the specific data requirements for some of the most common hardware reliability assessment methods to transfer them from expert-driven towards data-driven. We found that the data required can be categorized as either **state data**, **event data** or **condition monitoring data**. All three data types have in common that they have a time series format and thus each data record consists of a time stamp (ts) and type-specific data.

Figure 2 displays the identified data types. We, furthermore, provide a mapping in terms of data requirements of the data types and reliability assessment methods reviewed in Subsection 2.2. The potential level of detail of the generated reliability models increases when more data is available.

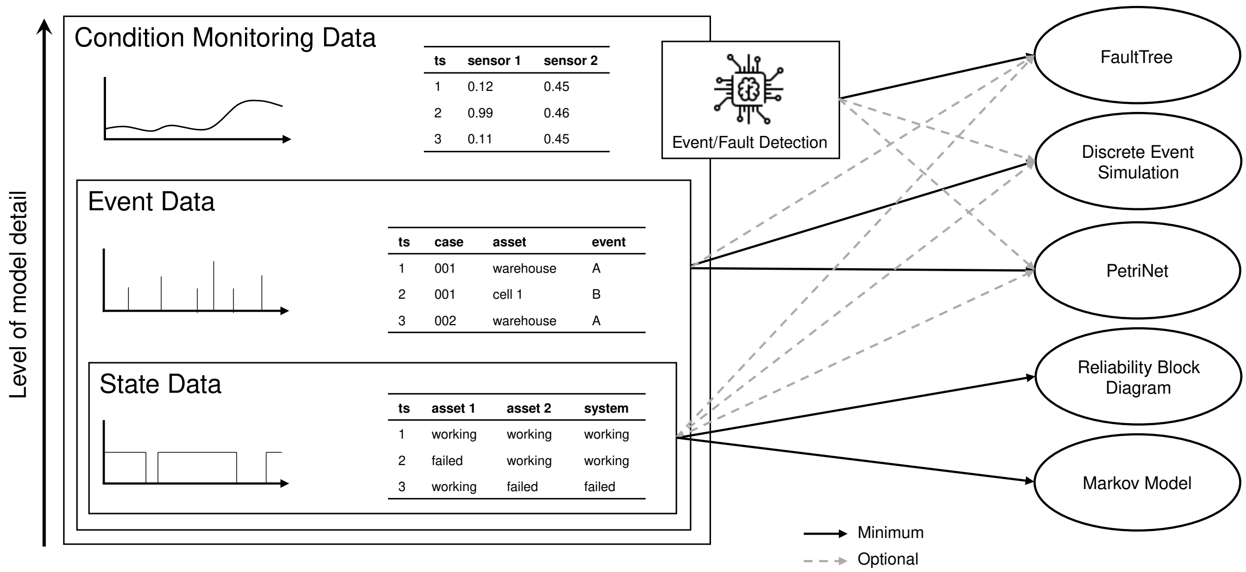

Figure 2: Proposed requirements for data-driven reliability modeling and simulation of Smart Manufacturing Systems. For clarification, we provide exemplary data for each data type. On the right, data requirements of common reliability models are mapped to these data types.

State data provides a record of the different states of the individual assets of a SMS, as well as the SMS itself. The record could consist of the up- and down-times or, adding more detail, working, idle and failed states. On the one hand, this data can be considered as a very low-level source of information as no explanations are provided about what is happening in the system and why assets and the system are failing. On the other hand, the lack of detail reduces the effort involved in providing this type of data. When designing and developing middleware for SMS, it must be decided whether the asset state data is propagated by the asset itself or if the middleware requests the state in regular time intervals.

Event data provides a record of discrete events generated by the assets and the system. In this context, discrete events mark the beginning and ending of activities in the production that are relevant to a given simulation study. These activities could be, for instance, the preparation of raw material in a warehouse, the transport of material or operations at assembly cells. For each event record, a case identifier should be provided. The case identifier is used to group events that belong to the same case, i.e., a trace of events about the production of one product instance. On the one hand, event data provides valuable insights about what is happening in the manufacturing system and, thus, supports the generation of more realistic and accurate reliability and simulation models. On the other hand, the effort involved in providing this type of data increases as more detailed data needs to be extracted from the assets. This implies interfacing with assets from potentially different manufacturers. Furthermore, in case assets themselves can not provide event data, sensors have to be installed for the generation of such.

