Assessing the suitability of traditional event log standards for IoT-enhanced event logs*

Yannis Bertrand[1 [0000−0002−6407−7221]], Jochen De Weerdt[1 [0000−0001−6151−0504]], and Estefanía Serral[1 [0000−0001−7579−910X]]

Research Centre for Information Systems Engineering (LIRIS), KU Leuven
Warmoesberg 26, 1000 Brussels, Belgium
{firstname.lastname}@kuleuven.be

Abstract. Since IoT devices supporting business processes (BPs) in sectors like manufacturing, logistics or healthcare can collect data on the execution of the processes, there is a growing awareness of the opportunity to use IoT data for process mining (PM). However, several challenges need to be addressed to enable IoT-enhanced PM, among which the need for a standard IoT-enhanced event log model. In this paper, we identify four main challenges for the integration of IoT data in event logs, from which 10 requirements for an IoT-enhanced event log model are derived. We then analyse the two main current event log models, the extensible event stream (XES) and the object-centric event log (OCEL), and confront them with the requirements, showing that both present important limitations to storing IoT-enhanced event logs. Based on this analysis, we conclude that a comprehensive data model is required to develop more advanced IoT-enhanced PM techniques.

Keywords: Process Mining · IoT · IoT-enhanced event log

1 Introduction

As IoT devices, i.e., sensors and actuators, are more widely used to support the execution of business processes (BPs) , there is a growing awareness of the opportunity to use the data collected by these devices for process mining (PM). Currently available PM methods capable of incorporating IoT data are consistent in terms of strategy: IoT data are preprocessed in such a way that they can be integrated in a classical event log format. This necessitates abstraction techniques, either data- or expert-driven. Although this is a good first step in including IoT data in PM, given the fact that it allows for application of the existing set of control-flow and data-aware methods, this approach does not use IoT data to their full potential. Most of the time, the derived high-level event log omits context information (i.e., properties that can influence the execution of the process, see [29]) that could be derived from the IoT data, or is limited in the extent to which context information can be incorporated. Moreover,

* This research was supported by Flanders Research Fund (FWO) in the scope of project number G0B6922N.

by separating the abstraction phase from the discovery/analysis phase, the real potential of advanced algorithms to jointly optimise abstraction and model discovery, cannot be exploited. For instance, developing an IoT-enhanced decision mining algorithm would require direct access to lower-level IoT data in order to learn the most relevant features directly on the source data, instead of an error-prone and lossy event abstraction step.

This observation relates strongly to limitations of the most commonly used event log standards, i.e. the extensible event stream (XES, see [13]) and the object-centric event log (OCEL, see [12]). Accordingly, in this paper, we present four concrete contributions to further research on IoT-enhanced process mining. First, we motivate the problem by describing possible new techniques that would use IoT data to their full potential and demonstrate the impossibility to apply them to a typical event log. Second, we put forward four specific challenges that pertain to the integration of IoT data in event logs for PM, and illustrate these challenges with a running example. Third, based on these challenges, we present 10 key requirements that a suitable model for IoT-enhanced event logs should fulfil. Finally, the fourth contribution of this work consists of an assessment of XES and OCEL in function of the identified requirements, to determine their suitability and shortcomings.

Thus, this paper is structured as follows. Section 2 provides the motivation. Sections 3 discusses related work, before the running example is introduced in Section 4. The challenges and requirements are put forward in Sections 5 and 6, respectively. Section 7 analyses XES and OCELbefore concluding the paper in Section 8.

2 Motivation

Table 1 provides a high-level overview of methods and techniques in the PM field. The table is structured vertically along the three main process mining task types. In the horizontal dimension, the evolution from a classical control-flow perspective to data-aware and IoT-enhanced approaches is depicted. The second and third column cover the lion's share of process mining research so far. However, an important gap is still present in terms of developing IoT-enhanced process mining techniques. The most prominent existing stream of research relates to activity mining, i.e. approaches to derive higher level process events from low-level IoT data [10, 17, 21, 28, 34, 36]. Nonetheless, as illustrated by an only evocative set of so far unaddressed research opportunities, denoted in italics in Table 1, the widespread and grounded development of IoT-enhanced process mining techniques, covering all process mining tasks and possible applications, is far from realised yet. This is mainly due to the fact that the very source of the opportunities of IoT data is their fine granularity, which enables a deeper understanding of the process and the detection of finer patterns. As illustrated by a non-exhaustive set of examples in the table, there exist many potential new techniques that could be developed to take advantage of IoT data. For example:

