
















From Internet of Things Data to Business Processes: Challenges and a Framework

Jürgen Mangler^{}, Ronny Seiger^{}, Janik-Vasily Benzin^{},
Joscha Grüger^{}, Yusuf Kirikkayis^{}, Florian Gallik^{},
Lukas Malburg^{}, Matthias Ehrendorfer^{}, Yannis Bertrand^{},
Marco Franceschetti^{}, Barbara Weber^{}, Stefanie Rinderle-Ma^{},
Ralph Bergmann^{}, Estefanía Serral Asensio^{}, and Manfred Reichert^{}

Abstract The IoT and Business Process Management (BPM) communities co-exist in many shared application domains, such as manufacturing and healthcare. The IoT community has a strong focus on hardware, connectivity and data; the BPM community focuses mainly on finding, controlling, and enhancing the structured interactions among the IoT devices in processes. While the field of Process Mining deals with the extraction of process models and process analytics from process event logs, the data produced by IoT sensors often is at a lower granularity than these process-level events. The fundamental questions about extracting and abstracting process-related data from streams of IoT sensor values are: (1) Which sensor values can be clustered together as part of process events?, (2) Which sensor values signify the start and end of such events?, (3) Which sensor values are related but not essential?

Jürgen Mangler, Janik-Vasily Benzin, Matthias Ehrendorfer, Stefanie Rinderle-Ma
Department of Informatics, Technical University of Munich, Garching, Germany,
e-mail: {juergen.mangler, janik.benzin, matthias.ehrendorfer, stefanie.rinderle-ma}@tum.de

Ronny Seiger, Marco Franceschetti, Barbara Weber
Institute of Computer Science, University of St.Gallen, St.Gallen, Switzerland
e-mail: {ronny.seiger, marco.franceschetti, barbara.weber}@unisg.ch

Joscha Grüger, Lukas Malburg, Ralph Bergmann
Business Information Systems II, University of Trier, Trier, Germany
e-mail: {grueger, malburgl, bergmann}@uni-trier.de

Joscha Grüger, Lukas Malburg, Ralph Bergmann
German Research Center for Artificial Intelligence (DFKI), Branch University of Trier, Germany
e-mail: {joscha.grueger, lukas.malburg, ralph.bergmann}@dfki.de

Yusuf Kirikkayis, Florian Gallik, Manfred Reichert
Institute of Databases and Information Systems, Ulm University, Ulm, Germany
e-mail: {yusuf.kirikkayis, florian-1.gallik, manfred.reichert}@uni-ulm.de

Yannis Bertrand, Estefanía Serral Asensio
Research Centre for Information Systems Engineering (LIRIS), KU Leuven, Leuven, Belgium,
e-mail: {yannis.bertrand, estefania.serralasensio}@kuleuven.be

This work proposes a framework to semi-automatically perform a set of structured steps to convert low-level IoT sensor data into higher-level process events that are suitable for process mining. The framework is meant to provide a generic sequence of abstract steps to guide the event extraction, abstraction, and correlation, with variation points for plugging in specific analysis techniques and algorithms for each step. To assess the completeness of the framework, we present a set of challenges, how they can be tackled through the framework, and an example on how to instantiate the framework in a real-world demonstration from the field of smart manufacturing. Based on this framework, future research can be conducted in a structured manner through refining and improving individual steps.

Key words: Process Mining, Internet of Things, Business Process Management, Event Abstraction, Smart Manufacturing

1 Introduction

In IoT environments, large amounts of procedural data are generated from IoT devices, information systems, and other software applications. The use of this data can foster the development of innovative applications in process control [63, 75, 56, 54, 35, 52, 42, 68], process conformance checking [23, 81, 83, 28], and process enhancement [67, 59], among others. Particularly, the use of process mining techniques to analyze not only process data but also IoT-collected data could provide important insights into processes and interactions as shown in different applications in the manufacturing domain, such as [58, 75, 56, 59, 67]. In these applications, IoT actuators are used to realize and execute process activities, while IoT sensors and smart tags are used to closely monitor the execution environment and involved resources [79, 75, 26, 37, 54]. IoT technology can therefore capture the context in which certain process tasks are performed, allowing process mining techniques to better understand and analyze the processes [7, 76, 12].

As such, besides the procedural data generated from the process execution systems, the data captured by IoT should also be considered an integral part of the process execution in the form of *IoT-enriched event logs* [57, 53]. Both the procedural nature of sensor logs, and the tight integration of these with the process executions and the executing resources [24] makes sensor data an integral part of process-based application scenarios in IoT [76, 75, 7].

However, the integration of IoT data and process data to be used for process mining is still often done ex-post in a manual fashion during a separate pre-processing phase [95, 73, 53]. In these cases, the data from the IoT environment is still collected and stored separately, and only later it is explicitly connected to the notion of a process, which is non-trivial as pointed out in the challenge “Bridging the Gap Between Event-based and Process-based Systems” in the BPM-IoT manifesto [37]. In case no explicit process orchestration or monitoring happens via a BPM system, and thus no explicit process data is available [30], the IoT data is the sole provider

of process-related data, which has to be separately identified, extracted, abstracted, and correlated [19, 73, 76].

We define the following two extreme states of a spectrum that IoT data sets may be in regarding their inclusion of process-related data (called **Process Awareness**), with certain levels of process awareness in between:

- An IoT data set (log) exhibits **Full Process Awareness**, when all relations between sensor events and process events are known.
- An IoT data set (log) exhibits **No Process Awareness**, when it only contains raw sensor events, and no process events at all.

In between *full process awareness* and *no process awareness*, IoT data sets may exhibit some known correlations with the process executions, e.g., the type of a process activity that a specific sensor event belongs to might be known, but not the specific instance, or the correlation of a process-level event identified from the sensor events with a specific process instance might be unknown [95].

In this work, we propose a framework to transition IoT data sets from **No Process Awareness** to **Full Process Awareness**, in a structured, non-exploratory fashion. The contribution of the paper is three-fold:

- We highlight relevant challenges regarding the automatic derivation and connection of process data with IoT data to allow for a better subsumption of future work in this area of research.
- We propose a generic framework to address the identified challenges and to derive process data from IoT data step-by-step to facilitate process mining.
- We showcase the framework using a real-life example from smart manufacturing.

This work is structured as follows: Section 2 introduces the notion of *Process Awareness* in detail. Section 3 discusses related work. Section 4 presents challenges regarding the connection of process and IoT data. Section 5 presents our framework to (semi-)automatically increase the level of process awareness in IoT data sets with a real-world example from smart manufacturing. Section 6 concludes the paper and shows opportunities and starting points for future work.

2 IoT Data and Process Awareness

This work aims at bringing the world of BPM together with the world of IoT and cyber-physical systems (CPS). The conceptual model linking the worlds of BPM (highlighted in grey) with IoT/CPS that we base our work on is depicted in Fig. 1. A *Business Process* contains *Events*, *Decision Points* and *Activities* [20]. It involves *Actors* playing an active part in the process and *Objects*. In the context of our work, components of a CPS (e.g., a production machine) can also be actors that are able to execute process activities, which in turn can be observed through *Sensor Events*.

IoT data is typically generated by a set of sensors that measure a set of values emitted as *Sensor Events* [4]. These sensors can be standalone or part of a larger CPS

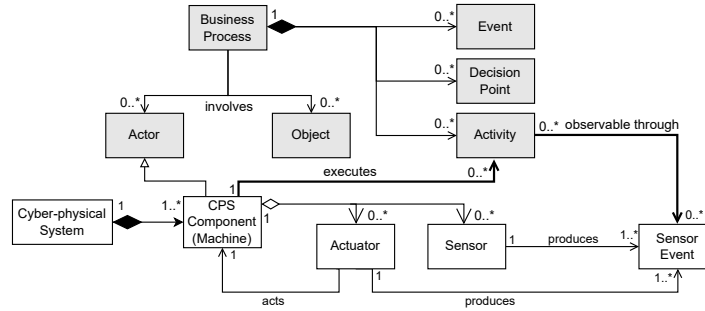


Fig. 1 Relation between concepts from BPM (in grey) and CPS/IoT (adopted from [73]).

component (e.g., a production machine or production cell) resulting in a particular *sensor topology* for a CPS. Actuators interact with the physical world and manipulate its state. As actuators may also produce data regarding their current state and actions, we consider data produced from actuators as relevant *Sensor Events* for our analysis, too. Fig. 2 shows a Fischertechnik model of a smart factory (i.e., an implementation for simulation purposes) that we use for research¹. It represents a CPS consisting of several (highlighted) CPS components that again consist of a multitude of sensors and actuators. Sensor events may also be directly related with actors and objects. However, we put process-related concepts in the center of our investigations, which is why the key relationship for our work to bridge the BPM and IoT worlds is the *observable through* association between *Activity* and *Sensor Event* in Fig. 1.

The sensor events are typically either directly pushed to some IoT data log by the sensor, or collected and stored in an IoT data log by an external mechanism. Thus, a set of IoT data logs, given some data retention policies, holds historical information regarding all timestamped observations from a set of sensors. For each sensor, the stream of events (called *Raw Sensor Stream* in this work) is available as a time series [16].

2.1 Process Activity–Sensors–Machines Relationship

When associating data collected from IoT with process data to achieve full process awareness, one has to consider that sensors can be related to process activities without explicitly knowing the corresponding actions, i.e., which machine or human triggered a certain chain of events. This is because one activity might trigger, e.g., the start of a machine, while other activities might fine-tune the parametrization of the machine at runtime. A third activity might then wait for an event signifying the end of the operation. This also means that one sensor at any given time might produce data relevant to many activities, while one activity might be enacted in the context of many

¹ <https://www.fischertechnik.de/en/products/industry-and-universities/training-models>, last access: 2024-05-13

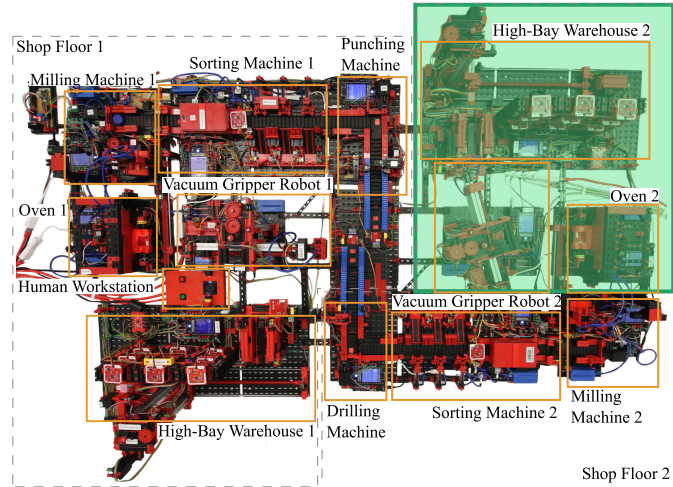


Fig. 2 Factory model representing a CPS with highlighted CPS Components [56].

sensors (cf. Fig. 1). The same holds true for machines: a sensor might at any given time produce data relevant to the operation of many machines (e.g., a temperature sensor might deliver data relevant to the operation of multiple machines), while one machine might operate in the context of multiple sensors [59, 73, 41, 44].

