

Case Report

Joscha Grüger*, Lukas Malburg and Ralph Bergmann

IoT-enriched event log generation and quality analytics: a case study

<https://doi.org/10.1515/itit-2022-0077>

Received December 23, 2022; accepted May 16, 2023;
published online June 1, 2023

Abstract: Modern technologies such as the Internet of Things (IoT) are becoming increasingly important in various fields, including business process management (BPM) research. An important area of research in BPM is process mining, which can be used to analyze event logs e.g., to check the conformance of running processes. However, the data ingested in IoT environments often contain data quality issues (DQIs) due to system complexity and sensor heterogeneity, among other factors. To date, however, there has been little work on IoT event logs, DQIs occurring in them, and how to handle them. In this case study, we generate an IoT event log, perform a structured data quality analysis, and describe how we addressed the problems we encountered in pre-processing.

Keywords: data quality in event logs; datastream XES extension; iot; IoT-enriched event log; physical smart factory; process mining.

ACM CCS: Information systems → Information systems applications → Decision support systems → Data analytics.

1 Introduction

Combining *Business Process Management (BPM)* methods with the *Internet of Things (IoT)* promises numerous

Joscha Grüger and Lukas Malburg contributed equally to the work.

***Corresponding author: Joscha Grüger**, Artificial Intelligence and Intelligent Information Systems, University of Trier, 54296 Trier, Germany; and German Research Center for Artificial Intelligence (DFKI), Branch University of Trier, 54296 Trier, Germany, E-mail: grueger@uni-trier.de
Lukas Malburg and Ralph Bergmann, Artificial Intelligence and Intelligent Information Systems, University of Trier, 54296 Trier, Germany; and German Research Center for Artificial Intelligence (DFKI), Branch University of Trier, 54296 Trier, Germany, E-mail: malburgl@uni-trier.de (L. Malburg), bergmann@uni-trier.de (R. Bergmann)

benefits for both sides [1]. The environment captured and driven by IoT devices can benefit from process modeling methods to control data capture and drive resource functionalities [2, 3]. *Process Mining (PM)* can be used in smart environments such as manufacturing to control conformance of production or to adjust and optimize process run times. BPM can benefit from systematic data collection and the variety of IoT data, e.g., actuator-related data, context data, and other forms of sensor data. However, the high volume, variety, and the temporal relationship between data points in IoT sensor streams poses challenges that other traditional domains occasionally do not. Thus, BPM methods and tools should be modified to achieve appropriate analysis results with this complex IoT data.

Due to the volume of data, the speed at which the data volumes are generated and the wide range of data types, data quality problems, which are typical for big data applications, also occur in IoT environments. These problems arise mainly due to the large number of sensors that are often integrated and their heterogeneity in terms of type, format, configuration, and susceptibility to errors [4]. During transformation of IoT datasets into event logs, these *Data Quality Issues (DQIs)* must be analyzed and addressed. To date, there is some research on data quality in procedural data [5–7] but little attention is paid to the IoT domain.

Therefore, we investigate the quality of data of an IoT-enriched event log in a structured way in this paper. In addition, we describe the handling of DQIs found in the analysis during data preprocessing. The data used in this case was generated in a physical smart factory. A small-scale smart factory is used, since only a few logs are currently available for process-oriented research in this domain. However, the data generated from the smart factory is very similar to that of large real-world production environments and allows us to provide the log in pre- and post-processing status, i.e., with and without DQIs, and make it available for other research projects [8].

The paper is structured as follows: Section 2 discusses related work regarding the smart factory used, using and representing knowledge of IoT environments, XES for