- *IoT context-aware trace clustering*: cluster traces based on the values of IoT-derived context parameters.
- *IoT-driven root cause analysis*: delve into IoT measurements to discover the root cause of process deviations.
- *IoT-enhanced decision mining*: incorporate IoT-derived context parameters in decision mining.
- *IoT-enhanced resource mining*: use IoT data (e.g., location, tags) to mine the fine-grained behaviour of resources in a process.
- *IoT-enhanced predictive process monitoring*: use IoT sensor data measurements to facilitate real-time predictive process monitoring.

A common denominator of these envisioned IoT-enhanced PM techniques is the fact that simply abstracting fine grained IoT data into a classical XES or OCEL event log considerably limits analysis possibilities. Accordingly, this paper provides a foundation for the development of a new standard adapted to IoT-enhanced event logs, which is intended to give enough flexibility to develop new techniques taking advantage of the possibilities of IoT data.

Table 1: Positioning of IoT-enhanced techniques within the PM field, extending control-flow and data-aware approaches for all three main PM task types.

PM task types	**Control-flow**	**Data-aware**	**IoT-enhanced**
Discovery	•Control-flow discovery [1, 19, 37] •Trace clustering [31, 6]	•Data-aware process discovery [20, 23] •Multi-perspective trace clustering [14] •Multi-perspective process discovery [22]	•Activity mining [10, 17, 21, 28, 27, 34, 36] •*IoT context-aware trace clustering* •IoT-enhanced data-aware process discovery [2]
Conformance checking	•Control-flow conformance checking [9]	•Data-aware conformance checking [38]	•IoT-enhanced deviation detection [26] •*IoT-driven root cause analysis*
Extension		•Decision mining [3] •Resource mining [24] •Predictive process monitoring [8] •Performance analysis [5]	•*IoT-enhanced decision mining* •*IoT-enhanced resource mining* •*IoT-enhanced predictive process monitoring* •*Real-time predictive process monitoring*

To further substantiate our motivation, let us consider a lifelike example to illustrate the main limitations of current standards. This example consists in a smart distribution centre (DC), which receives products from warehouses, assesses their quality and distributes them to supermarkets (adapted from [35]).

In this process, several instances (e.g. shipments, crates) are processed in parallel, several decisions are made at different points, involving data from multiple sensors and following more complex rules than simple thresholds.

The process begins with the arrival of a container loaded with crates of products at the smart DC. The first activity is a manual check of the quality and freshness (based on, e.g., firmness, colour, damages) of a sample of the products. The results are entered in the system by the worker performing the check. After this, information about the product (e.g., name, harvest or production date) is retrieved by scanning the label (i.e., the QR code) of the shipment, together with data on the transport conditions recorded by sensors (e.g. temperature, humidity, shocks). Based on the worker's assessment and the evaluation of the overall transport conditions, each crate of products is judged proper for consumption or not. If it is, the crate is registered and stored in a suitably acclimatised refrigerated area to maximise product conservation. Otherwise, the crate is rejected and discarded.

After the first product quality check, a second one is performed over a sample of the products at the laboratory. If bacteria are detected, an alarm is triggered and the crate is discarded. Otherwise, the crate can be shipped. Depending on the quality of the products (as determined by the worker, the transport and storage conditions and the lab analysis), the crate will either be moved to the non-priority shipment area (if the products are excellent), or the crate will be set for priority shipment to be sold as fast as possible, at a discount (if the quality is not excellent). Finally, pallets are registered as shipped when they have left the DC to the supermarket.

It is known that the human evaluation of the freshness of the products is highly correlated with the transport conditions tracked by sensor data. We would like to discover a threshold value from the data (i.e., using IoT-enhanced decision mining), but without more domain knowledge, it is necessary to store the whole time series sensor data in the process event log to look for relationships (correlations) between temperature and humidity values and the decision. However, it is not possible to store both data types in a typical event log. Some sort of aggregation or abstraction needs to be applied to the time series sensor data first, which requires knowing the relationship between sensor data and the process flow beforehand; but that is not the case here, as this relationship is what we would like to mine.

