Towards IoT-driven Process Event Log Generation for Conformance Checking in Smart Factories

Ronny Seiger*, Francesca Zerbato*, Andrea Burattin†, Luciano García-Bañuelos‡ and Barbara Weber*
*School of Computer Science, University of St. Gallen (HSG), St. Gallen, Switzerland
Email: {ronny.seiger | francesca.zerbato | barbara.weber}@unisg.ch
†Technical University of Denmark (DTU), Lyngby, Denmark
Email: andbur@dtu.dk
‡School of Engineering and Sciences
Tecnologico de Monterrey, Puebla, Mexico
Email: luciano.garcia@tec.mx

Abstract—**The Internet of Things (IoT) enables software-based access to vast amounts of data streams from sensors measuring physical and virtual properties of smart devices and their surroundings. While sophisticated means for the control and data analysis of single IoT devices exist, a more process-oriented view of IoT systems is often missing. Such a lack of process awareness hinders the development of process-based systems on top of IoT environments and the application of process mining techniques for process analysis and optimization in IoT. We propose a framework for the stepwise correlation and composition of raw IoT sensor streams with events and activities on a process level based on Complex Event Processing (CEP). From this correlation we derive refined process event logs–possibly with ambiguities–that can be used for process analysis at runtime (i. e., online). We discuss the framework using examples from a smart factory.**

Index Terms—**Conformance Checking, Complex Event Processing, Internet of Things, Process Mining, Smart Factories**

1. Introduction

The Internet of Things (IoT) enables the digitization of the physical world through interconnected devices, thereby providing access to vast amounts of data that can be used to develop digital services in several application domains including Business Process Management (BPM) [1]. The data generated by sensing devices and smart objects allows for the continuous monitoring of the IoT devices and their surroundings, providing new opportunities for analysis and optimization of the processes performed in IoT environments, e. g., through process mining approaches [2], [3].

However, IoT infrastructures mostly rely on siloed devices and are integrated with applications which are not necessarily process-aware [4]. Indeed, such IoT applications are often built upon proprietary control software relying on non-standardized interfaces that thus hinder the use of process-enabled enactment systems [1], [5]. Moreover, IoT data is usually produced in the form of low-level event streams having little to no meaning for high-level activities and processes [6] and, often, requires to be analyzed as soon as it is generated (i. e., online [7]).

Despite recent research efforts having contributed to advance the integration of IoT and BPM technologies (e. g., [8]–[11]), IoT technology is not readily integrated into industrial-strength BPM systems [1]. Indeed, events generated from IoT sensors (i) do not directly correspond to meaningful process activities and (ii) do not carry information about process instances, and (iii) IoT entities and processes are rarely represented in an explicit way in and through process models. However, (i)–(ii) are crucial pre-requisites for generating event logs suitable for process mining, and (iii) is necessary to assess the conformance of such a log with a reference process model [12]. Besides, the increased process awareness in IoT also facilitates process planning, optimization and adaptation [9]. This is of high relevance in dynamic IoT systems (e. g., smart factories) where physical resources are constrained and multiple processes are executed in parallel [1], [13].

With this work-in-progress paper, we address the challenge of bridging the gap between event-based and process-based systems (cf. C13 in [1]), mainly focusing on the generation of event logs from streams of IoT sensor data with smart factories as an application domain. Indeed, since IoT sensor data usually refers to the low-level states of IoT devices and the physical properties of their surroundings, the *online correlation* with higher-level process events and the subsequent detection of activities and activity instances becomes a non-trivial task. Building upon the assumption that IoT environments lack process awareness, we present a novel framework for deriving process events from low-level IoT data streams into a process event log, which can be stored and used for offline conformance checking or used directly for online analysis [7]. We rely on Complex Event Processing (CEP) as the key technology to achieve this online correlation in (near) real-time and refine the detected events in multiple stages [14]. Besides, since the correlation of IoT sensor data with process events and activities may not always be clear and achievable through one-to-one mappings [15], we also consider the presence of uncertainties and *ambiguities* during event log generation.

The paper is structured as follows: Sect. 2 discusses related work; Sect. 3 introduces the smart factory simulation model that we use in our research; Sect. 4 presents the framework for IoT-driven event log generation; Sect. 5 elaborates on our vision towards IoT-driven conformance checking; Sect. 6 concludes the paper and shows starting points for future work.

2. Related Work

Enriching process execution environments with IoT technologies brings several benefits but does not come without challenges [1]. One of the main benefits is undoubtedly the improved recognition of (manual) activities and processes from sensor data, e. g., with the help of multi-modal approaches such as the one by Rebmann et al. [16] that combines motion and vision sensors with user feedback for detecting and disambiguating known activity types in real-time. Activity recognition is often combined with process mining for extracting knowledge to analyze and optimize the underlying processes [17]. 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 [1].

