An Expert-Validated Bridging Model for IoT Process Mining

Yannis Bertrand[1]*, Jochen De Weerdt[1], and Estefanía Serral[1]

[1]Research Centre for Information Systems Engineering (LIRIS), KU Leuven,
Warmoesberg 26, 1000 Brussels, Belgium

*To whom correspondence should be addressed; E-mail: yannis.bertrand@kuleuven.be

Contextualisation is an important challenge in process mining. While Internet of Things (IoT) devices are collecting increasing amounts of 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. Accordingly, in this paper, a consolidated model to bridge the conceptualisation gap between the IoT and process mining fields, based on IoT ontologies and business process context models, is presented. This consolidation based on an initial model was obtained after an extensive validation both with an expert panel and with case studies. The results of the expert survey show that the model properly describes the links between the IoT and process mining and that it has added value for IoT process mining. Furthermore, the model was refined according to the experts' feedback. Accordingly, this paper's key contribution consists of a common reference model that can instigate true interdisciplinary research connecting IoT and process mining.

Keywords - *Process mining; Internet of Things; IoT process mining; Contextualised process mining; Event log model*

Acknowledgements: This research was supported by the Flemish Fund for Scientific Research (FWO) with grant number G0B6922N.

1 Introduction

The Internet of Things (IoT) is defined as a "group of infrastructures interconnecting connected objects and allowing their management, data mining and the access to the data they generate" (Dorsemaine, Gaulier, Wary, Kheir, and Urien (2015)). IoT data are therefore all the data collected by the objects belonging to connected infrastructures and 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, and the heart rate of a patient.

Recently, research has become aware of the potential of these data for process mining (PM). To better understand how IoT data can be used in PM, consider a smart factory where car parts are manufactured. The production line makes use of sensors to track parameters such as the pressure exerted on parts, the temperature in the room and the presence of workers. Process analysts would like to use these data in PM to analyse the process. A first idea is combining pressure data to data about critical steps of the production of a part to better understand the impact of pressure on the outcome of the process (i.e., if the part is acceptable or not). Another possibility is creating an event log with events derived from data collected by motion sensors or cameras and using PM to gain insight from this log. These examples show two types of benefits of using IoT data in PM. First, it enables contextualisation of processes with IoT data to deepen their understanding (e.g., by performing decision mining as in Banham and Wynn (2021)). Second, IoT data enable the application of PM to cases where it has thus far not been applicable because no log of the process is registered.

However, the integration of IoT data in PM is still in its infancy and there remain challenges to overcome to reach this vision. Important concepts, such as context or event, are often understood differently in the fields of IoT and PM. The low-level nature of IoT data makes them useless in most PM projects without an extensive, nontrivial, and error-prone abstraction step. Moreover, the understanding of the relationships between the concepts of PM and the IoT remains incomplete as IoT data may not only be linked to a single process event or even a single process case.

These conceptual issues cause 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, often making them unpractical and unfaithful to reality. For example, Dees, Hompes, and van der Aalst (2020) showed that translating three types of events of the Sepsis dataset (Mannhardt (2016)) into so-called

contextual data (i.e., not including them in the control-flow) reduced the number of discovered variants to one-fourth of the original, while annotating the model with the contextual data 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: first, mining models at an unsuitable level of granularity, as IoT data have to be abstracted to an adequate semantic level (i.e., that of the process); and second, obtaining models that confuse the context and the control-flow, as data on the control-flow of the process have to be carefully distinguished from data on the execution context of the process.

For these reasons, an initial model linking process mining and IoT concepts was created by Bertrand, De Weerdt, and Serral (2021), following the design science research framework of Hevner, March, Park, and Ram (2004). In this paper, this initial model is further refined and validated by means of an expert survey and case studies. As such, the core contribution consists of the first expert-validated conceptual bridging model to connect the IoT and process mining, advancing a common reference model that can instigate true interdisciplinary research.

The remainder of the paper is organised as follows. In Section 2, we outline important problems for using IoT data in PM, relating to the understanding of the concepts of context and event. Section 3 then provides some background in the representation of IoT data capture and business process (BP) context. Next, in Section 4 the methodology followed to create, validate and improve the model, based on the design science research paradigm, is presented. The initial model linking IoT and PM itself (Bertrand et al. (2021)) is presented and discussed in Section 5. Then, we provide a double validation of the model: first, we conduct a validation survey completed by experts in IoT PM (Section 6); based on the results of this expert validation, an improved version of the model is presented (Section 7) and validated with three case studies (Section 8). Finally, we discuss the validation of the model (Section 9) and put it in perspective with respect to related works (Section 10) before concluding with some ideas for future research (Section 11).

2 Problem statement

Bridging the IoT and PM requires a solution for two specific problems: (1) the mismatch between the intrinsic nature of IoT data being low-level vs. event logs in PM typically logging data at a business-meaningful level and (2) the difficulty distinguishing control-flow from context.

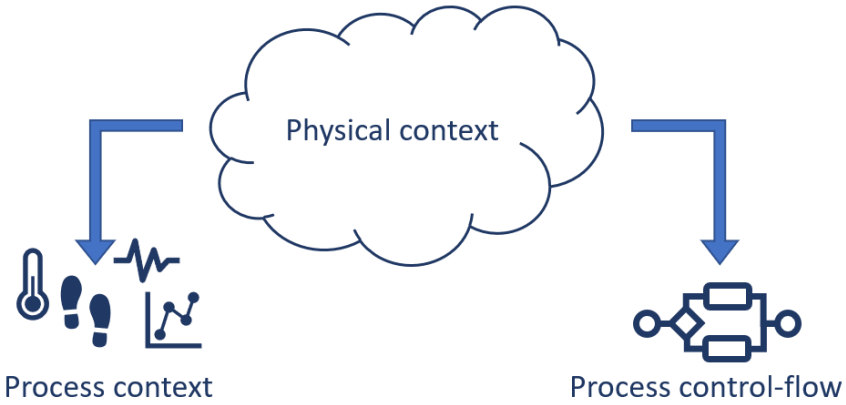


Figure 1: Relationship between IoT data and business processes.

2.1 Context in PM vs context in IoT

Context is broadly defined by Dey (2001) as “any information that can be used to characterise the situation of an entity”. In PM, the context that is taken into account is typically the context of the process. This translates to factors that impact the design or the execution of BPs. At the same time, IoT devices collect data about the physical environment of a system. As such, IoT data are often context data, making systems *context-aware*, in the sense that they make digital systems aware of their physical surroundings. While there are clear overlaps between both understandings, they are not identical, and several grey areas have to be clarified.

Figure 1 shows the links between the set of parameters tracked by IoT devices (i.e., the physical context) and the process. There are two main options to use IoT data in PM. First, IoT data can be used to determine the process context. This corresponds to, e.g., the temperature of a refrigerated area where fresh products are stored, or the pressure in a tank containing a chemical product being produced. This is usually done by abstracting the IoT data to the proper level for contextualising the process execution. Once the IoT data are at the context level, these data are analogous to non-physical process context parameters (i.e., "digital" context parameters), e.g., an order quantity or the resource executing an activity.

In addition to deriving process context information from IoT data, there is a second important case to mention: the use of IoT data to derive the control-flow of the process. This is the area most PM literature has focused on thus far, and it consists of physical context parameters that relate to the progress of the process. These parameters directly correlate with an activity of the control-flow of a process and their value can indicate that an activity was executed. E.g., the location of a package in a delivery process can

be used to derive that the package has been loaded on a truck, or the temperature of an oven decreasing in an industrial process can indicate that a phase of the process is over. IoT context data therefore eludes the usual control-flow/context boundary, and a need exists to correctly identify and model these IoT context parameters.

Moreover, the IoT context is often inherently more dynamic than the traditional non-IoT BP context and it is typically only loosely coupled with the process: e.g., the temperature in the refrigerated area may fluctuate independently of the activities that occur nearby and have interesting values (i.e., too high or too low) at any point between two process events. This contrasts with usual digital context parameters, which are often fixed and can be updated only with an event of the process, e.g., the value of a claim being modified by a "reevaluate claim amount" activity. This scenario poses the question of how to capture and represent these occurrences, which are not events nor event attributes in the traditional sense in PM, but are relevant to the process and should be taken into account in PM.

2.2 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 (Koschmider, Janssen, and Mannhardt (2020); Tax, Sidorova, Haakma, and van der Aalst (2018)). This distinction is usually limited to placing events from IoT at a lower abstraction level than that of events in PM. However, both definitions differ in more than that if we consider a more detailed evaluation.

On the IoT side, an event can be defined as a time-value set Serpanos and Wolf (2018):

$$< key, value, destination, generation_time, release_time >$$

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

On the PM side, a common definition of the term event is that 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." (Gunther and Verbeek (2014)). Such events are typically of the following format:

$$< caseID, eventID, timestamp, activity[, lifecycle_step, attributes] >$$

We can especially note that an event is more broadly defined in the IoT literature than in the PM literature, as an event in the IoT can represent a very wide range of things, depending on the key-value

pair. An event as understood in the IoT can be linked with either an event of the process (e.g. a patient taking a blood test) or a context variable (e.g. the patient's insulin level), whereas an event as understood in PM can correspond to only the first case. Moreover, PM algorithms require that each event has a case ID, which is not necessarily logged with events as understood in IoT. Therefore, PM algorithms cannot simply be run on IoT data. In addition, an event in IoT that relates to a context variable can impact multiple instances of the process, contradicting the usual assumption in PM that each event is linked to one and only one case.

The conceptual model presented and validated in this paper was developed to help build a common understanding of the structure of information in the IoT and PM.

3 Background

In this section, previous works on the modelling of IoT and PM are introduced as part of the relevance cycle of the design science research approach (see Section 4.1). Relevant IoT ontologies are discussed, before addressing BP context models. Literature on process mining in IoT environments is discussed in detail in Section 10.

3.1 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, see `https://www.w3.org/TR/2017/REC-vocab-ssn-20171019/`) to the development of new ontologies that are simpler and more practical. Two such ontologies are the Sensor, Observation, Sample and Actuator (SOSA) ontology (Janowicz, Haller, Cox, Le Phuoc, and Lefrançois (2019)) and IoTStream (Elsaleh et al. (2020)).

SOSA proposes three perspectives: the sensor, observation and actuator perspectives Janowicz et al. (2019). IoTStream is a more specific ontology, inspired by SOSA, that focuses on the treatment of streaming data Elsaleh et al. (2020). Both 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 the IoT relies on.