The reason for collecting IoT data typically is to monitor the effects of a series of actuations, e.g., as part of a production process, the treatment of a patient, or as part of a building automation scenario. The collected data can be used to [84, 76, 43, 37, 54, 18]:

- monitor, visualize, and track progress,
- find deviations from rules,
- find drifts in relation to goal functions.

While these analysis targets are very generic and universally applicable, they can only be reached with detailed knowledge of the scenario, its rules, goal functions, and the underlying **Business Process** [20].

All IoT scenarios have in common that a series of actuations, potentially influenced by humans and the environment, are based on a blueprint that defines when something should happen, and which order relations might exist between actuations. This blueprint, the **Process** [37, 20], might either be:

- Implicit, i.e., the result of the interactions of some hard-coded software components with potential human involvement, or
- Explicit, i.e., a known artifact (e.g., the **Process Model**), which is the basis for enacting a certain scenario, e.g., through a process engine.

The relationship between IoT data and processes can be understood as IoT data tracking changes in the observed environment while processes and activities are executed, and thus allow to observe their effects (cf. Fig. 1 [4]).

Albeit, some caveats to complicate the situation exist:

- Sensors, e.g., temperature sensors, might collect data while no scenario is enacted. Thus, data might not be relevant for the process [73].
- Not all sensor data can be directly assigned to a specific scenario, only the combination of a set of individual observations from different IoT data logs (e.g., temperature sensors combined with the status of an oven) might allow identifying significance for a particular scenario [76].
- As IoT data from different sources is available in different granularity, substantial effort for aggregation of the data has to be made [80, 76].

Consequently, as briefly introduced in Section 1, we define **No Process Awareness** as a situation where only raw sensor streams from a set of data sources are available (cf. Fig. 3) [73]. **Full Process Awareness**, on the other hand, **is the goal** as it represents the full contextualization of all data regarding a certain scenario, which is represented explicitly by a set of process models [59]. Each event in each available sensor stream time series has been either discarded due to irrelevance or assigned to an observation of an actuation. Each actuation can be related to many process-level events, and these events might be related to multiple process activities [32, 73]. For the remainder of this section, we further explore the two extreme states of the Process Awareness spectrum with different levels in between.

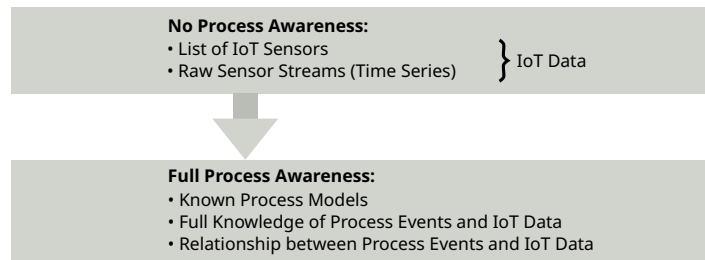


Fig. 3 Relation between IoT data and process awareness.

2.2 Example: Smart Manufacturing Process

In order to better exemplify the concepts discussed in this section, and following sections, for the remainder of this work we will focus on a specific slice of the scenario displayed in **Fig. 2 (top right area, green box)**. This **running example** is displayed in simplified form in **Fig. 4** and describes the interaction between a **Conveyor** which transports raw parts to a **Crane** (“Vacuum Gripper Robot 2”), which can move the parts to an **Oven** (“Oven 2”) for surface treatment, and subsequently to a **Stockpile** (“High-Bay Warehouse 2”).

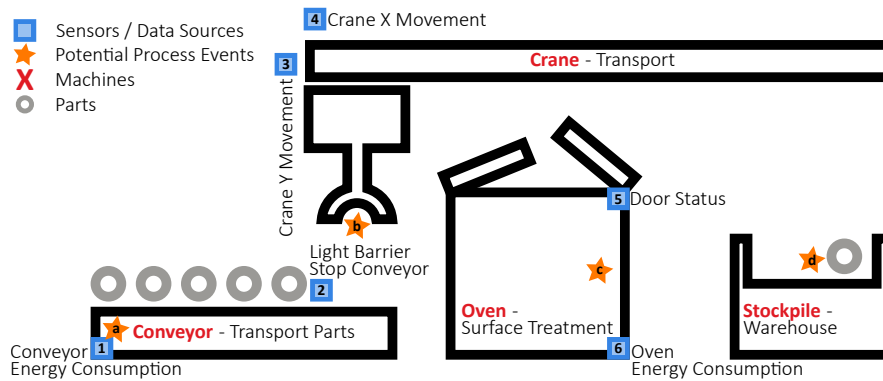


Fig. 4 Running example, part of factory simulation model shown in Fig. 2.

Our running example utilizes a total of 6 different sensors, respectively data sources, which deliver the following values:

- 1** is a continuous sensor that emits the energy consumption levels of the **Conveyor** in watts every second. When the **Conveyor** is not running, the sensor emits the value (close to) 0 every second.
- 2** is a binary sensor that switches from 0 to 1 whenever a **part** is in its field. It is also used by the internal programming of the factory to start/stop the **Conveyor** to ensure that whenever a **part** is taken by the **Crane**, a new part is moved into a position where it can be picked up.
- 3** is a continuous sensor that monitors the y position of the **Crane**, i.e., how far the gripper is extended downwards. It delivers values every second whenever the position changes, it delivers no values, when the position is unchanged. Fully extended the y value for the **Crane** is 100, fully retracted the value is 0.
- 4** is a continuous sensor that monitors the x position of the **Crane**, i.e., in which position above the **Conveyor**, **Oven**, or **Stockpile** the crane is. It delivers values every second whenever the position changes, it delivers no values, when the position is unchanged. When the **Crane** is above the **Conveyor** the value is 0, when it is above the **Oven** it is 30, and when it is above the **Stockpile** it is 60.
- 5** is a binary sensor that monitors if the door of the **Oven** is opened (1) or closed (0).
- 6** is a continuous sensor that emits the energy consumption levels of the **Oven** in watts every second. When the **Oven** is not running, the sensor emits the value (close to) 0 every second. As the **Oven** is always kept at temperature, positive energy consumption does not necessarily mean that a part is currently processed.

Between all of these sensor values, a wide range of process events could be specified, leading to more or less fine-granular process models that describe the interaction between the **Conveyor**, the **Crane**, the **Oven**, and the **Stockpile**. Potential process events that we want to highlight for our running example are:

- ★ a The **Conveyor** moves a **part** into position, which can be deduced from **1**.
- ★ b The **Crane** grabs a **part**, which can be deduced from certain values of **2**, **3**, and **4**.
- ★ c A **part** is surface treated in the **Oven**. If **3**, **4** have certain values, and **5** is 1, then the oven is currently being loaded. If **3**, **4** have certain values, and **5** is 0, then the oven is processing a **part**.
- ★ d The **Crane** drops a **part** into the **Stockpile**, which can be deduced from certain values of **3**, and **4**.

2.3 No Process Awareness

This state of Process Awareness typically occurs in brownfield scenarios. IoT data collected from existing complex IoT environments is partially analyzed regarding certain domain-specific preordained, non-process related questions. Whenever new questions arise, the data aggregation is refined to derive the required answers. This state is typically prone to breakages as the following properties might hold true:

- Unknown Side-Effects: During the operation of the IoT environment, actuations might occur which are not foreseen in the data analysis. The software steering the actuations, or the human's or environment's influence on the process might not be fully understood [74, 62].
- Breaking Changes: Such side-effects might come into existence whenever changes to the IoT environment are made. Changes not only include the addition of new actuators or sensors, but also the (software) upgrade of existing ones [1].

Consequentially, all changes have to be tracked, monitored, and all data analysis tasks have to be rechecked if they still deliver the required answers, or have to be adapted to do so. We can sum up the properties of this state as follows, while not all of them must apply to all scenarios [76, 80, 75, 64, 73, 29, 5, 97]:

- Hard-coded software.
- No or only partial knowledge of the process.
- Event correlation ambiguities: no clear assignment of events to individual actuations.
- Partial monitoring and tracking of data.
- Partial knowledge about the dependencies between sensors (i.e., sensor topology), and how they influence different scenarios.
- Incomplete data, i.e., sensors might be missing to monitor relevant aspects of the process executions.
- Inconsistent data resulting from different data granularities of different sensors.

Table 1 shows an excerpt from a typical IoT data log referring to the example scenario described in Section 2.2 which only contains sensor readings with a minimal

set of sensor attributes, sensor value, and timestamp of the sample. There is no process-related information available.

Table 1 Excerpt from an IoT data log with no process awareness.

	Sensor ID	Sensor value	Sensor timestamp	Case	Process event ID	Event timestamp	Lifecycle transition	Label
1	1	512	1	-	-	-	-	-
2	2	0	1	-	-	-	-	-
3	1	0	2	-	-	-	-	-
4	2	1	2	-	-	-	-	-
5	3	100	3	-	-	-	-	-
6	4	0	3	-	-	-	-	-
7	4	-	-	-	-	-

2.4 Full Process Awareness

This state of Process Awareness typically occurs in greenfield scenarios, when new IoT environments are planned and realized. Furthermore, it might occur in brownfield scenarios, when process management software is introduced as part of an automation effort, such as [59, 71]. Full process awareness is often characterized by [59, 32, 75, 56, 53, 18]:

- Full knowledge of the relevant processes in the form of a set of process models.
- Imperative control or at least full monitoring through a process engine.
- Sensors which are linked to process activities, i.e., sensors might contribute data to one or more activities.
- Complete Data: if data is missing, the process is not specified well enough.
- Logs, which are automatically generated as part of the execution, and containing events and process data at a suitable, homogeneous level of granularity [97]; complemented by IoT data [57].

Table 2 Excerpt from an IoT-enriched event log with full process awareness related to control flow.

	Sensor ID	Sensor value	Sensor timestamp	Case	Process event ID	Event timestamp	Lifecycle transition	Label
1	1	512	1	part 1		1	start	move part
2	1, 2	0, 1	2, 2	part 1		2	complete	move part
3	2, 3, 4	1, 100, 0	2, 3, 3	part 1		3	start	grab part
4	part 1

In this case, data stores for individual sensors might still exist, but they are no longer used because all data is collected as part of the execution of process instances

stored in an IoT-enriched process event log [32]. Table 2 contains the same sensor readings as Table 1, but these are now fully contextualized in the control flow of the process execution (cf. scenario process in Section 2.2) as indicated by the additional process attributes according to the XES standard [93]. In this example, the sensor events are clearly related to specific process events for one case, i.e., an executed process instance, namely the workpiece “part 1”. Furthermore the following important concepts are exemplified in Table 2:

- One or multiple sensors and thus one or multiple sensor events might contribute to one process event.
- Row 1 and row 2 describe different events in the lifecycle of the same process task (start/complete of *move part*). Multiple combinations of sensor events might point to the same process task, i.e., in our example **1** is sufficient for start, but the combination of **1** and **2** with different values point at the and also is more robust (i.e., more sensors involved means better distinction).

Thus it becomes clear, that a process event is defined by a combination of sensors and the specific values they have at a given point in time.