First of all, *event abstraction* is needed to map multiple fine-grained sensor data to coarse-grained process events [18], e. g., with the help of CEP [6], which also supports data pre-processing and context enrichment [2]; or with machine learning approaches such as clustering [19] or supervised learning [20]. Wanner et al. combine these two methods based on expert knowledge and observations in the context of a smart factory [21]. Then, 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* [15], [22]. We discuss work about event log generation from IoT events that mostly relates to our contribution and refer the interested reader to [6], [18] for a broader literature overview.

In [23] 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 [24] 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. In [25] Koschmider et al. investigate the role of context-awareness in event-to-activity mappings and propose a framework to support the inclusion of context information in the event logs and the comparison of different mapping approaches. The authors also propose and discuss a modular framework for enabling process discovery from sensor data, explicitly considering the data aggregations needed for event correlation, activity discovery, event abstraction, and process discovery [26]. In [22] Baier et al. present an approach for semi-automatically bridging the event abstraction levels required for process mining based on domain knowledge. Van der Aa et al. propose a probabilistic conformance checking technique that considers all potential event-to-activity mappings–the *behavioral space*–and identifies conforming traces in the presence of mapping uncertainty [15]. Ehrendorfer et al. present in [13] 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.

Our approach contributes to the body of research focusing on correlating low-level sensor events with process events and mapping them to process activities. The goal is to generate an event log directly from streamed sensor data that can be used for (online) conformance checking [7]. Compared to foundational research focusing on the abstraction of raw IoT sensor data to process events and activities, we take a practical view and aim to achieve these multi-stage correlations in an online manner, starting directly from the IoT data streams and combining them with domain and process knowledge. The goal of realizing this online analysis in (near) real-time goes in line with the nature of IoT systems that produce high amounts of live streaming data and often lack the presence of a process management system [1]. The smart factory model presented in Sect. 3 thereby serves as perfect hands-on experimentation platform [27] that allows for a stepwise increase of sensor to process event correlations starting from single sensors and process instances to multi-modal sensor correlations and multiple process instances executed in parallel. With that we are also able to control the level of ambiguities and uncertainties introduced into the derived process event log.

3. Smart Factory Simulation Model

In our research we focus on smart factories as representatives of IoT environments [17]. Smart factories feature IoT-enabled production devices of varying complexity that consist of heterogeneous sensors and actuators. With the advancements of Industry 4.0 technologies these components are increasingly controlled by microprocessors, which enable communication with other components and the software-based access to sensor data and actuator functionality [28]. While processes are inherently prevalent in the production lines of smart factories, the process-awareness of these IoT environments remains limited, i. e., a process is rarely viewed as “a collection of inter-related events, activities, and decision points that involve a number of actors and objects” [29]. The individual production machines are usually programmed–similar to many other IoT devices–in isolation and on a low level resulting in inflexible hardwired production processes and access to sensor data and actuator functionality on a very fine-grained technical level [4].

Fig. 1 shows the factory simulation model *Training Factory Industry 4.0*[1] developed by Fischertechnik that we

[1]https://www.fischertechnik.de/en/service/elearning/teaching/lernfabrik-4

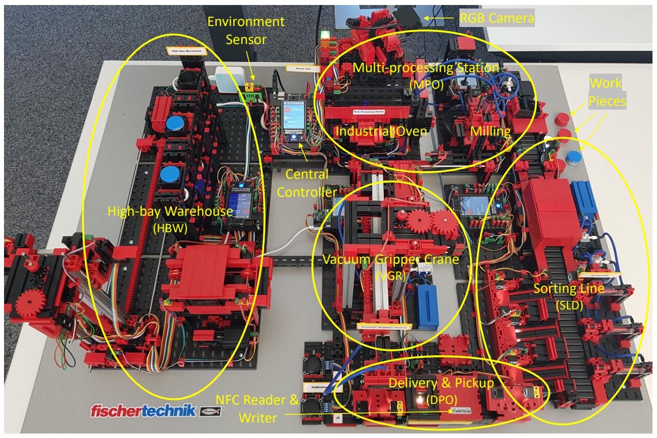

Figure 1: Physical smart factory simulation model as IoT environment.

