A Bridging Model for Process Mining and IoT

Yannis Bertrand(✉), Jochen De Weerdt, and Estefanía Serral

Research Centre for Information Systems Engineering (LIRIS), KU Leuven,
Warmoesberg 26, 1000 Brussels, Belgium
{yannis.bertrand,jochen.deweerdt,estefania.serral}@kuleuven.be

Abstract. Contextualisation is an important challenge in process mining. While Internet of Things (IoT) devices are collecting more and more data on the physical context in which business processes are executed, the IoT and process mining fields are still considerably disintegrated. Important concepts, such as *event* or *context*, are not understood in the same way, which causes confusion and hinders cooperation between the two domains. Based on IoT ontologies and business process context models, this paper proposes a model to bridge the conceptualisation gap between the IoT and the process mining fields. The model defines the necessary concepts and relationships to build process mining techniques that take the physical context into account. As a first validation, the model is used to describe a lifelike process example, showing how IoT data and process events are related. Using this conceptualisation, both practitioners and researchers from the IoT and the process mining communities can reason about the use of IoT data in process mining and find support for data understanding, event abstraction and IoT and process data integration.

1 Introduction

Although the potential of IoT data for process mining (PM) has been recognised, the relationships between IoT data and event logs has not been made explicit yet. This lack of deeper knowledge about these relationships, at the conceptual level, is part of a more general conceptual issue in PM boiling down to the question: what is an event? Previous works by different researchers have identified various conceptions of "event", which differ on their semantic level, e.g. micro events, high- versus low-level events, etc.; on their "scope", e.g. context events or process/control-flow events; or on whether the event includes the data associated with it, as in XES [9], etc. In addition to this, the same kind of conceptual challenges arise when using IoT data to retrieve the context of a process, as the understanding of context differs in the fields of IoT and PM.

These conceptual issues causes practical problems in PM. Process models discovered from event logs at an inadequate semantic level, i.e., too detailed or too coarse, can be too complex or too simple, which often make them unpractical and unfaithful to the reality. Then, confusing process and context events can also have an impact on the resulting process model, by either omitting activities of

© The Author(s) 2022
J. Munoz-Gama and X. Lu (Eds.): ICPM 2021 Workshops, LNBIP 433, pp. 98–110, 2022.
https://doi.org/10.1007/978-3-030-98581-3_8

the process, or over-complexifying the model. An example of this can be found in Dees et al. [4], who showed that translating three types of events of the Sepsis dataset [14] into context events, reduced the number of discovered variants to one-fourth, while annotating the model with information on these context events made the model as informative as a model considering all events as process events. Employing IoT data in PM can cause both types of problems: models at an unsuitable level of granularity, or models that confuse the context and the control-flow, as IoT data have to be abstracted to the adequate semantic level (i.e. that of the process) and data on the control-flow of the process have to be carefully distinguished from data on the context of the process.

The goal of this paper is to discuss the important concepts of event and context in PM, highlighting the difference in their understanding in the domains of IoT and PM. Using these concepts, we propose a model defining the links between these concepts and between the IoT and PM conceptual views, based on IoT ontologies, context models from business process management (BPM) and PM data models. The rest of the paper is structured as follows. Section 2 reviews the existing literature, focusing on IoT ontologies and business process (BP) context models. Next, in Sect. 3, the ambiguities in some important concepts are analysed. Section 4 presents a conceptual model defining and linking important concepts of IoT and PM. After this, a use-case of the model is presented in Sect. 5, and a comparison with some related works is done in Sect. 6 Finally, Sect. 7 provides a brief conclusion with some propositions for future works.

2 Background

In this section, previous works on the modelling of IoT and PM are introduced. First, relevant IoT ontologies are discussed, before addressing BP context models. Literature on process mining in IoT environments is discussed in detail in Sect. 6.