Condition monitoring data provide a record of relevant health data of a SMS. This includes data from sensors that are either built into the individual assets (e.g., torque sensors in collaborative robots) or installed at critical locations along the production line. The sensors used could include, for example, pressure and vibration sensors that provide continuous data, as well as image and spatial sensors. The condition monitoring data itself already increases the level of detail of derived reliability and simulation models. However, performing event or fault detection based on sensor data enables deeper insights in the system at hand and, for example, generation of detailed fault models.

RBDs and MMs focus on representing the structure and states a system can have respectively. Thus, they require solely state data. Very simplistic versions of DES models and PNs can also be constructed from state data since state changes of systems can also be represented as discrete events. However, event data is required in order to unfold the full modeling potential of these methods. State data and condition monitoring data, as well as detected events and faults, can contribute to a higher model detail. FTA requires data about faults that happen in a system. Therefore, data on detected faults based on condition monitoring data are required for this method.

4 CASE STUDY AND INFORMATION BACKBONE

This Section is structured as follows: We, first, describe the Industry 4.0 laboratory at the University of Southern Denmark, the drone production line and the assets involved in the production sequence. Second, we present the I4.0 Information Backbone (I4.0-IB) with a high-level overview of the architecture and the concept of asset integration to enable communication between the I4.0-IB and the software of the assets. Thirdly, we introduce a simulation model of the Industry 4.0 lab and present data output matching the previously defined requirements. The knowledge gained during the development of the I4.0-IB is continuously used as input to further improve and refine both the simulation model and the I4.0-IB.

4.1 Industry 4.0 Laboratory

The I4.0 lab at the University of Southern Denmark is a strategic initiative that targets the support of research, industry collaboration, innovation, and education in the latest I4.0 technologies. Furthermore, the lab is a demonstration window for industries working to adapt I4.0 technologies into their own production. The lab is currently in its initiating phase, in which an infrastructure of hardware and software is built, and experience with the different assets is being gained. The I4.0 lab integrates knowledge from different parts of the university as well as assets from different technology vendors. The research areas from the university are robotics, operations management, software, mechanics, etc. and the technology vendors are robot manufacturers, magnetic track manufacturers, warehouse manufacturers, automated guided vehicle (AGV) manufacturers, software vendors, etc.

An ongoing iterative project has been started to gain knowledge and experience with the assets. The goal is to produce a simplified drone in a production sequence. One of the ideas is building a middleware infrastructure (I4.0-IB) that facilitates information flow between the different assets. Doing so gains experience and knowledge with the overall middleware architecture, the asset technologies, data design, and interface design needed to integrate to the assets. Figure 3 illustrates an excerpt of the assets involved in the production sequence, which we describe in detail in the following.

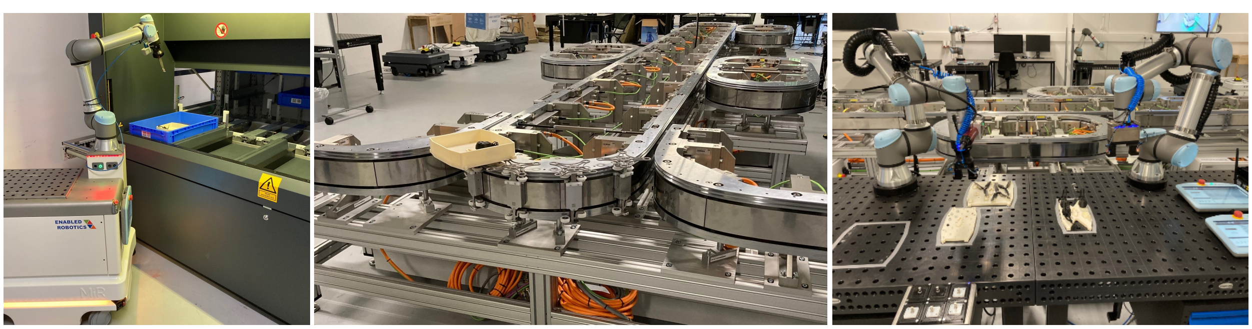

Figure 3: Assets in the I4.0 lab. Left) the AGV picks up drone parts from the opening at the warehouse, middle) the magnetic track that transports the drone parts using the shuttles, and right) the production cell assembles landing gear on drone motor.