use in our research. The factory features a complete production line consisting of five workstations: a delivery & pickup station (DPO) with NFC reader/writer; a high-bay warehouse (HBW) with 3 x 3 slots for storage of workpieces in containers; a central vacuum gripper (VGR) for transport; a multi-processing station (MPO) with an industrial oven and a milling station; and a sorting line (SLD) with color detection. Each station consists of a multitude of sensors (e. g., light barriers) and actuators (e. g., motors, compressors). An RGB camera and environmental sensor collect additional data. To simulate the production, small cylindrical workpieces (height = 1,4 cm, diameter = 2,6 cm) of varying colors (blue, red, white) are moved through the factory. Each workpiece is equipped with an NFC tag containing information regarding the individual piece (e. g., an identifier, the color and timestamped production history). Six networked micro-controllers run the programs (written in C/C++) to control the individual stations. In the factory's standard configuration the central controller hosts an additional MQTT broker[2] that allows remote publish/subscribe access to sensor, machine and production-related data (e. g., machine states, production states, warehouse inventory, environment and NFC readings) from MQTT clients. MQTT (Message Queuing Telemetry Transport) [30] is a light-weight industry standard protocol that is specifically tailored to enable messaging in IoT environments with resource-constraint devices, e. g., the factory's controllers. For our work we assume that the data produced by the factory is fixed, i. e., we cannot change or extend its content or interfaces, as it is often the case for IoT platforms and production machines due to limited access to their underlying control programs and source code [5]. The amount and quality of low-level sensor data can be improved via *retrofitting* of production machines with additional sensors as discussed in [31]. However this does not directly influence the information available on the higher abstraction level of processes, which requires additional steps of data fusion and processing.

In its default configuration the factory supports the

[2]https://github.com/fischertechnik/txt_training_factory/blob/master/TxtSmartFactoryLib/doc/MqttInterface.md

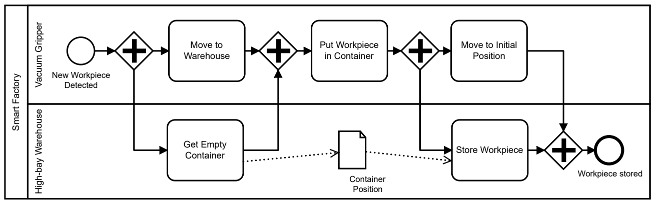

Figure 2: Simplified storage process in BPMN 2.0.

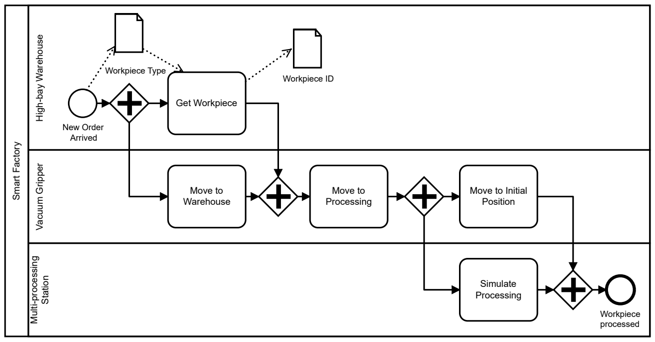

Figure 3: Simplified production process in BPMN 2.0.

following two processes, which are implemented statically as C++ programs–a process-based control system is not available. The process models have been created based on observations and from the smart factory's documentation.

- **Storage Process**: This process (cf. Fig. 2) starts when the VGR station detects a new workpiece. The gripper moves the new workpiece to the high-bay warehouse. In parallel the warehouse retrieves an empty container from a specific slot (position). The vacuum gripper then puts the workpiece into the container. Afterwards the workpiece is stored in the HBW at the container's original slot. In parallel the gripper returns to its initial position.
- **Production Process**: This process (cf. Fig. 3) starts when a new order for a specific type (i. e., color) of product is received. The HBW retrieves a container with a suitable workpiece. In parallel the vacuum gripper moves to the HBW to collect the workpiece. Then, the gripper moves it to the multi-processing station. The MPO then simulates the production of the workpiece while, in parallel, the vacuum gripper moves to its initial position.

4. IoT-driven Event Log Generation using CEP

In this work we propose a framework for generating process event logs based on data from IoT sensors without having to rely on a BPM system to manage process executions [6]. An overview of the proposed framework is depicted in Fig. 4. Our framework is grounded in CEP, which enables the live (online) analysis of high volume data streams from various sources and features rich mechanisms to correlate, aggregate, filter, map and process events from

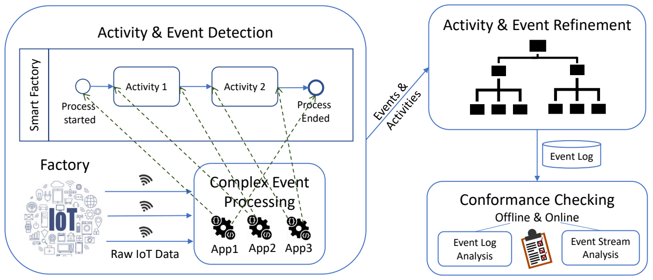

Figure 4: Framework for event log generation & conformance checking.