2.1 IoT Ontologies

Recently, the focus in IoT ontologies has shifted from the creation of ontologies that are as complete as possible (e.g. the Semantic Sensor Network (SSN) ontology[1]) to the development of new ontologies that are simpler and more practical (e.g. IoTStream [7]). Two such ontologies are the Sensor, Observation, Sample and Actuator (SOSA) ontology [10] and IoTStream [7].

SOSA proposes three perspectives: the sensor, observation and actuator perspectives [10]. IoTStream is a more specific ontology, inspired by SOSA, that focuses on the treatment of streaming data [7]. Both of these ontologies are *event-centric*, in the sense that they focus on data generation and treatment, and less attention is paid to the devices and platforms IoT relies on.

[1] https://www.w3.org/TR/2017/REC-vocab-ssn-20171019/.

2.2 Business Process Context Modelling

One of the first BP context models was proposed by Rosemann et al. [16]. In this paper, the authors described an onion model where context was split in four layers (listed from closest to farthest from the process): immediate context, internal context, external context and environment context. van der Aalst developed an akin onion model a few years later [1]. Another relevant representation was proposed by Ghattas et al. [8], who extended the generic process model (GPM) with a context model $C = <I, X>$ that links each instance of the process with 1) I, the initial state of its variables and 2) X, the inputs from the external environment that affect the instance.

In a recent review paper, Brunk [3] proposed a taxonomy for BP context data with six dimensions: time, structure, origin, relevance, process relation and runtime behaviour. The dimensions proposed describe traditional BP context accurately, but they are not suitable for IoT data. For instance, typical IoT context variables such as *temperature*, can hardly fit in the origin dimension.

Another approach was followed by van der Werf et al. [23], who represented the context of a BP in a domain model.

However, these papers do not discuss context based on sensor data in particular. This is done by Koschmider et al. [12], who model context information in a hierarchy that contains three elements: raw data, simple context information and complex context information.

3 Conceptual Ambiguity in IoT and PM

To bring the IoT and PM fields of study together, there needs to be an agreement on some common fundamental concepts. However, a recurrent issue when trying to bridge IoT and PM is that some common concepts are not understood homogeneously across both domains, such as the concept of *context*. This lack of homogeneity can create confusion and undermine the integration of the two fields. In this section, we start by defining the concept of IoT data, explaining next the concepts of *context* and *event*. We especially highlight the differences in understanding of those concepts by the IoT and PM fields.

3.1 IoT Data

To understand IoT data, we start with the concept of IoT. A profusion of definitions exists, and the one we retain is from Dorsemaine et al., which is synthetic and explicitly mentions the various aspects of IoT: IoT is a "Group of infrastructures interconnecting connected objects and allowing their management, data mining and the access to the data they generate." [6]. Relying on this definition of IoT, we can say that IoT data are all the data collected by the objects belonging to connected infrastructures. These data describe physical objects (the so called *Things*) or the physical environment. Examples of IoT data are the temperature in a refrigerated area, the location of a package in a warehouse, the heart rate of a patient, etc.

3.2 Context in PM vs Context in IoT

Context was defined by Dey [5] as: "any information that can be used to characterise the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves." This broad definition is referred to in both IoT and PM, but what it means in practice substantially differs in each domain, as the situation is viewed from different angles.

In IoT, the notion of context refers to the physical context of a system. Data about the physical context are usually gathered by sensors that measure e.g. the location of objects, the ambient temperature, the movement of people or objects, etc. In PM and BPM, the context that is typically taken into account is the context of the BP that is analysed. In general, this translates to factors that impact the design or the execution of the process [1,16]. However, within the PM field, the understanding of context can differ from this definition. Most papers describe context-aware PM approaches (e.g. [1–3,23]), and understand context in the same way as in BPM. But we can also find some papers describing PM approaches applied to context-aware environments (e.g. [13,20]), which understand context like in IoT, as the physical context.

There is thus a discrepancy in the understanding of context. As a consequence, some IoT context data do not fit in taxonomies describing business process context variables; e.g., a variable such as the temperature in a room can hardly fit in any of the categories defined by Rosemann et al. [16]. However, such a variable can be useful to describe the context of a business process, and should be taken into account in PM.

