

An Event-Centric Metamodel for IoT-Driven Process Monitoring and Conformance Checking

Marco Franceschetti¹[0000–0001–7030–282X], Ronny Seiger¹[0000–0003–1675–2592],
and Barbara Weber¹[0000–0002–6004–4860]

Institute of Computer Science, University of St.Gallen, St. Gallen, Switzerland
{marco.franceschetti|ronny.seiger|barbara.weber}@unisg.ch

Abstract. Process monitoring and conformance checking analyze process events describing process executions. However, such events are not always available or in a form suitable for these analysis tasks, for example for manual processes and (semi-)automated processes whose executions are not controlled by a Process-Aware Information System. To bridge this gap, we propose to leverage Internet of Things (IoT) technologies for sensing low-level events and abstracting them into high-level process events to enable process monitoring and conformance checking. We propose an event-centric metamodel for monitoring and conformance checking systems that is agnostic with respect to process characteristics such as level of automation, system support, and modeling paradigm. We demonstrate the applicability of the metamodel by instantiating it for processes represented by different modeling paradigms.

Keywords: Process monitoring · Conformance checking · IoT.

1 Introduction

Process monitoring and conformance checking are important tasks of Business Process Management that analyze the execution of a process in terms of *events*. Typically, process monitoring is concerned with the run-time analysis of events describing the execution of a process, emitted by a Process-Aware Information System (PAIS) that coordinates the process execution [6]. In turn, conformance checking analyzes the relation between a normative specification of a process behavior, e.g., a process model, and the actual process behavior recorded in the form of a collection of events, e.g., an event log, describing its execution [4].

The ability to monitor and check the conformance of processes is frequently challenged by the nature of the processes at hand and their domains. These present a spectrum of heterogeneity, differing in dimensions such as process structuredness, modeling paradigm, automation level, or process-awareness, with consequences for the availability and quality of the respective events describing the process executions. At one end of the spectrum are greenfield domains in which full automation is possible, activity execution is process-driven with the support of a PAIS, and all relevant events signifying a relevant state change in

a process are generated automatically at the right level of abstraction for analysis: here, monitoring and conformance checking can be realized seamlessly [22]. At the other end of the spectrum are domains characterized by high flexibility, little to no automation, no PAIS support, and no process awareness: here, no process-relevant events are available out of the box and ad-hoc solutions must be deployed to detect events and enable monitoring and conformance checking [27]. Between these extremes of the spectrum are legacy domains, e.g., those of smart factories combining modern and legacy machines. Here, some degree of automation is possible, but the execution is not (fully) controlled by a (single) PAIS, and events are available at varying levels of abstraction [2], ranging from low-level events with no process-awareness to high-level events associated with relevant process state changes: thus, for monitoring and conformance checking, event data collection and processing are required [25].

As indicated in the BPM-meets-IoT manifesto [11], Internet of Things (IoT) technologies can provide a lightweight, transparent, and relatively inexpensive solution to automatically capture relevant execution aspects of processes with limited or no support from a PAIS. Integrating the IoT with processes allows sensing low-level events with no process-awareness; processing these events, e.g., by aggregation and abstraction, allows detecting high-level events that correspond to relevant process state changes, thus enabling monitoring and conformance checking. This may be realized, at design-time, by enriching process specifications with *monitoring points* [9], annotations to the control flow elements associating them with low-level events generated at run-time by the IoT stack. At run-time, the monitoring points capture these low-level events and abstract them into high-level process events describing the process execution [26].

The goal of this work is to define a general IoT-driven process monitoring and conformance checking framework that is agnostic with respect to characteristics such as process structuredness, modeling paradigm, automation level, and PAIS support. To this end, we define an event-centric conceptual metamodel, which only assumes observability of low-level events from IoT for event detection, and availability of a process specification for conformance checking. We validate the metamodel by instantiation in the healthcare and smart manufacturing domains.

This paper is structured as follows: in Section 2, we motivate our work with scenarios from the healthcare and manufacturing domains; in Sect. 3, we propose an IoT-based conceptual metamodel for process monitoring and conformance checking; in Sect. 4, we validate the metamodel; in Sect. 5, we discuss related work; in Sect. 6, we draw conclusions and indicate future research directions.