But how are the sensor values collected? We differentiate between three options of realizing the data collection.

2.4.1 Option 1: Explicit Modeling of Sensor Data Collection

This way is about modeling the process in the most fine-grained and detailed way possible, including the explicit collection of sensor data as process tasks (activities) that are part of the process model [59]. This has the advantage that

- every process engine can support it,
- and models are better to follow as everything is made explicit in an already existing modeling language (e.g., BPMN 2.0).

On the downside, the resulting models are complex and complicated. Furthermore, the data collection might be hindered by the expressiveness of the modeling language, e.g., BPMN 2.0 requires complex synchronization modeling if data relevant to multiple in-parallel running tasks has to be collected in separate process instances [89]. While this downside can be alleviated by modeling the process in a more abstract way (fewer tasks, higher abstraction, “more knowledge” in lower levels), this requires increased effort to maintain a good modeling quality and decreases the level of detail of monitoring.

2.4.2 Option 2: Implicit IoT Data Collection

In this approach, the process model contains additional information/annotations that allow data collection implicitly, independent of the traditional process execution,

but still automatically. For example, if process activities contain information about linked sensors, and how to collect data from them, this data collection can be enacted automatically in parallel to the normal process execution [54, 41].

The main advantage is that the process models stay simple, but on the downside some additional knowledge describing the connection to the IoT environment (e.g., in the form of annotations [54, 41]) has to be provided.

2.4.3 Option 3: Raw IoT Data Collection

In this approach, the process notion is completely ignored, and data is collected directly from all sensors. This is the most common way of data collection. In contrast to the two other collection methods, where the data can be used for compliance checking and process refinement, here the goal is to discover and create a new process model. The downside is that for getting the correct granularity of the process models, some of the steps in the framework proposed in this paper (cf. Section 5) become much more reliant on domain expert input and validation [73].

3 Related Work

Augmenting business processes with IoT technologies brings several benefits, but does not come without challenges [37, 70, 17]. In this section, we discuss related research addressing IoT data processing as well as event extraction and event abstraction [19] in the context of IoT with the goal of increasing the level of process awareness associated with IoT data, and ultimately, to enable process mining [2].

One of the main benefits is the improved recognition of activities and processes from sensor data, e. g., with the help of multi-modal approaches such as the one by Rebmann et al. [69] that combines motion and vision sensors with user feedback for detecting and disambiguating known activity types/tasks in real-time. Other approaches employ camera-based object detection to track workpieces within a smart factory and correlate them to the execution of individual process activities [55]. Activity recognition is often combined with process mining for extracting knowledge to analyze and optimize the underlying processes [60, 14]. Applying process mining, in turn, requires bridging the gap between low-level sensor data and the event logs needed for process mining—a well-known issue, which gives rise to succeeding challenges [37].

The data used for process mining can reside in different data sources. *Event Extraction* includes the identification of the event data and its extraction [19]. To leverage the opportunities IoT brings for improved recognition of (manual) activities and processes from sensor data [80], differences in granularity at which the data is recorded and at which it shall be analyzed [19] need to be bridged, also referred to as *Event Abstraction*. In turn, *Event Correlation* aims at associating event data extracted from data sources to *cases* of a process [12]. An overview of non-IoT

related event abstraction techniques in the field of process mining is provided in a literature review by van Zelst [96]. A paper by Diba et al. [19] provides an overview of different methods for event abstraction and correlation in the traditional process mining context.

3.1 Event Extraction

Identifying relevant data for process mining and extracting it from different sources is part of *Event Extraction* [19]. Existing approaches use database schema, process documents, domain models, event models, and domain knowledge for the identification and extraction of relevant data (cf. [19, 38]) and mostly focus on the extraction of event data from databases and traditional information systems. Recently, several works looked into the use of IoT as data source for process mining (e.g., [2, 21, 84, 47, 23, 7, 50]). Similarly, in our work we are moving towards IoT sensors as new data sources for event extraction and process correlation [76], which requires the sensor streams to be accessible and linked to processes [13, 65].

The IoT provides new sources of access to real-time data and thus opportunities to develop advanced monitoring approaches, e.g., for condition monitoring, predictive maintenance [36] or verification of process outcomes [74]. Large amounts of data, different data formats, sampling rates, and data quality are among the challenges that come with using the IoT as a new data source [17, 2, 40]. With our framework, we focus on using IoT data for event extraction, abstraction, and correlation where we relate raw IoT event streams to the execution of process activities [39]. Most works assume rather high levels of process knowledge when discovering activities and processes from sensor data, i. e., existing activity labels only have to be connected to the raw events [19, 59]. With almost no process knowledge and limited IoT topology knowledge, most approaches are not applicable.

3.2 Event Abstraction

Event abstraction in the context of BPM focuses on the *abstraction gap* between the granularity at which the data is recorded and at which it is analyzed [37, 19, 5]. When considering sensor data, the challenge of mapping fine-grained sensor data to more abstract process activities becomes more pronounced [18, 37, 96]. In literature [96], various approaches exist to bridge this abstraction gap using, e.g., Complex Event Processing (CEP) [72, 76, 80] or machine learning (ML) [27, 88, 39].

Event abstraction is needed to map multiple fine-grained sensor data to coarse-grained process events [96, 18], e. g., with the help of CEP [80], which also supports data pre-processing and context enrichment [87]; or with machine learning approaches such as clustering [27, 18, 25] or supervised learning [88]. Wanner et al. combine these two methods based on expert knowledge and observations in the

context of a smart factory [94]. Then, to be suitable for the application of process mining techniques, such coarse-granular events need to be matched to the process activities in a process model or correlated to the process instances through *event-to-activity mappings* [3, 91]. We discuss work about the development of frameworks for IoT processes and event log generation from IoT events that mostly relates to our contribution, and refer to [80, 96, 17, 2] for a broader literature overview.

In [77], Senderovich et al. propose a knowledge-driven approach based on interactions mining to transform historical logs of sensor data into standardized event logs including information about process instances. In [61], Mannhardt et al. propose a supervised method for event abstraction based on activity patterns. These patterns are derived from low-level events using domain knowledge and then used to build an abstracted event log by exploiting existing alignment techniques. The authors remark the challenges stemming from the distributed, continuous, and ambiguous nature of IoT data, which makes event abstraction a necessary step for event log creation. Moreover, Koschmider et al. [47] discuss a modular framework for obtaining process event logs from sensor data, explicitly considering the data aggregations needed for event correlation, activity discovery, and event abstraction. The lack of explicit process information in the low-level IoT data may lead to *uncertainties* when being correlated to specific cases, as patterns within IoT data can often be associated with multiple activities and processes [47]. Ehrendorfer et al. present in [23] an approach for conformance checking and classification of manufacturing log data from a smart factory, thereby raising the analysis of low-level IoT data to a process-oriented perspective.

3.3 Conclusions from Related Work

From our investigations of related work at the intersection of BPM and IoT, we can conclude that the application of IoT technology in the domain of BPM has received increasing attention over the past years [37, 17]. Several approaches investigate aspects regarding IoT as new data source for enriching process event logs and for supporting process mining [2]. Our discussions in Section 2 have shown that detecting process-level events and activities from raw IoT data is a multi-step process. Related work either focuses on using IoT data in early stages of activity detection (e.g., event extraction), on using specific sensors to detect specific types of activities independent of the process context, or on parts of the event abstraction [18]. In later stages of event and activity detection (e.g., event abstraction, event correlation, process mining), IoT data plays a less significant role. The goal of this work is to derive a list of challenges that have to be addressed when going the entire way to increase the level of process awareness of IoT data as defined in Section 2 from *no process awareness* to *full process awareness*. From these challenges, which are backed up by additional related work and literature, we intend to develop a holistic framework consisting of a generally applicable sequence of steps that address these challenges in order to increase process awareness of IoT data. Thereby, the

framework should serve as a skeleton defining generic, abstract steps and their expected inputs and outputs. The particular techniques and approaches discussed in the aforementioned related work can then be plugged in as concrete mechanisms to implement one or multiple steps of the framework.

4 Challenges

We have identified the following challenges regarding the automatic connection and derivation of process data and IoT data. These challenges are based on aforementioned related work and additional literature [37, 11, 85, 8] as well as on our own experience from implementing real-world use cases regarding the integration of BPM and IoT data in smart factories [59, 56, 75, 43, 41], smart healthcare [82, 28], and smart homes [74, 71].

The most relevant basis for this work is the BPM-IoT manifesto [37], which introduces 16 high-level challenges related to the interaction between the IoT and BPM. The manifesto presents these challenges along a similar spectrum as we have introduced in Section 2, going from rather low level raw event data to higher level process-related knowledge. We provide a more in-depth view on some specific challenges related to IoT and (business) process data posed in the manifesto. Specifically, we address aspects referring to 1) *Placing Sensors in a Process-aware Way*; 2) *Connection of Analytical Processes With the IoT*; 3) *Integrating the IoT With Process Correctness Checks*; and most prominently 4) *Bridging the Gap Between Event-Based and Process-Based Systems* [37]. With that, we cover all major challenges that are related to collecting and analyzing raw event data from sensors and actuators with the goal of deriving higher-level knowledge. This knowledge is then used for tasks in the BPM context, e.g. the discovery of new processes, prediction and adaptation of processes, or enacting responses [37].

We have identified the following list of 10 challenges:

- C0** Data delivered by IoT sensors is often incompletely described, e.g., the measurement procedure affecting the granularity and the units are missing, and sensors are not aware of their location [78, 33]. Moreover, missing knowledge about the granularity of sensors hinders the combination and aggregation of different sensors, which is crucial for conducting process mining on IoT data [8].
- C1** Typically, IoT environments produce a plethora of data from heterogeneous sources [11], both regarding volume and variety [10], which can only be stored and analyzed with considerable computational effort [40, 48, 9].
- C2** Data delivered by IoT sensors has different meaning in different scenarios, e.g., a certain temperature reading from one sensor might point towards an error in one scenario/process, while it signifies nominal operation for a second scenario/process [41, 34]. Thus, it is desirable that the implemented processes are stronger coupled with the corresponding data produced during execution (cf. Challenge 6 in [9]).

- C3** Given a scenario and an IoT environment, the complete set of sensors and sensor information required for process analysis cannot be easily derived [11, 44], which is mainly due to the missing integration of process and IoT data within a common event log format or model [8, 32]. This results in potential ambiguities regarding the process–IoT data correlation [29, 76, 61, 8].
- C4** Different sensors are relevant for different scenarios, both regarding if they are required at all, and regarding their significance to the scenario/context [73, 34, 51, 90]. In addition, specific IoT data can be relevant in more than one scenario (cf. Challenge 3 – *Scope of Relevance* in [8]).
- C5** Data analysis is often dependent on visualization and human involvement [73, 95], which makes many of these interactive approaches infeasible, for example for small lot-sizes in manufacturing scenarios or for high-volume, data-intensive production processes [40]. This leads to data analysis being a highly demanding task and thus process refinements being difficult to achieve (cf. Challenge 9 in [9]).
- C6** In semi-structured domains with high human involvement, it is challenging to get complete process/event and IoT data in a desirable and uniform granularity [28, 5, 40, 11]. Moreover, monitoring of human activities quickly raises privacy challenges [15] and is, in general, difficult to achieve [45].
- C7** Refining existing process scenarios requires in-depth domain knowledge, and is currently neither well-supported by a methodology nor by (semi-)automatic decision support systems [37, 75, 90]. The automatic correlation of IoT data streams, process instances, and process models for the purpose of process refinement is currently not possible.
- C8** In some scenarios, the explicit knowledge of processes might not be complete [37, 76, 11, 31]. Processes might be emerging and evolving [62, 74]. This makes predicting which IoT data is required for analysis difficult.
- C9** Process outcome prediction, given explicitly known processes, is non-trivial given the high number of influencing IoT devices and data streams as well as physical surroundings (context), thus leading to emergent behavior [46, 37, 9].