On the other hand, not all the parameters that are measured by sensors in context-aware environments are relevant for PM: only those that impact the process are. The notion of context that should be used in PM using IoT is therefore the business context as understood in BPM, including relevant physical parameters, which have so far been largely overlooked.

3.3 Process Event vs IoT Event

Fundamental to PM, the concept of event is also very important in IoT. A distinction between the two acceptations of the term is recognised [11,20]. This distinction is usually limited to placing events from IoT at a lower abstraction level than events in PM. However, both definitions differ in more than that if we do a more detailed evaluation.

On the IoT side, an event can be defined as a time-value set [17]:

$$<key, value, destination, generation\_time, release\_time>$$

where the semantics of a particular event are specified by the key-value pair. Notice that this is a *data* definition, characterising a data construct.

On the PM side, a common definition of the term event is the one of the XES Standard: "Events represent atomic granules of activity that have been observed during the execution of a process. As such, an event has no duration" [9].

We can especially notice that an event is more broadly defined in the IoT literature than in the PM literature, as an event in IoT can represent a very wide range of things, depending on the key-value pair. An event as understood in IoT can be linked either with an event of the process (e.g. a patient taking a blood test), or with a context variable (e.g. the patient's insulin level), whereas an event as understood in PM can only correspond to the first one. This means that, as such, PM algorithms cannot simply be run on IoT data.

To connect these two concepts, and to reconcile the views of the IoT and PM domains on context, we developed a conceptual model to help to build a common comprehension of the structure of information to IoT and PM.

4 Connecting IoT and Process Mining: A Conceptual Model

In this section, we propose and discuss a conceptualisation that shows the link between IoT and PM, based on: (1) the concepts defined in the previous section, (2) IoT ontologies, and (3) BP context data models. To create this model, we took inspiration from the methodology described by Noy and McGuinness [15].

Following this methodology, as a first step, we formulated the requirement that our model had to fulfil: our goal is to model the link between data generated from, or captured by, IoT devices and PM event logs. The model should be able to represent the different concepts involved in IoT-enhanced PM, and to distinguish different concepts (i.e., different types of events) that are often confused in PM.

Then, we reviewed existing models, focusing on lightweight and event-centric IoT ontologies (IoTStream [7], SOSA [10]), as well as context data models (e.g., [3,16]).

After this, our third step was to look for recurrent terms and concepts. We searched for concepts that were often present in IoT ontologies and for concepts that were often present in the BP context literature. We proposed archetypal classes of objects in IoT and BP context, as well as concepts that were common to both IoT ontologies and BP context models. Recurrent concepts in IoT ontologies are sensor/device, observation, observable property, and analytics, while recurrent terms in BP context are context variable, event log, and data.

Events are central in both IoT and PM. However, as mentioned in Sect. 3, this concept is difficult to grasp and it is not understood in the same way in both fields. We propose to use a generic definition of event, which both IoT and PM experts can accept: "An **Event** is an actual occurrence or happening that is significant (i.e. it falls within a domain of interest to the system), instantaneous (i.e. it takes place at a specific point in time), and atomic (i.e. it either occurs or not)" [22] . This definition acknowledges that any occurrence that is actual (i.e. happens in the real world), atomic and instantaneous, only needs to be significant to a certain purpose or in a certain application to be an event. Events in PM are significant for the execution of the process, while events in IoT measure relevant factors of the physical environment. Examples of events complying with this definition include e.g. the termination of an activity, a report on daily sales, the entrance of a person in a certain area, or the switching on or off of a lamp.

The fourth step was to create the classes of our conceptual model, and to link them together. The result can be seen on Fig. 1. The model, built as a UML class diagram, is constituted of two main parts: the first one, from *Observable Property* to *Event*, describes how data are captured and managed by IoT devices (following IoT ontologies), and the second one, from Event to *Event Log Entry* (including *Process-Aware IS* and *IS data entry*) shows how data are processed in PM to create a contextualised event log. A link is made through the common construct of Event. Next, we define the different terms represented in the model.