3.2 Business process context modelling

One of the first BP context models was proposed by Rosemann, Recker, and Flender (2008), where the authors described an onion model in which context was split into four layers (listed from closest to farthest from the process): immediate context, internal context, external context and environment

context. van der Aalst and Dustdar (2012) developed a similar onion model a few years later . Another relevant representation was proposed by Ghattas, Soffer, and Peleg (2010), 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 (2020) proposed a taxonomy for BP context data with six dimensions: time, structure, origin, relevance, process relation and runtime behaviour. The proposed dimensions describe the 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, Verbeek, and van der Aalst (2012), who represented the context of a BP in a domain model. da Cunha Mattos, Santoro, Revoredo, and Nunes (2014) went further and proposed the use of three metamodels: a context metamodel, a business process metamodel, and a domain metamodel.

4 Methodology: a design science research approach

This research follows the design-science research approach from Hevner et al. (2004), which consists of three cycles: relevance, rigour and design (see Fig. 2).

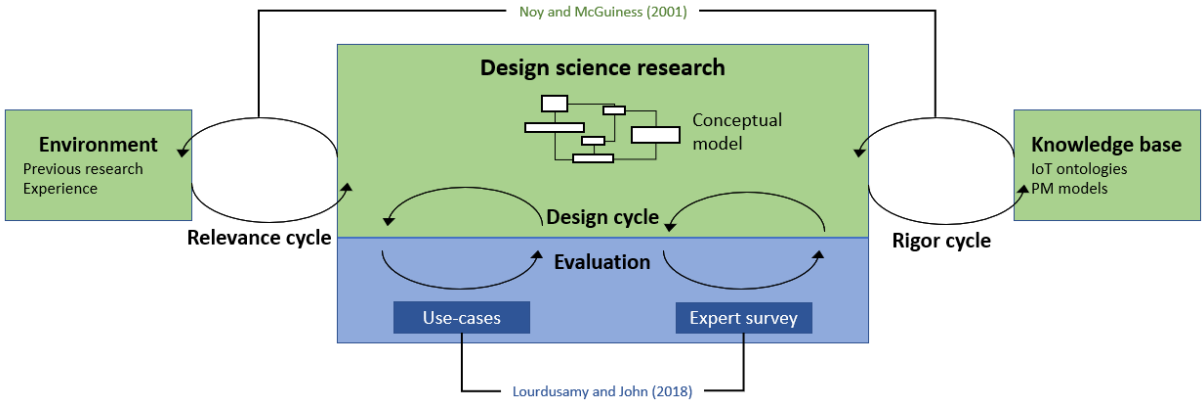


Figure 2: Design science research approach.

4.1 Relevance cycle

The relevance cycle aims to define requirements and methods used to evaluate the model based on the requirements. In this research, we address one main issue: the ambiguity of certain concepts, i.e., event and context, and their inconsistency in the literature about IoT and PM. We therefore propose to create a new conceptual model to define these concepts and link them to clarify their interrelationships. We

derived two requirements from the observations of Section 2: 1) the model has to represent the concepts of IoT and PM and link them together; 2) the model has to disambiguate important concepts, i.e., event and context, that are often misunderstood.

4.2 Rigour cycle

The rigour cycle reviews existing knowledge and methods to construct and evaluate a new solution. We first searched the literature and reviewed existing related models: IoT ontologies, BP models and PM models (see Section 3). We then scanned the literature for ontology creation methods and chose to follow the methodology described in Noy and McGuinness (2001). The evaluation methods are also ontology validation techniques, as reviewed in Lourdusamy and John (2018). Specifically, we chose to perform user-based validation (with an expert survey) and case-based (or application-based) validation, the other techniques being hardly applicable.

4.3 Design cycle

The design cycle consists in creating and evaluating alternative solutions. In this cycle, we created the model basing on the methodology proposed in Noy and McGuinness (2001), and building upon existing models in the knowledge base. We first gathered the requirements, which were were presented in Section 4.1. Secondly, two types of existing models were reviewed: IoT ontologies (in particular, lightweight and event-centric ones) and PM models. The third step was to look for recurrent concepts: on the one hand, archetypal classes in the domains of the IoT and PM were searched for and, on the other hand, concepts that are common to IoT and PM and could form a junction between both domains were looked for. Next, the classes of the conceptual model were created, drawing from existing models and supplementing existing classes with new concepts to make a consistent whole. Additionally, relationships and cardinalities between the classes were defined. Finally, instances were created for validation purposes representing the use case presented in Section 8.1. Four iterations were performed to improve the model until all the instances of the case study could be satisfactorily represented.

After the first version of the model was created, we performed a two-level evaluation based on an expert survey and two additional case studies which we present in this paper; see overall steps in Figure 3. These types of evaluations were chosen because the quality criteria of a conceptual model are difficult to define and there is often some subjectivity in the assessment of these criteria, which we mitigated by surveying a series of experts in the domain. The feedback obtained within this evaluation was used to

improve and validate the model, which was further validated by performing an additional case study with the improved model.

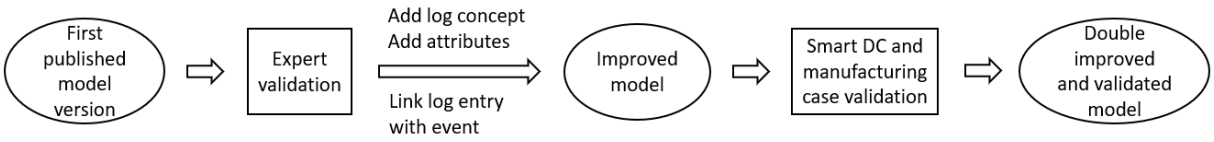


Figure 3: Overview of the methodology to validate and improve the model.

Expert-based validation. We conducted an expert-based validation that consisted of showing the model to experts or end users and gauging their opinion on both the overall structure and the extent to which the model achieves the tasks it is designed to fulfil. The survey was first tested and improved based on the feedback of two specialists before being sent out to the final set of experts (from which the two pre-testers were excluded). In this case, key experts in process mining with IoT data were invited to evaluate and comment on the model via a survey.

Case-based validation. Based on the feedback collected by the expert survey, some improvements were made to the model. These improvements were validated with case-based validation, which consists in demonstrating that a model fulfils its requirements in a case study Lourdusamy and John (2018). According to Porzel and Malaka (2004), it is a suitable approach to evaluate the relevance of newly added concepts and substituted concepts, which corresponds to the modifications performed on the model. One of the authors had previously gathered a repository of published IoT-enhanced BP cases, which served as a baseline to design the two first scenarios presented in Section 8, while the last one depicts a real-life process in a production company. Note that the case study outlined in Section 8.1 was already presented by Bertrand et al. (2021) and is updated in this paper. Instances to represent both cases were created using the model, validating as such the final improved version.

5 A bridging model for IoT and process mining

5.1 General introduction

The model (Bertrand et al. (2021)), shown in Figure 4, is built as a UML class diagram and consists of two main parts. The first, from ObservableProperty to Event, describes how data are captured and managed by Sensors (following two IoT ontologies: IoTStream (Elsaleh et al. (2020)) and SOSA (Janowicz et al. (2019))). The second, from Event to Event Log Entry (including Process-Aware IS and IS data entry)

shows how data are processed in PM to create an event log (following the traditional PM view as in Gunther and Verbeek (2014)). A link is made through the common construct of Event. Next, we define the different terms represented in the model.

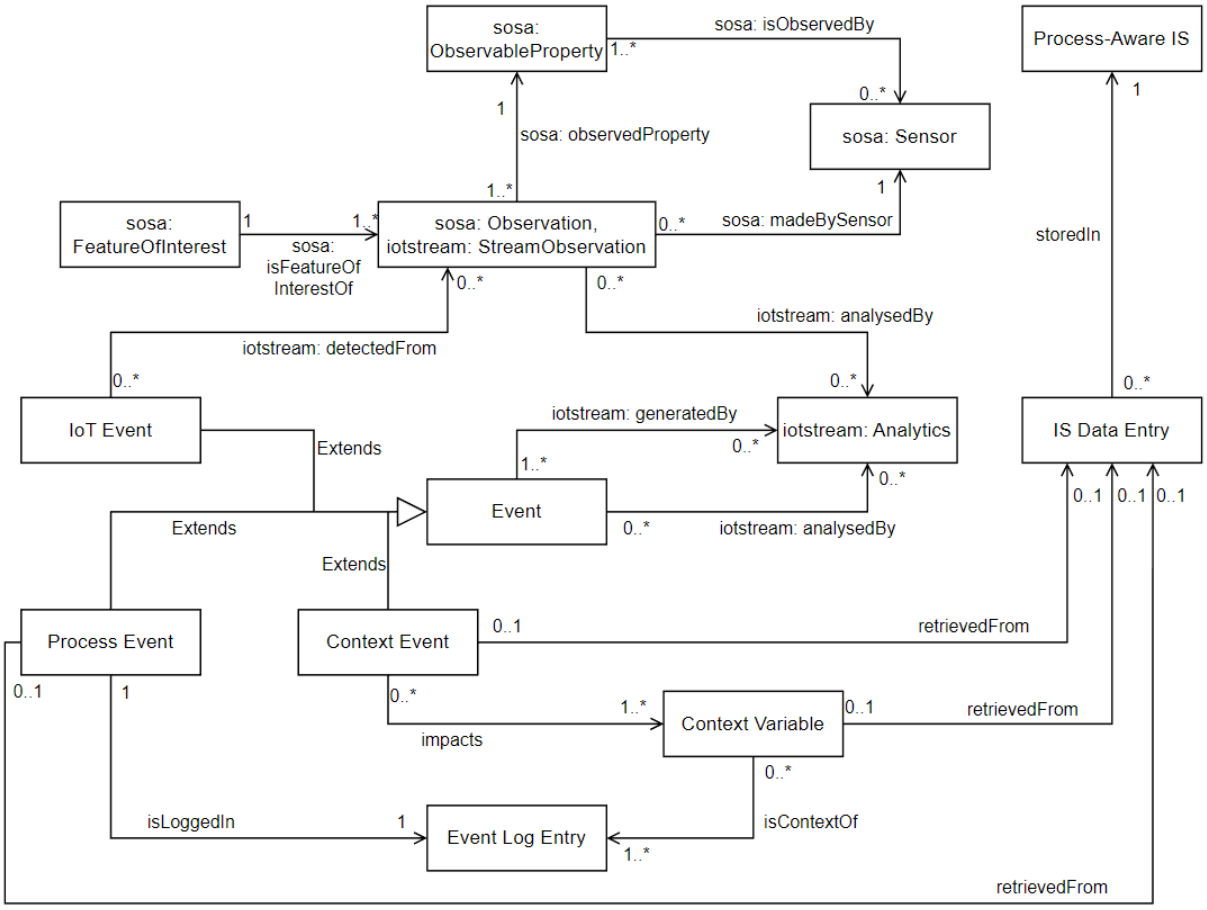


Figure 4: Core of the model linking the IoT with PM.