ERP (Enterprise Resource Planning). The ERP system is a planning system that, among others, contains purchasing-, inventory-, and order management. The order management contains incoming customer orders.

AGV (Automated guided vehicle). The AGV, consists of an autonomous mobile robot with a robotic arm. The AGV is able to load and unload objects, as well as transport objects from one place to another (Enabled Robotics 2021).

Magnetic track. The magnetic track is a high-speed transport system with precision positioning. The magnetic track has exchangeable shuttles (transport devices) that are attached magnetically to the track. An object can be mounted on each of the shuttles (i.e., shuttles are equipped with boxes that can hold drone parts). The magnetic track is highly configurable to support different production flows (B&R Automation A/S 2021).

Production cell. The production cell contains two collaborative robotic arms capable of carrying out a specific task. In the presented case study, the production cell performs partial drone assembly. The production cell is highly configurable, depending on the task it performs (Universal Robots A/S 2021).

Warehouse. The warehouses are automated storage units that have efficient order picking rates of up to 250 boxes per hour (Effimat A/S 2021; Kardex Remstar 2021).

Based on the assets mentioned above, the production sequence is as follows:

1. Production order triggers the I4.0-IB from the ERP
2. Warehouse 1 prepares the parts for the order
3. AGV picks the parts from the Warehouse 1 and transports the parts to the Magnetic track
4. Magnetic track transports the parts to the Production cell 1
5. Production cell 1 executes an assembly step
6. Magnetic track transports the semi-assembled parts to the Production cell 2
7. Production cell 2 executes an assembly step
8. Warehouse 2 puts the product into storage

4.2 Industry 4.0 Information Backbone

In the following, we provide a description of the I4.0-IB, a high-level overview of the architecture, as well as an elaboration of the asset integration between the I4.0-IB and the machine software.

The Industry 4.0 Information Backbone. I4.0-IB is a middleware (Sommer et al. 2018) that enables information exchange between the assets (e.g. production floor machine and sensors) and corresponding enterprise applications. Therefore asset high-level interfaces must be available to asset lower-level operations (e.g. control signals to mechanical parts) from an I4.0 middleware perspective. Figure 4 illustrates a simplified sequence diagram and high-level operations that are invoked on assets seen from an I4.0-IB perspective. Recall in Section 1 that an asset is defined as anything that adds value to an I4.0 solution. The I4.0-IB will not only be limited to integrate assets such as machines and sensors on the production floor, but also be able to integrate other information sources that add value e.g. product development related data (Plattform Industrie 4.0 2021b).

I4.0-IB Architecture. Figure 5 illustrates the production floor layout with the assets and an evolving architecture for the I4.0-IB. Note how enterprise applications access the middleware regardless of the traditional hierarchical structure. The architecture supports a variety of integration technologies that are needed to support heterogeneous assets. Heterogeneous assets vary on different parameters, such as integration technologies and communication patterns, e.g., push-pull information and information frequency. The I4.0-IB architecture is based on a distributed event-driven approach, where services use a publish-subscribe communication pattern through message bus technology (Sommer et al. 2018). The distributed event-driven approach supports both different asset integration technologies, as well as different communication patterns. In addition, the event-driven approach supports the collection of data that meet the requirements for data-driven reliability modeling and simulation, as defined in Section 3.