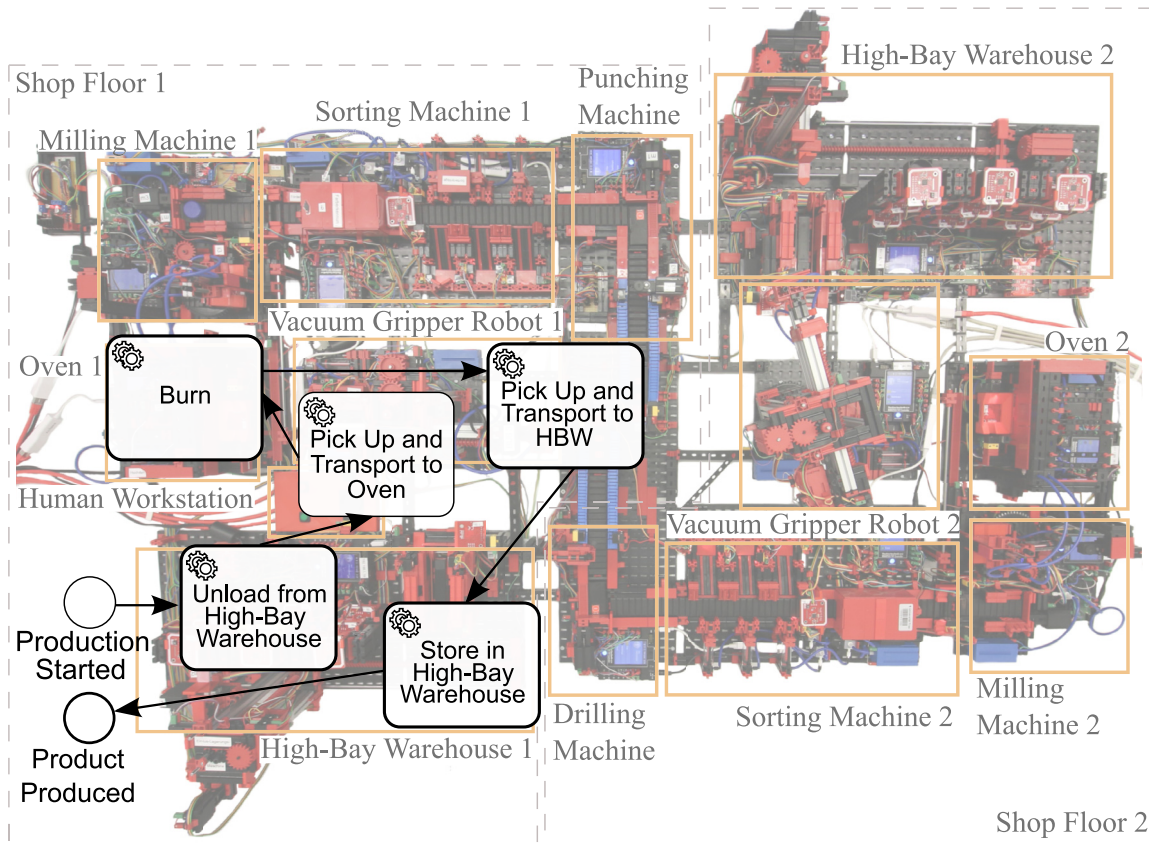


Figure 1: Process-based control of the Fischertechnik smart factory. (Based on: [3]).

representing event logs, and data preprocessing methods in process mining. Section 3 describes the procedure for generating event logs based on the data generated by the smart factory. Section 4 introduces a descriptive evaluation in which we (1) assess the quality of the generated log by using several metrics and (2) present important lessons learned from the applied procedure. Finally, Section 5 concludes the work and an outlook is given.

2 Foundations and related work

In the following, we describe the foundations for this paper: Section 2.1 describes used small-scale Fischertechnik physical smart factory. The representation of knowledge in smart environments and the XES DataStream Extension [9] is presented in Section 2.2 and Section 2.3 respectively. In Section 2.4, we present relevant related work on data preprocessing and quality issues in the context of *Process Mining (PM)*.

2.1 Fischertechnik physical smart factory

In our work [2, 3, 10, 11], we use a physical Fischertechnik¹ smart factory to conduct BPM research. The factory consists of several machine resources that can be controlled in a process-oriented fashion by a *Workflow Management System (WfMS)*. For this purpose, a semantic service-oriented architecture [2, 3, 11] is used that abstracts from low-level control commands and encapsulates the capabilities of the machine resources in web services. In addition, a web server is used that handles incoming requests from high-level systems and initiates the corresponding calls of the capabilities on the resources. Figure 1 depicts the used smart factory and its components. The smart factory is equipped with several sensors, including 31 light barriers, 32 switches, 2 acceleration sensors, 2 gyroscopes and 6 capacitive sen-

¹ Fischertechnik is a company producing modules for small scale simulating factories. Information can be found at <https://www.fischertechnik.de/en/simulating/industry-4-0> and information regarding our custom model at <https://iot.uni-trier.de>.

sors for control of the actuators consisting of 32 motors, 8 compressors, and 16 valves. The machine resources are enhanced with sensors mounted on moving parts, motors, and compressors for condition and pressure monitoring. Moreover, NFC readers/writers are integrated into the stations, resulting in 28 communication points. During process execution, the smart factory generates a lot of data through the multitude of sensors and actuators. This data is streamed by each resource with approx. 10 data points per second into Apache Kafka² and stored for a short period of time. All in all, we store the data in 38 Kafka topics, of which each resource has its own topic and the individual types of sensors are grouped into corresponding topics. Based on these topics, an event log can be generated consisting of the process related data and the corresponding IoT sensor data.

2.2 Structure and representing knowledge in smart environments

Machine-readable knowledge representations are becoming increasingly important in research and practice. They enable the development of advanced methods for data analysis or, in general, of research artifacts. To structure and represent the knowledge of smart environments, several approaches in the form of ontologies have been proposed. The focus in these ontologies has shifted from trying to be as complete as possible (e.g., the *Semantic Sensor Network (SSN)*³ ontology [12] or the *CREMA Data Model, Core module (CDM-Core)* [13]) to be simpler and more practical in real-world applications (e.g., the IoT-Lite [14] ontology). The *Sensors, Observations, Samples, and Actuators (SOSA)* ontology [15] is a compact version of the SSN ontology and describes the relationships between sensors and actuators as well as their measured observations in IoT data. By using this ontology, it is possible to represent, for example, a relationship between a machine resource and the sensors that monitor its condition. A further ontology especially tailored for streaming data is the IoT-Stream ontology [16]. It is a more specific ontology, inspired by SOSA, that focuses on the treatment of streaming data. Both the SOSA and the IoT-Stream ontologies are event-centric, in the sense that they

focus on data generation and treatment, and less attention is paid to the devices and platforms on which IoT relies, such as in the CDM-Core ontology. The *Manufacturing's Semantics Ontology (MASON)* [17] is used for representing manufacturing domains. The ontology is based on three head concepts: *entities*, *operations*, and *resources*. Based on the hierarchy in this ontology, we developed a domain ontology FTOnto [18] for structuring and representing the knowledge for Fischertechnik smart factories. The physical parts of the factory are represented as individuals in the ontology. Object properties are used to model relations between the individuals. To model the relationships between actuators and their corresponding sensors, we use the well-established SOSA ontology [15].

2.3 DataStream extension

eXtensible Event Stream (XES) [19] is the *de facto* standard for representing process data in event logs. However, this standard has not been developed for representing IoT sensor data besides the process-related event data. For this purpose, the XES DataStream extension [9] enables to connect IoT data to process events. In addition, it is possible to link the data to a set of semantic annotations to describe the scenario and environment during data collection. Therefore, the extension describes a metamodel for adding sensor data in the context of

- single events: time-series of sensor data from at least one sensor connected to an activity.
- group of activities: time-series of sensor data from at least one sensor connected to a set of activities.
- trace: time-series of sensor data from at least one sensor connected to a trace.