5.2 Concept-by-concept description

5.2.1 Core of the model

A **Sensor** (e.g., a temperature sensor) is an IoT device that measures the state of a real-world phenomenon, named *ObservableProperty* in SOSA (Janowicz et al. (2019)). An **ObservableProperty** is an observable quality (property, characteristic; e.g., the temperature) of a **FeatureOfInterest**, which is the thing whose property is being estimated or calculated in the course of an observation (e.g. the refrigerator whose temperature is measured). An **Observation** is a measurement of an ObservableProperty; it provides the result of sensing or calculating a value of an observable property (e.g., 5°C) .

The model contains different types of *Events*. An **Event** is defined as "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)" (Wasserkrug, Gal, Etzion, and Turchin (2008)).

Event can be specialised as *IoTEvent*. An IoTEvent can be derived from one or several Observations, and is defined as an instantaneous change in a real-world phenomenon that is monitored by a Sensor. Several IoTEvents can be detected from the same Observation, e.g., an observation of the Observable Property "temperature" can trigger the IoTEvents "temperature decreases to 0°C" and "it is freezing". Likewise, an IoTEvent can be triggered 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 one or several 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 (Elsaleh et al. (2020)) that is used here to describe any technique that enables the extrapolation of an Event from an Observation or another Event (or several), such as event abstraction, complex event processing, database query and stream annotation.

An Event can also be specialised in two other types of Events, namely *ContextEvent* and *ProcessEvent*. These typically have a granularity level that is that of the process and can be abstracted from IoTEvents using Analytics. A **ProcessEvent** is defined as an instantaneous change of state in the transactional lifecycle of an activity. This type of event corresponds to the traditional notion of an event in PM, but it decouples the occurrence of the change of state in the activity lifecycle and the attributes that are usually present in event data structures, e.g., in XES. These types of attributes are represented using *ContextVariables* and ContextEvents. A **ContextVariable** is a variable that is relevant for the execution of a process, while a ContextEvent is an instantaneous change in a ContextVariable (deduced from one or several IoTEvents or *ISDataEntries*). Note that ContextVariables might be included at the level of activities, process instances, or even the overall process. Examples of ContextVariables include the location of a package in a delivery process, the vital signs of a person in a health monitoring process and the resource executing a task.

In other words, a ProcessEvent marks the progress of the control-flow (execution of activities); hence, it relates to an activity, whereas a ContextEvent is a change in the environment/process parameters that may determine the control-flow path and how activities are executed. The distinction between ProcessEvent and ContextEvent is to a certain extent dependent on the use-case and requires judgement. This relates to the issue of determining what the process that one wants to analyse actually is, which activities

are part of the control-flow, and which activities may occur in the surroundings of the process, i.e. in its context. This may depend on e.g., the case notion that is retained, the perspective taken (data, resource, etc.), and the type of analysis that has to be performed.

5.2.2 The event hierarchy

As mentioned above, Events in the model follow a hierarchy based on their level of granularity. A higher-level Event (ProcessEvent or ContextEvent) can be deduced from one or several lower-level Events (IoTEvents); similarly a lower-level event can be the basis of one or several higher-level events. This mechanism is inspired by CEP (Wasserkrug et al. (2008)), as was also suggested by Soffer et al. (2019). The model goes further than this with the notion of Analytics, which represents any technique used for event abstraction. Examples include rule-based reasoning techniques like CEP, data-driven models, etc. Concrete instances are presented in the cases in Section 8. IoTEvents can cascade until a deduced event has a direct relationship with the process, i.e., it is a ProcessEvent or a ContextEvent.

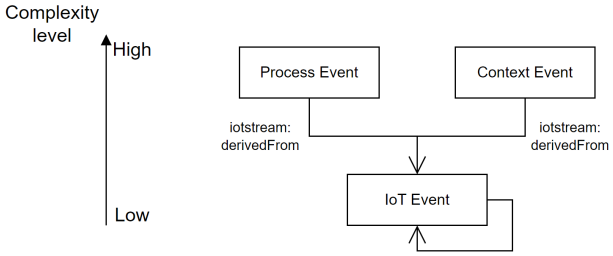


Figure 5: Hierarchy of events in the model.

5.2.3 Linking the model with traditional process mining data management

ProcessAwareInformationSystem (PAIS) and *ISDataEntry* represent the traditional PM data source. A **PAIS** is any information system that records process data, such as a CRM or an ERP system. The link between PAIS and ProcessEvent 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. An **ISDataEntry**, which represents a row in a relational database table, can relate to three classes: ProcessEvent, ContextEvent and ContextVariable. Usually, in PM, data used as ContextVariables are data retrieved from the IS and can only change with ProcessEvents (and hardly change at all in practice). However, this does not mean that such ContextVariables may not be subject to change; e.g., the value of a claim in a claim handling process is typically assumed fixed, but in reality it can be modified during the treatment of the claim.

Finally, **EventLogEntry** is the point where ContextVariables can be linked with Events in the **Event-**

Log, which contains records of events (possibly together with the context in which they occurred). Notably, although EventLogEntry and EventLog are not linked with Analytics in the model, it does not necessarily mean that Analytics techniques are not used. Analytics is linked with Observation and Event to emphasize the importance of Analytics techniques to derive Events from Observations or other Events, but it may be the case that, e.g., event correlation or data fusion techniques are necessary to match a ProcessEvent with the relevant ContextVariables or to derive an IoTEvent from an Actuation. This consideration is omitted in the figure for the sake of clarity.

6 Expert-based validation

As explained in Section 4.3, a two-level validation of the model is conducted.

As a main evaluation instrument, we constructed a survey containing questions to evaluate the completeness, conciseness, accuracy and clarity of the model, as well as the experts' level of agreement with the proposed model. Moreover, the experts were asked to provide feedback on the strengths, weaknesses and usability of the model.

6.1 Survey design

The survey consists of 24 questions, with the distribution in terms of evaluation dimensions and question types illustrated in Table 1. The different types of questions are: scaling questions, asking the respondent to evaluate their agreement with a given statement, on a scale from one to seven; exclusive multiple choice questions; long open questions, used to collect feedback on the model at large; and short open questions, which asked the opinion of the respondents about specific aspects of PM with IoT.

This final set of questions incorporates feedback from a test run with two senior researchers acquainted with the topics of IoT and PM. This test phase confirmed that the main structure of the survey was suitable for the evaluation of the model and brought important improvement to the survey, e.g., to make it clearer and less burdensome for the respondents. A one-page extract is shown in Appendix A.

Profile and generic questions were used to gather data about the respondents themselves, e.g., their job title and how they would assess their knowledge of both the IoT and PM. Completeness, conciseness and accuracy questions aimed to assess the completion of the first requirement: the model should identify and clarify the main concepts of the IoT and PM and link them. Then, questions about the clarity of the model and agreement of the experts were asked to assess to what extent the second requirement

	Scaling question	Exclusive multiple choice question	Long open question	Short open question
Profile & Generic	1	1	1	5
Completeness	2			
Conciseness	2			
Accuracy	2			
Agreement	4			
Clarity	2			
Feedback			4	

Table 1: The number of questions according to evaluation dimension and question type.

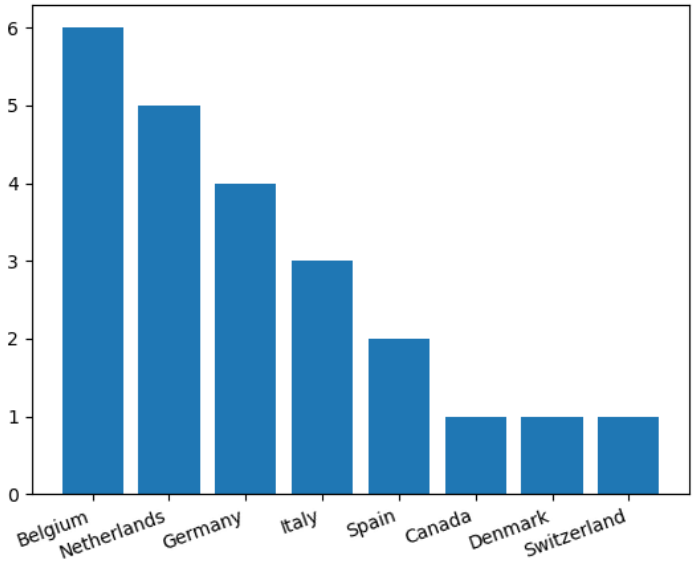


Figure 6: Distribution of experts contacted per country.

(disambiguate the complex concepts of event and context) is fulfilled. Finally, feedback questions aimed to identify potential points of improvement of the model.

6.2 Respondents profile

Experts were sought for their combined knowledge of the IoT and PM or BPM. Potential respondents were selected from the program committees of relevant events, such as the BP-meets-IoT workshop held each year at the BPM conference, from the authors of publications in process mining using IoT data and from the personal network of the authors. The respondents are all academics since this domain of study thus far remains scarcely deployed in industry.

A total of 23 experts were contacted via email to invite them to complete the survey (see the country where they are based in Figure 6). Of this small but targeted pool of experts, 12 eventually completed

the survey: 9 professors and 3 researchers (pre- and post-doctoral). Most of the respondents assessed their knowledge of PM and BPM as good or excellent, and they evaluated their proficiency in IoT data processing mostly as good or fair. Knowledge of IoT technologies was more varied, with half of the respondents describing it as good or excellent and the other as fair or little knowledge. Other research interests ranged from software engineering to blockchain technologies, also comprising data mining and artificial intelligence.

6.3 Validation results

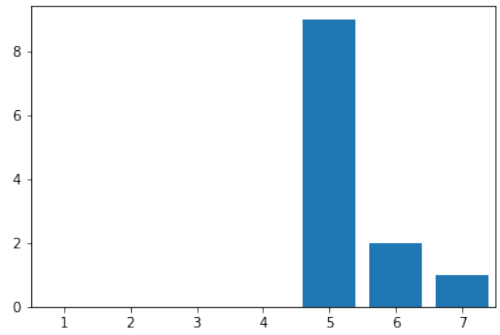
An overview of the quantitative results is presented in Table 2. The model is globally positively evaluated. As shown in Figure 7, all respondents believe that the model properly describes the links between the IoT and PM (5.33/7), and ten of twelve believe it has added value for PM using IoT data (5.5/7).