2 Motivation

Process monitoring and checking the conformance against a normative specification are important aspects of BPM, irrespective of the operational support from a PAIS orchestrating the execution. However, these are challenging tasks for manual processes with high human involvement, e.g., in healthcare, but also for processes with high level of automation, e.g., in smart manufacturing, in which

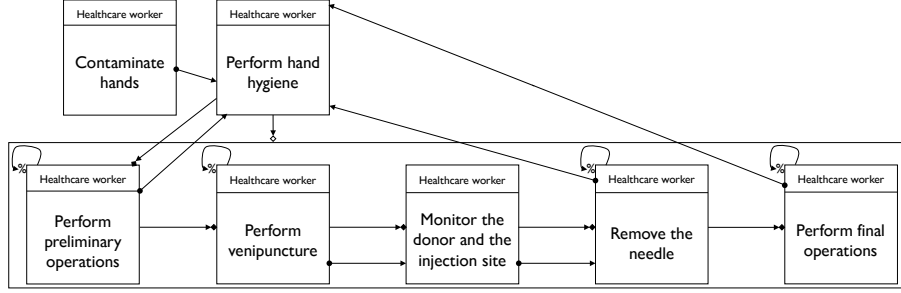


Fig. 1. Example blood donation process represented in DCR graphs notation

data with varying levels of abstraction and little or no process-awareness make it difficult to abstract low-level events to high-level process events [25].

An example from the healthcare domain is the phlebotomy process for blood donation, as defined in the guidelines in [20]. A simplified version of the process composed of five steps, with four additional steps for performing hand hygiene following the indications in [23] is shown in Figure 1 in DCR graphs notation [10] (cf. Sect. 4.1). The process consists of the following sequence of steps: *Perform hand hygiene (before touching donor)*, *Perform preliminary operations*, *Perform hand hygiene (before aseptic procedure)*, *Perform venipuncture*, *Monitor donor and injection site*, *Remove needle*, *Perform hand hygiene (after risk of exposure to body fluids)*, *Perform final operations*, *Perform hand hygiene (after touching donor)*. The process is fully manual and cannot be automated with current technologies. Moreover, since typically healthcare settings are highly unpredictable, additional, unforeseen hand hygiene steps may be required, for example, due to nurses having to deal with emergencies that might contaminate their hands during the course of the process (cf. activity *Contaminate hands* in Fig. 1). Due to their high flexibility, such processes are often specified through guidelines in natural language or declarative models [17]. Frequently, traces of events of these processes are produced offline by the healthcare workers by manually documenting the operations in some Information System at the end of the work shift. However, this approach is prone to tampering, errors, or imprecision; it is also a cause for increased workload for the healthcare workers, with negative effects on their well-being and operations [8]. Thus, monitoring and checking the conformance of such processes is challenging, yet important, e.g., to ensure that whenever a hand hygiene indication is triggered, nurses disinfect their hands before proceeding with other process steps to reduce the risk of infections [23].

An example from the smart manufacturing domain is the storage process in [25]. It consists of the sequence of steps *Get workpiece from pickup*, *Read color*, *Move to High-bay Warehouse*, *Hold at High-bay Warehouse + Store workpiece* (in parallel), *Calibrate*. Unlike the phlebotomy, this process is fully automated and highly structured and predictable, which makes it suitable for modeling with imperative languages such as BPMN [25]. For such processes, sensor events generated by machines at run-time might be available; however, frequently these

events are low-level or at different abstraction levels and lack process-awareness. Thus, at least some event aggregation, abstraction, and correlation steps are required to enable monitoring and conformance checking [25].

Prior work points to the IoT as a possible technology to integrate with processes like the above to enable automated process monitoring and conformance checking [11,25]. In the following, we propose a metamodel for such an IoT-driven monitoring and conformance checking framework.