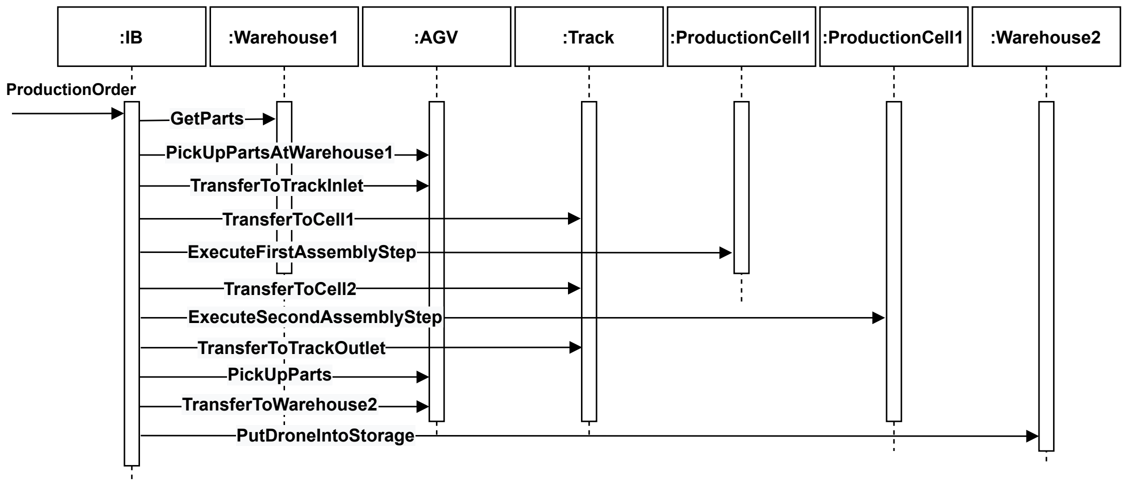

Figure 4: The I4.0-IB invokes high-level operations (e.g., assemble landing gear on a motor) on assets, after which the assets are responsible for invoking lower-level operations internally (e.g., control signals to mechanical parts).

Asset Integration. The I4.0-IB supports integration to SOAP (Leitão et al. 2015), REST (Cavalieri et al. 2019), OPC UA (Cavalieri et al. 2019), and MQTT (Sommer et al. 2018) technologies, as seen in Figure 5 and they will be continuously expanded with new integration technologies. The service structure contains, among others, an *Outer Interface* which contains implementation-specific asset integration code (e.g. SOAP). The implementation also contains primitive ping functionality to detect reachability of machines. The service has a structure for representing available operations callable from other services. The structure is a guideline for integrating new assets in the middleware and, at the same time, supports consistent generation of data for the simulation when when new assets are added. The *Inner Interface* includes components for how to receive and create messages from other services.

Listing 1 is an example of a service request. The code shows how a JavaScript Object Notation (JSON) file is passed to the *Warehouse1* service invoking the *prepareMaterialStart* operation on the asset, after which the warehouse fetches the box with *productId* 1008. The *id* is a globally unique identifier (GUID) that makes the message unique, the *type* indicates the message type, the *aasOriginId* indicates the sender of the message, the *aasTargetId* is the reciever of message, the *orderId* indicates the order id which the message is part of, the *operation* is the operation invoked on the asset, and the *parameters* are optionally parameters needed for a given operation.

Listing 1: Service request

```
Topic: "Storage"
{
  "@id":"d11972cc-247a-4107-b899-5e2e29ab4257",
  "@type":"operation",
  "aasOriginId":"i4.sdu.dk/Middleware/Orchestrator",
  "aasTargetId":"i4.sdu.dk/Storage/Warehouse1",
  "orderId":"123456789",
  "operation":"prepareMaterialStart",
  "parameters":
    {
      "productId":"1008"
    }
}
```

Listing 2: Service response

```
Topic: "Storage"
{
  "@id":"ee8c0d1f-c652-4cfb-acf8-677e35e47381",
  "@type":"response",
  "operationId":"d11972cc-247a-4107-b899-5e2e29ab4257",
  "aasOriginId":"i4.sdu.dk/Storage/Warehouse1",
  "aasTargetId":"i4.sdu.dk/Middleware/Orchestrator",
  "response":
    {
      "effimatReferenceId":"1250125",
      "carrierSlot":2,
      "success":true
    }
}
```

Listing 2 is an example of service response. The JSON file contains the data for successfully invoking the *GetParts* on the *Warehouse1* service. Service requests and responses are linked through *@id* and *operationId*. The data is structured as a linked list of events that mark beginnings and completions of activities.