Evaluation dimension	Question	Average score
Completeness	All important concepts are present in the model presented above.	5.00
Completeness	All important relationships are present in the model presented above.	4.92
Conciseness	The model contains no superfluous concepts.	5.58
Conciseness	The model contains no superfluous relationships.	5.83
Accuracy	The model is at the right level of granularity, i.e., neither too simple nor too complex.	5.41
Accuracy	The model is well balanced between its IoT part and its process mining part.	4.50
Clarity	The distinction between the three different types of events is clear.	4.50
Agreement	I agree with the distinction between the three different types of events.	4.58
Clarity	The presented hierarchy of events is clear.	5.25
Agreement	I agree with the presented hierarchy of events.	5.17
Agreement	The conceptual model properly describes the links between IoT data and process data.	5.33
Agreement	The proposed model has an added value for process mining using IoT data.	5.50

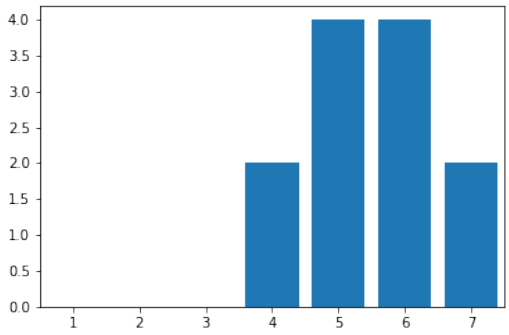
Table 2: Overview of the results of the scaling questions.

Specifically, questions assessing the completeness of the model scored 4.96, with one expert somewhat disagreeing with each statement. Moreover, the conciseness of the model received a score of 5.7, with only one respondent somewhat disagreeing with the statement that there are no superfluous concepts in the model. In agreement with these results, the level of granularity of the model was generally assessed as appropriate (5.41/7), with one respondent somewhat disagreeing and finding the model too simple.

The balance between the IoT and PM sections of the model was slightly less agreed upon, with an average score of 4.5: three respondents found the model too IoT-oriented while one found it too PM-

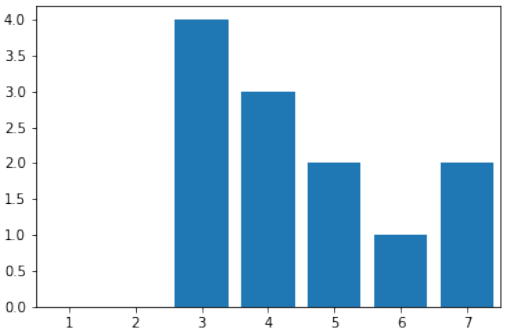


(a) The conceptual model properly describes the links between IoT data and process data.

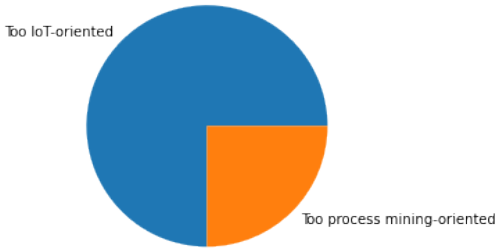


(b) The proposed model has an added value for process mining using IoT data.

Figure 7: Global evaluation of the model.



(a) The model is well balanced between its IoT part and its process mining part.



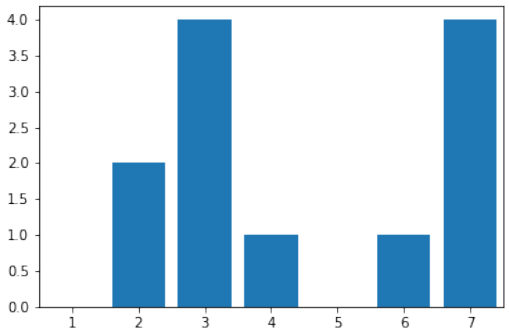
(b) Do you think it is too much IoT-oriented or process mining-oriented?

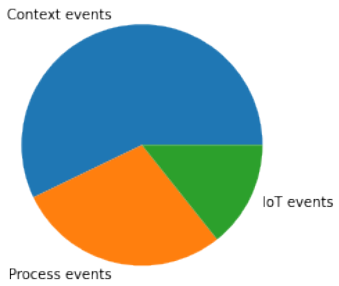
Figure 8: Assessment of the balance between the IoT and PM parts of the model.

oriented (Figure 8).

A second point of disagreement, with the model and between experts, was the distinction made between different types of events: context events, IoT events and process events (see Figure 9. Half of the respondents found the distinction between different types of events unclear. In particular, four found context events unclear, two found process events unclear and one found IoT events unclear. However, four respondents strongly agreed with the statement that the distinction between the event types is clear. Therefore, there is some disparity in the responses, with some experts doubting the distinction between event types and others strongly supporting it.

Next to this, the hierarchy between the event types saw greater agreement. Only two respondents found it unclear, and one disagreed with it (average scores of 5.25 and 5.17, respectively).





(a) The distinction between the three different types of events (IoT, process, context) is clear.

(b) If you disagree, which type(s) of events do you find unclear?

Figure 9: Understanding of the distinction between different event types.

6.4 Improvement recommendations

Finally, the experts were requested to provide suggestions for improving the model. Table 3 summarises their proposals, as well as the steps taken to address them. Please observe that recommendations 6 and 7 were deemed unnecessary since the model already incorporates them.

7 Improved model

In this section, we present the modifications made to the model shown in Figure 4 to improve it, based on the feedback of the expert survey (see Tables 2 and 3). Although the model presented was generally accepted by the respondents, we incorporated minor yet significant modifications to improve the final model. The quantitative evaluation revealed two key points: 1) the model was too IoT-oriented, and 2) that the distinction between different types of events was not entirely clear. Additionally, the open-ended questions resulted in concrete recommendations R1-9.

7.1 Event log concept

To improve the balance between the IoT and PM components of the model, a new concept was added: EventLog. It is related to all the event log entries relevant for the process. Note that the relationship between EventLog and EventLogEntry is many-to-many and not a collection (like in, e.g., XES) because IoT or context events can be related to several processes executed in the same physical environment.

Code	Recommendation	# Experts	Action undertaken
R1	Discuss process attributes/variables	3	Clarified model description as process attributes are represented in model as context variables
R2	Discuss how to represent event abstraction/aggregation	3	Clarified model description as this takes place with the concepts of event and analytics; attributes added to observation, analytics and event.
R3	Add concept of case	1	CaseID added as an attribute of process event
R4	Add concept of activity	1	Activity added as an attribute of process event
R5	Add relationship between context and IoT events and event log entry	1	Replaced link between process event and event log entry with link between event and event log entry
R6	Add relationship between process and IoT events	2	None: process and IoT events are already related via analytics, adding such a relationship would induce redundancy
R7	Add relationship between context and IoT events	2	None: context and IoT events are already related via analytics, adding such a relationship would induce redundancy
R8	Provide more detail regarding the concept of event	1	Attributes added to all three types of events
R9	Clarify the distinction between event types	2	Attributes added to all three types of events to clarify the differences between them

Table 3: Summary of the recommendations made by the experts and how they were addressed.

7.2 Main attributes of the concepts

We propose to clarify the different event types and refine the model by defining typical attributes for the classes (see Figure 10), addressing R3-4 and R8-9. We decided to focus on the most important attributes to keep the model as comprehensive as possible. Moreover, we avoided replicating attributes in different concepts to limit redundancy across concepts.

Regarding IoT ontologies, SOSA (Janowicz et al. (2019)) does not describe a large number of attributes; most of the information is modelled as relationships, while IoTStream (Elsaleh et al. (2020)) gives more details (sometimes through referencing other ontologies). From SOSA, we retrieve the following attributes of Observation: phenomenonTime, resultTime, hasSimpleValue. The coexistence of two time attributes stresses the fact that one observed phenomenon may not be instantaneous (hence phenomenonTime can be an interval), but is recorded at the moment of its completion (resultTime). Six attributes of Analytics are extracted from IoTStream: methods, parameter (a list of length n), paramValues (a list of length n), windowStart and windowEnd. The two latter are time attributes, which represent the first and last Observations taken into account by the Analytics technique to derive a higher-level event.

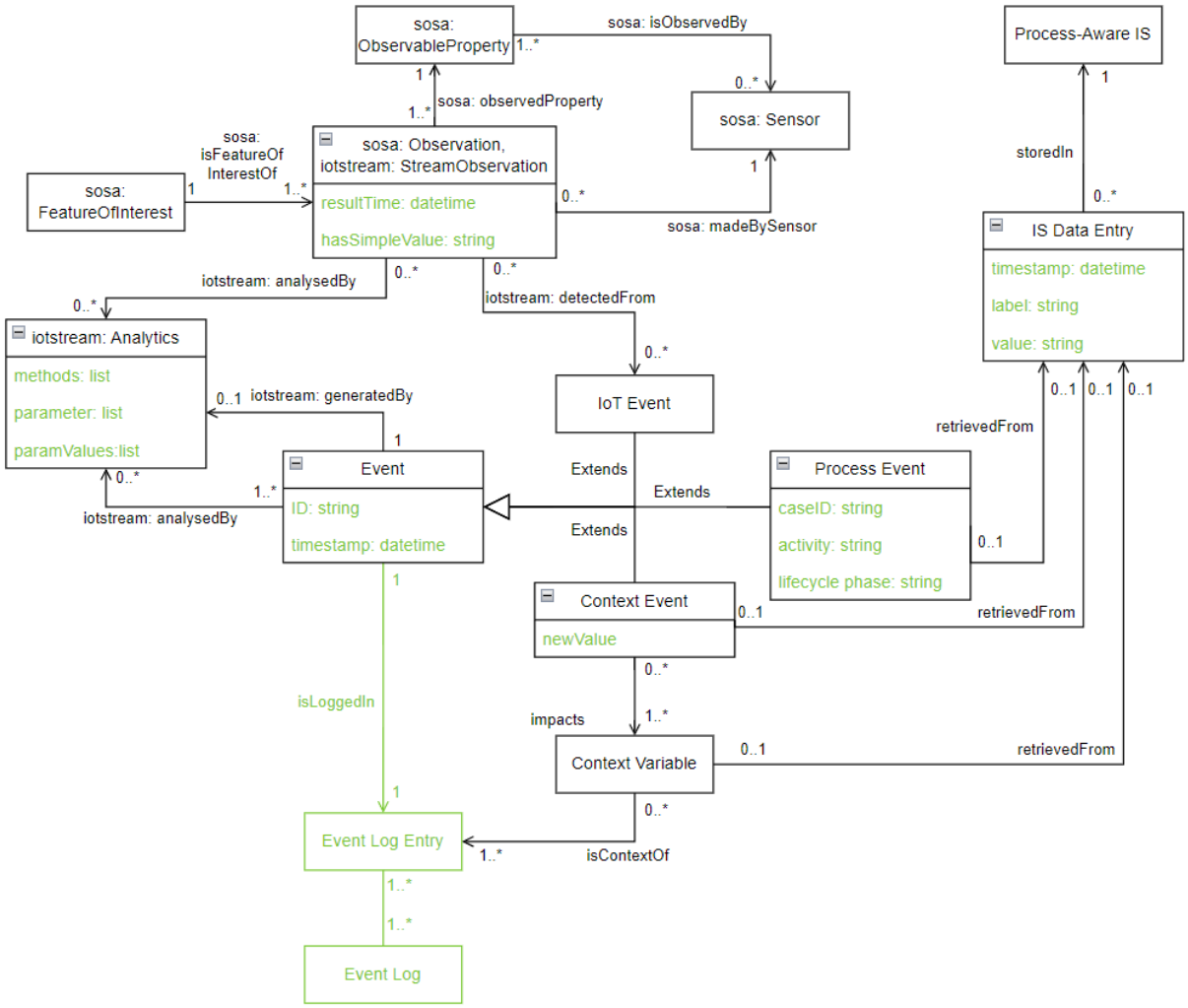


Figure 10: Final expert-validated bridging model for IoT and PM.