3 Conceptual Metamodel

3.1 Methodology and Requirements

This work is grounded in the Design Science Research principles and inspired by the metamodel development method proposed in [16]. In line with [16], we held a workshop at the Cantonal Hospital of St. Gallen to identify concerns relevant to stakeholders from the healthcare domain. The outcomes of the workshop are the healthcare scenario presented in Section 2 and the requirements for an IoT-driven monitoring and conformance checking framework. The scenario and requirements were developed independently by the authors and validated by the healthcare stakeholders in a second workshop. The healthcare scenario was used by the authors to develop the metamodel presented here as the basis for a monitoring and conformance checking framework that is agnostic with respect to the characteristics discussed in Sect. 2. Both the healthcare scenario and the smart manufacturing scenario from [25] were used as representatives of the fully manual and legacy domain to validate the metamodel and its agnostic nature.

The workshops identified the suitability of the phlebotomy process described in Sect. 2. In the process, it is important to identify when healthcare workers are required to perform a hand hygiene as a pre- or post-condition for certain phlebotomy steps, as well as when hand hygiene is performed. The identification should be realized using IoT sensors capable of sensing low-level events in the physical world (e.g., the dispensing of hand sanitizer). These low-level events should be processed (e.g., aggregated or translated using complex event processing) into high-level process events associated with the respective activities (e.g., phlebotomy and hand hygiene activities), to produce event logs suitable for process monitoring and conformance checking [26].

3.2 CAIRO Events

Events at the process level are at the core of the proposed metamodel. They are the manifestation of the progress of a process instance and refer to the execution of activities (e.g., the completion of a blood drawing activity) or the occurrence of some happening (e.g., the reception of a message), hence they must be at the basis of process monitoring and conformance checking. A widely accepted definition states that an event is an actual happening that is significant, atomic, and instantaneous [28]. Here, we put forward the idea that, in addition, to enable IoT-driven monitoring and conformance checking in processes like the phlebotomy

process in Sect. 2, an event has to be detectable (*observable*) and it should carry context information. Indeed, events that cannot be detected do not contribute to monitoring or conformance checking, while neglecting context information might lead to ambiguities in interpreting a trace (cf. [15]): for instance, establishing the right event-activity-case associations might not be obvious without context information. Accordingly, we propose to extend the definition of [28]:

Definition 1 (CAIRO Event). *A CAIRO event is an actual (real world) occurrence that is:*

- Contextualized: *it happens within a context, which characterizes it;*
- Atomic: *it either occurs or not, i.e., it has all-or-nothing semantics;*
- Instantaneous: *it happens at a specific point in time;*
- Relevant: *it refers to a relevant process state change (“significant” in [28]);*
- Observable: *its occurrences can be detected.*

Each of the five CAIRO properties is necessary to support comprehensive and unambiguous monitoring and conformance checking of processes, since:

- “AIRO” non-contextualized events might admit uncertainty about their properties and the process instance they belong to, leading to ambiguous trace interpretations. For example, assume a hospital ward with concurrent processes instances involving several healthcare workers, in which a hand hygiene execution event is detected, but the healthcare worker performing it is unknown. In this case, it might be impossible to associate the hand hygiene event with the right resource and process instance with certainty, and to reliably detect hand hygiene indications relevant for conformance checking.
- “CIRO” events lacking the Atomicity property do not conform to the all-or-nothing semantics. These are incompatible with process monitoring and conformance checking, which require clearly identifiable events corresponding to state changes at the process level to match with a process specification.
- “CARO” events are not Instantaneous, i.e., they are not associated with a single timestamp, but to a time interval, leading to uncertainty about their specific time of occurrence (cf. [21]). Consider the phlebotomy process: if the completion of preliminary operations and performance of hand hygiene were not timestamped but had overlapping intervals associated, it would not be possible to identify with certainty in which order these steps were performed.
- “CAIO” events are not Relevant, i.e., they do not reflect process-level state changes. Irrelevant events in a trace might result in ambiguities in the interpretation of the process execution. This is the case, for example, of low-level events: assume, for instance, that a low-level event “*Nurse_proximity < 0.5*” were part of a trace for the phlebotomy process. The interpretation of this event is ambiguous, since it is unclear to which process state change it refers.
- “CAIR” events are not Observable, thus they are not detected, resulting in their absence from the process traces. Incomplete traces missing (unobservable) events might jeopardize monitoring and conformance checking results.

We now propose a conceptual metamodel with CAIRO events at its core.