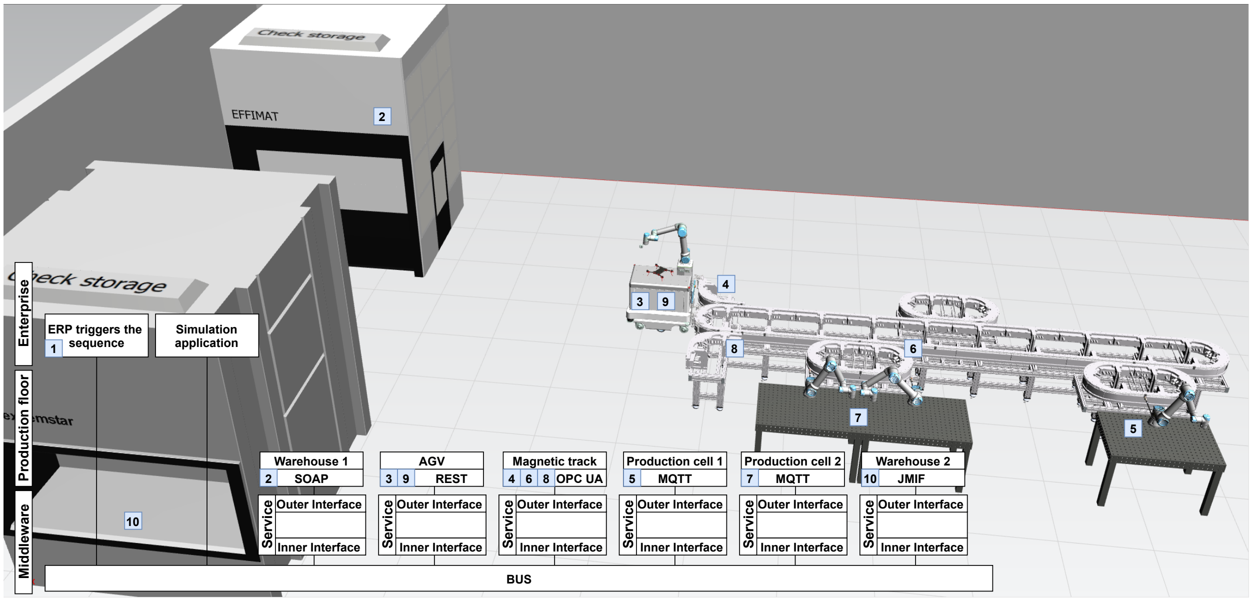

Figure 5: Production floor layout with assets, and a high-level overview of the I4.0-IB architecture. Numbers indicate production sequence for a simplified drone (Jepsen et al. 2021).

State-, event- and condition monitoring data. The state data of the assets is made available by the ping functionality of the services. Table 1 illustrates event data extracted from a service request and response. Supporting additional sensor data would require extending the JSON structure or adding new services that integrate with new assets. In general, the architectural design and data design of the I4.0-IB supports different application scenarios that aim to leverage or analyze the information flow for an improved production flow.

OrderId	Sender	Reciever	Asset	Event
1	Orchestrator	Warehouse1Service	Warehouse1	prepareMaterialStart
1	Orchestrator	Warehouse1Service	Warehouse1	prepareMaterialEnd
1	Orchestrator	AGVService	AGV	transportToTrackStart
1	Orchestrator	AGVService	AGV	transportToTrackEnd

Table 1: Event data extracted from linked data messages. Each activity is started and ended by an event.

The experiences and the development of I4.0-IB is an iterative process maturing I4.0-IB functionality, I4.0-IB software architecture as well applications using the I4.0-IB. The functionality for applications integrating directly to the I4.0-IB is under development. The next Section 4.3 will, therefore, elaborate on a simulation model emulating parts of the behavior (e.g., extraction of data) of the I4.0-IB described in this section, supporting the data requirements, presented in Section 3, for reliability assessment and simulation of manufacturing systems.

4.3 Case Study Simulation and Exemplary Data Output

We developed a simulation of the case study introduced in Section 4.1 using the discrete-event simulation framework *SimPy* (SimPy 2021). The simulation emulates the behavior of the production system in the lab and is able to generate synthetic state and event data. By doing so, the data requirements for data-driven reliability assessment became more apparent and graspable, which helps us in further development of the previously presented I4.0-IB. The implementation of the simulation can be found on GitHub (Friederich 2021).