Listing 1: CEP app for detecting the start of "Simulate Production".

```
@source(type = 'mqtt', topic = 'f/i/state/mpo',                 1
    client.id = 'siddhi-mpo', @map(type = 'json'))               2
define stream Mpo (active int, ts string);                       3
                                                                 4
define stream ProductionDetection(type string, ts string);       5
                                                                 6
@info(name='SimulateProductionStartDetectionQuery')              7
from every e1=Mpo, e2=Mpo[(e1.active==0) AND (e2.active==1)]     8
select 'start' as type, e2.ts                                    9
insert into ProductionDetection;                                 10
```

multiple streams to detect patterns of event occurrences and derive complex events from these analyses via data fusion [14]. CEP has also been applied in the context of IoT [21], [32] and BPM [8], [10], [11], [33] for deriving higher level events from IoT data. As the detection of process events from IoT sensors is not always straightforward and unambiguous [1], we propose to go through multiple stages within our framework: *Activity & Event Detection*; *Activity & Event Refinement*; *Event Log Generation*.

4.1. Activity & Event Detection from IoT Data

For the detection of process activities and events, we assume that domain knowledge about the IoT data sources and processes, and their correlations, exists [22], [23]. Domain experts and process engineers provide models for the IoT processes, e. g., as presented in Figures 2 and 3 for two exemplary smart factory processes. These experts also develop CEP applications that link the raw IoT data to the individual activities and events of the modeled processes–raising the abstraction of IoT events to the process level, e. g., by specifying queries for identifying the start and end of activities (*Monitoring Points* [34]). This can be partially automated using machine learning as proposed in [21].

In our research we use the CEP platform *Siddhi* for event processing as it provides a comprehensive language for stream processing including filtering, aggregation, enriching and merging of event streams [35]. In contrast to similar tools, it can be used as a standalone application proving a graphical editor or as light-weight library embedded into third party applications. The main ingredients of a *Siddhi App* are *Streams* containing *Events*, and SQL-like *Queries* executed on these streams. External data *Sources* can be defined as producers of events for a stream (e. g., from IoT

Listing 2: CEP app for detecting the start of "Move to Warehouse".

```
@source(type = 'mqtt', topic = 'f/i/state/vgr',                  1
    client.id = 'siddhi-vgr', @map(type = 'json'))               2
define stream Vgr (active int, target string, ts string);        3
                                                                 4
define stream ToWareHouseDetection(type string, ts string);      5
                                                                 6
@info(name='MoveToWarehouseStartDetectionQuery')                 7
from every e1=Vgr, e2=Vgr[(e1.active==0) AND (e2.active==1)]     8
select 'start' as type, e2.ts                                    9
having e2.target=='Hbw'                                          10
insert into ToWareHouseDetection;                                11
```

sensors) and external *Sinks* can be defined as consumers of events from a stream (e. g., for IoT actuators). Siddhi already supports many standardized messaging and web service protocols to connect to various streams as sources and sinks, which makes it also easy to establish a bi-directional communication with BPM systems.

Listing 1 presents an exemplary Siddhi app for detecting the start of the "Simulate Production" activity from the production process (cf. Fig. 3). First the connector to the external event source is defined. Here we refer to the MPO's state data published via MQTT on the given topic (Line 1) to the event stream `Mpo` having the `active` flag and timestamp (`ts`) as event attributes (Line 3). We define a new stream containing activity-related events (`ProductionDetection`) with events having a `type` (e. g., `start` or `end` of the activity) and corresponding timestamp as attributes (Line 5). Starting on Line 7 the query to process the low-level machine state and derive the start of the activity is defined. We look at the changes of the `active` attribute in two subsequent events `e1` and `e2` (Line 8). According to the MPO's state machine[3] the production starts when the `active` attribute changes from `0` to `1`. Upon detection of this pattern we insert a new event with type `start` and the timestamp of event `e2` as attributes (Line 9) into the `ProductionDetection` stream (Line 10).

Listing 2 presents a Siddhi app to detect the vacuum gripper's movement to the warehouse, which is part of both exemplary factory processes. Similar to the previous example we use the changes of the station's (here: `Vgr`) activation state to derive the high-level activity start event. As the VGR is the central transportation unit and, thus, is activated very often, we consider the additional `target` attribute (here: `Hbw`) contained within the low-level sensor events (Line 3) to distinguish between different possible types of gripper movements (Line 10).

4.2. Activity & Event Refinement based on Context

While the correlation, fusion and abstraction of low-level sensor events to higher level process-related events described in the previous section is rather easy to achieve for IoT systems with only a small process landscape and a single instance running at a time, the activity detection in more

[3]https://github.com/fischertechnik/txt_training_factory/blob/master/README.md

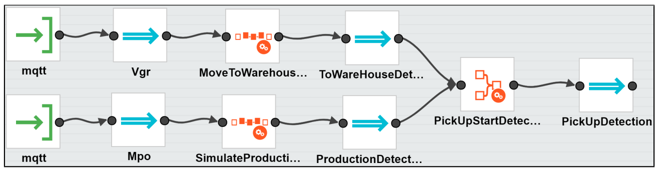

Figure 5: Joining event streams for refinement of the VGR movement.