The core of the extension is `stream:point`. It contains all attributes that allow to represent individual sensor values as XES artifacts. This element, represented as a list, includes a unique identifier, a source of the sensor data, a timestamp, the recorded value and optional descriptive metadata. To provide additional semantic information, `stream:point` can be annotated with a variety of information describing the many semantic aspects of the data collection. In particular, references to ontological knowledge from the SOSA/SSN ontology and IoT systems-specific ontologies can be used for this purpose (see Section 2.2). In Listing 1.1, an exemplary stream point in the XES representation is depicted.

² <https://kafka.apache.org/>.

³ <https://www.w3.org/TR/vocab-ssn/>.

Listing 1.1. *Sample XES (XML serialization) stream:datastream Nesting.*

```

<trace>
  <string key="concept:name"
    value="Process 1"/>
  <list key="stream:datastream">
    <list key="stream:point">
      stream:system="Oven 1"
      stream:system_type="sosa:Sensor"
      stream:interaction_type="
        sosa:Observation">
        <date key="stream:timestamp"
          value="2021-11-04T15:22:19"/>
        <string key="stream:value"
          value="62.5"/>
        [...]
      </list>
    </list>
  </list>
</trace>

```

In the `stream:point`, it is specified by using the `system` attribute that the data has been gathered by a sensor from the Oven. In this case, the sensor is not concretely specified, as the general `sosa:Sensor` class is used as reference in the `system_type` attribute. However, it is also possible to use a concrete sensor, which is attached to the Oven, e.g., a temperature sensor modeled as individual in the FTOnto.⁴ By using a concrete sensor, it is possible to gather more information regarding data acquisition, e.g., min and max data values. The `interaction_type` is based on SOSA and can reference on a concrete `sosa:Observation` or `sosa:Actuation`. For this purpose, a key-value pair consisting of a date and a string with the measured value is used.

2.4 Related work

The event log is the most important input for process mining techniques. Therefore, the quality of the log has a great impact on the results and preprocessing has an important role in process mining pipelines [20]. Preprocessing takes up a large part of the effort in process mining projects and remains a challenge. Possibilities to improve this include the standardization of data formats as well as the development and improvement of data transformation pipelines [21]. A lot of effort has been put into developing uniform standards for the presentation of process logs, as well as for the preprocessing of data and the classification and handling of data quality issues. Many proposals have been made in the past for storing process logs. MXML, for example, as a simple XML format for audits and trails in process goods

information systems [22]. Also, XML-based is the current standard model for event logs, XES, which is widely used in both industry and academia [19, 23]. In addition, since the requirements for event logs vary depending on the application and domain, XES can be extended via so-called extensions. Recently, the introduction of new technologies and the increasing maturity of process mining have increased the development of alternative event log models. To this end, several proposals have been made to relax the assumptions of XES and thus bring more flexibility to the storage of event data [24, 25]. One development is Object Centric Event Log (OCEL), which is designed to store event data from relational databases and is now considered the main competitor to XES. In contrast to XES and its central case, OCEL uses the concept of an object as a generalization of the case concept. For this purpose, an event is associated with several objects instead of a case. A second noticeable difference from XES is the explicit inclusion of the concept of activity in OCEL, which is absent in XES.

More specifically in the direction of IoT, in addition to the `DataStream` approach used, there is also an approach in the form of a data model for representing IoT data in the process mining context. However, this is only conceptual so far [26]. In Ref. [27], a method to incorporate IoT data into XES logs by concentrating on a restricted but established group of attributes from the BPM field, namely physical object, location, time, identity, and environment, and by explicitly gathering the data through the addition of collection tasks to existing process models, which results in various issues, like not considering IoT standardization and the focus only on IoT data on trace level.

In preprocessing, there is a lot of preliminary work from the field of data mining [28]. In addition to basic preprocessing techniques, there are approaches formalizing the preprocessing process [29]. Proposes a five-step procedure consisting of Data Cleaning, Data Integration, Data Transformation, Data Reduction, and Data Discretization. The framework contrasts the Knowledge Discovery in Databases (KDD) approach by Ref. [30] in the interpretation of the transformation step. This considers the transformation as a downstream step from preprocessing.

Many preprocessing steps address the correction or removal of errors or quality issues in data. The process mining manifesto [5] highlights the need for high-quality event logs. For this purpose, the manifesto defines 5 levels of maturity. The one-star (*) level describes logs in which events were recorded manually, whereas a five-star log (*****) is a log that is considered complete and accurate and contains only automatically recorded events. Most real-life logs are two, three, or four-star logs [31].

⁴ SSCMRRN03PD2A3_Pi_1_Temperature_Sensor_3 is a temperature sensor that is attached to the Oven on the first shop floor.

Numerous frameworks and approaches exist for identifying, classifying, and analyzing event log quality and quality issues. In Ref. [6], the authors describe 27 event log quality issues, divided into the problem categories *missing*, *incorrect*, *imprecise*, and *irrelevant* data. Missing data describes the absence of, e.g., an event in the log. Incorrect data refers to the fact that data is wrong, e.g., wrong attribute values. Irrelevant data describes data that is irrelevant to the analysis in its current form, but another relevant entity may need to be derived/obtained from the logged entities, e.g., through filtering/aggregation.

Another approach describing event log quality was proposed by Ref. [7]. The literature-based taxonomy builds on the taxonomy by Refs. [6, 32]. Based on the literature review, Verhulst defines 12 dimensions for data quality in event logs. These include dimensions such as the completeness or correctness of the data in the event log.

Another work by Ref. [31] propose the event log imperfection patterns. In total, Suriadi et al. derive 11 imperfection patterns from real-life case study experiences in a variety of domains. These event log specific patterns include unanchored event, i.e., when timestamps are recorded in a format different from the one expected, or Elusive case, i.e., when events are not linked to a case.

In contrast to related work, this work focuses on adapting existing data quality analysis frameworks for event logs with complex IoT data, identifying and classifying the data quality issues, as well as correcting them. This is implemented based on real data.