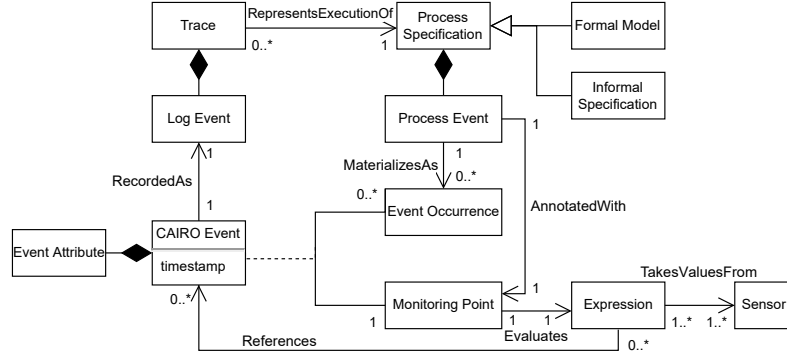


Fig. 2. UML class diagram for the proposed metamodel

3.3 Monitoring and Conformance Checking Metamodel

Figure 2 illustrates the proposed metamodel developed following the principles in [16] and centered around the concept of CAIRO event. A *CAIRO Event* (cf. Sect. 3.2) is a timestamped event at process level of abstraction corresponding to a relevant state change detected by a *Monitoring Point*.

A *Monitoring Point* is an executable object (e.g., an application or a service, cf. [24]) responsible for the abstraction of low-level sensor events with no process-awareness by emitting a *CAIRO Event* at the process level, and the contextualization and correlation of this *CAIRO Event* with a process instance. In line with [1], a *Monitoring Point* evaluates one *Expression* (e.g., a Complex Event Processing query [26]), whose variables are taken from the values of one or more *Sensors* sensing relevant features of the process environment. An *Expression* might *Reference* zero or more prior *CAIRO Events* (and indirectly their event attributes) to identify events unambiguously. This might be useful when knowing prior occurred events allows discriminating between new process-level events. For example, phlebotomy steps *Perform preliminary operations* and *Monitor donor and injection site* might be identified by the same low-level sensor events, but knowing that *Perform venipuncture* occurred before allows identifying the current step as *Monitor donor and injection site*. The *Expression* associated with a *Monitoring Point* is evaluated at run-time based on *Sensor* readings, and a *CAIRO Event* is emitted if it evaluates to **true**.

A *Process Event* represents any control flow element in a *Process Specification* reflecting a relevant state change at process level, e.g., the start of an activity. At design-time, a *Process Event* is *AnnotatedWith* a *Monitoring Point*; one *Monitoring Point* annotates exactly one *Process Event*. At run-time, a *Process Event* may *MaterializeAs* zero or more *Event Occurrences*. For example, *Process Event* “*Perform hand hygiene*” materializes multiple times during one execution of the process in Fig. 1, resulting in multiple instances of *Event Occurrence*. Each such occurrence is detected (by evaluating its *Expression*) by the *Monitoring Point* that annotates the respective *Process Event*; the *Monitoring Point* emits a new

CAIRO Event timestamped with the time of materialization of the *Process Event* as *Event Occurrence*. As *CAIRO Event* is an association class, each of its instances can be reconducted to the *Monitoring Point* that emitted it, and, via *MaterializesAs*, to the *Process Event* whose *Event Occurrence* was detected.

A *Process Specification* is the description of a process defining at least the control flow perspective, e.g., activities and events. To enable monitoring and conformance checking of a broad range of processes (cf. Sect. 2), the metamodel supports processes of different representation paradigms. Hence, the metamodel admits both process specifications in the form of informal specifications in natural language in which *Process Events* can be explicitly and unambiguously identified, such as the activities specified in the guidelines in [20], and formal specifications, e.g., in BPMN. Thus, a *Process Specification* may be either a *Formal Model*, e.g., in BPMN, or an *Informal Specification* in natural language.

A *CAIRO Event* might be associated with *Event Attributes*, whose values are assigned by the respective *Monitoring Point* based on *Sensor* readings. *Event Attributes* are useful for establishing event-instance correlations by storing the process instance context, e.g., as a *CaseId* attribute or through several attributes forming a candidate key identifying the process instance. A *CAIRO Event* is *RecordedAs* a *Log Event*, part of a (potentially partial) *Trace* describing the (potentially partial) execution of an instance of the *Process Specification*. This trace, along with the process specification, makes conformance checking possible.