The assets Warehouse 1 (wh1), Cell 1 (cell1), Cell 2 (cell2) and Warehouse 2 (wh2) are implemented as classes which include all the logic needed to simulate their respective behavior. The AGV (agv) and the magnetic track (track) are implemented as *SimPy* resources with variable capacity. To model and simulate the processes inherent in the aforementioned assets, we used Python generator functions as prescribed by *SimPy*. We also provide a configuration file allowing the user of the simulation program to adjust parameters such as the simulation runtime, whether the simulation should run in real time, asset operation times, and asset break and repair times.

Tables 2 and 3 show excerpts of exemplary event and state logs that were generated by the simulation program. For simplifying reasons, for this example, each operation was set to take one time unit. The event log shows the generated events, the corresponding assets, the case identifier and a timestamp (ts). Note that multiple events occur at the same time and new production runs (cases) are initiated while previous runs have not yet been finished. The state log has been generated based on the same simulation run as the event log. The state log provides the asset states for each time unit. The states of the AGV and the magnetic track are not included in the table for reasons of simplicity. Note that at time 5, Warehouse 1 fails and resumes operation at time 10. Cell 1 still assembles the first three production orders and then remains idle due to the lack of new raw material.

ts	case	asset	event
0	case_0	wh1	prepareMaterialStart
1	case_0	wh1	prepareMaterialEnd
1	case_0	agv	transportToTrackStart
2	case_0	agv	transportToTrackEnd
2	case_0	track	transportToCell1Start
2	case_1	wh1	prepareMaterialStart
3	case_0	track	transportToCell1End
3	case_0	cell1	assemblyStart
3	case_1	agv	transportToTrackStart
3	case_1	wh1	prepareMaterialEnd
4	case_2	wh1	prepareMaterialStart

Table 2: Exemplary event log.

ts	wh1	cell1	cell2	wh2
0	working	idle	idle	idle
1	idle	idle	idle	idle
2	working	idle	idle	idle
3	idle	working	idle	idle
4	working	idle	idle	idle
5	failed	working	working	idle
6	failed	idle	idle	working
7	failed	working	working	idle
8	failed	idle	idle	working
9	failed	idle	working	idle
10	working	idle	idle	working

Table 3: Exemplary state log.

The simulation program, which emulates the production line described in Section 4.1 is able to generate state and event data. Some parameters of the simulation can be adjusted in order to run experiments and to generate custom synthetic datasets. This data can be used to aid the development of data-driven methods for reliability assessment and simulation.

5 SUMMARY AND OUTLOOK

We substantiated and defined data requirements for a novel middleware for Smart Manufacturing Systems, in support of data-driven reliability modeling and simulation. We further categorized the data requirements by the model level of detail (i.e., state data, event data, condition monitoring data) and matched them to the popular reliability modeling approaches. Meeting this data requirements for a SMS would imply that new applications for data-driven reliability and simulation can be enabled. We, furthermore, provided a case study within the I4.0 lab at the University of Southern Denmark and presented an novel middleware architecture focusing on asset integration and how the architecture supports state, event, and conditional data requirements. Within the case study, we developed a simulation program of the behavior of the drone production line in the I4.0 lab to generate synthetic data matching the formulated data requirements.

In future, we intend to develop novel methods for data-driven reliability assessment and simulation (e.g., data-driven RBDs, FTA, PNs). Data generated by the simulation program supports this process. The simulation program is continuously developed further to generate more detailed data, e.g. synthetic condition monitoring data. The further development of the simulation model will continue in line with the progress of the I4.0-IB, ensuring that the simulation model reflects the production line in the I4.0 lab. In addition, we will develop and test novel methods for data-driven reliability assessment and simulation using real data provided by the I4.0-IB. The research of the I4.0-IB architecture, among others, will in future work focus on the quality attributes (i.e. reconfiguration, modifiability, availability) requirements for the infrastructure supporting different enterprise applications needs.

REFERENCES

Adamyan, A., and D. He. 2002, June. "Analysis of Sequential Failures for Assessment of Reliability and Safety of Manufacturing Systems". *Reliability Engineering & System Safety* 76(3):227–236.

Ali, S. R. 2019. "Hardware Reliability Modeling". In *Next Generation and Advanced Network Reliability Analysis: Using Markov Models and Software Reliability Engineering*, edited by S. R. Ali, Signals and Communication Technology, 29–57. Cham: Springer International Publishing.