3 Related Work

3.1 Existing standards for event logs

XES [13], the current standard event log model, is an XML-based model that mainly consists of the notions of event, case, and log. It proposes standard attribute types to contextualise the events, e.g. the resource executing an activity, the cost of an activity, etc. A standard activity lifecycle is defined together with XES, based on which the status of an activity could be mapped with events

relating to this activity. XES also allows the definition of new data attribute types through the notion of extensions, thereby increasing the flexibility of the model. Several implementations coexist, the main one being OpenXES [1], which is used by many event logs described in the literature.

Recently, the uptake of new technologies and the gain in maturity of the PM field have increased the urge to create alternative models. Multiple propositions that relax some assumptions of XES and allow for more flexibility in event data storage have been presented, e.g., in [25, 12]. Among them, the OCEL [12] was designed to be more suitable for storing event logs extracted from relational databases and is widely considered as the main challenger of XES today. It replaces the strict notion of case with the concept of object, which generalises it by allowing one event to be linked with multiple objects instead of a single case. This removes the necessity to "flatten" the event log by picking one case notion from the several potential case notions that often coexist in real-life processes. A second noticeable difference with XES is the explicit inclusion of the concept of activity in OCEL, which is absent in XES.

3.2 Process mining using IoT data

As mentioned in Section 2, the vast majority of the process mining literature involving IoT data has focused on mining high-level events of the process from low-level IoT data to create XES event logs (so-called *activity mining* in Table 1). Traditional process mining techniques can then be applied to these event logs to, e.g., discover control-flow models of the processes.

Trzcionkowska and Brzychczy describe a framework to create an event log from industrial IoT data in four steps: data preprocessing, clustering low-level data, classification to derive events from clusters and creation of the final event log [34]. The event log obtained is in XES format and contains no data attribute. Also focusing on industrial applications, Seiger et al. propose to transform raw IoT data into an XES event log using complex event processing (CEP) and event detection and refinement techniques [28]. They apply this approach to a smart manufacturing case to mine a production process in a follow-up paper [27]. In Valencia-Parra et al., a domain-specific language is developed to extract different XES event logs from IoT data by specifying the case and activity identifiers [36]. Koschmider et al. [18] propose a model to go from low-level events captured by sensors to instances of the process, by aggregating low-level events into high-level events using methods like CEP. In a further work, the authors showed a more systematic and complete framework [17], which assumes a sensor event log as input, and consists of three event log creation steps: (1) event correlation, (2) activity discovery, (3) event abstraction. This approach generates an XES event log, from which a process model can be mined, and was applied to a smart home scenario by Janssen et al. [16].

Although most of the existing literature is in activity mining, some of the other possible techniques have also been investigated. Banham et al. [2] proposes

[1] https://www.xes-standard.org/openxes/start

to perform data-aware process discovery with IoT-based attributes. A data Petri Net is discovered from two real-life event logs, and rules behind some decisions are mined based on IoT-derived attributes. The framework proposed requires abstracting the IoT data to integrate them in an XES event log. A second work is proposed by Rodriguez-Fernandez et al. [26], who present an approach for IoT-enhanced deviation detection. In their paper, they argue that traditional conformance checking cannot take into account data that can change over time independently of the events of the process (i.e., time series data). They propose a method to detect patterns in the time series data directly (in a so-called *time-series log*). Remark that these papers bumped into the limitations of traditional event logs: Banham et al. [2] had to abstract the IoT data, which implied making important assumptions, and Rodriguez-Fernandez et al. [26] only used the time series data as input to their technique because they could not integrate them in an event log.

4 Challenges

Using this running example as illustration, we pinpoint four key challenges of IoT data that make it difficult to integrate them in a traditional event log without having prior knowledge about the process or making non-trivial assumptions. These challenges are established based on an in-depth analysis of the literature (see Section 3) and based on the authors' experience in a number of case studies [33]. Observe that the literature also mentions other challenges than the ones listed here, e.g., data quality or data volume. However, we focused on the challenges that 1) have a direct influence on the mining and 2) pertain to the format of the event log.