3 Procedure for generating IoT-enriched event logs

Generating event logs in the IoT context is a complex process, as the underlying IoT architecture in each case must be taken into account and data is collected from a variety of sources [33]. The event log generation follows several steps: First, all possible data sources were identified. After a short analysis of the data sources, the relevant sources have been selected (see Section 3.1). In an extended data quality analysis (see Section 3.2), the quality of the data and data quality issues have been investigated. Based on the results of the analysis, strategies in the form of transformations for addressing the DQIs have been developed and applied (see Section 3.3). The process of data quality analysis, development of strategies to address the issues, and implementation of these had to be repeated three times because new issues, not identified in the data quality analysis, have been identified during development. These were identified late because

the DQIs could only be identified in the combined view of multiple data sources and by using the semantic representation of the smart factory (see Section 2.2). Thus, the complete procedure of event log generation (see Section 3.4) is inspired by the KDD process [30] and is depicted in Figure 2. The following chapters describe the implementation and application.⁵

3.1 Data selection

The event log is based on the factory system described in Section 2.1. The production is controlled by a WfMS. The WfMS stores data for each execution of a workflow. This includes the start and end time of each activity that the WfMS triggers, the assignment to a case, start and end time as well as expected and actual execution time, the resource on which the activity was executed and a status code indicating whether the execution was successful. In addition to the WfMS, there is a web server that serves as middleware between the WfMS and the factory system. The web server manages, besides others, the queue of the individual actuators and logs, their status, error messages, and all responses and requests triggering the execution of certain activities, including their parameters. In total, the factory system comprises 15 actuators, having a separated event log. Each actuator is directly or indirectly associated with sensors. The sensor data is stored in 30 log files. In addition to the log files, there is also an ontology that semantically describes the factory system, the sensors and the actuators.

After a first analysis, we included all logs except the web server queue log. However, we still needed this later in preprocessing for validation and reconstruction of events.

3.2 Data analysis

The investigation of data quality has been conducted along the event log quality issues, described in [6]. We went through the event log quality issues from I01 to I27 for each of the files in a structured way. It was important for us to detect as many errors as possible before compiling the event log. This allowed us to work with high-quality data during log generation. Below we list the errors and problems we identified:

- **Missing Events (I2):** in WfMSs there are events that were not logged or only partially logged (the request is missing, but the response is present). When comparing

⁵ Log containing DQIs and final log available at <https://zenodo.org/record/7795547>.

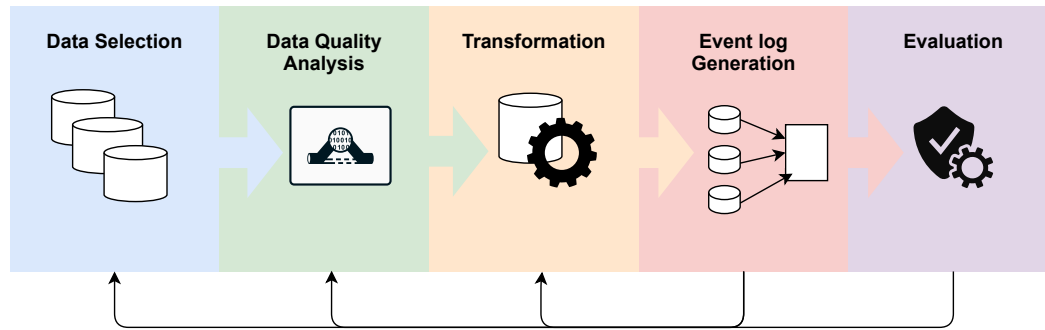


Figure 2: Process of event log generation. (Based on: [30]).

with the web server logs and the sensor logs, many of the events were found to be fully and correctly executed even if they did not appear in the log. In addition, there are events that only appear in the sensor logs.

- **Missing Relationships (I3):** No unique mapping to the current case exists in the logs for the sensor data. Likewise, no mapping exists uniquely associating the sensors with entities in the ontology, which should be included in the log for better semantic description.
- **Missing Event Attributes (I9):** For some events, no status code was logged. The status codes indicate whether the event was successful or whether an error occurred, and if so, which one.
- **Incorrect Events (I11):** Single events appear twice in the WfMS log, although both the web server log and the sensor logs clearly show that they were executed only once.
- **Incorrect Timestamps (I16):** The operating times of the resources differ from those in the web server log. Also, there is a constant shift of +15 s in the logged data in the sensor logs.
- **Incorrect Event Attributes (I18):** the recorded sensor data is not harmonic. Individual sensors measure metrically, while others measure in inches. The same applies to units that relate a value to time, e.g., sensors measuring the same thing, but in one case in terms of seconds and in another case in terms of milliseconds.
- **Missing Events (I2):** in the WfMS and the web server log, each event is represented by a request and a response. The analysis indicated that some events have a response but no request in the WfMS log, and some not appear at all. Since they are recorded in the web server log, they could be reconstructed. In some cases where the high-bay warehouse (HBW) was the resource used, there were events that are not in the WfMS log and also not in the web server log. However, knowing the process model, the store floor, and the fact that the events show up again in the log after this event, it was clear that they must have occurred. We also found evidence of this in the sensor data. So we were able to reconstruct the event data using the sensor data (to identify start and end time).
- **Missing Relationships (I3):** The missing relation between the Sensorlogs and the entities in the underlying ontology was established manually. For this purpose, a mapping was constructed that uniquely mapped the name of the sensors to entities in the ontology. This mapping and the semantics behind it could then be used to create a mapping of sensors to specific resources. Using the knowledge of the resource used in an event, the associated sensors and the start and end time of the event, the sensor data points could then be uniquely assigned to individual events.
- **Missing Event Attributes (I9):** The missing status codes in the WfMS log could be generated in case of a positive status code from the knowledge about the further course of the respective trace (status code 200). In the case of a negative status code (e.g., 401, 417, 418), this could be reconstructed from the error message logged in the web server log (e.g., The High Bay Warehouse does not contain any workpiece with the given business_key WF_103is a status code 401).