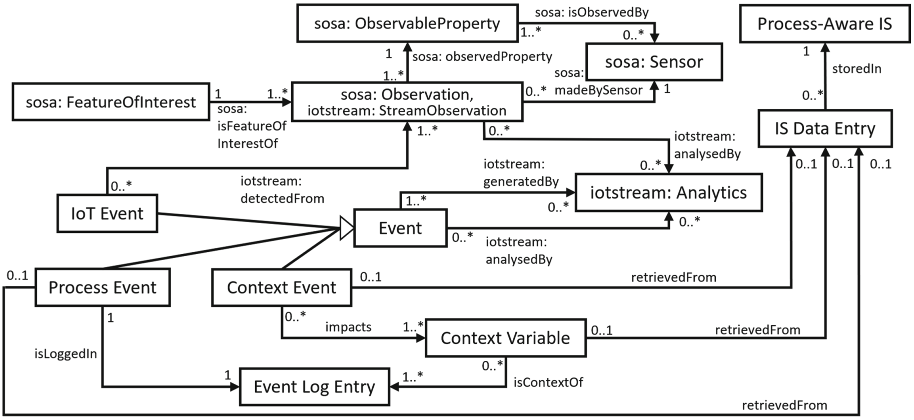


Fig. 1. Core of the model linking IoT with PM

A **Sensor** is an IoT device that measures the state of a real-world phenomenon, named Observable Property in SOSA [10]. An **Observable Property** is an observable quality (property, characteristic) of a *Feature Of Interest*. Examples of Observable Properties are: the outside temperature, the location of a truck, the weight of a container. A **Feature Of Interest** is the thing whose property is being estimated or calculated in the course of an observation (e.g. the container whose weight is measured). An **Observation** is a measurement of an Observable Property; it provides the result of estimating or calculating a value of an observable property (e.g. the measured weight of the container). The case of an actuator (an IoT device that can interact with the environment) generating the data can be modelled similarly, with actuator, actuatable property and actuation classes that mirror the sensor, observable property and observation. To avoid overloading the model, this is omitted in the figure.

IoT Events can be derived from Observations or other IoT events, and are a specialisation of Event that is defined as an instantaneous change in a real-world phenomenon that is monitored by a Sensor. Several IoT Events can be detected from the same Observation, e.g., an observation of the Observable

Property "temperature" can trigger the IoT event "temperature decreases to 0 °C" and "it is freezing".

Likewise, an IoT Event can be created directly by a change in the Observations of a Sensor (e.g. "temperature reaches 23 °C" is directly linked to the observation "23" of a temperature sensor), or it can be derived by processing one or several observation(s) from the same sensor (e.g. "temperature has increased" results from the processing of two temperature observations), with *Analytics* techniques. **Analytics** is an umbrella term from the IoTStream ontology [7] used here to describe any technique that allows the extrapolation of an Event from an Observation or another Event, such as e.g. event abstraction, complex event processing, database query, stream annotation, activity recognition, event-activity and event-case correlation, aggregation techniques, filtering techniques or machine learning algorithms.

Two other types of Events (*Context Event* and *Process Event*), which typically have richer semantics, can be derived from IoT Events using Analytics. A **Process Event** is an instantaneous change of state in the transactional lifecycle of an activity. This type of event corresponds to the usual notion of event in PM. Note that we decouple the occurrence of the change of state in the activity lifecycle and the attributes that are usually present in event data structures. Conceptually, we consider the attributes independent of the existence of the process event, and we model them separately (with the Context Event and Context Variable classes). An example of Process Event is the arrival of a package at a storage facility, which could have as attribute the size or the weight of the package.

A **Context Event** is an instantaneous change in a real-world phenomenon (deduced from an IoT Event or an IS Data Entry), that has an impact on the execution of the process (i.e. it impacts a *Context Variable*), but that does not change its control-flow state. Examples of Context Variables include the location of a package in a delivery process, the vital signs of a person in a health monitoring process, etc. An example of Context Event would be, e.g., a package has arrived at a certain area, which makes the package ready for pick up.