C1: Granularity. The granularity of the sensor can be considered a topmost challenge [4, 10, 17, 28, 30, 39]. Process decisions are usually not made based on the raw sensor values, but rather depend on aggregations of the value of a sensor over a certain time or on combinations of the values of several sensors. E.g., the first decision is based on the humidity and temperature inside the refrigerated truck over the whole transport. Granularity is usually dealt with in the event abstraction step, i.e. it is considered as preprocessing. However, without prior knowledge, there is no way to know which aggregation or combination of sensor data should be applied to retrieve the relevant high-level context parameter that determines the decision, and the event abstraction is a mining step, which should be done based on the event log.

C2: Perspective convolution. IoT data can relate to events of the control-flow or to the context of the process, or to both [2, 15, 30]. E.g., accelerometer data could tell that crates of a product are being shaken during transport (context) or that the crates are being loaded in the truck (control-flow), a spike in fridge temperature can indicate that the fridge is opened to put in or take out crates (control-flow), etc. However, in existing event log standards, it is required to

choose between using the IoT data as process events or attributes of a case or an event.

C3: Scope of relevance. The relevance of the IoT data, especially when it relates to the context of a process, may not always be limited to an event or a case, but can relate to several events or several cases [10, 15, 17]. E.g., the temperature in the truck does not necessarily impact only one crate of products, but usually many crates of different types of products transported in the same truck. And logging the value of temperature (and other variables) separately for each event of each product or crate potentially leads to duplicating huge chunks of data.

C4: Dynamicity. IoT data are inherently dynamic and not always synchronised with the process [26]. The dynamicity of context parameters is usually coped with by allowing data attributes to be updated in process events. However, and as also shown with the A-B-C process example, IoT data are often loosely coupled with the process. In the motivating example, the temperature is sensed at regular intervals during the transport of the products, and not only when events happen in the process. Although there is no guarantee that the sensor data will have relevant values when process events are logged, this is usually the only point when context parameters can be updated. But without previous knowledge, it is not possible to know which sensor measurement(s) is or are relevant. E.g., the observation of temperature that impacts the decision to keep a crate of fresh products or not can be made at any moment before the decision point, and there is no way to log it without an event. But this event 1) would not be linked with a process activity and 2) this requires prior knowledge on the impact of temperature on the decision, which we assume we do not have, as it is the sort of effects that we would like to discover with PM.

5 Requirements for an IoT-enhanced event log model

Based on the challenges identified previously, we put forward 10 requirements that a suitable data model for IoT-enhanced event logs should meet.

- R1: Store high-level events
- R2: Store low-level events
- R3: Store intermediary/mixed-level events
- R4: Enable traceability between high-level and low-level events
- R5: Represent context at event level
- R6: Represent context at activity level
- R7: Represent context at case/object level
- R8: Represent context at process level
- R9: Update context parameters independently from process events
- R10: Update context parameters at a higher frequency

R1-4 connect to the ability to store data at different granularity levels (to meet C1). This means that data at either the typical low-level of raw IoT sensors,

or the traditional high level of the process, or in-between levels, should be able to coexist in an IoT-enhanced event log. From this, it follows that the traceability of the IoT data should be guaranteed, to keep track of which raw low-level IoT data have been abstracted into which high-level process events or context parameters, and of possible in-between steps, e.g., by defining links between events at different levels. Meeting these requirements would enable process mining to leverage IoT data to dig deeper into the working of the process.

Another effect of these requirements is that event abstraction is recognised as a mining step, and not just as a preprocessing step, since the low-level events are present in the event log itself. E.g., a process analyst can more easily test different possibilities to derive high-level events from the IoT data and compare the resulting process models.

R5-10: relate to having a suitable representation of the (physical) context of the process. These requirements can be split in two subgroups. First, R5-8 state that context parameters should have different scopes (event, activity, case/object, and process; see C3). E.g., consider a process where many instances and multiple process activities take place in the same environment (i.e., in the same physical space). Some environmental phenomena are tracked by sensors s_1, s_2, ..., s_n. As such, these sensors may be collecting data that is relevant for different processes, different instances of a process, different activities within a process, etc. In a typical XES event log, the values of all the sensors (or of a parameter derived by abstracting them) would be logged for each event happening in this environment for each case (see Table 2). Better defining the scope of the IoT data could simplify the event log and help focus on the more relevant values of the sensors. Moreover, it could help to find links between different instances of the process that occur simultaneously.