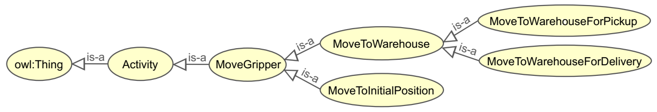

Figure 6: Excerpt of the domain ontology for gripper movement.

Listing 3: Excerpt of sample output event log showing the detected start of the gripper movement activity in XES format.

```
<?xml version="1.0" encoding="UTF-8" ?>
<log xes.version="1.0" xes.features="nested-attributes"
     openxes.version="1.0RC7">
...
 <trace>
   <string key="concept:name" value="ProductionProc_001"/>
   <event>
     <string key="concept:name"
          value="MoveToWarehouseForPickup"/>
     <boolean key="active" value="true"/>
     <date key="time:timestamp"
          value="2020-09-09T08:20:02.223+01:00"/>
     <string key="org:resource" value="R108"/>
     <string key="org:role" value="VacuumGripper"/>
     <string key="lifecycle:transition" value="complete"/>
   </event>
   ...
 </trace>
 ...
</log>
```

complex IoT settings becomes increasingly difficult. The low-level sensor readings are usually not sufficient to always clearly identify process-related events and correlate them with a specific process (model) and execution (instance), e. g., when the same activity appears in one or more process models at multiple points, or multiple instances containing the same activity are executed in parallel [1], [26].

Solving these uncertainties requires the consideration of additional process-related factors and context data. Within our framework we propose to extend the CEP apps with additional rules and queries (e. g., regarding temporal, logical and spatial dependencies of the process/activity executions) and intermediate event streams with higher level process (domain) events that can be detected and joined for activity and event refinement [22], [36]. On the model level, logical dependencies (e. g., the detected following process step) as described within the respective process models could for example be used to refine the activity detection; on the instance level, temporal aspects (e. g., execution duration) or spatial aspects (e. g, interactions in the same location [23], [37]) could be used for refinement. These rule-based correlations between activities can also be used to detect unintended behavior, errors and deviations of the process execution [21].

In the smart factory we are able to detect the vacuum gripper moving as an activity, which can be part of several processes or even appear multiple times in the same process (e. g, the movement to its initial position). By considering the additional `target` attribute in the low-level sensor data we can already limit the candidates of VGR activities to the "Move to Warehouse" activity in the storage process (for delivery of a new workpiece) or in the production process (for pickup of the workpiece to be processed). By considering the activities detected subsequently, we can further refine the type of movement on the *model level*. Joining the two output streams defined in Listings 1 (Line 5) and 2 (Line 5) and introducing the condition that the start event of "Simulate Production" is preceded by the "Move to Warehouse" start event in a new CEP query–looking at their subsequent occurrence similar to the examples in line 8 of both listings–allows us to further refine the movement activity to "Move to Warehouse for Pickup", which is associated with the production process (cf. Fig. 5). Fig. 6 shows an excerpt from the ontological representation of the refinement possibilities for the gripper movement activity. With this form of hierarchical process knowledge, the corresponding CEP queries and event streams can be incrementally combined to increase the granularity of activity detection [38], [39].

The *instance level* event and activity correlation (i. e., deriving of case-IDs) requires more fine-grained and detailed knowledge about the executions, which has to be reflected in additional intermediate event streams and more precise queries producing complex events with a `case-ID` as additional attribute [40], e. g., considering the events' timestamps and linking two subsequent activities to the same instance based on their close temporal proximity. Listing 3 shows the sample output of applying these refinements to concretize the VGR movement event and associate it with a case-ID in a trace of the event log. However, despite these refinements the available IoT data and process knowledge may still not be sufficient to clearly resolve all uncertainties about the activity detection and event correlations, which results in *ambiguities* that need to be reflected in the event log [26].

4.3. Event Log Generation with Ambiguities

The refined and case-related process events and activities can be either stored in a classical event log containing execution traces for offline analysis (e. g., control flow discovery or conformance checking [12]) or streamed as part of an online event log for online analysis. However, due to the limited availability and capabilities of IoT sensors (e. g., being to coarse grained or imprecise) as well as limited means for an online analysis of the complete process instance context (e. g., regarding future process steps or long passed executions), the refinement of events and activities may not be possible to clearly associate the complex events with a specific process model and instance [1]. We therefore have to consider *ambiguities/uncertainties* regarding the detected process elements in the event logs and propose to introduce a *candidate list* and *confidence metric* [3].