A similar approach was followed to define attributes for the PM part, considering models such as XES (Gunther and Verbeek (2014)). All Events share two attributes: an ID and a timestamp, which is unique, showing that the Event is instantaneous. Next to these common attributes, different event types have different specific attributes, resulting in structural differences between them. ProcessEvents have three specific attributes: case ID, activity and lifecyclePhase. The traditional attributes of events in PM (as defined in XES, e.g., resource and cost) are modelled as ContextVariables here, which are updated by ContextEvents (with the value newValue) and are linked with relevant ProcessEvents via EventLogEntry. Finally, ISDataEntry, a generic concept, has generic attributes: ID, timestamp, label and value. Label can correspond to the name of a ContextVariable or the name of an activity in the case of a ProcessEvent, while the value is the value of the ContextVariable (newValue of a ContextEvent) or the lifecycle state of the activity of a ProcessEvent.

7.3 Relationship EventLogEntry - Event

A second improvement to clarify event types is to link EventLogEntry with Event instead of with ProcessEvent (R5). This allows the representation of different types of events in the log itself. This way, even if the links between IoTEvents and higher-level Events are unknown, all the data can be represented together in the log and the links can be discovered later (always with Analytics techniques). This also contributes to tackling the comment from a part of the experts that the distinction between the different types of Events is unclear, by giving them all a similar importance in the model.

8 Case-based validation

The capability of the model to represent a relevant use case was already demonstrated by Bertrand et al. (2021) in the case study described in Section 8.1. To verify that the first version's claims remain valid, we replicated this case study using the improved version of the model. Moreover, in addition to the replication study, we conducted two additional case studies (outlined in Sections 8.2 and 8.3) to generalise the conclusions of the first case study and showcase the model's capabilities . The case studies were carefully selected to 1) demonstrate the entire model's end-to-end functionality, including its various capabilities and the different paths to obtain process or context events, and 2) depict IoT-enhanced BPs from different domains, leveraging different sensor types to generate diverse data. Notably, Figures 11, 12 and 13 include name labels for instances of concepts that do not have attributes, enhancing clarity.

8.1 Case study 1: Vaccine transportation

Consider the process of transporting vaccines from their production facility to patients. The vaccines are manufactured in a main production plant before being shipped to a central storage facility. The vaccine crates are then dispatched to local vaccination centres where each dose is administered to a patient. Because the vaccines are 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 to improve the process to minimise this number. Figure 11 shows how to use the model to map different concepts from the raw output of an IoT sensor to an entry in the event log.

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

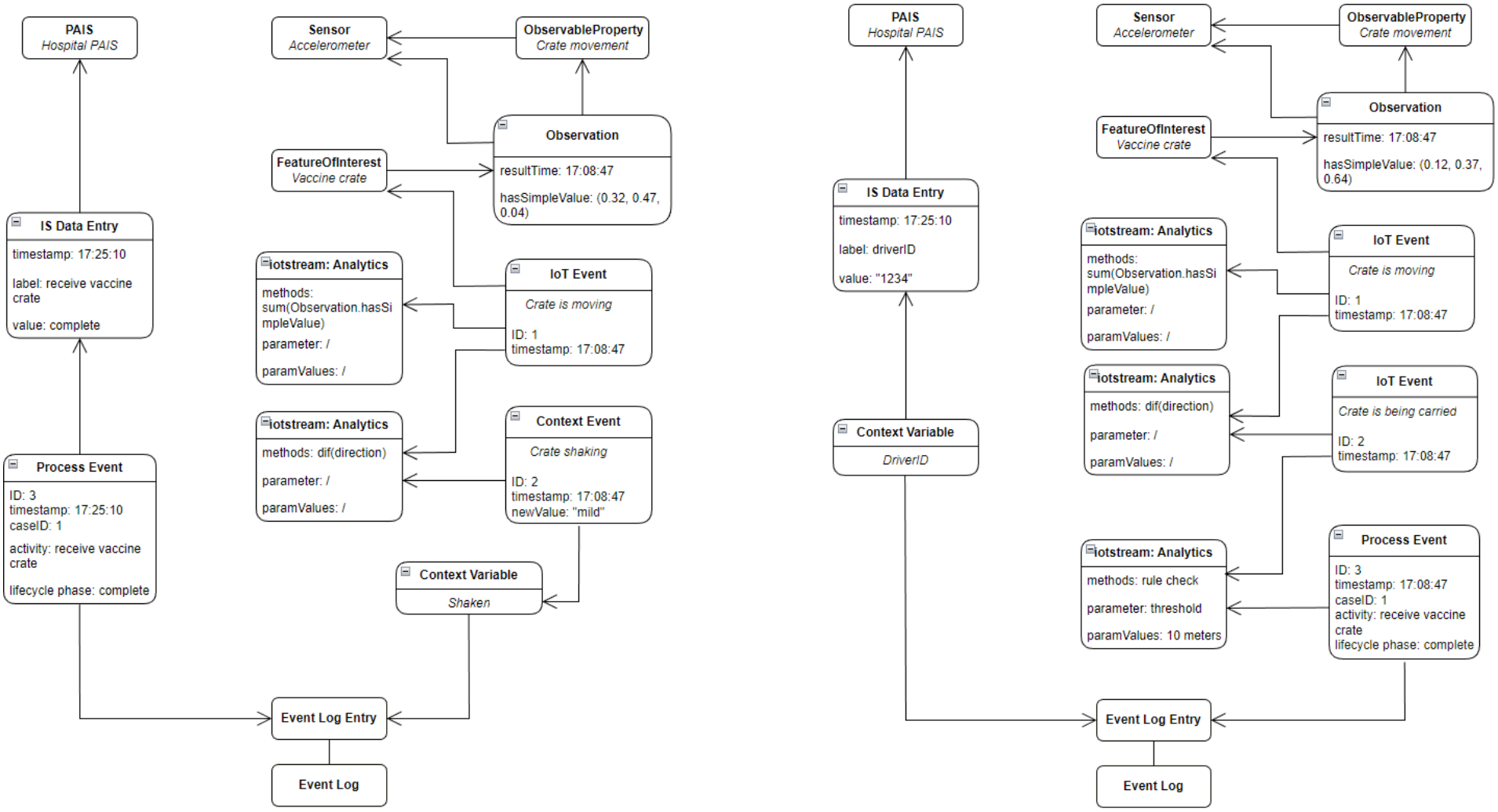


Figure 11: Example instances for the vaccine shipment process.

from such an Observation. Comparison of the movement with previous movements can indicate whether the crate is being shaken (which corresponds to the Context Event *Crate is shaking*, Figure 11 (a)) or if it is being displaced (which could detect a Process Event *Crate is loaded*, Figure 11 (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) in Figure 11).

Recording this Context Variable with each Process Event enables identification of 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) in Figure 11).

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

More specifically, this case study illustrates multiple capabilities of the model. First, we can see the different possibilities to obtain higher-level events, either by deriving them from lower-level events (i.e., IoT events) or by retrieving them in the IS supporting the process. Then, this case study also shows that observations from the same sensor can be used to derive both process and context events, with different instantiations of analytics. Compared to the version of the case study with the original model (Bertrand et al. (2021)), the introduction of attributes greatly clarified the concepts and in particular events and analytics. Moreover, the links and differences between events of different types are also highlighted by each event's attributes, e.g., process events with activity, context events with newValue. Finally, the new event log concept is useful to show and reason about the final content of the log.

8.2 Case study 2: Smart DC

This example consists of a smart distribution centre (DC) that receives products from warehouses, assesses their quality and distributes them to supermarkets (adapted from Valderas, Torres, and Serral (2022)).

The process begins with the arrival of a container loaded with crates of products at the smart DC. The

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

The monitoring of the transport conditions of products received in the smart DC is modelled as follows. *Temperature* and *Humidity* Sensors track the Observable Properties *Pallet temperature* and *Pallet humidity* of the Feature of interest *Pallet*. Each recording of the sensors is compared with the previous recording, and each change in value constitutes an IoT event *Increase* or *Decrease* of temperature or humidity. When such IoT events are detected, a second level of Analytics compares the new value with a threshold to determine whether both values are within acceptable limits for the product being transported. If not, a Context event is triggered, which modifies the Context Variable *Suitable conditions* and makes it false. This information is retrieved and registered as an Event log entry when the pallet is scanned upon arrival to the smart DC.

The main point of this second case study is to illustrate a more complex interplay between events and analytics. Indeed, two levels of IoT events are derived from observations from two sensors and combined to detect a context event, showing the case where data from multiple sensors have to be analysed to derive a high-level event.

8.3 Case study 3: Production environment

The last case study tackles a real-life process of a production company. Their production process can be summarised in four main steps:

1. Preparing raw material and loading the tank;
2. Mixing the raw material in the tank;
3. Circulating the product through filters to remove impurities;
4. Bottling and packing the finished product.