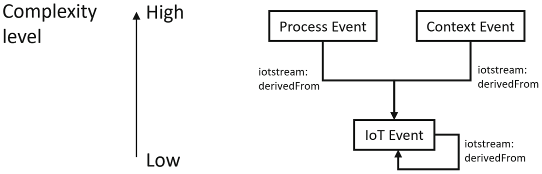


Fig. 2. Hierarchy of event specialisations in the model

Events in the model follow a hierarchy based on their complexity, as shown on Fig. 2. A higher-level event can be deduced from one or several lower-level events, and similarly a lower-level event can be the basis of one or several higher-level events. This mechanism is inspired by CEP [22], as was also suggested by

Soffer et al. [18]. IoT Events can cascade until a deduced event has a direct relationship with the process, i.e. it is a Process or a Context Event. Note that an Analytics technique, possibly trivial, is required to derive an Event from one or several other Events.

As stated earlier, a **Context Variable** is a parameter that has an influence on the execution of the process. Brunk [3] distinguished four categories of context variables, depending on their relationship with the process: activity-related, process event-related, control-flow-related, and artefact-related. Note that Context Variables might be at the level of activities, process instances, or even the overall process.

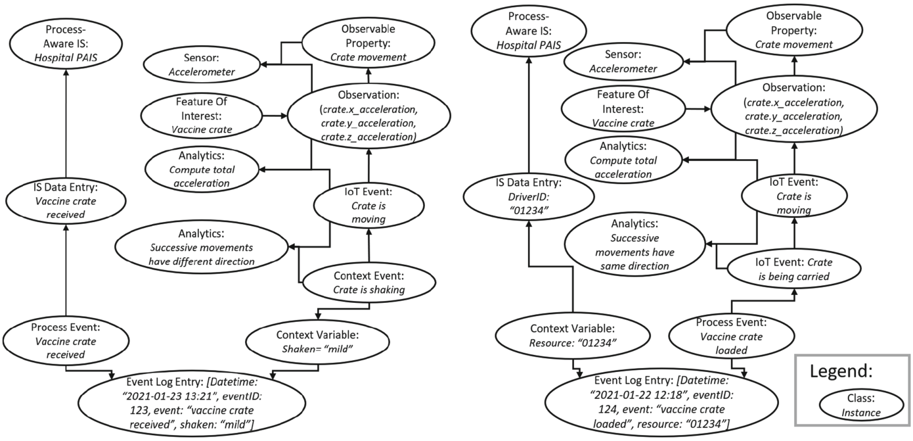
Process-Aware Information System (PAIS) and **Information System (IS) Data entry** represent the traditional PM data sources. A PAIS is an IS that records process data (i.e., IS data entries). An IS Data entry can relate to three classes: Process Event, Context Event and Context Variable. The link between PAIS and Process Event is the usual path of data used in PM, which are entered in the PAIS at runtime and later extracted to form an event log. Usually, in PM, data used as context variables are data retrieved from the IS and are considered rather static. But this does not mean that such context variables may not be subject to change. Take, for example, the amount of a claim in a claim handling process. The claim amount is usually assumed fixed, but it can actually change, as a result of, e.g., a reevaluation of the claim by an expert. This is why IS Data Entry is linked with both Context Variable and Context Event.

Finally, **Event Log Entry** is the point where Context Variables are linked with Process Events in the contextualised event log, which would contain logs of process events together with the context in which they took place. Note that, although these classes are not linked with Analytics in Fig. 1, it does not necessarily mean that Analytics are not used. Analytics is linked with Observation and Event to emphasise the importance of Analytics techniques to derive Events from Observations or other Events, but it may be that, e.g., event correlation or data fusion techniques are necessary to match a Process Event with the relevant Context Variables, or to derive an IoT Event from an Actuation. This is omitted in the figure for the sake of clarity.

5 Use Case Validation

In this section, we present a lifelike use case showing how the path between IoT data and a PM event log can be represented using this conceptualisation.