Table 2: Example XES event log with sensor data being duplicated.

Case ID	Event ID	Activity	Timestamp	Sensor 1	Sensor 2	...	Sensor n
1	1	A	12-01-22.09:28	42	0.2	...	10
2	2	A	12-01-22.09:28	42	0.2	...	10
1	3	B	12-01-22.09:30	42	0.3	...	10
3	4	A	12-01-22.09:31	42	0.3	...	10.1
3	5	B	12-01-22.09:31	42	0.3	...	10.1
1	6	C	12-01-22.09:32	40	0.2	...	10.1
2	7	B	12-01-22.09:34	40	0.2	...	10.1
4	8	A	12-01-22.09:34	40	0.2	...	10.1
...	...	...	...	...	...	...	...

Second, R9-10 stipulate that context parameters should be able to evolve with time at their own rhythm (C4), independently of control-flow events (C2). This aims at catering for situations such as the one depicted by the A-B-C process example, where sensor data that are relevant for a decision can be observed at

different moments during the execution of the process, i.e., typically at any point within a certain time frame, and not only when events happen.

6 Assessment of current event log standards

6.1 Comparison of existing models

In this section, we confront existing event log models (XES and OCEL) with the requirements listed in Section 5 to point out their limitations.

Table 3: Comparison of XES and OCEL with respect to the Requirements.

		XES	OCEL	Source
Requirements	R1: Store high-level events	✓	✓	C1
	R2: Store low-level events			C1
	R3: Store intermediary/mixed-level events			C1
	R4: Enable traceability between high-level and low-level events			C1
	R5: Represent context at event level	✓	✓	C3
	R6: Represent context at activity level			C3
	R7: Represent context at case/object level	✓	✓	C3
	R8: Represent context at process level	✓		C3
	R9: Update context parameters independently from process events			C2
	R10: Update context parameters at a higher frequency			C4

XES. So far, nearly all event logs used in IoT-based PM were XES logs. However, these event logs were already abstracted and most of the time IoT data were only used to retrieve process events and not to add context information.

XES typically stores data at a high level of abstraction, as events or attributes (fulfilling R1), which means that IoT data need to be abstracted into events or attributes to be recorded in a XES log, potentially losing important bits of information to the abstraction step. Moreover, abstracting the IoT data considerably restricts the spectrum of analyses that can be performed with the IoT data. This entails that XES cannot meet R2-4, which stipulate that it should be possible to store data at different granularity levels in a same event log and allow for traceability between them.

The elements that store all the information are the attributes. However, attributes in XES are defined over an event, a trace or the log (R5,7,8; R6 is not fulfilled). This makes it difficult for an IoT context variable, which can often impact several cases at the same time, to be related to all the cases it applies to (R9). Moreover, this means that new or updated context information cannot be recorded outside of a process event, while R10 stipulates that context

parameters (especially physical phenomena tracked by IoT devices) sometimes have interesting values at arbitrary moments.

OCEL. In line with XES, OCEL typically stores data at a high level of granularity (R1), which means that IoT data need to be abstracted into events or attributes to be saved in an OCEL log too, incurring the same risk of losing important information and restricting the possibilities of analysis of the IoT-enhanced log. This contradicts R2-4.

Then, like in XES, all context information is represented by means of attributes of events or objects (R5,7; R6 and R8 not fulfilled). Moreover, the only way to log a new value of an attribute is with an event, which goes against R9-10.

7 Conclusion

In this paper, we pleaded for a wider look at the possible uses of IoT data in process mining. While the existing literature largely tackles the issue of extracting process events from IoT data (so-called *activity mining*), other potential uses of IoT data listed in Table 1 remain almost entirely unexplored. One of the main hurdles complicating the development of new techniques is the restrictions that are imposed by the current main standards for event logs, namely XES and OCEL. All the data have to be stored as high-level events or attributes of high-level events or cases (objects in the case of OCEL), leaving context as a second-class citizen. This makes it necessary to abstract the low-level IoT data to store them in event logs, thereby losing information and making important assumptions on the influence of the phenomena tracked by IoT devices on the process. Moreover, to make these assumptions, an extensive knowledge of the working of the process is required, which may not be available when doing process mining, as acquiring this deeper knowledge of the process is often the goal of process mining itself.