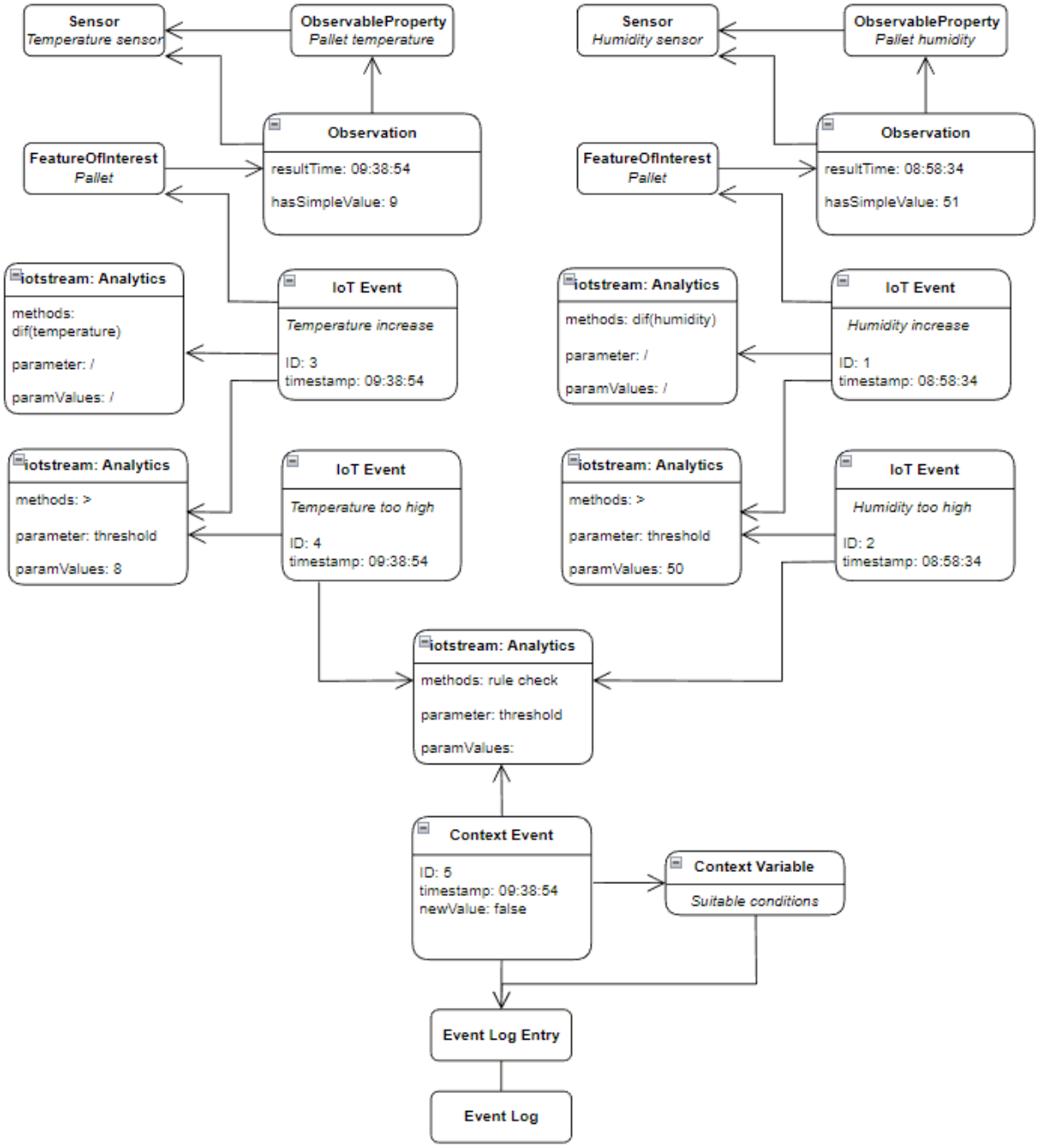


Figure 12: Instantiation of the model for the smart DC example.

To ensure that the product meets the specifications given by the customer, samples of the product are taken during the circulation step. In case the product does not meet the requirements, an adjustment of the composition of the product has to be performed. Sampling events are interesting to better understand the working of the process, but they are not logged in the system. However, taking a sample affects the circulation of the product and is visible as two shortly followed peaks in the time series of the flow of the product in the tank. Based on this, it is possible to retrieve sampling events by analysing sensor data, as depicted in Figure 13.

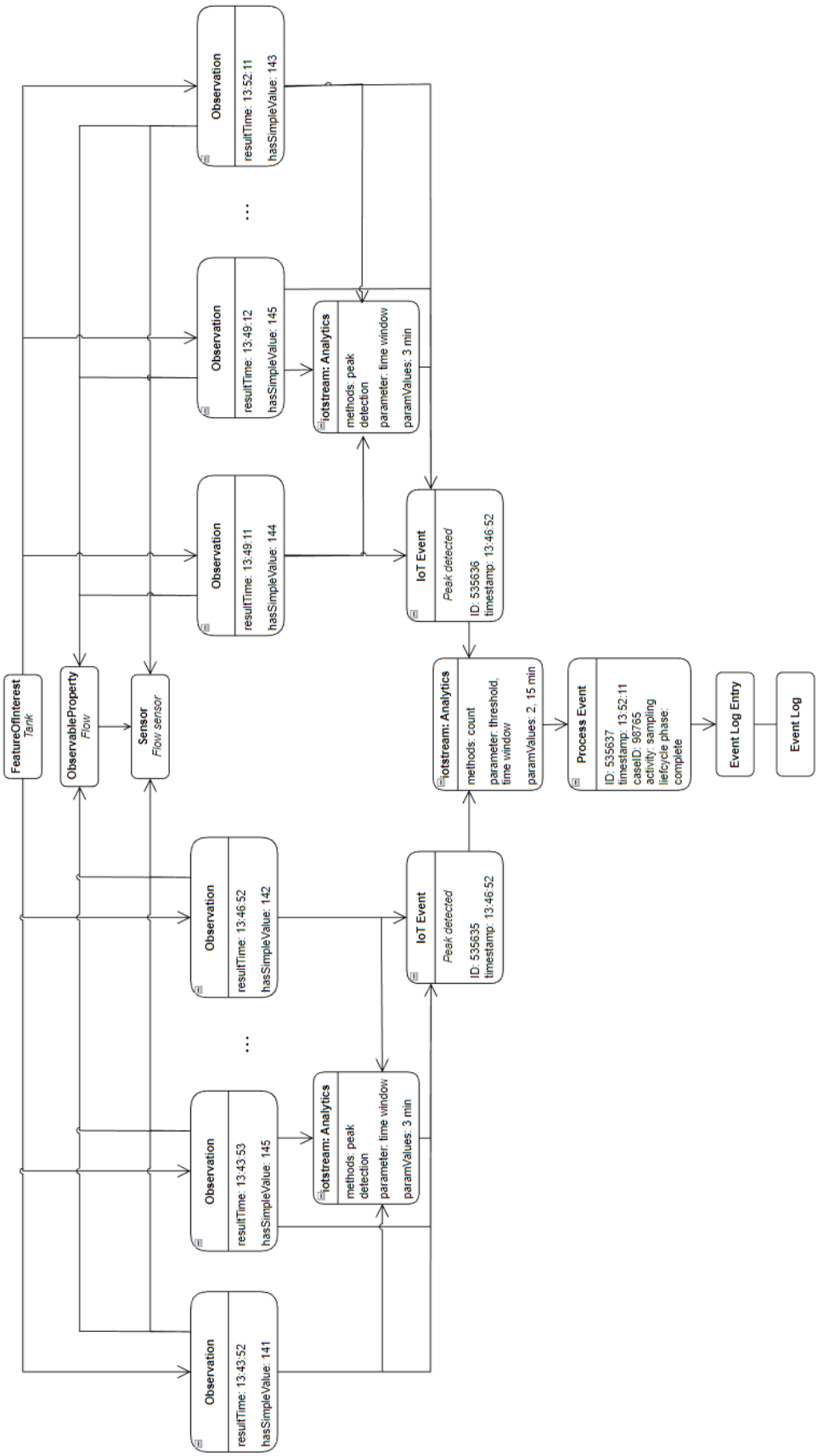


Figure 13: Instantiation of the model for the production process.

Observations of the Observable Property *Flow*, relating to the Feature of interest *Tank*, are made every second by the *Flow* Sensor. Observations are monitored by an Analytics instance to detect *Peak* IoT events within windows of three minutes. Based on this, another Analytics counts the number of peaks and derives a *Sampling* Process event when two peaks are detected within 15 minutes.

This last case study shows that the model is capable of depicting real scenario's in complex processes. Moreover, it illustrates the use of more data-driven Analytics (a peak detection algorithm, a frequently used technique in time series data analysis), on a longer sequence of Observations (180 consecutive Observations, although only a few are shown in Figure 13 for simplicity).

9 Discussion

Previous models such as those presented by Ghattas et al. (2010)) and Gunther and Verbeek (2014)) cannot properly represent the subtlety of IoT data, e.g., the different relationships that IoT data can have with the process (e.g., with the control-flow or the context, with several instances of the process). The treatment of the data is usually simple and, apart from a few exceptions (e.g., Seiger, Zerbato, Burattin, Garcia-Banuelos, and Weber (2020)), only the control-flow perspective is considered. The model presented in this paper is the first that is capable of conceptually merging the domains of the IoT and PM, from sensing to logging. No preexisting model can represent raw IoT data, the event log, and the intermediary steps (i.e., IoT events) in between. This contributes to addressing two challenges identified in Janiesch et al. (2020): 1) connection of analytical processes with the IoT (C3) by linking IoT data and traditional process data and 2) detecting new processes from data (C8) by proposing a new way of modelling IoT and process data that can help discover new processes. Additionally, existing models on the PM side often focus on the control-flow and do not provide a proper representation of the context of the process. Moreover, in joining the IoT and PM, this model brings clarity to the critical zone where IoT data are prepared to make them suitable for PM, as context data or as process events. The important concepts of event and context, which differ in PM and the IoT, are compared and represented together in a single model, creating a common vision on the concepts of IoT and PM.

The two-fold validation of the model demonstrated evidence of community support as derived from an expert survey and its usability through three case studies. All experts agree to some extent that the model properly links the concepts of IoT data and process data, and the vast majority believe that it provides added value for process mining with IoT data. In particular, the model is perceived as complete,

concise, and clear. However, some of the experts expressed moderate disagreement on two points. First, some experts consider the model to be overly oriented towards the IoT. While the authors acknowledge that the number of concepts taken from IoT ontologies is higher than the number of their PM counterparts, they primarily see this as a consequence of the higher level of refinement of IoT ontologies compared to PM models. More precisely, some of the respondents proposed adding additional concepts from the PM domain, mainly case and the event log itself, as in XES and OCEL (see Ghahfarokhi, Park, Berti, and van der Aalst (2021); Gunther and Verbeek (2014)). Case was subsequently included as an attribute of Process event, and Event log was added and linked to Event in Section 7.