3.4 CAIRO Events in the Metamodel

In the metamodel, a *CAIRO Event* is contextualized through the composition association with one or more *Event Attributes*, each describing a contextual characteristic. For example, a *CAIRO Event* for a hand hygiene activity may be associated with attributes: *Resource*, indicating the nurse, *Room* indicating the activity location, and *Dispenser* indicating the specific dispenser used.

Atomicity is enforced in the metamodel by the cardinality constraints associating each occurrence of a *CAIRO Event* with exactly one *Process Event*.

A *CAIRO Event* is instantaneous, since each instance of it has a timestamp denoting the point in time when the respective *Process Event* occurred. For example, a *CAIRO Event* instance with UNIX epoch time 1679309356 places the occurrence of the event on March 20, 2023 at 10:49:16.

Relevance is reflected by the fact that a *CAIRO Event* is associated with a *Monitoring Point* for the detection of the *Event Occurrences* of the corresponding *Process Event*, and recorded as a *Log Event* in a *Trace*.

The requirement of being observable is expressed by the chain of associations relating each *CAIRO Event* with at least one *Sensor* deployed in the process environment (via *Monitoring Point* that *Evaluates* an *Expression*). For example, to detect a hand hygiene event, one could deploy a proximity sensor to detect the presence of a healthcare worker near a sanitizer dispenser, and a press sensor to detect the dispensing of the sanitizer. An expression evaluates the changes in the values of these sensor readings, resulting in the emission of a *CAIRO Event* if the values satisfy some condition (e.g., `proximity<0.5 && press_force>0.3`).

4 Metamodel Validation

The proposed metamodel is applicable to the spectrum of domains discussed in Section 2, irrespective of process structuredness, modeling paradigm, automation level, and PAIS support. To demonstrate such general applicability, we instantiated it for the cases of healthcare and smart manufacturing as two representatives of the fully manual domain and legacy domain, respectively [7].

4.1 Healthcare Domain

For the healthcare domain instantiation, we consider the phlebotomy process described in Section 2. The process, like several in this domain, is specified through guidelines in natural language (*Informal Specification* in the proposed metamodel). One could annotate the guidelines by marking all the relevant *Process Events* and annotating them with *Monitoring Points*, based on annotator tools such as the one in [19]. At run-time, these *Monitoring Points* would emit corresponding *CAIRO Events* matching with the annotations in the *Informal Specification*. Here, indeed starting from the guidelines and annotating them with the tool proposed in [19], we derived a formal model in DCR graphs notation [10]: we discuss a metamodel instantiation based on this notation. DCR graphs are a declarative event-based process modeling language, which is particularly suitable for representing healthcare processes requiring a high degree of operational flexibility, e.g., to support the repeated execution of activities. As DCR graph notation is a formal modeling language with defined semantics, the representation of a process in this language is a formal model. Thus, the *Formal Model* concept in the metamodel in Figure 2 corresponds to the *Process concept* in the DCR graph metamodel (cf. [18]). The *Process Event* concept in our metamodel corresponds to the *Event* concept of the DCR graph metamodel.

Fig. 3 shows an excerpt of an instantiation of the metamodel for the case of the phlebotomy process modeled as a DCR graph like in Fig. 1, limited to the *Perform hand hygiene* event. Fig. 3a) shows that at design-time the event is annotated with a *Monitoring Point Hand hygiene detection service*, which has an *Expression* that evaluates the values for the healthcare worker's proximity to the patient ($proximity < 0.5$) and the force exerted on the sanitizer dispenser by pressing on it ($press_force > 0.3$). These values are provided by a proximity sensor and a load cell, respectively. At run-time (cf. Fig. 3b)), once the expression evaluates to **true**, a *CAIRO Event Perform hand hygiene completed*, with

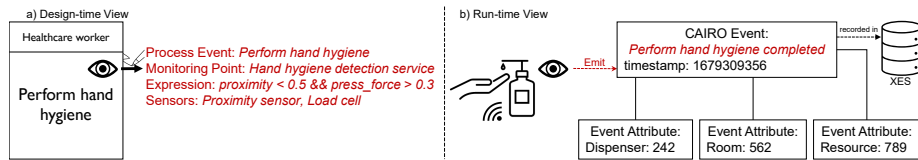


Fig. 3. Metamodel instantiation for an excerpt of a process in DCR graphs notation

associated attributes indicating the dispenser (242), the room (562), and timestamp (1679309356) is emitted. The CAIRO event is recorded as a log event in a trace in a XES log to enable conformance checking against the model in Fig. 1.