3.3 Transformation

In the transformation step, the quality issues are addressed and problems are corrected. Most of the problems could be corrected by taking the data from another log, since a lot of information was recorded twice (e.g., in web server log and WfMS log). In addition, the ontology could be used to establish unique relations between sensors and actuators.

- **Incorrect Events (I11):** In case of double recorded events in the WfMS log, they were removed from the log.
- **Incorrect Timestamps (I16):** The difference in the recorded time for the events in the log of the resources and in the web server log or in the WfMS log is due to a differently configured time-server and could thus be corrected. The shift of constant +15 s in the log of the sensors is caused by a faulty central time-server of the control units and could be corrected.
- **Incorrect Event Attributes (I18):** The differences in the units of measurement were corrected via a script. For this purpose, it was specified how a sensor should record the data and, in case of a deviation from the target, a conversion to the desired unit of measurement was performed.

3.4 Event log generation

The underlying control flow and case data is available in two data sources: the first one contains the information about actuators performing the process activities in the smart factory. By using this data source, it is possible to derive the event lifecycle. The second data source consists of 25 GB sensor data that is related to the actuation and all time series data of sensors in the smart factory. The two data sources are combined, and, finally, they have been transferred to an XES log.

In the first step, we start by generating an event log containing the process activities from the WfMS log executed by the actuators. From the information when an event was placed in a queue of a resource and the start and end times of the actual execution and the status codes, the lifecycle was mapped for each event using the XES lifecycle extension. The resulting log contains 9471 events in 301 cases. This log is referred to as the MainLog in the following. As we take this data from the first data source, it currently is not enriched with IoT sensor data.

In the resource event logs, i.e., the second data source, the activities performed are described again in finer granularity by time series sensor data. In the case of the oven, for example, instead of the temper activity in the MainLog, each step of opening the door, retracting the workpiece, burning itself, etc. is described here (see [3] for a further example). From the resource event logs, a sublog is created for each event in the MainLog with `lifecycle:state` value `InProgress`. This sublog and the subtrace described in it are referenced from the MainLog using a UUID in the event attribute `SubProcessID`. Each sublog contains exactly one trace, which is also uniquely identified by the UUID. The

subprocesses in total contain 13,424 subevents. Figure 3 illustrates the described resolving of relationships between main activity and subprocesses for the temper activity of the oven.

The sensor data was included at the sublog level. The Data Stream XES [9] extension was used to represent the complete sensor data, including semantic enrichment via ontologies. The DataStream XES extension enriches the XES standard with the ability to integrate complex sensor data into the event log. For this purpose, the extension introduces the `datastream` schema. This schema enables the semantic description of sensor data using the SOSA and SNN ontology. The schema enables the enrichment of event logs with sensor data at different levels of the log. In this log, all sensor data is associated with events. For each sensor value, the data type, interaction type (observation or actuation), receiving system type, and relation to the underlying FTonto⁶ [18] ontology and timestamp were recorded.

The creation of the event log was done in Python. The library PM4PY and lxml were used for this purpose. However, we reached the limits resource-wise when we wanted to add the sensor data to the log using the DOM-based library. Instead of using DOM based XML, we generated the sensor data using genshi MarkupTemplates and wrote the log line by line at file level.

4 Descriptive evaluation

The descriptive evaluation is carried out along quality dimensions to determine the degree to which the event log quality issues could be addressed by the selected interventions. Thus, the goal of the evaluation is to measure the quality of our proposed event log generation procedure and to provide a high-quality event log for process-oriented research. During the evaluation, we focus on the final, error-free event log as it builds the basis for developing and evaluating BPM research methods and tools. Finally, a discussion, limitations, and lessons learned from the case study are derived and described at the end of this section.

4.1 Quality metrics

The log is evaluated using the event data quality dimensions described in Ref. [7]. These describe the twelve dimensions of completeness, uniqueness, timeliness, validity, accuracy/correctness, consistency, believability/credibility,

⁶ <https://gitlab.rlp.net/iot-lab-uni-trier/ftonto>.

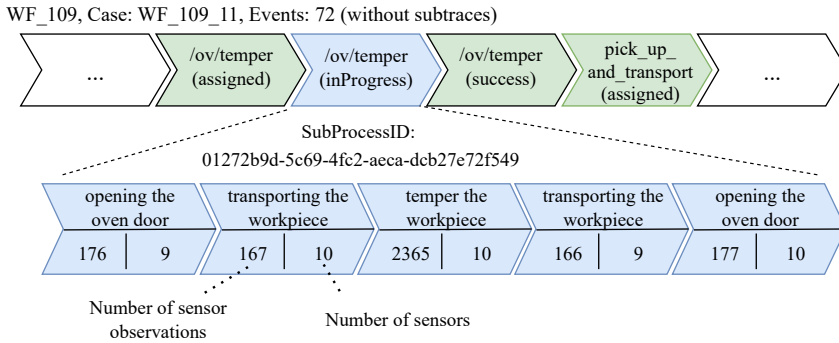


Figure 3: Excerpt of a process instance of workflow 109 including resolving the relationship to the subprocess of the event “/ov/temper” in the lifecycle state “(InProgress)” via the SubProcessID.

relevance, security/confidentiality, complexity, coherence, representation/format. The framework in Ref. [7] is based on the framework developed by Ref. [6] describing event log quality issues, which was used in this work to identify and address the issues. In addition [7], also describes, for the most dimensions, measurement methods for quantifying them. So, the main distinction between the work of Refs. [6, 7] is that Ref. [6] focused more on the process of event log generation and the problems and issues that might occur. In contrast to this, the metrics by Ref. [7] describe concrete measurements that can be applied for assessing the resulting quality of the event log.