Assuming that the complete process landscape and all event and activity refinements are known (e. g., described by a domain ontology [39]), we can provide a list of possible refined candidates (children) of a detected ambiguous activity at the specific point of a trace in the log accompanied by a confidence metric indicating the probability of the candidate's occurrence (e. g., by assuming an equally likely appearance of each candidate) [41]. Referring to the gripper movement from the example processes and assuming that the detected "Move to Warehouse" activity cannot be further refined through CEP, we are able to determine from the ontology presented in Fig. 6 that two refined activity candidates exist. The derived event log would contain a list with both activities and a confidence of 50 % for each activity at this point in the trace. In case of multiple instances running in parallel in the IoT environment the handling of ambiguities again becomes more complex. Similar to the consideration of ambiguities on the model level, we propose to also include a list of possible instances (cases) a detected uncertain event or activity may belong to–including the confidence value–at the specific point in the log.

5. Towards IoT-driven Conformance Checking

Conformance checking addresses the analysis and comparison between the intended behavior of a process described in a normative process model and its observed execution captured in an event log [12], where also a mapping between the events included in the event log and the elements of the process model is required [15]. Following the framework for IoT driven event log generation proposing the event-to-activity mappings based on domain knowledge described in Section 4, existing conformance checking approaches can be used in subsequent steps if no ambiguities are present in the generated log.

Traditionally, conformance checking approaches have focused on the *offline* analysis of completed process traces, thus detecting violations after they occurred. This is also supported within our framework by means of streaming and persisting the derived process events into an event log. In the presence of ambiguities for specific activities and events in the log, conformance checking has to be extended to also consider the possible candidates and their probabilities at the specific points in the traces. This may result in multiple possible alignments of varying fitness regarding the detected variants of executions and their conformance with the corresponding process models.

Online conformance checking approaches seem more attractive as IoT data is often generated and transmitted in streams and at a high volume, velocity and variety [1]. Recently, online or *streaming* conformance checking approaches [7] (e. g., based on prefix alignments [42] or behavioral patterns [43]) have been proposed to enable the immediate analysis of event streams and allow for violation detection and compensation before the process is completed. Applying these approaches in IoT environments on event streams created with the proposed framework and subsequently streamed for conformance checking (e. g., via the *MQTT-XES* protocol [44]) again requires adaptations to also consider the possible ambiguities contained within the detected process events, which increases complexity and with that reduces the capabilities of near real-time analysis. New approaches are needed here–possibly also relying on CEP–to enable the online alignment of the detected, incomplete and ambiguous executions (traces) with the process models.

6. Conclusion and Future Work

In this work we presented a framework for the detection of process events from IoT data, their refinement and associated event log generation. CEP proves to be a suitable technological basis for realizing this correlation with an arbitrary level of abstraction from sensor data to process events and activities based on domain knowledge. Multiple streams can be combined and chained hierarchically to derive higher level (complex) events from patterns related to the process execution in online settings. However, due to the nature of IoT environments and available sensors, ambiguities and uncertainties may remain as part of the event log that have to be considered in subsequent analysis stages (e. g., in conformance checking).

In future work, we will move to more complex process landscapes–including human activities–and higher numbers of parallel instances executed in the smart factory to derive more sophisticated means of handling ambiguities. While this paper mostly discussed the control flow perspective, we will also consider other perspectives measurable via IoT for event correlations. Moreover, we will investigate new means for an end-user oriented development of CEP applications and for their automatic generation from observations.

References

[1] C. Janiesch, A. Koschmider, M. Mecella, B. Weber, A. Burattin *et al.*, "The Internet-of-Things meets Business Process Management. A manifesto." *IEEE SMC Magazine*, 2020.

[2] T. Sztyler, J. Carmona, J. Völker, and H. Stuckenschmidt, *Self-tracking Reloaded: Applying Process Mining to Personalized Health Care from Labeled Sensor Data*. Berlin, Heidelberg: Springer, 2016, pp. 160–180.

[3] M. Pegoraro and W. M. P. van der Aalst, "Mining uncertain event data in process mining," in *2019 International Conference on Process Mining (ICPM), Aachen, Germany*. IEEE, Jun 2019, pp. 89–96.

[4] L. Monostori, "Cyber-physical production systems: Roots, expectations and challenges," *Procedia CIRP*, vol. 17, pp. 9–13, 2014.

[5] M. Noura, M. Atiquzzaman, and M. Gaedke, "Interoperability in Internet of Things: Taxonomies and open challenges," *Mobile Networks and Applications*, vol. 24, no. 3, pp. 796–809, 2019.

[6] P. Soffer, A. Hinze, A. Koschmider, H. Ziekow, C. Di Ciccio *et al.*, "From event streams to process models and back: Challenges and opportunities," *Information Systems*, vol. 81, pp. 181 – 200, 2019.