Consider the process of transporting *Moderna* vaccines from their production facility to the patients in Belgium. The vaccines are manufactured in a main production plant in the US, before being shipped to a central storage facility in Belgium. The vaccine crates are then dispatched to local vaccination centres where each dose is administered to a patient. The vaccines being particularly fragile, one would like to keep track of shocks and bumps experienced by the crates during transport, to detect during which activities most shocks are incurred, and improve the process to minimise this number. Figure 3 shows how to use our model to map different concepts from the raw output of an IoT sensor to an entry in the event log.


Process-Aware IS: *Hospital PAIS*
Observable Property: *Crate movement*
Sensor: *Accelerometer*
Feature Of Interest: *Vaccine crate*
Observation: (*crate.x_acceleration, crate.y_acceleration, crate.z_acceleration*)
Analytics: *Compute total acceleration*
IS Data Entry: *Vaccine crate received*
IoT Event: *Crate is moving*
Analytics: *Successive movements have different direction*
Context Event: *Crate is shaking*
Process Event: *Vaccine crate received*
Context Variable: *Shaken= "mild"*
Event Log Entry: *[Datetime: "2021-01-23 13:21", eventID: 123, event: "vaccine crate received", shaken: "mild"]*
Process-Aware IS: *Hospital PAIS*
Observable Property: *Crate movement*
Sensor: *Accelerometer*
Feature Of Interest: *Vaccine crate*
Observation: (*crate.x_acceleration, crate.y_acceleration, crate.z_acceleration*)
Analytics: *Compute total acceleration*
IS Data Entry: *DriverID: "01234"*
IoT Event: *Crate is moving*
Analytics: *Successive movements have same direction*
IoT Event: *Crate is being carried*
Context Variable: *Resource: "01234"*
Process Event: *Vaccine crate loaded*
Event Log Entry: *[Datetime: "2021-01-22 12:18", eventID: 124, event: "vaccine crate loaded", resource: "01234"]*
Legend:
Class: *Instance*

(a) (b)

Fig. 3. Example instances for the vaccine shipment process.

While the Features Of Interest *Vaccine crates* are being handled, their Observable Property *Crate movement* is recorded by an *Accelerometer* Sensor. Observations of this sensor are triplets (x, y, z) containing the acceleration in the three dimensions of space. The IoT Event *Crate is moving* can be derived from such an Observation. Comparing the movement with previous movements can tell if the crate is being shaken (which corresponds to the Context Event *Crate is shaking*, Fig. 3(a)) or if it is being displaced (which could detect a Process Event *Crate is loaded*, Fig. 3(b)), depending on the direction of consecutive movements (consecutive movements in the same direction correspond to a displacement, while consecutive movements in different directions indicate a tremor). The Context Event *Crate is shaking* impacts the Context variable *Shaken*, which after a certain amount of shocks becomes equal to "mild", to reflect the magnitude of shaking undergone by the crate (part (a) on Fig. 3).

Recording this Context Variable with each Process Event allows determining which activities shake the crates most. It can also be crossed with other Context Variables (e.g. the *Resource* driving the truck transporting the vaccines, an activity-related Context Variable that can be found as an IS Data Entry of the PAIS), to determine under which circumstances shocks are minimised (part (b) on Fig. 3).

This helps in 1) retracing the sources of the event log (IoT and PAIS), and 2) getting a deeper understanding of the links between the raw accelerometer data and the process events and context variables in the event log, as well as 3) distinguishing process events (e.g. *Vaccine crate loaded*) from context information (e.g. *Shaken*).

6 Related Work

Most of the literature that tackles PM using IoT data proposes step-by-step frameworks to extract an event log from low-level IoT data, such as those proposed by Koschmider et al. [11], Trzcionkowska and Brzychczy [21] or Soffer et al. [18]. There are differences from one framework to the other, but typical steps included in these frameworks are preprocessing the raw data, activity recognition or discovery and event abstraction. These works differ from ours as 1) they focus on the processing of the data (the "how") while we concentrate on the data themselves (the "what"), and 2) although contextual sensor data are included, their use is limited to supporting the discovery of activities or the abstraction of events, as in [12] or [19], i.e. the IoT data are not used to mine the context of the process model.