Second, the distinction between event types is not unanimously accepted, the concept of context event in particular being questioned by a fraction of the respondents. Interestingly, the opinion of the experts is very divided on this issue. Some agree with the different types of events, some disagree, and a large portion is neutral. This disagreement appears to be the result of a lack of understanding of one or several event type(s) (mainly context events). Reasons for this could be that context events are more "novel" than process and IoT events, which *de facto* coexist in virtually all the IoT PM literature. Another possible explanation is that context in general is not yet frequently studied in IoT PM and, as shown in Section 2, the conception of the context can vary among researchers. Finally, the disagreement could be a consequence of the brevity of the description of the event types necessary to keep the survey length within reasonable bounds. The authors tried to clarify the different types of events in this paper by providing more extensive explanations in Section 5 and contrasting three case studies in Section 8 where context events are modelled. Moreover, the definition of attributes and linking Event Log Entry with Event can also lead to a better understanding of the different types of events.

Then, one respondent proposed to link the model explicitly with XES, the most well-known standard event log model. After careful consideration, the authors decided not to do so to avoid anchoring the conceptual model to a given event log model, thereby ensuring greater flexibility. In addition, the authors have reservations regarding the suitability of XES for the storage of IoT-enhanced event logs, following Bertrand, De Weerdt, and Serral (2023). Indeed, XES assumes that all events are linked to one (and only one) case, which does not hold with IoT events and context events. Regardless, the addition of the concept of Event Log is a step towards well-known event log standards such as XES and is considered by the authors to be an intermediary solution between linking the conceptual model with XES and keeping it completely separate. Although the model presented in this paper is the only one which has demonstrated

suitability for IoT-enhanced logs, other models recently proposed in the PM literature, such as OCEL (Ghahfarokhi et al. (2021)), OCED (OCED working group (2023)) or event graphs, could also turn out to be appropriate for storing IoT data. In particular, the notion of object, which is more flexible than the concept of case, may be suitable for IoT and context events that may not necessarily relate to a single instance of the process. More specifically, features of interest could be represented as objects and could delimit the scope of observable properties or context parameters by being linked with events that pertain to these objects. As an example, IoT and context events derived from the temperature and humidity sensors could be related to Pallet and Refrigerated Truck objects in the second case study. This being said, not all concepts from our model can be translated into OCEL/OCED concepts in a straightforward manner, without restricting the model. In doing so, many concepts described in our model could be instantiated as events, objects or object attributes, yielding a complex log structure and removing most of the semantics and the ability to reason with the concepts of the model. Figure 14 depicts a matching of some of the concepts of our model with the concepts of the latest OCED standard published on the website of the PM task force[1]. Although many concepts and relationships of the model comply with OCED, some constructs pose problems. It is for instance difficult to transcribe analytics into OCEL/OCED terms, and implementing different types of events could also complicate the architecture of the log. Next to this, some relationships are also complex to represent in OCED, e.g., observation, as an object attribute value in OCED, should only be linked with one object (either feature of interest or sensor). In addition to these difficulties, the novelty of object-centric event log models makes them rather unstable, compared to the well-established XES format, and subject to series of improvement rounds (e.g., Goossens, De Smedt, Vanthienen, and van der Aalst (2023)), making it hazardous to ground our model on them at this moment. Moreover, OCEL, like XES, currently does not satisfy the requirements of an IoT-enhanced storage format (Bertrand et al. (2023)).

Finally, another comment that was formulated was whether it was possible to represent causal relationships with the model, for instance, to express that a context event could "trigger" a process event. This is not explicitly stated in the presentation of the model Bertrand et al. (2021), but a link between an IoT event and process or context events can be seen as a causal relationship. The addition of an explicit link between context and process event in the model should be further explored.

[1] https://www.tf-pm.org/upload/1678694478319.pdf

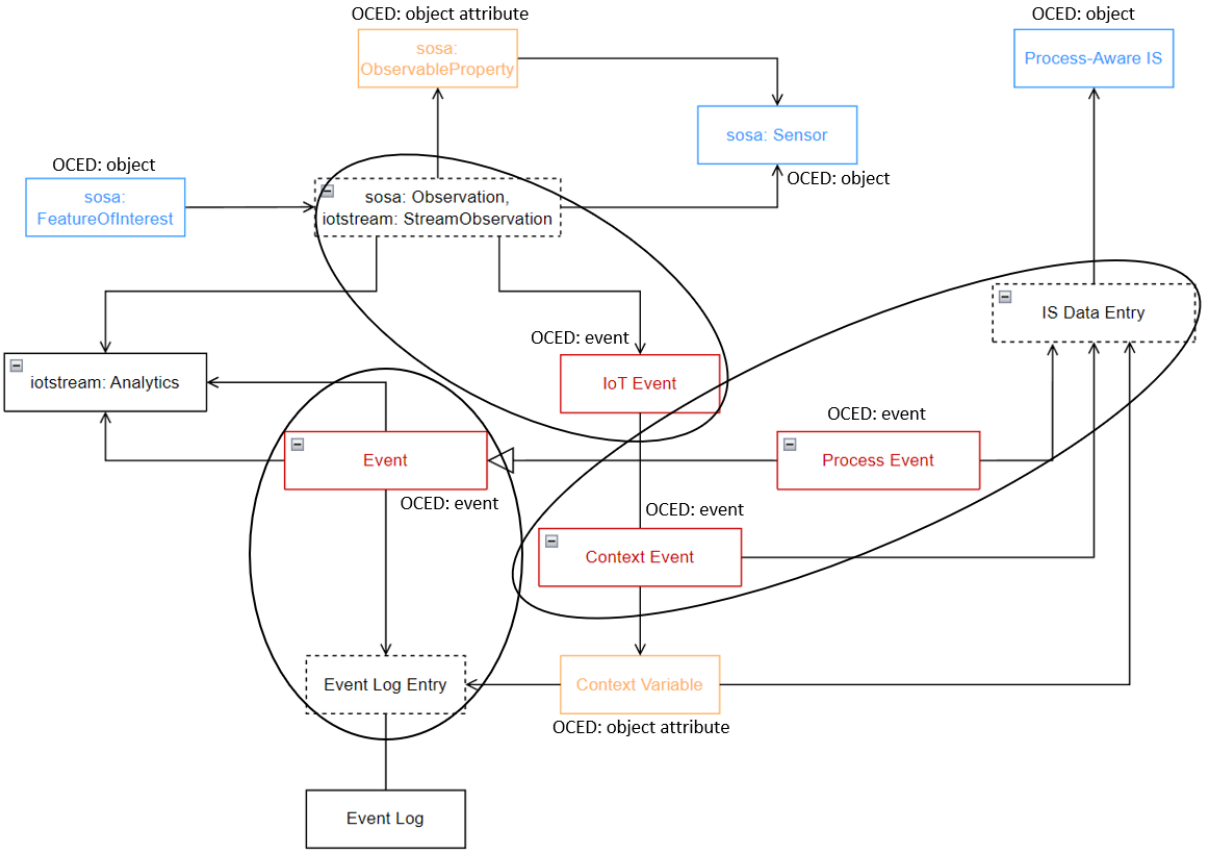


Figure 14: Correspondence of the concepts of our model with the OCED framework. Red concepts would be OCED events, Blue concepts would be OCED objects, Orange concepts would be OCED object attributes. Observation would be merged with IoT event, IS data entry would be merged with process and context events, and event log entry would be merged with event. Event log would be an overarching concept containing all the other concepts of the model. Analytics has no direct equivalent in OCED. Note that some details of the model (relationship names, cardinalities, attributes) are omitted to improve the readability of the figure.

10 Related Work

As stated in the previous section, to the best of the knowledge of the authors, there is no other existing model that unites the concepts of the IoT and PM. Therefore, in this section we present an overview of previous research efforts in using IoT data in process mining.

Despite the potential of IoT data to capture the context of processes, this topic has received little research interest. Most of the literature on using IoT data in PM focuses on activity recognition, i.e., the derivation of an event log from IoT data, to which existing PM techniques are then applied. The main advantages of this approach is that organisations typically collect large quantities of data with sensors, which means that a large amount of information about processes is available (albeit requiring preprocessing). However, several challenges must be addressed to make this possible. In particular, the gap between sensor measurements and event logs must be bridged (Leotta, Mecella, and Mendling

(2015); van Eck, Sidorova, and van der Aalst (2016)) and the logs must be properly segmented (Leotta et al. (2015)).

van Eck et al. (2016) propose to create an event log from sensor data in two main steps: mapping sensor measurements to process activities and then grouping activities into instances of the process. An event log can also be created by applying activity recognition techniques to IoT data from a smart space (Sztyler, Carmona, Völker, and Stuckenschmidt (2016)).

Soffer et al. (2019) highlight the possible benefits of using complex event processing (CEP) to perform process mining on sensor data, with a four-step methodology: 1) data preprocessing, 2) event abstraction, 3) enriching with context information and 4) finding recurrent patterns. Focusing on an industrial application, Seiger et al. (2020) propose to transform raw IoT data into events and activities using CEP and event/activity detection and refinement techniques. Another framework to create an event log from industrial IoT data is proposed by Trzcionkowska and Brzychczy (2018). The four steps of their approach are data preprocessing, clustering low-level data, classification to find activities and events from clusters, and creation of the final event log.

Koschmider, Mannhardt, and Heuser (2019) propose a model to go from low-level events captured by sensors to instances of the process, by aggregating low-level events into high-level events using methods such as CEP. High-level events are contextualised to form activity instances, which are then correlated to create process instances. In further work, the authors present a more systematic and complete framework (Koschmider et al. (2020)) that assumes a sensor event log as input and consists of four steps: (1) event correlation, (2) activity discovery, (3) event abstraction, and (4) discovery of processes, which outputs a process model. This approach was applied to a smart home scenario by Janssen, Mannhardt, Koschmider, and van Zelst (2020), who use clustering techniques to discretise continuous sensor time series to extract activities.

Note that the main steps of these frameworks can be linked with the model; see Figure 15.