To solve this issue, we listed requirements for a suitable model for IoT-enhanced event logs. Then, we confronted XES and OCEL, the two main standards for event logs at the moment, with these requirements, and showed that they are both currently unsuitable for the storage of IoT-enhanced event logs. We therefore claim that a dedicated comprehensive event log model is needed to develop more advanced IoT-enhanced PM techniques

Note that some of the challenges are not only encountered when dealing with IoT data. The representation of the context in XES and OCEL can be a limitation to more traditional process mining as well, e.g., some non-IoT context parameters can change outside of process events and a blurry definition of the context can lead to complex process models (see [7]) Moreover, the issue of granularity is found in many other processes and is discussed outside of the literature on PM with IoT data (e.g., [11, 32]).

In future works, we plan to define and formally implement a new model for IoT-enhanced event logs. We also intend to investigate additional real-life examples and use cases to show the use of a new standard. At a later stage,

we would like to develop algorithms that can be applied to such event logs to realise the IoT-enhanced PM techniques evoked in Section 2. This may also require to examine the suitability of current modelling languages to represent such IoT-enhanced processes, and propose new ones, e.g., combining BPMN with ontologies to give semantics to the context of the process.

References

1. van der Aalst, W., Weijters, T., Maruster, L.: Workflow mining: discovering process models from event logs. TKDE **16**(9), 1128–1142 (2004)
2. Banham, A., Wynn, M.T.: xpm: A framework for process mining with exogenous data p. 12 (2021)
3. Bazhenova, E., Buelow, S., Weske, M.: Discovering Decision Models from Event Logs, LNBIP, vol. 255, p. 237–251 (2016)
4. Bertrand, Y., De Weerdt, J., Serral Asensio, E.: A bridging model for process mining and iot. In: ICPM Workshops proceedings (2021)
5. Bozkaya, M., Gabriels, J.: Process diagnostics: a method based on process mining p. 7 (2009)
6. De Weerdt, J., vanden Broucke, S., Vanthienen, J., Baesens, B.: Active trace clustering for improved process discovery. TKDE **25**(12), 2708–2720 (Dec 2013). https://doi.org/10.1109/TKDE.2013.64
7. Dees, M., Hompes, B., van der Aalst, W.M.: Events put into context (epic). In: ICPM. p. 65–72 (Oct 2020)
8. Di Francescomarino, C., Ghidini, C., Maggi, F.M., Milani, F.: Predictive process monitoring methods: Which one suits me best? arXiv:1804.02422 [cs] (Apr 2018)
9. Dunzer, S., Stierle, M., Matzner, M., Baier, S.: Conformance checking: a state-of-the-art literature review. In: S-BPM ONE '19 Proceedings. p. 1–10 (2019)
10. van Eck, M.L., Sidorova, N., van der Aalst, W.M.P.: Enabling process mining on sensor data from smart products. In: RCIS. p. 1–12 (Jun 2016)
11. Fazzinga, B., Flesca, S., Furfaro, F., Pontieri, L.: Process discovery from low-level event logs (2018)
12. Ghahfarokhi, A.F., Park, G., Berti, A., van der Aalst, W.M.P.: OCEL: A Standard for Object-Centric Event Logs, Communications in Computer and Information Science, vol. 1450, p. 169–175 (2021)
13. Günther, C.W., Verbeek, E.: Xes standard definition (Mar 2014)
14. Jablonski, S., Rögliner, M., Schönig, S., Wyrtki, K.M.: Multi-perspective clustering of process execution traces. EMISAJ p. 2:1–22 Pages (Feb 2019)
15. Janiesch, C., Koschmider, A., Mecella, M., Weber, B., Burattin, A.e.a.: The internet-of-things meets business process management: A manifesto. IEEE Systems, Man, and Cybernetics Magazine **6**(4), 34–44 (Oct 2020), arXiv: 1709.03628
16. Janssen, D., Mannhardt, F., Koschmider, A., van Zelst, S.J.: Process model discovery from sensor event data. p. 12 (2020)
17. Koschmider, A., Janssen, D., Mannhardt, F.: Framework for process discovery from sensor data. p. 8 (2020)
18. Koschmider, A., Mannhardt, F., Heuser, T.: On the Contextualization of Event-Activity Mappings, LNBIP, vol. 342, p. 445–457 (2019)
19. Leemans, S.J.J., Fahland, D., van der Aalst, W.M.P.: Discovering Block-Structured Process Models from Event Logs Containing Infrequent Behaviour, LNBIP, vol. 171, p. 66–78 (2014)