E.g., using the framework of Koschmider et al. to model the use-case in Sect. 5 would yield the following: in step 1, accelerometer data would be correlated with activity "vaccine crate loading". Step 2 would extract the rule that successive movements in the same direction characterise the "vaccine crate loading" activity, and step 3 would apply this rule to the whole sensor data to create an event log with the activity instead of the sensor data. The Process Events derived are similar to these described by our model, but many aspects, such as the context information, e.g. the Context Variable "shaken", are not included. The main steps of these frameworks can also be linked with some parts of the model; see Fig. 4.

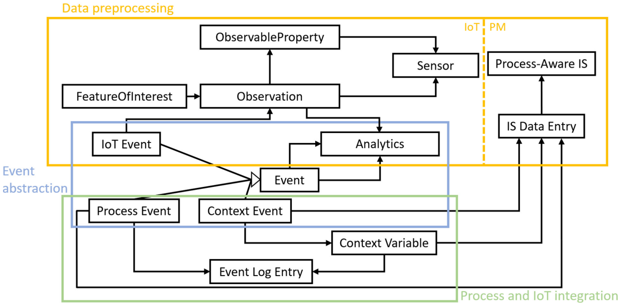


Fig. 4. Translation of typical IoT PM frameworks steps on our model

Furthermore, existing BP data models cannot model the use-case either. XES [9] is at a high level of abstraction, and is designed to store Process events only. Extensions of XES exist, among which the micro-event extension, which makes it possible to define a hierarchy of events, but it does not make it possible to link multiple higher-level events to a lower-level event[2]. The object-centric event log (OCEL)[3], a new PM data paradigm based on the concept of object, is also unsuitable for context events, as each event has to be linked with one and only one activity, which is not the case for many context events, such as e.g. the weight of a package. The context-aware GPM [8] can represent Context events, but does not distinguish them from Process events and is more coarse-grained than our model. For instance, using the context-aware GPM [8], the context in the vaccine shipment example would be modelled with: I = {{crate_shaken}, {resource}} and X = {{crate is shaking}, {vaccine crate loaded}, {vaccine crate received}}. This representation includes all the final elements of the context but, again, it misses the traceability provided by our model, and cannot include IoT metadata. Lastly, neither XES nor the context-aware GPM can represent the hierarchy of events.

7 Conclusion

In this paper, we pleaded for the use of IoT as source of context information in PM. After analysing the existing relevant models and current ambiguities affecting very important concepts, we proposed a conceptual model that defines and connects IoT and PM. As such, the model provides definitions to foster understanding between the IoT and PM community, and enables traceability between the two types of data. This is a first step towards properly understanding the relationship between IoT data and process data in order to improve their further analysis using PM. Also note that the reuse of ontologies and models from the literature automatically enables the possibility to add other additional concepts, e.g., to conceptualise the ecosystem and platforms that exist around an IoT device as described in IoT ontologies. We hope that this conceptualisation inspires others to investigate further the uncharted spaces at the intersection of IoT and PM.

In future works, we plan to complete the model by adding attributes to the classes and to make it actionable and reusable by others. To this second end, we foresee two possibilities: implementing it in OWL, or translating it into an extension of the XES Standard . The presented model also needs to be further validated. We plan to validate it with additional real-life cases and to conduct an expert-based evaluation. Finally, we also aim at researching analytics and machine learning techniques that can automatically learn the influence of IoT data on process execution and discovery.

[2] http://www.xes-standard.org/xesstandardextensions.

[3] http://ocel-standard.org/1.0/specification.pdf.