Banks, J., J. C. II, B. Nelson, and D. Nicol. 2009, June. *Discrete-Event System Simulation*. 5. Edition ed. Upper Saddle River: Pearson.

B&R Automation A/S 2021. "B&R Industrial Automation GmbH Homepage". https://www.br-automation.com/en/#. Accessed: 2021-04-11.

Cavalieri, S., M. G. Salafia, and M. S. Scroppo. 2019. "Integrating OPC UA with Web Technologies to Enhance Interoperability". *Computer Standards & Interfaces* 61:45–64.

Chakraborty, S., and B. Ankiah. 1989, January. "Assessment of Manufacturing System Reliability: A Case Study". *Journal of the Operational Research Society* 40(1):55–63.

Dalenogare, L. S., G. B. Benitez, N. F. Ayala, and A. G. Frank. 2018, October. "The Expected Contribution of Industry 4.0 Technologies for Industrial Performance". *International Journal of Production Economics* 204:383–394.

Effimat A/S 2021. "Effimat". https://effimat.com/. Accessed: 2021-04-11.

Enabled Robotics 2021. "Enabled Robotics Homepage". https://www.enabled-robotics.com/. Accessed: 2020-12-09.

Farahani, Saeed 2021. "SDU Industry 4.0 Lab". https://www.sdu.dk/en/forskning/i40lab. Accessed: 2021-04-10.

Friederich, J. 2021, April. "SDU Industry 4.0 Lab Simulation". https://github.com/jo-chr/i40lab-simulation. Accessed: 2021-04-10.

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

Han, X., Z. Wang, Y. He, Y. Zhao, Z. Chen, and D. Zhou. 2019, January. "A Mission Reliability-Driven Manufacturing System Health State Evaluation Method Based on Fusion of Operational Data". *Sensors* 19(3):442. Number: 3 Publisher: Multidisciplinary Digital Publishing Institute.

Jepsen, S. C., T. I. Mørk, J. Hviid, and T. Worm. 2020, December. "A Pilot Study of Industry 4.0 Asset Interoperability Challenges in an Industry 4.0 Laboratory". In *2020 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, 571–575.

Jepsen, S. C., T. Worm, T. I. Mørk, and J. Hviid. 2021. "Industry 4.0 Middleware Software Architecture Interoperability Analysis". In *Proceedings of the IEEE/ACM 43nd International Conference on Software Engineering Workshops*.

Kampa, A., G. Gołda, and I. Paprocka. 2017, February. "Discrete Event Simulation Method as a Tool for Improvement of Manufacturing Systems". *Computers* 6(1):10.

Kardex Remstar 2021. "Kardex Remstar Homepage". https://www.kardex-remstar.com/en.html. Accessed: 2021-04-11.

Khorshidi, H. A., I. Gunawan, and M. Y. Ibrahim. 2016, June. "Data-Driven System Reliability and Failure Behavior Modeling Using FMECA". *IEEE Transactions on Industrial Informatics* 12(3):1253–1260. Conference Name: IEEE Transactions on Industrial Informatics.

Kuo, W., and T. Kim. 1999, August. "An Overview of Manufacturing Yield and Reliability Modeling for Semiconductor Products". *Proceedings of the IEEE* 87(8):1329–1344. Conference Name: Proceedings of the IEEE.

Kusiak, A. 2018, January. "Smart Manufacturing". *International Journal of Production Research* 56(1-2):508–517.

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., N. Mohamed, and H. R. Shaker. 2017, April. "Reliability Modeling of Cyber-Physical Systems: A Holistic Overview and Challenges". In *2017 Workshop on Modeling and Simulation of Cyber-Physical Energy Systems*, 1–6.

Lazarova-Molnar, S., P. Niloofar, and G. K. Barta. 2020. "Automating Reliability Analysis: Data-driven Learning and Analysis of Multi- state Fault Trees". In *European Safety and Reliability Conference Conference (ESREL)*, Volume 30, 8.

Leitão, P., J. Barbosa, M.-E. C. Papadopoulou, and I. S. Venieris. 2015. "Standardization in Cyber-Physical Systems: The ARUM Case". In *2015 IEEE International Conference on Industrial Technology (ICIT)*, 2988–2993. IEEE.