With these 10 challenges we provide a comprehensive, though non-exhaustive, more detailed list and discussion—compared to the BPM-IoT manifesto [37]—of aspects that emerge when trying to collect and correlate BPM and IoT data to increase process awareness within the IoT. In the following section, we will propose a framework to address these challenges with the goal of moving from no process awareness to full process awareness within a given IoT data set or data stream.

5 Framework: From No Process Awareness to Full Process Awareness of IoT Data

In this section, we describe a data-driven process deduction framework consisting of eight steps to guide the analysis of a given IoT dataset from no process awareness

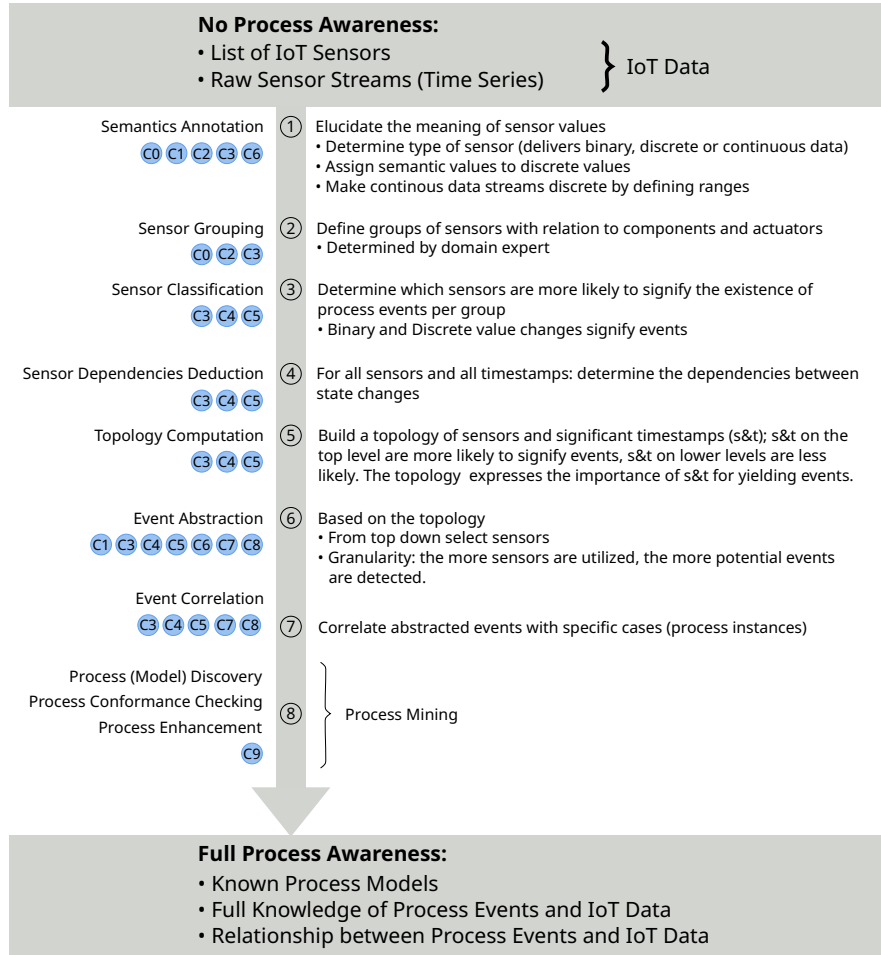


Fig. 5 Framework for (semi-)automatic process deduction from IoT data, including references to challenges addressed by each step.

to full process awareness in accordance with the definitions given in Section 2, and as depicted in Fig. 5. In the framework we provide a sequence of generally applicable steps and specify their interfaces including their expected inputs and outputs to increase process awareness of the datasets and intermediate artifacts. The goal is to (semi-)automatically transform the raw IoT sensor streams and augment them with process models, knowledge of process events and IoT data, and their relationships. As a starting point, we assume that only the IoT data containing a list of IoT sensors and the respective raw sensor streams are given, i.e., there is no process awareness (cf. Section 2). For each step we provide a conceptual description and an instantiation of the framework based on the running example from smart manufacturing (cf. Section 2.2). We then discuss which of the challenges from

Section 4 are addressed by the individual step. Note that the instantiations of the framework's individual steps are intended to show just one possible method or technique to achieve the desired outcome in this step, either relying on the domain expert or already on some (semi-)automated means. Other methods or techniques may be possible to implement the specific steps, as well. The goal of the framework is to streamline the acquisition of process awareness in a series of generalized abstract steps with a high degree of automation potential. The conceptualization and implementation of all possible techniques and algorithms for each step is left for future investigations.

Throughout this section we will continue to use the example introduced in Section 2.2. As already mentioned, how the manufacturing process is executed / running, is not important: it could be executed through a series of PLC programs that may or may not include the sensors, through a process engine enacting an for us unknown process, or through a static Python program that controls all the machinery.

5.1 Semantics Annotation

5.1.1 Concept:

In Step ① the domain expert elucidates semantics and characteristics of the sensors such as the type of the sensor values or relations after discretizing continuous streams. For instance, the process expert can give labels to the values of binary sensors. These semantics guide the subsequent steps in deducing how changes in the sensor stream values signify process-level events and thus the execution of activities. For each sensor the semantics are stored according to the schema defined in [32].

5.1.2 Challenges:

Here we address challenges **C0** and **C2**, as well as **C3** to some extent, by enriching the set of sensor streams with additional information describing the sensor characteristics to enable the (semi-)automatic process deduction through subsequent steps. For **C1**, we assume that through systematically discretizing continuous data, the amount of stored values can be drastically reduced while producing equal or better results with less computational effort. The privacy aspect in **C6** on the other hand can be supported by annotating data with privacy levels, which can be utilized further on to (a) always identify when privacy sensitive data is involved, and (b) to regulate the visibility of such data. This step will be hard to conduct automatically, as it requires deep semantic knowledge.

5.1.3 Example:

Table 3 is the result of the semantic annotations done by the domain expert for the set of 6 sensors from the smart factory introduced in the example in Fig. 4 (cf. Section 2.2). In our particular example we can see that all values typically have a semantic meaning. Not being able to map all continuous sensor values (i.e., “Source Type”) to discrete values (i.e., “Target Type”) is no problem in our framework (as is exemplified in Step 4 – *Sensor Dependencies Deduction*). But having a clear mapping to discrete values helps in understanding why certain process events occur, and greatly reduces the computational complexity of potentially automatically carried out subsequent steps. From the example it also becomes clear that with a sufficiently trained AI or Artificial General Intelligence (AGI) this step could even be conducted automatically.

Table 3 Sensor annotation, with the source type and target type, and the explanation of the values.

ID	Component	Sensor	Source Type	Target Type	Value Mapping and Semantics	
					Source → Target	– Semantics
1	Conveyor	Energy Consumption	continuous	discrete	0 → 0 1..1000 → 1 > 1000 → 2	– stopped – running – stuck
2	Conveyor	Light Barrier	binary	binary	0 → 0 1 → 1	– unbroken – triggered
3	Crane	Y Movement	continuous	discrete	0 → 0 100 → 1 1..99 → 2	– up – down – moving
4	Crane	X Movement	continuous	discrete	0 → 0 30 → 1 60 → 2 1..29 → 3 31..59 → 3	– Conveyor pos – Oven pos – Stockpile pos – moving – moving
5	Oven	Door Status	binary	binary	0 → 0 1 → 1	– closed – open
6	Oven	Energy Consumption	continuous	discrete	0 → 0 1..1000 → 1 > 1000 → 2	– off – heating – error

5.2 Sensor Grouping

5.2.1 Concept:

In Step ② the domain expert defines groups of sensors based on the setup of the physical CPS components including their sensors and actuators, which are henceforth also considered as *sensors* producing sensor events (cf. Fig. 1). This should be

rather straight-forward in traditional manufacturing scenarios (e.g., grouping for one production machine or one production cell) or medical settings as the grouping is usually done based on physical proximity of sensors and their belonging to one specific CPS component (cf. Fig. 1). This grouping marks which sets of sensors are impacted by the same activities and should therefore be analyzed together to detect these activities. Groups that only contain one sensor may also exist.

5.2.2 Challenges:

Here we address the challenges **C0**, **C2**, and **C3** in a similar way as in Step ①: by further adding information that guides the subsequent steps. Having a clear semantic relationship-based grouping of sensors will again help in increasing understandability of how process events could be deducted from sensor events, as process events are more likely to occur as a result of interactions *between* sensor groups (i.e., sensor groups are often equivalent to production cells).

5.2.3 Example:

Sensor groups are formed by the domain expert through mostly considering the location of the sensors and the associated machine. For example, the sensor group *Buffer* contains different sensors which might encompass one production cell (cf. Table 4). For the purpose of this example we added an additional sensor **7**, which is not part of the original running example (cf. Fig. 4). **7** measures the ambient temperature on the shop floor, and in our case is considered important for the correct functioning of the conveyor and the crane. This exemplifies that . . .

- . . . one sensor can be part of multiple groups, and
- . . . not all sensors must be part of, or directly associated with a component.

Table 4 Grouping of sensors and assignment of new group identifiers.

Sensor ID	Component	Sensor	Sensor Group (G)	GID
1	Conveyor	Energy Consumption	Buffer	B
2	Conveyor	Light Barrier	Buffer	B
3	Crane	Y Movement	Buffer	B
4	Crane	X Movement	Buffer	B
7	Ambient	Temperature	Buffer	B
5	Oven	Door Status	Oven	O
6	Oven	Energy Consumption	Oven	O
3	Crane	Y Movement	Stockpile	S
4	Crane	X Movement	Stockpile	S
7	Ambient	Temperature	Stockpile	S

As can be seen, wherever groups (e.g., production cells) interact, sensors might be part of multiple logical groups (cf. ambient temperature, crane movement sensors in our example). Defining the correct granularity of the groups will not impact the overall result quality of subsequent analysis steps, but can contribute significantly to visualization and understandability of the results. For our running example we chose the conveyor (representing a buffer), the oven, and the stockpile as the logical groups, but other groups could also make sense. The oven group for example could also profit from containing values from crane sensors.

5.3 Sensor Classification

5.3.1 Concept:

In Step ③ each sensor is ranked according to the significance regarding the existence of process-level events (e.g., the start or end of an activity) associated with it. Often, sensors with binary value changes are the most significant, followed by sensors with discrete value changes, followed by sensors with continuous value changes [73, 76].