The evaluation (see Table 1) of **completeness** is based on the fact that all data necessary for an evaluation is available. In generalized terms, this describes that no data is missing at the granularity level of the log. This is the case for the log. For checking the transaction information, the XES extension *lifecycle* is used. The framework describes that the more lifecycle stages used, the more complete the log. The log uses the lifecycle transitions scheduled, start and complete as well as the states assigned, inProgress and success or failure. This is evaluated in the framework as very complete. Taking **uniqueness** into account, it can be said that the event log contains most attribute values multiple times, with each attribute being unique within an event. This is due to the size of the event log. The same is true

for the repetition of events, which is due to the fact that identical workflows are executed multiple times or activities are part of different executed workflows, but results in a low score for uniqueness. **Timeliness** is a measure of how current the data is, in the time frame in which it is expected to be. Event timestamps are checked to see if they fall within the time frame specified by the user. It can be said that only data between 2021-06-23 and 2022-02-27 are part of the event log, which corresponds to the defined and expected time frame. Data are **valid** if they comply with the syntax (format, type, range) of their definitions. This is given in large parts. The use of lists in XES is defined there, but is not supported by many tools. The individual logs (sublogs and mainlog) are valid with the addition of the extensions. The use of the subprocess IDs, however, is not part of the standard. The **accurateness** and **correctness** of the log is related to, that the values obtained are close to the (unknown) true values. The obtained values in the log are all in the estimated value ranges, and there is no more indication of incorrect values. From a **consistency** perspective, it was checked if there were outlier among the individual attributes. These only existed under the sensor data as part of the logged calibration processes (see Figure 4). The **Believability** was checked, among other things, with the outlier analysis (see Consistency). The **Relevancy** is high following the framework, since the evaluated attributes

Table 1: Overview of log quality dimensions and evaluation results.

Quality dimension	Evaluation
Completeness	Very complete
Uniqueness	Repetition of events lowers score
Timeliness	Data falls within the expected time frame
Validity	Large parts of the log are valid, subprocess IDs are not part of standard
Accuracy/correctness	All values fall within estimated value ranges with no indication of incorrect values
Consistency	Outliers exist in sensor data due to calibration processes
Believability/credibility	Outlier analysis supports believability
Relevance	Evaluated attributes appear in almost all events
Security/confidentiality	No personal or critical data included
Complexity	301 traces in mainlog and 3118 subtraces, with an average of 31 events per trace in mainlog

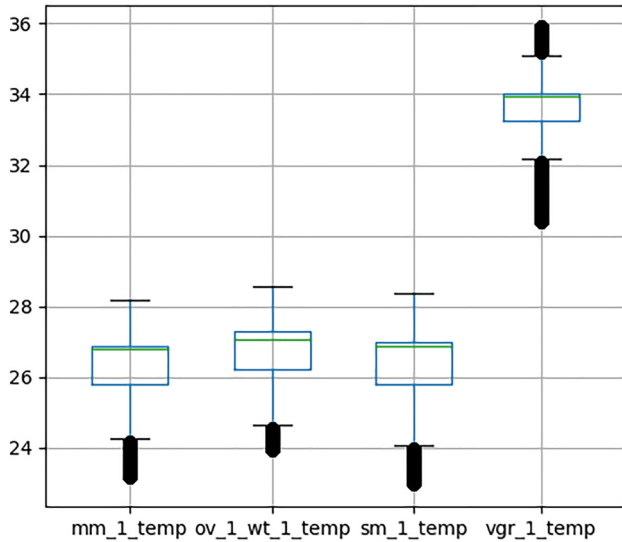


Figure 4: Box plot of the values of the temperature sensors on shop floor 1. Here it is clear that there are some outliers. This is due to the fact that calibration processes were also recorded.

appear in almost all events and there are no attributes in events that are included in less than 5 % of the attributes. The **Security** is not critical in this log, as no personal data or critical data is included. The **Complexity** is determined by the number of traces (301 in the mainlog, 3118 subtraces), the number of events (9471 in the mainlog, 13,424 in the sublogs), the average of events per trace (31 events in the mainlog, 8 events in the sublog), all traces in the mainlog start with an unload from the high-bay warehouse, and there are 231 trace variants in total (see Table 2).

4.2 Discussion, limitations, and lessons learned

During the process of generating the event log, we identified a few aspects that can be addressed to improve the overall process. In this paragraph, we briefly discuss the results of the case study and present limitations and lessons learned of the applied procedure. During applying the proposed procedure, we observed that the implementation effort was mainly affected by the issues detected during the

manual inspection. For inspecting the generated event log after recording, the workload was very high. In addition to this aspect, a lot of knowledge about the processes and the smart factory is needed to suitably address DQIs. Therefore, it would be beneficial to formalize this domain knowledge and experiential knowledge from domain experts and use it for automated detection of DQIs, for example. Based on this, issues can be resolved, resulting in much lower efforts for addressing issues afterward. In this context, utilizing semantic annotations of the smart factory or the processes can also help to detect DQIs more reliable. This knowledge can also be exploited to resolve the quality issues in the log. The observed DQIs result from different sources: on the one side, DQIs are caused by technical defects such as a faulty sensor or on the other side by intervention of humans during the production process leading to inconsistent data. For this reason, the proposed and structured way for identifying such DQIs is appropriate to detect both sources of quality issues. However, using the proposed procedure for other event logs can be a demanding task, as the required knowledge to detect and to resolve quality issues partly in an automated way might not be available. This results in manual efforts by domain experts to detect and to resolve the quality issues manually or to model the required knowledge beforehand. In addition, we currently only determine the quality metrics for the error-free event log in the proposed application scenario for smart factories. However, we assume that other IoT domains have similar characteristics as the smart manufacturing domain, but it might be that some other quality issues might occur, and the proposed approach is not completely appropriate for fixing them.