4.2 Smart Manufacturing Domain

For the smart manufacturing scenario, we considered the production process from [26]. In a metamodel instantiation with processes encoded in BPMN 2.0 (here the modeling language is chosen according to the discussion in Sect. 2), the *Process Event* concept in our metamodel corresponds to the *Flow node* concept of the BPMN metamodel, which entails BPMN process events, gateways, and activities. The *Formal Model* concept in our metamodel corresponds to the *Process* concept in the BPMN metamodel. The *Monitoring Point* concept in our metamodel corresponds to the *Monitoring* concept in the BPMN metamodel.

While the BPMN 2.0 standard states that process events may be also activity events (e.g., the event associated with the start of an activity), it does not define in its metamodel any such event. Here, for simplicity and correspondence to the metamodel, we consider the execution of a BPMN activity to be associated with—and detected at—the completion of the activity: the activity completion detected by a *Monitoring Point* marks the occurrence of a *CAIRO Event* (timestamped with the activity completion time to ensure the “I” property). This is compatible with many event logs found in the literature, which associate activity executions with their completion time only (cf. [4] for examples). Nevertheless, if fine-grained activity event detection, e.g., differentiating between start and end, is a requirement, the BPMN metamodel could be extended in line with the extension proposed in [1].

Fig. 4 shows the instantiation of the metamodel for the simplified production process in BPMN 2.0 [26]. Fig. 4a) depicts the design-time view regarding the extension of a standard BPMN model with a *Monitoring Point* to detect the end of the *Burn Workpiece* activity. The *Expression* associated with the *Monitoring Point* refers to the oven temperature sensor and the status of its door. Fig. 4b) presents the run-time view of executing one instance of the *Burn Workpiece* activity. Here, the *Monitoring Point* detects the end of the activity based on the modeled expression and emits a *CAIRO Event Burn Workpiece ended* with the corresponding timestamp.

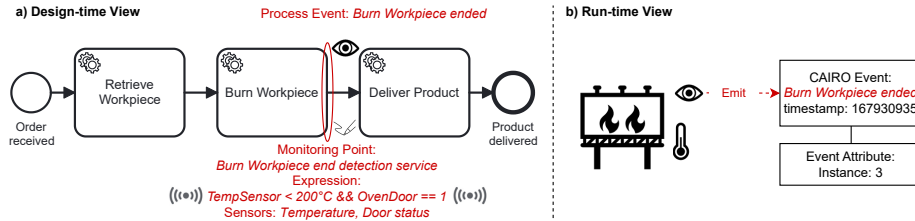


Fig. 4. Instantiation of the metamodel for a BPMN 2.0-based process

5 Related Work

A pivotal work calling for the integration of Internet of Things (IoT) technologies with Business Process Management (BPM) is the IoT-meets-BPM manifesto [11]. Our work relates to several of the challenges indicated in the manifesto, such as *Placing Sensors in a Process-Aware Way*, *Support for Managing Manually Executed Physical Processes*, and *Improving Online Conformance Checking*.

In [5] a systematic literature review can be found, which surveys existing modeling notations for IoT-aware business processes and potential application domains for the integration of IoT technologies with BPM. These could be used as a basis for choosing the most adequate language for modeling a process to be augmented with monitoring points based on IoT technologies. Another work towards the BPM-IoT integration is [13], which proposes an extension of BPMN 2.0 to model, enact, and monitor IoT-aware processes with an addition for IoT-driven events presented in [14]. The work builds on the assumption that a PAIS is available for the orchestration of processes. While the metamodel for monitoring and conformance checking proposed here was motivated by domains in which a PAIS is not available, the methods proposed in [13] could still be applied to a system built following our metamodel. The metamodel proposed in [3] is aimed at bridging process mining with IoT, and focuses on how low-level data is captured by IoT devices and how these data are used to generate an event log suitable for offline process mining tasks. Here, we do not assume monitoring points and the transformation of low-level IoT data into events relevant in the process execution to preclude online monitoring and online conformance checking.