While this ordering can be a good first choice, the classification should be confirmed by a domain expert or a more automated technique that analyses sensor data for its relevance. The classification in this step is done per sensor group, whereas inter-group relationships are analyzed in Step ④. The resulting classification ranks each sensor per group. A lower ranking sensor (i.e., higher priority) is more likely to indicate its contribution to process-level events. Such a classification can be derived automatically by the information available from steps ① and ② through analyzing the discretizations, and the respective sensor value stream characteristics within a group.

5.3.2 Challenges:

This step addresses challenges **C3**, **C4** and **C5** by semi-automatically deriving a classification signifying the relevance of each sensor per group for the scenario and the data analysis.

5.3.3 Example:

The information from the previous steps is analyzed to classify the sensors within a group resulting in Table 5.

As can be seen, we propose to rank sensors for each group that was defined in the previous step to simplify the ranking. For example for the first group, the binary sensor has ranking (priority) 1, while discrete sensors are ranked 2 (lower priority).

Table 5 Ranking of sensors per group (a lower ranking means a higher priority/significance of the sensor).

Sensor ID	GID	Sensor	Target Type	Sensor Ranking
1	B	Energy Consumption	discrete	2
2		Light Barrier	binary	1
3		Y Movement	discrete	2
4		X Movement	discrete	2
5	O	Door Status	binary	1
6		Energy Consumption	discrete	2
3		Y Movement	discrete	2
4		X Movement	discrete	2
3	S	Y Movement	discrete	1
4		X Movement	discrete	2

Refined rules might include that the discreteness dimension (binary is a special case of discrete with 2 potential values) is further considered: e.g., X Movement of the crane is considered less important because it has 4 discrete values in comparison to Energy Consumption or Y Movement. Even more refined rules might consider the range of source values, e.g., X Movement is more important because it only spans values from 1 to 59, as compared to the Y Movement which covers values from 0 to 99, or Energy Consumption which covers source values from 0 to greater than 1000.

Finding a good set of rules will be subject of future research, i.e., multiple ranking algorithms with their own strength and weaknesses might emerge.

5.4 Sensor Dependencies Deduction

5.4.1 Concept:

In Step ④ dependencies across all sensors and sensor groups are deduced. The goal is to identify important timestamps, where multiple sensors show significant *joint changes*. While discrete sensors might just change their values a few times, continuous sensors typically change their values all the time. In order to identify the importance of a change in sensor values at a particular timestamp, as many as possible sensors, weighted by their importance as calculated in Step ③, have to contribute.

This analysis can be done manually by the domain expert, but it also shows high potential for automation based on techniques and metrics for sensor similarity and correlation calculation (e.g., using dynamic time warping [66], clustering, or more sophisticated machine learning approaches).

Nevertheless, the concept of detecting joint changes comes with a set of interesting questions:

- In which time frame do joint changes occur? If changes occur within, for example, 1 second, are they joint? How big does the time window need to be?
- Should we operate on the original data, or on the discretized data from Step ①?
- If a sensor that provides continuous data (and thus changes all the time) is part of every change, how can we minimize its influence or exclude its dependency to everything?

5.4.2 Challenges:

This step addresses the same challenges C3, C4 and C5 as Step ③ by revealing dependencies among sensors across groups. The biggest challenge in this step is, that changes do not always happen at the same time, but some sensors might only show a delayed reaction, which makes the change detection for a group of sensors dependent on thresholds and thus computationally expensive.

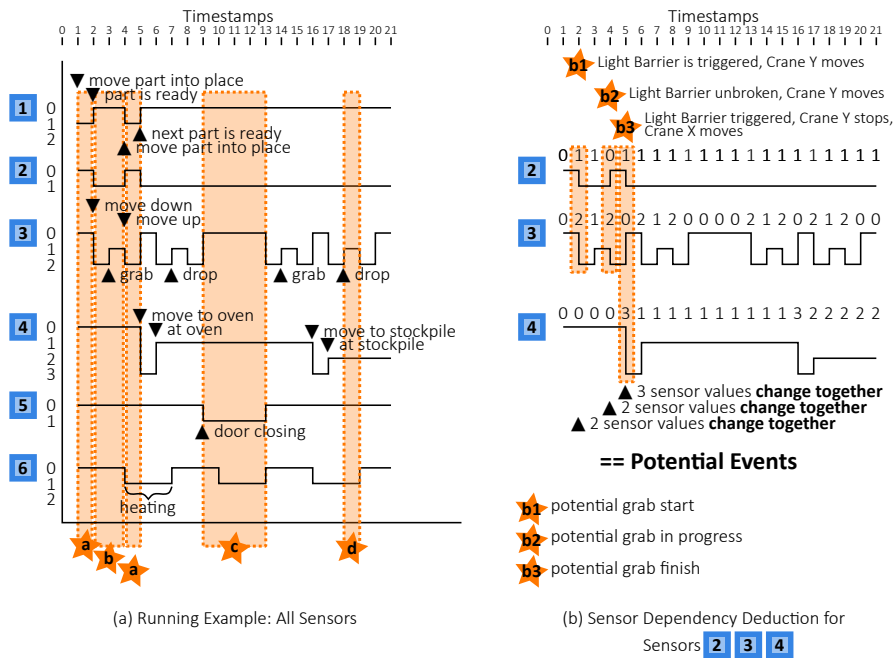


Fig. 6 Dependency Deduction: Sensor values change together

5.4.3 Example:

Sensor dependency deduction infers dependencies between all sensors across groups by means of finding joint changes. Figure 6(a) shows the discretized changes of all sensors for the complete running example scenario, while Fig. 6(b) exemplifies the dependencies by showing how the values change for a set of sensors. Figure 6 also hints at potential solutions for the questions asked above. Joint changes rely on temporal proximity. We presume that by calculating solutions for multiple time frames (utilizing multiple thresholds), the granularity of dependencies and thus the granularity of potentially detected process events can be controlled. Working with discretized data makes the algorithm much faster, but by pursuing automatic discretization of original data (e.g., by utilizing Symbolic Aggregate approximation [86] to map to integer values), again the granularity of dependencies and thus the granularity of potentially detected process events can be controlled. Thus a wide range of potential algorithms and optimizations for calculating dependencies seem possible.

For our running example we assume a naive algorithm and rely on the discretized data from Step ①. We assume changes happen exactly at the same time. Table 6 shows the changes for all sensors shown in Fig. 6(b).

Table 6 Dependencies among sensors.

Timestamp	Joint Change
0	–
1	–
2	2, 3
3	3
4	2, 3
5	2, 3, 4
6	3, 4
...	...

While for Table 6 we only used sensors 2, 3, and 4, to get the all dependencies, all sensors and all timestamps have to be considered, leading to a large table of timestamps with linked sensor activity.

5.5 Topology Computation

5.5.1 Concept:

In Step ⑤ a list of sensors that represents the sensor *topology*, i.e., the hierarchy and the dependencies among sensors and components and their significance, for a particular process is derived. Sensors are ordered with the goal of having sensors that are more likely to contribute process events on the top of the list, while sensors

that are less likely to produce process events (or would produce sensor events at a much finer granularity) at the bottom of the list.

We envision the topology to be automatically derived from information calculated in steps (3) and (4). This topology is intended to allow process analysts to fine-tune the number of potential process events derived from sensor events, by gradually including sensors from top to bottom of the list.

5.5.2 Challenges:

Here we address the same challenges **C3**, **C4** and **C5** as in Steps (3) and (4) by computing a topology of sensors in a list. Sensors closer to the top of the list indicate that value changes of these sensors are more relevant to the scenario and the analysis than sensors closer to the bottom.

5.5.3 Example:

A naive topology calculation for the event subset **2**, **3**, **4** and the first six timestamps (cf. Table 6, Step (4)) is depicted in Table 7.

Table 7 Sensor topology calculation.

Timestamp t	α_t	Sensor Ranking (SR_t)				Γ	Weighted Sensor Ranking (WSR_t)		
		2	3	4	2		3	4	
0	0					0	0	0	
1	0					0	0	0	
2	2	1	2		B	2	1	0	
3	1		$\frac{2+2}{2}$		B,O	0	0.5	0	
4	2	1	2		B	2	1	0	
5	3	1	2	2	B	3	1.5	1.5	
6	2		2	2	O	0	1	1	
						Σ	7	5	2.5
						#	3	5	3
						$\Sigma/\#$	2.30	1.00	0.83

For each timestamp t the base importance α_t is set as the number of joint changes, as can be seen in Table 6, Step (4). The sensor ranking SR_t at timestamp t is taken from Table 5, Step (3). As sensors can occur in multiple sensor groups, and can have a different importance in each group, the closest matching group for a combination of sensors is used as the source for the ranking. The sensor group id (GID) is denoted in column Γ . For example, for timestamp 2, Table 7 shows $\Gamma = B$, as sensors **2** and **3** only occur together in group B. For timestamp 3, with only sensor **3** changing, things are more difficult. Sensor **3** occurs in groups B and O. Thus the weights for

both groups have to be averaged (i.e., $\frac{2+1}{2}$). For our naive calculation, the weighted sensor ranking (WSR_t) for each timestamp t then amounts to:

$$WSR_t = \alpha_t \times \frac{1}{SR_t} \quad (1)$$

If a sensor ranking does not exist for a certain timestamp, its value becomes 0. To calculate the importance of a sensor in the topology, the sum (Σ) of all rankings per sensor is divided by the number of occurrences ($\#$) of a sensor in all timestamps. This results in the final topology (i.e., sorting sensors in a list by importance) as shown in Table 8.

Table 8 Naive Sensor Topology

Sensor ID	Importance	Timestamps
2	2.30	2, 4, 5
4	1.00	5, 6
3	0.83	2, 3, 4, 5, 6

The sensor topology calculation is naive, as similarly to Step (4), future research into more complex scenarios will determine if alternative calculation algorithms (e.g., considering aspects of the source data) can yield better results.

5.6 Event Abstraction

5.6.1 Concept:

Steps (1)–(5) served as event extraction and preparation of event abstraction. In Step (6) process-level events are abstracted from the information encoded in the topology calculated in Step (5) [19]. Due to their higher ranking and thus higher significance to indicate process-level events as explained in the previous steps, we assume that events from sensors at the top of the topology (list) represent process events related to process activities, which group all events from lower levels of the topology.

Just utilizing the top layer (sensors at the top position) of the topology will yield a coarse-grained process model to be mined in Step (8), which groups all existing events. By letting the domain expert specify that the first, second, or even deeper levels, and only a subset of sensors at a particular level should be utilized to determine process activities and thus grouping of events, the process models can be manually tweaked to yield the desired granularity.