Liu, T. S., and S. B. Chiou. 1997, August. "The Application of Petri Nets to Failure Analysis". *Reliability Engineering & System Safety* 57(2):129–142.

Liu, Y., S. Zhang, and X. Ye. 2018, April. "A Value-Based Maintenance Optimization Method for Failure Prevention Based on Reliability Modeling of a Hybrid Assembly System". *The International Journal of Advanced Manufacturing Technology* 95(9):3309–3318.

Lugaresi, G., and A. Matta. 2020. "Generation and Tuning of Discrete Event Simulation Models for Manufacturing Applications". 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, 2707–2718. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Miriyala, K., and N. Viswanadham. 1989, December. "Reliability Analysis of Flexible Manufacturing Systems". *International Journal of Flexible Manufacturing Systems* 2(2):145–162.

Plattform Industrie 4.0 2021a. "Plattform Industrie 4.0 - What is Industrie 4.0?". https://www.plattform-i40.de/PI40/Navigation/EN/Industrie40/WhatIsIndustrie40/what-is-industrie40.htm. Accessed: 2020-11-24.

Plattform Industrie 4.0 2021b. "Reference Architectural Model Industrie 4.0 (RAMI4.0) - An Introduction". https://www.plattform-i40.de/PI40/Redaktion/EN/Downloads/Publikation/rami40-an-introduction.html. Accessed: 2021-04-07.

Ruijters, E., and M. Stoelinga. 2015, February. "Fault Tree Analysis: A Survey of the State-of-the-Art in Modeling, Analysis and Tools". *Computer Science Review* 15-16:29–62.

SimPy 2021. "SimPy - Process-Based Discrete-Event Simulation Framework Based on Standard Python". https://simpy.readthedocs.io/en/latest/. Accessed: 2021-04-11.

Sommer, P., F. Schellroth, M. Fischer, and J. Schlechtendahl. 2018, dec. "Message-oriented Middleware for Industrial Production Systems". In *IEEE International Conference on Automation Science and Engineering*, Volume 2018-Augus, 1217–1223: IEEE Computer Society.

Universal Robots A/S 2021. "UNIVERSAL ROBOTS Homepage". https://www.universal-robots.com/da/. Accessed: 2021-04-11.

Čepin, M. 2011. "Reliability Block Diagram". In *Assessment of Power System Reliability: Methods and Applications*, edited by M. Čepin, 119–123. London: Springer.

AUTHOR BIOGRAPHIES

JONAS FRIEDERICH is a Ph.D. (2020-2023) student with the SDU Software Engineering Section at the Faculty of Engineering, University of Southern Denmark. In 2020 he obtained his Diplom in Information Systems from Dresden University of Technology, 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 modelling and simulation, process mining, machine learning, computer vision and data science in general. His email address is jofr@mmmi.sdu.dk.

SUNE CHUNG JEPSEN is a Ph.D. (2019-2022) student with the SDU Software Engineering Section at the Faculty of Engineering, University of Southern Denmark. His doctoral project aims to investigate software architectures for production systems in relation to Industry 4.0. The research includes architecture evaluation, functional and quality attribute requirements for infrastructure platforms that support smart manufacturing systems, as well as implementing and evaluating actual middleware for the systems. Software architecture and quality attribute requirements are the main focus in order to facilitate the integration of smart production in the I4.0 Laboratory and companies. His email address is sune@mmmi.sdu.dk.

SANJA LAZAROVA-MOLNAR is a Professor with the SDU Software Engineering Section at the Faculty of Engineering, University of Southern Denmark, researching in the areas of Modelling, Simulation and Data Analytics, in various decision support contexts. 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. She was also one of the Proceedings Editors for the Winter Simulation Conference 2019 and 2020. Her email address is slmo@mmmi.sdu.dk.

TORBEN WORM is an Associate Professor and Head of Section with the SDU Software Engineering Section at the Faculty of Engineering, University of Southern Denmark, researching in the areas of software architecture, software processes and blockchain technologies. He is former software company CEO and board member of Danish and Nordic boards. His email address is tow@mmmi.sdu.dk.