[7] A. Burattin, "Streaming process discovery and conformance checking," in *Encyclopedia of Big Data Technologies*. Springer International Publishing, 2018, pp. 1–8.

[8] G. Meroni, L. Baresi, M. Montali, and P. Plebani, "Multi-party business process compliance monitoring through IoT-enabled artifacts," *Information Systems*, vol. 73, pp. 61–78, 2018.

[9] A. Marrella, M. Mecella, and S. Sardiña, "Supporting adaptiveness of cyber-physical processes through action-based formalisms," *AI Commun.*, vol. 31, no. 1, pp. 47–74, 2018.

[10] R. Seiger, S. Huber, P. Heisig, and U. Aßmann, "Toward a framework for self-adaptive workflows in cyber-physical systems," *Software & Systems Modeling*, vol. 18, no. 2, pp. 1117–1134, 2019.

[11] S. Schönig, L. Ackermann, S. Jablonski, and A. Ermer, "IoT meets BPM: a bidirectional communication architecture for IoT-aware process execution," *Software and Systems Modeling*, pp. 1–17, 2020.

[12] J. Carmona, B. van Dongen, A. Solti, and M. Weidlich, *Conformance Checking*. Springer, 2018.

[13] M. Ehrendorfer, J.-A. Fassmann, J. Mangler, and S. Rinderle-Ma, "Conformance checking and classification of manufacturing log data," in *IEEE 21st Conference on Business Informatics (CBI), Moscow, Russia*, Aug 2019, pp. 569–577.

[14] D. Luckham, *The power of events*. Addison-Wesley Reading, 2002.

[15] H. Van Der Aa, H. Leopold, and H. A. Reijers, "Efficient process conformance checking on the basis of uncertain event-to-activity mappings," *IEEE Trans. on Knowledge and Data Engineering*, vol. 32, no. 5, pp. 927–940, 2019.

[16] A. Rebmann, A. Emrich, and P. Fettke, "Enabling the discovery of manual processes using a multi-modal activity recognition approach," in *Business Process Management Workshops*, ser. LNBIP, vol. 362. Springer, Jan 2020, pp. 130–141.

[17] F. Mannhardt, R. Bovo, M. F. Oliveira, and S. Julier, "A taxonomy for combining activity recognition and process discovery in industrial environments," in *Intelligent Data Engineering and Automated Learning (IDEAL 2018), Madrid, Spain*, ser. LNCS, vol. 11315. Cham: Springer International Publishing, Nov 2018, pp. 84–93.

[18] S. J. van Zelst, F. Mannhardt, M. de Leoni, and A. Koschmider, "Event abstraction in process mining: literature review and taxonomy," *Granular Computing*, May 2020.

[19] F. Folino, M. Guarascio, and L. Pontieri, "Mining predictive process models out of low-level multidimensional logs," in *International Conference on Advanced Information Systems Engineering (CAiSE), Thessaloniki, Greece*, ser. LNCS, vol. 8484. Springer, 2014, pp. 533–547.

[20] N. Tax, N. Sidorova, R. Haakma, and W. M. van der Aalst, "Event abstraction for process mining using supervised learning techniques," in *Proceedings of SAI Intelligent Systems Conference, London, UK*, ser. LNNS, vol. 15. Springer, Sep 2016, pp. 251–269.

[21] J. Wanner, L.-V. Herm, and C. Janiesch, "Countering the fear of black-boxed ai in maintenance: Towards a smart colleague," in *Proceedings of the 2019 Pre-ICIS SIGDSA Symposium*, 2019.

[22] T. Baier, J. Mendling, and M. Weske, "Bridging abstraction layers in process mining," *Information Systems*, vol. 46, pp. 123 – 139, 2014.

[23] A. Senderovich, A. Rogge-Solti, A. Gal, J. Mendling, and A. Mandelbaum, "The road from sensor data to process instances via interaction mining," in *International Conference on Advanced Information Systems Engineering (CAiSE), Stockholm, Sweden*, ser. LNCS, vol. 9097. Cham: Springer International Publishing, June 2016, pp. 257–273.

[24] F. Mannhardt, M. de Leoni, H. A. Reijers, W. M. P. van der Aalst, and P. J. Toussaint, "From low-level events to activities - a pattern-based approach," in *International Conference on Business Process Management (BPM), Rio de Janeiro, Brazil*, ser. LNCS, vol. 9850. Cham: Springer International Publishing, Sep 2016, pp. 125–141.

[25] A. Koschmider, F. Mannhardt, and T. Heuser, "On the contextualization of event-activity mappings," in *Business Process Management Workshops*. Cham: Springer International Publishing, Sep 2019, pp. 445–457.