5 Conclusions

In this paper, we presented the analysis of IoT data for data quality issues and the generation of an IoT-enriched event log with resolved data quality issues. For this purpose, we used a dataset generated with a physical small-scale smart factory that is comparable to other IoT datasets. The study used a five-step structured approach, based on the KDD process, to generate a DQI-free IoT-enriched event log.

Table 2: Overview over the logs (after preprocessing).

Data sets	Events	Cases	Activities	Resources	Variants	Actuators	Sensors	Data points	Trace len (avg, min, max)		
Main log	9471	301	21	15	231	–	–	–	31	3	69
Sublogs	13,424	3118	109	15	269	52	131	136,208,108	8	1	14
Total	22,895	3489	21	15	500	52	131	136,208,108	10	1	69

Bold values are the total sum over main log and sublogs.

To do so, we took the data quality issues presented in Ref. [6] and analyzed the dataset based on them. After resolving the DQIs, the event data quality dimensions described in Ref. [7] were used to determine the data quality of the final log. In addition, we discuss lessons learned from applying the proposed event log generation procedure. The limitations of the study lie in the used smart factory. However, this is used as a significantly cheaper alternative to real production environments while retaining real-world characteristics [3, 11].

In the future, we want to compare the DQIs of this log with DQIs of other IoT-enriched logs. Based on this, we want to identify common DQIs in IoT-enriched logs and develop best practices to avoid these DQIs during logging and correct them during log generation. In addition, it would also be interesting to investigate how far the OCEL structure allows identifying DQIs and if the non-flat structure of OCEL supports this.

Author contributions: All the authors have accepted responsibility for the entire content of this submitted manuscript and approved submission.

Research funding: This work is funded by the Federal Ministry for Economic Affairs and Climate Action under grant No. 01MD22002C EASY.

Conflict of interest statement: The authors declare no conflicts of interest regarding this article.

References

- [1] C. Janiesch, A. Koschmider, M. Mecella, et al., “The internet of things meets business process management: a manifesto,” *IEEE Syst. Man Cybern. Mag.*, vol. 6, no. 4, pp. 34–44, 2020.
- [2] L. Malburg, P. Klein, and R. Bergmann, “Semantic web services for AI-research with physical factory simulation models in industry 4.0,” in *1st IN4PL*, SciTePress, 2020, pp. 32–43.
- [3] R. Seiger, L. Malburg, B. Weber, and R. Bergmann, “Integrating process management and event processing in smart factories: a systems architecture and use cases,” *J. Manuf. Syst.*, vol. 63, pp. 575–592, 2022.
- [4] A. Gaddam, T. Wilkin, M. Angelova, and J. Gaddam, “Detecting sensor faults, anomalies and outliers in the internet of things: a survey on the challenges and solutions,” *Electronics*, vol. 9, no. 3, p. 511, 2020.
- [5] W. M. P. van der Aalst, A. Adriansyah, and A. K. A. de Medeiros, et al., “Process mining manifesto,” in *BPM Workshops*, Springer, 2012, pp. 169–194.
- [6] J. C. J. C. Bose, R. S. Mans, and W. M. P. van der Aalst, “Wanna improve process mining results?” in *CIDM*, IEEE, 2013, pp. 127–134.
- [7] R. Verhulst, “Evaluating quality of event data within event logs: an extensible framework,” Master thesis, 2016.
- [8] L. Malburg, J. Grüger, and R. Bergmann, “Dataset: an Iot-enriched event log for process mining in smart factories,” 2022.
- [9] J. Mangler, J. Grüger, L. Malburg, et al., “DataStream XES extension: embedding IoT sensor data into extensible event stream logs,” *Future Internet*, vol. 15, no. 3, p. 109, 2023.
- [10] L. Malburg, M. Hoffmann, and R. Bergmann, “Applying MAPE-K control loops for adaptive workflow management in smart factories,” *J. Intell. Inf. Syst.*, pp. 1–29, 2023. <https://doi.org/10.1007/s10844-022-00766-w>.
- [11] L. Malburg, R. Seiger, R. Bergmann, and B. Weber, “Using physical factory simulation models for business process management research,” in *BPM Workshops, LNBP*, vol. 397, Springer, 2020, pp. 95–107.
- [12] M. Compton, P. Barnaghi, L. Bermudez, et al., “The SSN ontology of the W3C semantic sensor network incubator group,” *J. Web Semant.*, vol. 17, pp. 25–32, 2012.
- [13] L. Mazzola, P. Kapahnke, M. Vujic, and M. Klusch, “CDM-core: a manufacturing domain ontology in OWL2 for production and maintenance,” in *8th KEOD*, 2016, pp. 136–143.
- [14] M. Bermudez-Edo, T. Elsaleh, P. Barnaghi, and K. Taylor, “IoT-lite: a lightweight semantic model for the internet of things,” in *UIC-ATC-ScalCom-CBDCom-IoP-SmartWorld 2016*, IEEE, 2016, pp. 90–97.
- [15] K. Janowicz, A. Haller, S. J. Cox, D. Le Phuoc, and M. Lefrançois, “SOSA: a lightweight ontology for sensors, observations, samples, and actuators,” *J. Web Semant.*, vol. 56, pp. 1–10, 2019.
- [16] T. Elsaleh, M. Bermudez-Edo, S. Enshaeifar, S. T. Acton, R. Rezvani, and P. Barnaghi, “IoT-stream: a lightweight ontology for internet of things data streams,” in *Global IoT Summit Proc*, IEEE, 2019, pp. 1–6.
- [17] S. Lemaignan, A. Siadat, J. Y. Dantan, and A. Semenenko, “MASON: a proposal for an ontology of manufacturing domain,” in *Workshop on Distrib. Intell. Syst.: Collect. Intell. and its Appl.*, IEEE, 2006, pp. 195–200.
- [18] P. Klein, L. Malburg, and R. Bergmann, “FTOnto: a domain ontology for a Fischertechnik simulation production factory by reusing existing ontologies,” in *21st LWDA*, vol. 2454, CEUR-WS.org, 2019, pp. 253–264.
- [19] C. W. Günther and E. Verbeek, *XES Standard Definition – Version 2.0*, 2014.
- [20] H. M. Marin-Castro and E. Tello-Leal, “Event log preprocessing for process mining: a review,” *Appl. Sci.*, vol. 11, no. 22, p. 10556, 2021.
- [21] M. T. Wynn, J. Lebherz, W. M. P. van der Aalst, et al., “Rethinking the input for process mining: insights from the XES survey and workshop,” in *3rd ICPM Workshops, LNBP*, vol. 433, Springer, 2021, pp. 3–16.
- [22] B. F. van Dongen and W. M. P. van der Aalst, “A meta model for process mining data,” in *EMOI-INTEROP*, vol. 160, 2005, p. 30.
- [23] B. F. van Dongen and S. Shabani, “Relational XES: data management for process mining,” in *CAiSE Forum*, 2015.
- [24] A. F. Ghahfarokhi, G. Park, A. Berti, and W. M. P. van der Aalst, “OCEL: a standard for object-centric event logs,” in *CCIS*, vol. 1450, Springer, 2021, pp. 169–175.
- [25] V. Popova, D. Fahland, and M. Dumas, “Artifact lifecycle discovery,” *CoRR abs/1303.2554*, 2013.
- [26] Y. Bertrand, J. De Weerd, and E. Serral, “A bridging model for process mining and iot,” in *Process Mining Workshops*, J. Munoz-Gama and X. Lu, Eds., Springer, 2022, pp. 98–110.