20. de Leoni, M., van der Aalst, W.M.P.: Data-aware process mining: discovering decisions in processes using alignments. In: SAC '13. p. 1454 (2013)
21. Mannhardt, F., Bovo, R., Oliveira, M., Julier, S.: A taxonomy for combining activity recognition and process discovery in industrial environments (2018)
22. Mannhardt, F.: Multi-perspective process mining. Ph.D. thesis (2018)
23. Mannhardt, F., de Leoni, M., Reijers, H.A., van der Aalst, W.M.P.: Data-Driven Process Discovery - Revealing Conditional Infrequent Behavior from Event Logs, LNCS, vol. 10253, p. 545–560 (2017)
24. Nakatumba, J.: Analyzing resource behavior using process mining. In: BPI'09. p. 69–80 (2010)
25. Popova, V., Fahland, D., Dumas, M.: Artifact lifecycle discovery. arXiv:1303.2554 [cs] (Mar 2013)
26. Rodriguez-Fernandez, V., Trzcionkowska, A., Gonzalez-Pardo, A., Brzychczy, E., Nalepa, G.J., Camacho, D.: Conformance checking for time-series-aware processes. IEEE TII **17**(2), 871–881 (Feb 2021)
27. Seiger, R., Malburg, L., Weber, B., Bergmann, R.: Integrating process management and event processing in smart factories: A systems architecture and use cases. Journal of Manufacturing Systems **63**, 575–592 (2022)
28. Seiger, R., Zerbato, F., Burattin, A., Garcia-Banuelos, L., Weber, B.: Towards iot-driven process event log generation for conformance checking in smart factories. In: EDOCW. p. 20–26 (Oct 2020)
29. Serral, E., De Smedt, J., Vanthienen, J.: Making business environments smarter: a context-adaptive petri net approach. In: UIC 2014. pp. 343–348 (2014)
30. Soffer, P., et al.: From event streams to process models and back: Challenges and opportunities. Information Systems **81**, 181–200 (Mar 2019)
31. Song, M., Günther, C.W., van der Aalst, W.M.P.: Trace Clustering in Process Mining, LNBIP, vol. 17, p. 109–120 (2009)
32. Tax, N., Sidorova, N., Haakma, R., van der Aalst, W.M.P.: Event abstraction for process mining using supervised learning techniques. arXiv:1606.07283 [cs] **15**, 251–269 (2018)
33. Torres, V., Serral, E., Valderas, P., Pelechano, V., Grefen, P.: Modeling of iot devices in business processes: A systematic mapping study. In: CBI 2020. vol. 1, pp. 221–230 (2020)
34. Trzcionkowska, A., Brzychczy, E.: Practical aspects of event logs creation for industrial process modelling. MAPE **1**(1), 77–83 (Sep 2018)
35. Valderas, P., Torres, V., Serral, E.: Modelling and executing iot-enhanced business processes through bpmn and microservices. JSS **184**, 111139 (Feb 2022)
36. Valencia-Parra, A., Ramos-Gutierrez, B., Varela-Vaca, A.J., Gomez-Lopez, M.T., Bernal, A.G.: Enabling process mining in aircraft manufactures: Extracting event logs and discovering processes from complex data p. 12 (2019)
37. Weijters, A., van der Aalst, W.M.P., Alves de Medeiros, A.K.: Process mining with the heuristics miner algorithms (2006)
38. van der Werf, J.M.E.M., Verbeek, H.M.W., van der Aalst, W.M.P.: Context-Aware Compliance Checking, LNCS, vol. 7481, p. 98–113 (2012)
39. Zerbato, F., Seiger, R., Federico, G.D., Burattin, A., Weber, B.: Granularity in process mining: Can we fix it? pp. 40–44 (2021)