[26] A. Koschmider, D. Janssen, and F. Mannhardt, "Framework for process discovery from sensor data," in *10th International Workshop on Enterprise Modeling and Information Systems Architectures (EMISA)*, vol. 2627. CEUR Workshop Proceedings, 2020, pp. 32–38.

[27] L. Malburg, R. Seiger, R. Bergmann, and B. Weber, "Using physical factory simulation models for business process management research," in *Business Process Management Workshops, in press*. Springer, 2020.

[28] H. Lasi, P. Fettke, H.-G. Kemper, T. Feld, and M. Hoffmann, "Industry 4.0," *Business & Information Systems Engineering*, vol. 6, no. 4, pp. 239–242, 2014.

[29] M. Dumas, M. La Rosa, J. Mendling, and H. A. Reijers, *Business process management*. Springer, 2013.

[30] A. Banks, E. Briggs, K. Borgendale, and R. Gupta, "MQTT V. 5.0," *OASIS Std.*, 2019.

[31] T. Lins and R. A. R. Oliveira, "Cyber-physical production systems retrofitting in context of industry 4.0," *Computers & Industrial Engineering*, vol. 139, p. 106193, 2020.

[32] C. Y. Chen, J. H. Fu, T. Sung, P.-F. Wang, E. Jou, and M.-W. Feng, "Complex event processing for the Internet of Things and its applications," in *2014 IEEE International Conference on Automation Science and Engineering (CASE), Taipei, Taiwan*. IEEE, Aug 2014, pp. 1144–1149.

[33] A. Baumgrass, C. Di Ciccio, R. M. Dijkman, M. Hewelt, J. Mendling *et al.*, "GET controller and unicorn: Event-driven process execution and monitoring in logistics." in *Proceedings of the BPM Demo Session Co-located with the 13th International Conference on Business Process Management (BPM 2015), Innsbruck, Austria*. CEUR Workshop Proceedings, 2015, pp. 75–79.

[34] N. Herzberg, O. Khovalko, A. Baumgrass, and M. Weske, "Towards automating the detection of event sources," in *Service-Oriented Computing - ICSOC 2013 Workshops, Berlin, Germany*, ser. LNCS, vol. 8377. Springer, 2013, pp. 111–122.

[35] S. Suhothayan, K. Gajasinghe, I. Loku Narangoda, S. Chaturanga, S. Perera, and V. Nanayakkara, "Siddhi: A second look at complex event processing architectures," in *Proceedings of the 2011 ACM workshop on Gateway computing environments*, 2011, pp. 43–50.

[36] N. Tax, B. Dalmas, N. Sidorova, W. M. van der Aalst, and S. Norre, "Interest-driven discovery of local process models," *Information Systems*, vol. 77, pp. 105–117, 2018.

[37] P. Grefen, N. Brouns, H. Ludwig, and E. Serral, "Co-location specification for IoT-aware collaborative business processes," in *Conf. Advanced Information Systems Engineering*. Springer, 2019, pp. 120–132.

[38] R. Seiger, C. Keller, F. Niebling, and T. Schlegel, "Modelling complex and flexible processes for smart cyber-physical environments," *Journal of Computational Science*, vol. 10, pp. 137–148, 2015.

[39] P. Klein, L. Malburg, and R. Bergmann, "Ftonto: A domain ontology for a fischertechnik simulation production factory by reusing existing ontologies." in *Lernen. Wissen. Daten. Analysen. (LWDA), Berlin, Germany*. CEUR Workshops Proceedings, 2019, pp. 253–264.

[40] S. Bülow, M. Backmann, N. Herzberg, T. Hille, A. Meyer *et al.*, "Monitoring of business processes with complex event processing," in *Business Process Management Workshops*, ser. LNBIP, vol. 171. Springer, 2013, pp. 277–290.

[41] M. Vitali and B. Pernici, "Interconnecting processes through IoT in a health-care scenario," in *2016 IEEE International Smart Cities Conference (ISC2), Trento, Italy*. IEEE, Sep 2016, pp. 1–6.

[42] S. J. van Zelst, A. Bolt, M. Hassani, B. F. van Dongen, and W. M. P. van der Aalst, "Online conformance checking: relating event streams to process models using prefix-alignments," *International Journal of Data Science and Analytics*, vol. 8, no. 3, pp. 269–284, Oct 2019.

[43] A. Burattin, S. J. van Zelst, A. Armas-Cervantes, B. F. van Dongen, and J. Carmona, "Online conformance checking using behavioural patterns," in *International Conference on Business Process Management (BPM), Sydney, NSW, Australia*, ser. LNCS, vol. 11080. Cham: Springer International Publishing, 2018, pp. 250–267.

[44] A. Burattin, M. Eigenmann, R. Seiger, and B. Weber, "MQTT-XES: Real-time telemetry for process event data," in *Business Process Management (PhD/Demos), in press*, 2020.