Our proposed metamodel leverages the concept of monitoring point for the detection of occurrences of process events. Monitoring points have been first introduced by Herzberg [9], and subsequently applied to BPMN for process monitoring in [1]. Authors in [1] extend the BPMN metamodel to enable a low-level detection of activity lifecycle events, thus their monitoring metamodel in [1] could be integrated with ours to achieve a fine level of granularity for process event detection. Monitoring points have been adopted also in [12] for conformance checking of processes in a hospital environment, showing the importance of engaging with domain experts for the definition of the monitoring points in a process model. Nevertheless, the work does not consider IoT technologies for the implementation of monitoring points, but manual methods. In contrast, the work in [24] proposes a method to automatically generate services for an IoT-based activity detection; thus, the method is a possible candidate for the generation of instances of the *Monitoring Point* concept in our metamodel.

In Sect. 3.2, we argued the importance of the *contextualized* property of CAIRO events. In line with our position is the work in [15], which proposes a framework for the contextualization of event-activity mappings—a task that is of high relevance when process activities are observed through the occurrence of low-level events detected by IoT sensors, as the proposed metamodel assumes. The proposed framework could be used as a basis for the identification of event attributes that must be detected to contextualize CAIRO events in systems designed following the proposed metamodel.

6 Conclusion and Future Work

Process monitoring and conformance checking are often challenged by process domain characteristics such as level of system support and process structuredness. We presented a general event-centric metamodel to support process monitoring and conformance checking driven by the Internet of Things for the detection of process events, overcoming challenges posed by these characteristics. At the core of the metamodel, we introduced the notion of CAIRO event, emphasizing the significance of the CAIRO event properties. We validated the metamodel by instantiating it for two significantly different scenarios, demonstrating the agnostic nature of the metamodel with respect to domain characteristics.

In future work, we plan to implement a monitoring and conformance checking approach for the phlebotomy scenario in a laboratory simulation environment, and to check process conformance based on informal process specifications. Additionally, we will further investigate the role of the *Contextualized* property of CAIRO events, in particular for the disambiguation of detected events.

Acknowledgments

This work has received funding from the Swiss National Science Foundation under Grant No. IZSTZ0_208497 (*ProAmbitIon* project). The authors thank Estefanía Serral Asensio and the workshop participants for their feedback.

References

1. Baumgraß, A., Herzberg, N., Meyer, A., Weske, M.: Bpmn extension for business process monitoring. EMISA 2014 (2014)
2. Beerepoot, I., Di Ciccio, C., Reijers, H.A., Rinderle-Ma, S., Bandara, W., et al.: The biggest business process management problems to solve before we die. *Computers in Industry* **146**, 103837 (2023)
3. Bertrand, Y., Weerdt, J.D., Serral, E.: A bridging model for process mining and iot. In: ICPM Intern. Workshops. LNBIP, vol. 433, pp. 98–110. Springer (2021)
4. Carmona, J., van Dongen, B., Solti, A., Weidlich, M.: Conformance checking. Switzerland: Springer. (2018)
5. Compagnucci, I., Corradini, F., Fornari, F., Polini, A., Re, B., Tiezzi, F.: Modelling notations for iot-aware business processes: A systematic literature review. In: BPM 2020 Intern. Workshops. LNBIP, vol. 397, pp. 108–121. Springer (2020)
6. De Medeiros, A.A., Pedrinaci, C., Van der Aalst, W.M., Domingue, J., Song, M., Rozinat, A., Norton, B., Cabral, L.: An outlook on semantic business process mining and monitoring. In: On the Move to Meaningful Internet Systems 2007: OTM 2007 Workshops, Proceedings, Part II. pp. 1244–1255. Springer (2007)
7. Franceschetti, M., Seiger, R., González González, M.J., García-Ceja, E., et al.: Proambition: Online process conformance checking with ambiguities driven by the internet of things. In: CEUR Workshop Proceedings. vol. 3413, pp. 52–59 (2023)
8. Golay, D., Karlsson, M.S., Cajander, Å.: Negative emotions induced by work-related information technology use in hospital nursing. *Computers, Informatics, Nursing* **40**(2), 113 (2022)