5.6.2 Challenges:

Here we address the same challenges **C3**, **C4** and **C5** as in the previous three steps by supporting the domain expert in selecting relevant sensors for analysis and event abstraction and automatically extracting the events based on the selection. **C1** can be addressed through having in-depth knowledge about the relationship between different sensors, leading to potential optimization regarding the required data velocity, but also relevance. In other words, data can potentially be discarded because it might not contribute to the data analysis tasks at hand. This step also allows us to control the granularity level of a process, thus for challenge **C6** allowing to stipulate a desired granularity of how human involvement in the process model must be represented. Challenges **C7** and **C8** are directly addressed by the development of this framework, and realizing Steps ① to ⑥. Generating knowledge and refining knowledge in a structured and semi-automatic way with the goal to refine and evolve processes is the purpose of the proposed framework. For example when the desired granularity of resulting processes is not fine enough, although we utilize the full topology generated in Step ⑤, we can deduct that we are missing sensors. On the other hand, the desired granularity of processes can be controlled by the mechanism introduced in Step ⑥.

5.6.3 Example:

Beginning at the top of the topology shown in Table 8, timestamps 2, 4, and 5 are identified to be points in time when process events occurred.

For our specific sensor subset **2**, **3**, **4** and sensor events for the first six timestamps (cf. Table 6), this yields process events when the light barrier shows changes. These events can be extracted as shown in Table 9.

Table 9 Extracted process events.

Event timestamp	Context
2	Group Buffer; Sensors 2 , 3
4	Group Buffer; Sensors 2 , 3
5	Group Buffer; Sensors 2 , 3 , 4





As can be seen, interpreting the context might require some further insights. When comparing to the large dependency deduction picture shown in Fig. 6(b), it becomes clear that the three potential events can be interpreted as *pickup*  events, but the lifecycle transitions associated with the process events are unclear. Including sensor **1** could have helped with further differentiation, but it was not included in the sample calculation in Steps ④ and ⑤. Nonetheless, the interpretation (i.e., by a human domain expert) shown in Table 10 seems feasible.

Table 10 Abstracted process events.

Process event ID	Event timestamp	Lifecycle transition	Label
 b1	2	start	grab part
 b2	4	unknown	grab part
 b3	5	complete	grab part

It is important to note that the dependency deduction in Step ④ shows that for timestamp 5 many more sensors are involved. Thus it is a stronger candidate for a process event. Also the semantic annotation in Step ① will provide further semantic information about the state of the involved sensors and thus further context for labeling of events and determining the process event lifecycle transition they represent.

Again, future research can show how all the structured information prepared in Steps ① to ⑥ can be incorporated in to either (1) a user interface to support domain experts, or (2) allow for automatic deduction of process events.

5.7 Event Correlation

5.7.1 Concept:

In Step ⑦ the extracted and abstracted events that were associated with specific process-level activities in Step ⑥ are correlated with a specific *case* (i.e., an executed process instance) to enable classic activity-centric process mining approaches [19]. This step may not be trivial due to the start and end of a case not being clearly identifiable from the IoT data log, and multiple executions of process instances in parallel being possible. Thus, ambiguities may emerge when trying to correlate abstracted events from Step ⑥ with a specific case [29, 50]. Additional sensors being annotated to provide data about the start and end of a case, or to track the execution of all activities belonging to one case (e.g., a camera [55]) might be necessary to support the correlation. The domain expert has to identify these sensors and specify their correlation with a case, e.g., what is the *object* that the execution of (a) process instance(s) is centered around. This could for instance be the *workpiece* in a production process or the *patient* in a treatment process, which would group all events into an executed case or an object-centric event log [6]. Moreover, context data from additional sensors and identified preceding process activities could be used to support the event correlation [12, 18].

5.7.2 Challenges:

In this step we address challenges **C3–C5**, **C7**, and **C8** as process-related data is added here to complement and complete the available information from previous steps. By correlating the extracted and abstracted events with a specific case or object, we achieve the full contextualization of these events within the specific process scenarios.

5.7.3 Example:

The domain expert performs the event correlation, i.e., adding the corresponding case identifiers (*Case IDs*) as additional attributes to the activities, by identifying all extracted events as pertaining to the production of the same workpiece and thus belonging to the same process instance (i.e., case). This step is straight-forward under the assumption that no parallel process executions are possible and that all executed production activities relate to the same workpiece (i.e., lot size 1) [73]. The result of this step is similar to the excerpt of an IoT-enriched event log depicted in Table 2 where all IoT sensor readings are fully contextualized in the specific process and activity instances.

5.8 Process Mining

5.8.1 Concept:

In Step ⑧ process mining techniques can be applied as soon as process activities are identified, which is achieved through *Event Abstraction* with a domain-expert providing a mapping for meaningful labels of the abstracted activities in Step ⑥ and the correlation of these activities with specific cases in Step ⑦. Case-based, artifact-based, object-centric, or other process mining techniques can be utilized to mine the processes from the event logs—to discover process models, check conformance, and enhance processes [92]—and thus to achieve full process awareness of IoT data combined with process-related artifacts (e.g., process models).

5.8.2 Challenges:

Challenge **C9** dealing with outcome prediction builds on all the data and relationships generated in Steps ① to ⑦. However, only if the resulting processes are known, potential outcomes can be identified, as they rely on the structure of the process and might only be realized at the very end of a process.

5.8.3 Example:

Applying the Inductive Miner [49] on the event log shown excerpt in Table 10—extended with additional entries from all sensors shown in Fig. 6(a)—but only considering the events with “start” lifecycle, the final process model could look as depicted in Fig. 7. This depends on the process granularity chosen in Step ⑥.

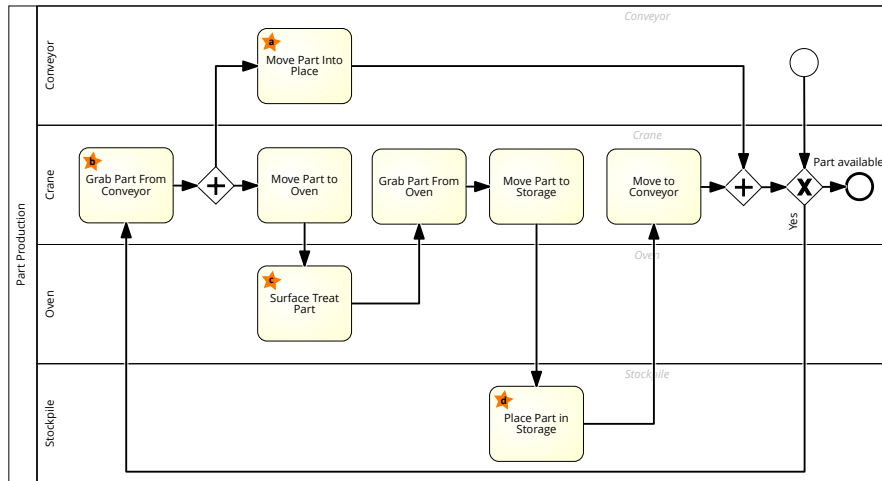


Fig. 7 Final process model of the production process in BPMN 2.0.

5.9 Discussion & Weaknesses

As implied throughout this chapter, the successful application of this framework relies on a variety of future advances. We will discuss potential problems that have to be solved for each step in the proposed framework.

If during Step ① neither domain experts, nor potential AI based annotation mechanisms can provide further insight into the characteristics and meaning of sensor values, it will be very difficult to identify potential process events during Step ⑥. The same holds for Step ②, although the physical location of sensors, combined with knowledge about the purpose of sensors might allow for automatic grouping.

Automatic discretization will have to be explored and can help with reducing human involvement while conducting Step ①.

We expect Steps ③ to ⑤ to run fully automatic without further human input, but the algorithms presented as part of this chapter need to be refined. Similar to the

multitude of process mining algorithms that exist today, for Steps ③ to ⑤ we can see multiple refined algorithms dealing with potential shortcomings of topologies.

The abstraction and correlation in Steps ⑥ and ⑦ will potentially always be conducted by domain experts as they allow fine-tuning the granularity of the resulting process model(s).

One corner case that has to be mentioned is when the framework is applied to a scenario with only one sensor. Here Step ① is the only step that affects the outcome. If the sensor delivers continuous data, the discretization in Step ① will determine all process events, as the topology in Step ⑤ will contain only this sensor, and all sensor events will be considered to be process events.

Regarding the applicability of the framework in other domains (e.g., healthcare) we do not foresee any problems. As the approach is very basic, depending on just time series data (i.e., timestamps and associated values) from sensors, nothing prevents the application of the approach to data from other domains. For example the approach for activity monitoring from IoT data in healthcare settings described in [28] exhibits the same characteristics within sensor data relevant for the framework presented in this work, namely time series data and sensor groups.

High volume sensor data will lead to a set of performance problems with the following steps being the most affected:

- Step ① will suffer from high computational cost for discretization, depending on the algorithm for automatic discretization, or the effort needed to transform data.
- Step ④ will suffer from performance problems, if a high number of sensors is included in the data, as all sensors have to be compared. The calculation of dependencies for different time windows further multiplies the computational costs.
- All other Steps are considered to not be problematic.

Furthermore, the development of user interfaces to support domain experts during Steps ⑥ and ⑦ (and potentially Step ①) will be crucial to maximize the utility of the framework, as well as the deduction of process events from IoT events.

6 Conclusion and Future Work

Data engineers, data scientists, and domain experts have to be supported in collecting, structuring, and analyzing IoT data to extract process models and event logs in order to (1) allow potential enhancements to materialize faster, (2) minimize the errors occurring during the analysis, and (3) minimize the required domain knowledge.

In the context of these goals, we identified a set of 10 current research challenges that can be associated with deriving process-relevant data from complex IoT data. To support the goals and address these challenges, we presented a framework consisting of 8 generalized steps that can guide the semi-automated data analysis to produce

process data and process models at different levels of granularity starting from raw IoT data (*no process awareness*) up to IoT data which is fully contextualized regarding a process (*full process awareness*). We envision future research to yield a wide variety of results for each step of the framework, including:

- Specialized tools for domain experts to annotate and group data artifacts.
- Custom visualizations to show dependencies between data artifacts in real-time.
- Refined algorithms to calculate (hidden) dependencies between hardware artifacts and sensor data.
- Refined data retention policies based on annotations and derived process intelligence.

To demonstrate the potential of the framework we also introduced a sample scenario and exemplary IoT dataset to discuss how a semi-automatically generated process model based on the steps in the framework can be (a) of similar quality as a process model provided by domain experts, (b) used to point out problems in existing process models, and (c) utilized to refine existing processes, e.g., by pointing out the necessity to introduce additional sensors in a certain context. We think that this framework can serve as means to focus research towards the overall goal of allowing domain experts to make IoT datasets fully process aware without the involvement of data engineers and data scientists [22].

Acknowledgments

This research was partly funded by the Austrian Research Promotion Agency (FFG) via the “Austrian Competence Center for Digital Production” (CDP) under the contract number 881843. This research was supported by the Pilot Factory Industry 4.0, Seestadtstrasse 27, Vienna, Austria. This work has received funding from the Swiss National Science Foundation under Grant No. IZSTZ0_208497 (*ProAmbition* project).