References

1. van der Aalst, W.M.P., Dustdar, S.: Process mining put into context. IEEE Internet Comput. **16**, 5 (2012)
2. Becker, T., Intoyoad, W.: Context aware process mining in logistics. Procedia CIRP **63**, 6 (2017)
3. Brunk, J.: Structuring business process context information for process monitoring and prediction. In: CBI, pp. 39–48. IEEE, June 2020
4. Dees, M., Hompes, B., van der Aalst, W.M.: Events put into context (EPiC). In: ICPM, pp. 65–72. IEEE, October 2020
5. Dey, A.K.: Understanding and using context. Pers. Ubiquit. Comput. **5**(1), 4–7 (2001)
6. Dorsemaine, B., Gaulier, J.P., Wary, J.P., Kheir, N., Urien, P.: Internet of things: a definition & taxonomy. In: NGMAST, pp. 72–77. IEEE, September 2015
7. Elsaleh, T., Enshaeifar, S., Rezvani, R., Acton, S.T., Janeiko, V., Bermudez-Edo, M.: IoT-stream: a lightweight ontology for internet of things data streams and its use with data analytics and event detection services. Sensors **20**(4), 953 (2020)
8. Ghattas, J., Soffer, P., Peleg, M.: A formal model for process context learning. In: Rinderle-Ma, S., Sadiq, S., Leymann, F. (eds.) BPM 2009. LNBIP, vol. 43, pp. 140–157. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-12186-9_14
9. Gunther, C.W., Verbeek, H.: XES standard definition, March 2014
10. Janowicz, K., Haller, A., Cox, S.J.D., Le Phuoc, D., Lefrançois, M.: SOSA: a lightweight ontology for sensors, observations, samples, and actuators. J. Web Semant. **56**, 1–10 (2019)
11. Koschmider, A., Janssen, D., Mannhardt, F.: Framework for process discovery from sensor data, p. 8 (2020)
12. Koschmider, A., Mannhardt, F., Heuser, T.: On the contextualization of event-activity mappings. In: Daniel, F., Sheng, Q.Z., Motahari, H. (eds.) BPM 2018. LNBIP, vol. 342, pp. 445–457. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-11641-5_35
13. Leotta, F., Mecella, M., Mendling, J.: Applying process mining to smart spaces: perspectives and research challenges. In: Persson, A., Stirna, J. (eds.) CAiSE 2015. LNBIP, vol. 215, pp. 298–304. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-19243-7_28
14. Mannhardt, F.: Sepsis cases - event log, December 2016
15. Noy, N.F., McGuinness, D.L.: Ontology development 101: a guide to creating your first ontology, p. 28 (2001)
16. Rosemann, M., Recker, J., Flender, C.: Contextualization of business processes. IJBPIM **3**(1), 47 (2008)
17. Serpanos, D., Wolf, M.: Internet-of-Things (IoT) Systems. Springer, Cham (2018). http://link.springer.com/10.1007/978-3-319-69715-4
18. Soffer, P., et al.: From event streams to process models and back: challenges and opportunities. Inf. Syst. **81**, 181–200 (2019)
19. Sztyler, T., Carmona, J., Völker, J., Stuckenschmidt, H.: Self-tracking reloaded: applying process mining to personalized health care from labeled sensor data. In: Koutny, M., Desel, J., Kleijn, J. (eds.) Transactions on Petri Nets and Other Models of Concurrency XI. LNCS, vol. 9930, pp. 160–180. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53401-4_8

20. 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, pp. 251–269 (2018)
21. Trzcionkowska, A., Brzychczy, E.: Practical aspects of event logs creation for industrial process modelling. Multidisc. Aspects Prod. Eng. **1**(1), 77–83 (2018)
22. Wasserkrug, S., Gal, A., Etzion, O., Turchin, Y.: Complex event processing over uncertain data. In: DEBS, p. 253. ACM Press (2008)
23. van der Werf, J.M.E.M., Verbeek, H.M.W., van der Aalst, W.M.P.: Context-aware compliance checking. In: Barros, A., Gal, A., Kindler, E. (eds.) BPM 2012. LNCS, vol. 7481, pp. 98–113. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32885-5_7

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.