9. Herzberg, N.: Integrating events into non-automated business process environments. Doctoral thesis, Universität Potsdam (2018)
10. Hildebrandt, T.T., Mukkamala, R.R.: Declarative event-based workflow as distributed dynamic condition response graphs. arXiv preprint arXiv:1110.4161 (2011)
11. Janiesch, C., Koschmider, A., Mecella, M., Weber, B., Burattin, A., Di Ciccio, C., et al.: The internet of things meets business process management: a manifesto. *IEEE Systems, Man, and Cybernetics Magazine* **6**(4), 34–44 (2020)
12. Kirchner, K., Herzberg, N., Rogge-Solti, A., Weske, M.: Embedding conformance checking in a process intelligence system in hospital environments. In: *BPM Joint Workshop, ProHealth/KR4HC. LNCS*, vol. 7738, pp. 126–139. Springer (2012)
13. Kirikkayis, Y., Gallik, F., Reichert, M.: A holistic framework for iot-aware business processes. In: *BPM Int. Workshops. LNBIP*, vol. 460, pp. 89–100. Springer (2022)
14. Kirikkayis, Y., Gallik, F., Seiger, R., Reichert, M.: Integrating iot-driven events into business processes. In: *Advanced Information Systems Engineering: 35th Intern. Conf., CAiSE 2023, Zaragoza, Spain, June 12-16, 2023. Springer* (2023)
15. Koschmider, A., Mannhardt, F., Heuser, T.: On the contextualization of event-activity mappings. In: *BPM 2018 International Workshops, Revised Papers. LNBIP*, vol. 342, pp. 445–457. Springer (2018)
16. Kurpjuweit, S., Winter, R.: Viewpoint-based meta model engineering. In: *Proceedings of the 2nd International Workshop on Enterprise Modelling and Information Systems Architectures (EMISA'07). LNI*, vol. P-119, pp. 143–161. GI (2007)
17. Lenz, R., Peleg, M., Reichert, M.: Healthcare process support: achievements, challenges, current research. *IJKBO* **2**(4) (2012)
18. López, H.A., Simon, V.D.: How to (re) design declarative process notations? a view from the lens of cognitive effectiveness frameworks. In: *PoEM-Forum 2022* (2022)
19. López-Acosta, H.A., Hildebrandt, T., Debois, S., Marquard, M.: The process highlighter: From texts to declarative processes and back. In: *CEUR Workshop Proceedings*. pp. 66–70. CEUR Workshop Proceedings (2018)
20. Organization, W.H.: Who guidelines on drawing blood: best practices in phlebotomy (2010), <https://apps.who.int/iris/handle/10665/44294>
21. Pegoraro, M.: Probabilistic and non-deterministic event data in process mining: Embedding uncertainty in process analysis techniques. arXiv preprint arXiv:2205.04827 (2022)
22. Rinderle-Ma, S., Mangler, J.: Process automation and process mining in manufacturing. In: *BPM 2021, Proceedings*. pp. 3–14. Springer (2021)
23. Safety, W.P., Organization, W.H., et al.: Who guidelines on hand hygiene in health care. Tech. rep., World Health Organization (2009)
24. Seiger, R., Franceschetti, M., Weber, B.: Data-driven generation of services for iot-based online activity detection. In: *International Conference on Service-Oriented Computing. Springer* (2023)
25. Seiger, R., Franceschetti, M., Weber, B.: An interactive method for detection of process activity executions from iot data. *Future Internet* **15**(2) (2023)
26. 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 Int. Enterprise Distr. Object Computing (EDOCW)*. pp. 20–26. IEEE (2020)
27. Tarallo, A., Mozzillo, R., Di Gironimo, G., De Amicis, R.: A cyber-physical system for production monitoring of manual manufacturing processes. *International Journal on Interactive Design and Manufacturing (IJIDeM)* **12**, 1235–1241 (2018)
28. Wasserkrug, S., Gal, A., Etzion, O., Turchin, Y.: Complex event processing over uncertain data. In: *DEBS 2008*. vol. 332, pp. 253–264. ACM (2008)