References

1. Ahmad, M.: Reliability models for the internet of things: A paradigm shift. In: 2014 IEEE International Symposium on Software Reliability Engineering Workshops. pp. 52–59 (2014)
2. Akhramovich, K., Serral, E., Cetina, C.: A systematic literature review on the application of process mining to industry 4.0. *Knowledge and Information Systems* pp. 1–48 (2024)
3. Baier, T., Mendling, J., Weske, M.: Bridging abstraction layers in process mining. *Information Systems* **46**, 123 – 139 (2014)
4. Bauer, M., Bui, N., De Loof, J., Magerkurth, C., Nettsträter, A., Stefa, J., Walewski, J.W.: Iot reference model. Enabling things to talk: designing IoT solutions with the IoT architectural reference model pp. 113–162 (2013)
5. Beerepoot, I., et al.: The biggest business process management problems to solve before we die. *Computers in Industry* **146**, 103837 (2023)

6. Berti, A., Koren, I., Adams, J.N., Park, G., Knopp, B., Graves, N., Rafiei, M., Liß, L., Unterberg, L.T.G., Zhang, Y., et al.: Ocel (object-centric event log) 2.0 specification. arXiv preprint arXiv:2403.01975 (2024)
7. Bertrand, Y., De Weerd, J., Serral, E.: A bridging model for process mining and iot. In: Munoz-Gama, J., Lu, X. (eds.) *Process Mining Workshops*. pp. 98–110. Springer International Publishing, Cham (2022)
8. Bertrand, Y., Weerd, J.D., Serral, E.: Assessing the suitability of traditional event log standards for IoT-enhanced event logs. In: *Business Process Management Workshops*. Springer (2022)
9. Beverungen, D., Buijs, J.C.A.M., Becker, J., Di Ciccio, C., van der Aalst, W.M.P., et al.: *Seven Paradoxes of Business Process Management in a Hyper-Connected World*. BISE (2020)
10. Beyel, H.H., Makke, O., Gusikhin, O., van der Aalst, W.M.: Analyzing behavior in cyber-physical systems in connected vehicles: A case study. In: *International Conference on Business Process Management*. pp. 92–104. Springer (2023)
11. Brauner, P., Dalibor, M., Jarke, M., Kunze, I., Koren, I., Lakemeyer, G., et al.: A Computer Science Perspective on Digital Transformation in Production. *ACM Trans. Internet Things* **3**(2), 1–32 (2022)
12. Brzychczy, E., Zuber, A., van der Aalst, W.: Process mining of mining processes: Analyzing longwall coal excavation using event data. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* (2024)
13. Burattin, A., Eigenmann, M., Seiger, R., Weber, B.: Mqtt-xes: Real-time telemetry for process event data. *BPM 2020 Best Dissertation Award, Doctoral Consortium, and Demonstration & Resources Track* pp. 97–101 (2020)
14. Cameranesi, M., Diamantini, C., Mircoli, A., Potena, D., Storti, E.: Extraction of user daily behavior from home sensors through process discovery. *IEEE Internet of Things Journal* **7**(9), 8440–8450 (2020)
15. Chanal, P.M., Kakkasageri, M.S.: Security and privacy in iot: a survey. *Wireless Personal Communications* **115**(2), 1667–1693 (2020)
16. Cook, A.A., Misrlı, G., Fan, Z.: Anomaly detection for iot time-series data: A survey. *IEEE Internet of Things Journal* **7**(7), 6481–6494 (2020)
17. De Luzi, F., Leotta, F., Marrella, A., Mecella, M.: On the interplay between business process management and internet-of-things: A systematic literature review. *Business & Information Systems Engineering* pp. 1–24 (2024)
18. Di Federico, G., Burattin, A.: Cvamos–event abstraction using contextual information. *Future Internet* **15**(3) (2023)
19. Diba, K., Batoulis, K., Weidlich, M., Weske, M.: Extraction, correlation, and abstraction of event data for process mining. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* **10**(3), e1346 (2020)
20. Dumas, M., La Rosa, M., Mendling, J., Reijers, H.A., et al.: *Fundamentals of business process management*, vol. 1. Springer (2013)
21. van Eck, M.L., Sidorova, N., van der Aalst, W.M.P.: Enabling process mining on sensor data from smart products. In: *2016 IEEE Tenth Int. Conf. on Research Challenges in Information Science (RCIS)*. pp. 1–12 (2016)
22. Ehrendorfer, M., Bertrand, Y., Malburg, L., Mangler, J., Grüger, J., Rinderle-Ma, S., Bergmann, R., Serral, E.: Internet of processes and things: A repository for iot-enriched event logs in smart environments. In: *International Conference on Business Process Management, Demonstration & Resources Forum*. pp. 92–96. CEUR (2023)
23. Ehrendorfer, M., Fassmann, J.A., Mangler, J., Rinderle-Ma, S.: Conformance checking and classification of manufacturing log data. In: *21st CBI*. vol. 1, pp. 569–577. IEEE (2019)
24. Ehrendorfer, M., Mangler, J., Rinderle-Ma, S.: Sensor data stream selection and aggregation for the ex post discovery of impact factors on process outcomes. In: *33rd CAiSE*. pp. 29–37. Springer (2021)
25. Ehrendorfer, M., Mangler, J., Rinderle-Ma, S.: Clustering raw sensor data in process logs to detect data streams. In: *International Conference on Cooperative Information Systems*. pp. 438–447. Springer (2023)

26. Elsaleh, T., Bermudez-Edo, M., Enshaeifar, S., Acton, S.T., Rezvani, R., Barnaghi, P.: IoT-Stream: A Lightweight Ontology for Internet of Things Data Streams. In: Global IoT Summit Proc. pp. 1–6. IEEE (2019)
27. Folino, F., Guarascio, M., Pontieri, L.: Mining predictive process models out of low-level multidimensional logs. In: Int. Conf. on advanced information systems engineering. pp. 533–547. Springer (2014)
28. Franceschetti, M., Seiger, R., González, M.J.G., Garcia-Ceja, E., Flores, L.A.R., García-Bañuelos, L., Weber, B.: Proambition: Online process conformance checking with ambiguities driven by the internet of things. In: CAiSE Research Projects Exhibition. pp. 52–59 (2023)
29. Franceschetti, M., Seiger, R., López, H.A., Burattin, A., García-Bañuelos, L., Weber, B.: A characterisation of ambiguity in bpm. In: International Conference on Conceptual Modeling. pp. 277–295. Springer (2023)
30. Franceschetti, M., Seiger, R., Weber, B.: An event-centric metamodel for iot-driven process monitoring and conformance checking. In: International Conference on Business Process Management. pp. 131–143. Springer (2023)
31. Gruhn, V., Han, Y., Hesenius, M., Reichert, M., Wang, G., Yu, J., Zhang, L.: BRIBOT: Towards a Service-Based Methodology for Bridging Business Processes and IoT Big Data. In: Service-Oriented Computing, vol. 13121, pp. 597–611. Springer (2021)
32. Grüger, J., Malburg, L., Mangler, J., Bertrand, Y., Rinderle-Ma, S., Bergmann, R., Asensio, E.S.: Sensorstream: An xes extension for enriching event logs with iot-sensor data (2022), <https://arxiv.org/abs/2206.11392>
33. Hirmer, P., Wieland, M., Breitenbücher, U., Mitschang, B.: Automated sensor registration, binding and sensor data provisioning. In: CAiSE Forum. pp. 81–88 (2016)
34. Hirmer, P., Wieland, M., Schwarz, H., Mitschang, B., Breitenbücher, U., Leymann, F.: Sitr-a situation recognition service based on modeling and executing situation templates. In: Proceedings of the 9th symposium and summer school on service-oriented computing. pp. 113–127 (2015)
35. Hoffmann, M., Malburg, L., Bergmann, R.: ProGAN: Toward a Framework for Process Monitoring and Flexibility by Change via Generative Adversarial Networks. In: BPM Workshops, LNBIP, vol. 436, pp. 43–55. Springer (2022)
36. Hoppenstedt, B., Pryss, R., Stelzer, B., Meyer-Brötz, F., Kammerer, K., Treß, A., Reichert, M.: Techniques and emerging trends for state of the art equipment maintenance systems—a bibliometric analysis. *Applied Sciences* **8**(6), 916 (2018)
37. Janiesch, C., Koschmider, A., Mecella, M., Weber, B., Burattin, A., Di Ciccio, C., Fortino, G., Gal, A., Kannengiesser, U., Leotta, F., et al.: The internet of things meets business process management: a manifesto. *IEEE Syst Man Cybern Mag* **6**(4), 34–44 (2020)
38. Jans, M., Soffer, P., Jouck, T.: Building a valuable event log for process mining: an experimental exploration of a guided process. *Ent. Inf. Syst.* **13**(5), 601–630 (2019)
39. Janssen, D., Mannhardt, F., Koschmider, A., van Zelst, S.J.: Process model discovery from sensor event data. In: Process Mining Workshops. pp. 69–81 (2021)
40. Kammerer, K., Pryss, R., Hoppenstedt, B., Sommer, K., Reichert, M.: Process-driven and flow-based processing of industrial sensor data. *Sensors* **20**(18), 5245 (2020)
41. Kirikkayis, Y., Gallik, F., Reichert, M.: Iotdm4bpmn: An iot-enhanced decision making framework for bpmn 2.0. In: 2022 International Conference on Service Science (ICSS). pp. 88–95 (2022)
42. Kirikkayis, Y., Gallik, F., Seiger, R., Reichert, M.: Integrating iot-driven events into business processes. In: International Conference on Advanced Information Systems Engineering. pp. 86–94. Springer (2023)
43. Kirikkayis, Y., Gallik, F., Winter, M., Reichert, M.: Bpmne4iot: A framework for modeling, executing and monitoring iot-driven processes. *Future Internet* **15**(3) (2023)
44. Klein, P., Malburg, L., Bergmann, R.: Ftonto: A domain ontology for a fischertechnik simulation production factory by reusing existing ontologies. In: Lernen. Wissen. Daten. Analysen. (LWDA), Berlin, Germany. pp. 253–264. CEUR Workshops Proceedings (2019)