11 Conclusion

The contribution of this paper is twofold: the validation of the conceptual model with 1) two additional case studies and 2) expert-based validation and the presentation of the improved version of the model (e.g., by adding attributes to the most important concepts). The refined model bridges the gap between IoT data capture and PM data logging and describes the concepts from both domains. Furthermore, it can

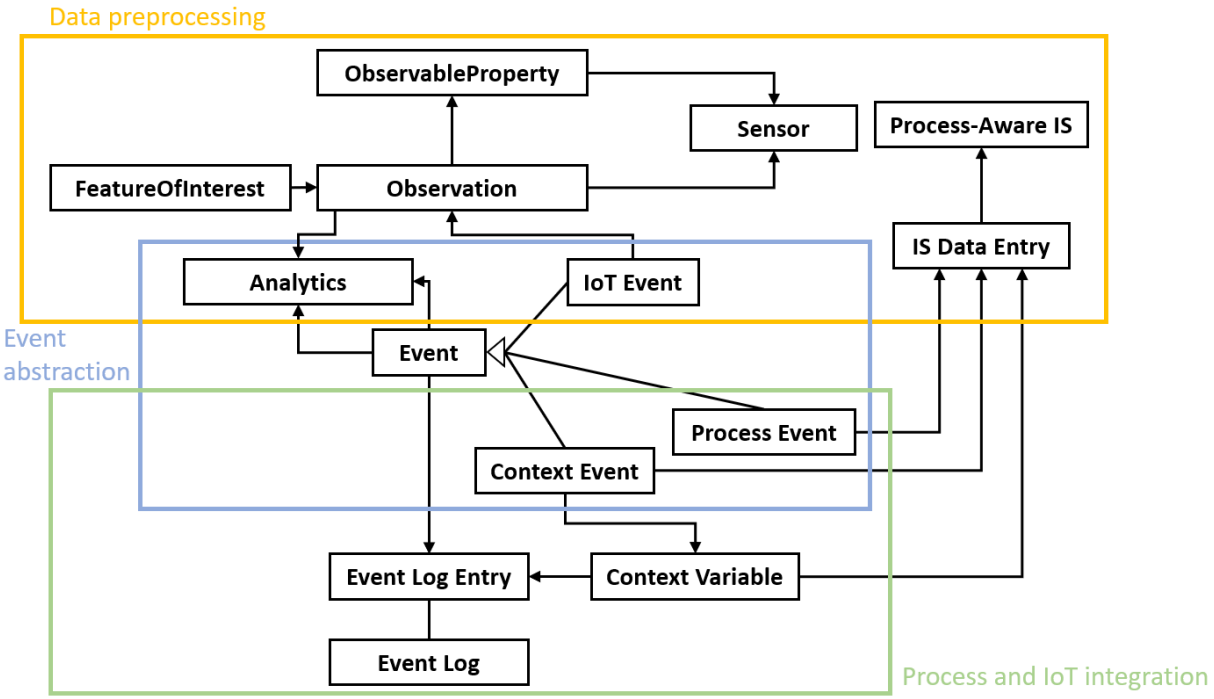


Figure 15: Translation of typical IoT PM frameworks on the conceptual model.

serve to model use cases where IoT data can be used in PM. We believe that this will help and encourage other researchers to advance the integration of IoT and PM and help them in their investigation of IoT PM.

In future work, we plan to consider further improving awareness and acceptance of the model in the community. More in-depth discussions (e.g., in workshops) could also be organised to reach a consensus on aspects on which the surveyed experts did not unanimously agree. We also plan to create a novel IoT-enhanced event log model based on this conceptual model to bring together process events, context events, and IoT events in a single log. We believe that this approach could also help create a consensus on the distinction of events proposed in the model. Finally, we intend to develop algorithms designed to mine IoT-enhanced event logs that would also take into account the discovery of the context.

References

Banham, A., & Wynn, M. T. (2021). xpm: A framework for process mining with exogenous data. , 12.

Bertrand, Y., De Weerdt, J., & Serral, E. (2021). A bridging model for process mining and iot. In *International conference on process mining* (pp. 98–110).

Bertrand, Y., De Weerdt, J., & Serral, E. (2023). Assessing the suitability of traditional event log

standards for iot-enhanced event logs. In *Business process management workshops: Bpm 2022 international workshops, münster, germany, september 11–16, 2022, revised selected papers* (pp. 63–75).

Brunk, J. (2020, Jun). Structuring business process context information for process monitoring and prediction. In *Cbi* (p. 39–48). IEEE.

da Cunha Mattos, T., Santoro, F. M., Revoredo, K., & Nunes, V. T. (2014). A formal representation for context-aware business processes. *Computers in Industry*, *65*(8), 1193–1214.

Dees, M., Hompes, B., & van der Aalst, W. M. (2020, Oct). Events put into context (epic). In *Icpm* (p. 65–72). IEEE.

Dey, A. K. (2001). Understanding and using context. *Personal and Ubiquitous Computing*, *5*(1), 4–7.

Dorsemaine, B., Gaulier, J.-P., Wary, J.-P., Kheir, N., & Urien, P. (2015, Sep). Internet of things: A definition & taxonomy. In *Ngmast* (p. 72–77). IEEE.

Elsaleh, T., Enshaeifar, S., Rezvani, R., Acton, S. T., Janeiko, V., & Bermudez-Edo, M. (2020, Feb). 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.

Ghahfarokhi, A. F., Park, G., Berti, A., & van der Aalst, W. M. P. (2021). Ocel: A standard for object-centric event logs. In *Ntdis* (Vol. 1450, p. 169–175).

Ghattas, J., Soffer, P., & Peleg, M. (2010). A formal model for process context learning. In *Business process management workshops* (Vol. 43, p. 140–157). Springer Berlin Heidelberg.

Goossens, A., De Smedt, J., Vanthienen, J., & van der Aalst, W. M. (2023). Enhancing data-awareness of object-centric event logs. In *Process mining workshops: Icpm 2022 international workshops, bozen-bolzano, italy, october 23–28, 2022, revised selected papers* (pp. 18–30).

Gunther, C. W., & Verbeek, H. (2014, Mar). Xes standard definition.

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS quarterly*, 75–105.

Janiesch, C., Koschmider, A., Mecella, M., Weber, B., Burattin, A., Di Ciccio, C., ... et al. (2020, Oct).

The internet-of-things meets business process management: A manifesto. *IEEE Systems, Man, and Cybernetics Magazine*, *6*(4), 34–44. (arXiv: 1709.03628)

Janowicz, K., Haller, A., Cox, S. J. D., Le Phuoc, D., & Lefrançois, M. (2019, May). Sosa: A lightweight ontology for sensors, observations, samples, and actuators. *Journal of Web Semantics*, *56*, 1–10.

Janssen, D., Mannhardt, F., Koschmider, A., & van Zelst, S. J. (2020). Process model discovery from sensor event data. In (p. 12).

Koschmider, A., Janssen, D., & Mannhardt, F. (2020). Framework for process discovery from sensor data. In (p. 8).

Koschmider, A., Mannhardt, F., & Heuser, T. (2019). On the contextualization of event-activity mappings. In *Business process management workshops* (Vol. 342, p. 445–457). Springer International Publishing.

Leotta, F., Mecella, M., & Mendling, J. (2015). Applying process mining to smart spaces: Perspectives and research challenges. In *Advanced information systems engineering workshops* (Vol. 215, p. 298–304). Springer International Publishing.

Lourdusamy, R., & John, A. (2018, Jan). A review on metrics for ontology evaluation. In *2018 2nd international conference on inventive systems and control (icisc)* (p. 1415–1421). IEEE. Retrieved from `https://ieeexplore.ieee.org/document/8399041/` doi: 10.1109/ICISC.2018.8399041

Mannhardt, F. (2016, Dec). *Sepsis cases - event log.* 4TU.ResearchData.

Noy, N. F., & McGuinness, D. L. (2001). Ontology development 101: A guide to creating your first ontology. , 28.

OCED working group. (2023, Mar). *Object-centric event data (oced) call for action: Reference implementations* (Tech. Rep.). Retrieved from `https://www.tf-pm.org/upload/1678694478319.pdf`

Porzel, R., & Malaka, R. (2004). A task-based approach for ontology evaluation. In *Ecai workshop on ontology learning and population, valencia, spain* (pp. 1–6).

Rosemann, M., Recker, J., & Flender, C. (2008). Contextualization of business processes. *IJBPIM*, *3*(1),

47.

Seiger, R., Zerbato, F., Burattin, A., Garcia-Banuelos, L., & Weber, B. (2020, Oct). Towards iot-driven process event log generation for conformance checking in smart factories. In *2020 ieee 24th international enterprise distributed object computing workshop (edocw)* (p. 20–26). IEEE.

Serpanos, D., & Wolf, M. (2018). *Internet-of-things (iot) systems*. Springer International Publishing. Retrieved from `http://link.springer.com/10.1007/978-3-319-69715-4`

Soffer, P., Hinze, A., Koschmider, A., Ziekow, H., Di Ciccio, C., Koldehofe, B., ... Song, W. (2019). From event streams to process models and back: Challenges and opportunities. *Information Systems*, *81*, 181–200.

Sztyler, T., Carmona, J., Völker, J., & Stuckenschmidt, H. (2016). Self-tracking reloaded: Applying process mining to personalized health care from labeled sensor data. In *Transactions on petri nets and other models of concurrency xi* (Vol. 9930, p. 160–180). Springer Berlin Heidelberg.

Tax, N., Sidorova, N., Haakma, R., & van der Aalst, W. M. P. (2018). Event abstraction for process mining using supervised learning techniques. *arXiv:1606.07283 [cs]*, *15*, 251–269. (arXiv: 1606.07283)

Trzcionkowska, A., & Brzychczy, E. (2018, Sep). Practical aspects of event logs creation for industrial process modelling. *Multidisciplinary Aspects of Production Engineering*, *1*(1), 77–83.

Valderas, P., Torres, V., & Serral, E. (2022). Modelling and executing iot-enhanced business processes through bpmn and microservices. *Journal of Systems and Software*, *184*, 111139. doi: https://doi.org/10.1016/j.jss.2021.111139

van der Aalst, W. M. P., & Dustdar, S. (2012). Process mining put into context. *IEEE INTERNET COMPUTING*, 5.

van der Werf, J. M. E. M., Verbeek, H. M. W., & van der Aalst, W. M. P. (2012). Context-aware compliance checking. In *Business process management* (Vol. 7481, p. 98–113). Springer Berlin Heidelberg.

van Eck, M. L., Sidorova, N., & van der Aalst, W. M. P. (2016, Jun). Enabling process mining on sensor

data from smart products. In *Rcis* (p. 1–12). IEEE.

Wasserkrug, S., Gal, A., Etzion, O., & Turchin, Y. (2008). Complex event processing over uncertain data. In *Debs* (p. 253). ACM Press.

A Questionnaire extract

The goal of this survey is to evaluate and validate a conceptual model that connects IoT data with process event logs.

A.1 Preamble: respondent profile

The goal of this section is to get some info on your background, to e.g. help us identify potential trends in the answers based on the main research interests of the respondents.

1. In which sector do you currently work?

   O Academia O Industry

2. What is your current job title?

   .....

3. What is your knowledge of

   - IoT technologies?

     O No knowledge at all O Little knowledge O Fair knowledge O Good knowledge O Excellent knowledge

   - Processs mining?

     O No knowledge at all O Little knowledge O Fair knowledge O Good knowledge O Excellent knowledge

   - Business process management?

     O No knowledge at all O Little knowledge O Fair knowledge O Good knowledge O Excellent knowledge

   - IoT data processing?

     O No knowledge at all O Little knowledge O Fair knowledge O Good knowledge O Excellent knowledge