- [27] J. Wei, C. Ouyang, A. H. ter Hofstede, and C. Moreira, “AMORETTO: a method for deriving IoT-enriched event logs,” CoRR abs/2212.02071, 2022.
- [28] S. A. Alasadi and W. Bhaya, “Review of data preprocessing techniques in data mining,” *J. Eng. Appl. Sci.*, vol. 12, pp. 4102–4107, 2017.
- [29] V. Agarwal, “Research on data preprocessing and categorization technique for smartphone review analysis,” *Int. J. Comput. Appl.*, vol. 131, no. 4, pp. 30–36, 2015.
- [30] U. M. Fayyad and G. Piatetsky-Shapiro, “Padhraic smyth: from data mining to knowledge Discovery in databases,” *AI Mag.*, vol. 17, pp. 37–54, 1996.
- [31] S. Suriadi, R. Andrews, A. ter Hofstede, and M. T. Wynn, “Event log imperfection patterns for process mining: towards a systematic approach to cleaning event logs,” *Inf. Syst.*, vol. 64, pp. 132–150, 2017.
- [32] W. M. P. van der Aalst, “Extracting event data from databases to unleash process mining,” in *Management for Professionals*, Cham, Springer International Publishing, 2015, pp. 105–128.
- [33] R. Seiger, F. Zerbato, A. Burattin, L. Garcia-Banuelos, and B. Weber, “Towards IoT-driven process event log generation for conformance checking in smart factories,” in *24th EDOC Workshops*, IEEE, 2020, pp. 20–26.

Bionotes



Joscha Grüger

Artificial Intelligence and Intelligent Information Systems, University of Trier, 54296 Trier, Germany
 German Research Center for Artificial Intelligence (DFKI), Branch University of Trier, 54296 Trier, Germany
grueger@uni-trier.de

Joscha Grüger, M.Sc. is a research associate and PhD candidate at the Chair of Artificial Intelligence and Intelligent Information Systems at the University of Trier and researches at the German Research Center for Artificial Intelligence (DFKI) Trier Branch. In 2019, he received his master’s degree in computer science from Trier University of Applied Science. His research focuses on the use of artificial intelligence methods, process mining, and business process management technologies, especially in the healthcare domain and in the context of the Internet of Things.



Lukas Malburg

Artificial Intelligence and Intelligent Information Systems, University of Trier, 54296 Trier, Germany
 German Research Center for Artificial Intelligence (DFKI), Branch University of Trier, 54296 Trier, Germany
malburg@uni-trier.de

Lukas Malburg, M.Sc. is a research assistant and PhD student at the Department of Artificial Intelligence and Intelligent Information Systems at Trier University. In addition, he is part of the German Research Center for Artificial Intelligence (DFKI) branch Trier since May 2021. In 2019, he obtained his Master degree in business informatics from the University of Trier. In his research, he examines the use of Artificial Intelligence methods, in particular (Process-Oriented) Case-Based Reasoning and AI Planning, in Cyber-Physical Production Systems and the Internet of Things combined with Business Process Management technologies.



Ralph Bergmann

Artificial Intelligence and Intelligent Information Systems, University of Trier, 54296 Trier, Germany
 German Research Center for Artificial Intelligence (DFKI), Branch University of Trier, 54296 Trier, Germany
bergmann@uni-trier.de

Prof. Dr. Ralph Bergmann is full professor at Trier University since 2004 and is directing a research group on business informatics with a strong focus on Artificial Intelligence. Since 2020 he is also topic-field leader for experience-based learning systems at the Trier Branch of the German Research Center for AI (DFKI). Over the past 30 years, he has significantly contributed to the foundations and applications of AI, including knowledge-based systems, knowledge representation and reasoning, case-based reasoning, machine learning, AI planning, and semantic technologies. With the current focus on experience-based learning systems he aims at developing hybrid AI-systems integrating data-oriented AI methods (machine learning and case-based reasoning) with semantic technologies for modeling explicit knowledge. He authored more than 250 refereed papers, including four books and 13 edited proceedings volumes and led more than 35 research projects.