45. Knoch, S., Ponpathirkoottam, S., Schwartz, T.: Video-to-Model: Unsupervised Trace Extraction from Videos for Process Discovery and Conformance Checking in Manual Assembly. In: 18th BPM, LNCS, vol. 12168, pp. 291–308. Springer (2020)
46. Kopetz, H., Bondavalli, A., Brancati, F., Frömel, B., Höftberger, O., Jacob, S.: Emergence in Cyber-Physical Systems-of-Systems (CPSoSs), pp. 73–96. Springer International Publishing, Cham (2016)
47. Koschmider, A., Janssen, D., Mannhardt, F.: Framework for process discovery from sensor data. In: EMISA. pp. 32–38 (2020)
48. Küfner, T., Schöning, S., Jasinski, R., Ermer, A.: Vertical data continuity with lean edge analytics for industry 4.0 production. *Computers in Industry* **125**, 103389 (2021)
49. Leemans, S.J.: Robust Process Mining with Guarantees: Process Discovery, Conformance Checking and Enhancement, vol. 440. Springer Nature (2022)
50. Li, C.Y., Joshi, A., Tam, N.T., Lau, S.S.F., Huang, J., Shinde, T., van der Aalst, W.M.: Rectify sensor data in iot: A case study on enabling process mining for logistic process in an air cargo terminal. In: International Conference on Cooperative Information Systems. pp. 293–310. Springer (2023)
51. Maamar, Z., Kajan, E., Guidara, I., Moctar-M’ Baba, L., Sellami, M.: Bridging the gap between business processes and IoT. In: 24th Symposium on Int. Database Eng. & Appl. pp. 1–10. ACM, New York, USA (2020)
52. Malburg, L., Bergmann, R.: Towards Adaptive Workflow Management by Case-Based Reasoning and Automated Planning. In: 30th ICCBR Workshops Proceedings. CEUR-WS.org (2022), accepted for Publication.
53. Malburg, L., Grüger, J., Bergmann, R.: An IoT-Enriched Event Log for Process Mining in Smart Factories. *CoRR* **abs/2209.02702** (2022)
54. Malburg, L., Klein, P., Bergmann, R.: Semantic web services for ai-research with physical factory simulation models in industry 4.0. In: Proceedings of the International Conference on Innovative Intelligent Industrial Production and Logistics - IN4PL. pp. 32–43. INSTICC, SciTePress (2020)
55. Malburg, L., Rieder, M.P., Seiger, R., Klein, P., Bergmann, R.: Object Detection for Smart Factory Processes by Machine Learning. *Procedia Comput. Sci.* **184**, 581–588 (2021)
56. Malburg, L., Seiger, R., Bergmann, R., Weber, B.: Using physical factory simulation models for business process management research. In: Business Process Management Workshops. pp. 95–107. Springer (2020)
57. Mangler, J., Grüger, J., Malburg, L., Ehrendorfer, M., Bertrand, Y., Benzin, J.V., Rinderle-Ma, S., Serral Asensio, E., Bergmann, R.: Datastream xes extension: Embedding iot sensor data into extensible event stream logs. *Future Internet* **15**(3) (2023)
58. Mangler, J., Rinderle-Ma, S.: CPEE - Cloud Process Execution Engine. *BPM Demo Sessions* **1295** (2014)
59. Mangler, J., Pauker, F., Rinderle-Ma, S., Ehrendorfer, M.: centurio.work - industry 4.0 integration assessment and evolution. In: 17th BPM Conf. pp. 106–117 (2019)
60. Mannhardt, F., Bovo, R., Oliveira, M.F., Julier, S.: A taxonomy for combining activity recognition and process discovery in industrial environments. In: Intelligent Data Engineering and Automated Learning (IDEAL 2018), Madrid, Spain. LNCS, vol. 11315, pp. 84–93. Springer International Publishing, Cham (Nov 2018)
61. Mannhardt, F., de Leoni, M., Reijers, H.A., van der Aalst, W.M.P., Toussaint, P.J.: From low-level events to activities - a pattern-based approach. In: International Conference on Business Process Management (BPM), Rio de Janeiro, Brazil. LNCS, vol. 9850, pp. 125–141. Springer International Publishing, Cham (Sep 2016)
62. Marrella, A., Mecella, M., Sardiña, S.: Supporting adaptiveness of cyber-physical processes through action-based formalisms. *AI Commun.* **31**(1), 47–74 (2018)
63. Messner, M., Pauker, F., Mauthner, G., Frühwirth, T., Mangler, J.: Closed loop cycle time feedback to optimize high-mix/low-volume production planning. *Procedia CIRP* **81**, 689–694 (2019)
64. Monostori, L.: Cyber-physical production systems: Roots, expectations and r&d challenges. *Procedia Cirp* **17**, 9–13 (2014)

65. Mottola, L., Picco, G.P., Oppermann, F.J., Eriksson, J., Finne, N., Fuchs, H., Gaglione, A., Karnouskos, S., Montero, P.M., Oertel, N., Römer, K., Spieß, P., Tranquillini, S., Voigt, T.: makesense: Simplifying the integration of wireless sensor networks into business processes. *IEEE Trans. Software Eng.* **45**(6), 576–596 (2019)
66. Müller, M.: Dynamic time warping. *Information retrieval for music and motion* pp. 69–84 (2007)
67. Pauker, F., Mangler, J., Rinderle-Ma, S., Ehrendorfer, M.: Industry 4.0 Integration Assessment and Evolution at EVVA GmbH: Process-Driven Automation Through *centurio*. work. In: *BPM Cases Vol. 2*, pp. 81–91. Springer (2021)
68. Pauker, F., Mangler, J., Rinderle-Ma, S., Pollak, C.: *centurio.work* - modular secure manufacturing orchestration. In: *16th International Conference on Business Process Management*. pp. 164–171 (2018)
69. Rebmann, A., Emrich, A., Fettke, P.: Enabling the discovery of manual processes using a multi-modal activity recognition approach. In: *Business Process Management Workshops. LNBP*, vol. 362, pp. 130–141. Springer (Jan 2020)
70. Rinderle-Ma, S., Mangler, J.: Process automation and process mining in manufacturing. In: Polyvyanyy, A., Wynn, M.T., Van Looy, A., Reichert, M. (eds.) *Business Process Management*. pp. 3–14. Springer International Publishing, Cham (2021)
71. Seiger, R., Assmann, U., Huber, S.: A case study for workflow-based automation in the internet of things. In: *2018 IEEE International Conference on Software Architecture Companion (ICSA-C)*. pp. 11–18 (2018)
72. Seiger, R., Franceschetti, M., Weber, B.: Data-driven generation of services for iot-based online activity detection. In: *International Conference on Service-Oriented Computing*. pp. 186–194. Springer (2023)
73. Seiger, R., Franceschetti, M., Weber, B.: An interactive method for detection of process activity executions from iot data. *Future Internet* **15**(2) (2023)
74. Seiger, R., Huber, S., Heisig, P., Aßmann, U.: Toward a framework for self-adaptive workflows in cyber-physical systems. *Software & Systems Modeling* **18**(2), 1117–1134 (2019)
75. Seiger, R., Malburg, L., Weber, B., Bergmann, R.: Integrating process management and event processing in smart factories: A systems architecture and use cases. *J. Manuf. Syst.* **63**, 575–592 (2022)
76. Seiger, R., Zerbato, F., Burattin, A., García-Bañuelos, L., Weber, B.: Towards iot-driven process event log generation for conformance checking in smart factories. In: *24th Intern. EDOC Workshop*. pp. 20–26. IEEE (2020)
77. Senderovich, A., Rogge-Solti, A., Gal, A., Mendling, J., Mandelbaum, A.: The road from sensor data to process instances via interaction mining. In: *International Conference on Advanced Information Systems Engineering (CAiSE)*, Stockholm, Sweden. LNCS, vol. 9097, pp. 257–273. Springer International Publishing, Cham (June 2016)
78. Shi, F., Li, Q., Zhu, T., Ning, H.: A survey of data semantization in internet of things. *Sensors* **18**(1) (2018)
79. Sisinni, E., Saifullah, A., Han, S., Jennehag, U., Gidlund, M.: Industrial Internet of Things: Challenges, Opportunities, and Directions. *IEEE Trans. Ind. Inf.* **14**(11), 4724–4734 (2018)
80. Soffer, P., Hinze, A., Koschmider, A., Ziekow, H., Di Ciccio, C., Koldehofe, B., Kopp, O., Jacobsen, A., Sürmeli, J., Song, W.: From event streams to process models and back: Challenges and opportunities. *Inf. Sys.* **81**, 181–200 (2019)
81. Stertz, F., Mangler, J., Rinderle-Ma, S.: Data-driven improvement of online conformance checking. In: *24th EDOC*. pp. 187–196. IEEE (2020)
82. Stertz, F., Mangler, J., Rinderle-Ma, S.: Balancing patient care and paperwork automatic task enactment and comprehensive documentation in treatment processes. In: *Enterprise Modelling and Information Systems Architectures*. vol. 15, p. 11 (2020)
83. Stertz, F., Mangler, J., Rinderle-Ma, S.: Temporal Conformance Checking at Runtime based on Time-infused Process Models. [arXiv:2008.07262](https://arxiv.org/abs/2008.07262) (2020)
84. Stertz, F., Rinderle-Ma, S., Mangler, J.: Analyzing process concept drifts based on sensor event streams during runtime. In: Fahland, D., Ghidini, C., Becker, J., Dumas, M. (eds.) *Business Process Management*. pp. 202–219. Springer International Publishing, Cham (2020)

85. Stoiber, C., Schöning, S.: Event-Driven Business Process Management Enhancing IoT – A Systematic Literature Review and Development of Research Agenda. In: *Innovation Through Information Systems*, vol. 48, pp. 645–661. Springer (2021)
86. Sun, Y., Li, J., Liu, J., Sun, B., Chow, C.: An improvement of symbolic aggregate approximation distance measure for time series. *Neurocomputing* **138**, 189–198 (2014)
87. Szttyler, T., Carmona, J., Völker, J., Stuckenschmidt, H.: *Self-tracking Reloaded: Applying Process Mining to Personalized Health Care from Labeled Sensor Data*, pp. 160–180. Springer, Berlin, Heidelberg (2016)
88. Tax, N., Sidorova, N., Haakma, R., van der Aalst, W.M.: Event abstraction for process mining using supervised learning techniques. In: *Proceedings of SAI Intelligent Systems Conference*. pp. 251–269. Springer (2016)
89. Traganos, K., Spijkers, D., Grefen, P., Vanderfeesten, I.: Dynamic process synchronization using bpmn 2.0 to support buffering and (un)bundling in manufacturing. In: Fahland, D., Ghidini, C., Becker, J., Dumas, M. (eds.) *Business Process Management Forum*. pp. 18–34. Springer International Publishing, Cham (2020)
90. Valderas, P., Torres, V., Serral, E.: *Towards an Interdisciplinary Development of IoT-Enhanced Business Processes*. BISE (2022)
91. Van Der Aa, H., Leopold, H., Reijers, H.A.: Efficient process conformance checking on the basis of uncertain event-to-activity mappings. *IEEE Trans. on Knowledge and Data Engineering* **32**(5), 927–940 (2019)
92. Van Der Aalst, W., Adriansyah, A., De Medeiros, A.K.A., Arcieri, F., Baier, T., Blickle, T., Bose, J.C., van den Brand, P., Brandtjen, R., Buijs, J., et al.: *Process mining manifesto*. In: *Business Process Management*. pp. 169–194. Springer (2011)
93. Verbeek, E.: IEEE standard for extensible event stream (XES) for achieving interoperability in event logs and event streams. *IEEE Std 1849-2016* pp. 1–50 (Nov 2016)
94. Wanner, J., Herm, L.V., Janiesch, C.: Countering the fear of black-boxed ai in maintenance: Towards a smart colleague. In: *Proceedings of the 2019 Pre-ICIS SIGDSA Symposium* (2019)
95. Weyers, F., Seiger, R., Weber, B.: Method to identify process activities by visualizing sensor events. In: *International Conference on Business Process Management*. pp. 76–88. Springer (2022)
96. van Zelst, S.J., Mannhardt, F., de Leoni, M., Koschmider, A.: Event abstraction in process mining: literature review and taxonomy. *Granular Computing* **6**(3), 719–736 (2021)
97. Zerbato, F., Seiger, R., Di Federico, G., Burattin, A., Weber, B.: Granularity in process mining: Can we fix it? In: *International Workshop on BPM Problems to Solve Before We Die*. pp. 40–44. CEUR-WS (2021)