International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. #, No.# (Mon-20..)

Applying Process Mining to Generate Business Process Models from Smart Environments

EL KODSSI Iman[1], Sbai Hanae[2] and Kabil mustapha[3]

[123]*Laboratory of mathematics, computer science and applications (LMCSA), FST Mohammedia, Hassan II university of Casablanca,Morocco.*

E-mail address: elkodssi.iman@gmail.com, hanae.sbai@fstm.ac.ma, kabilfstm@gmail.com

Abstract: The management of business processes went through several changes. On the one hand, business intelligence (BI) is becoming more popular among businesses as a way to cut costs, boost service quality, and enhance decision-making. On the other hand, we use business process management in a smart environment. So these data sources produced in this environment via sensors, actuators, and other devices are more varied and unstructured, so to apply the process mining techniques, it is necessary to transform them into a structured format. Several works have been done in this direction, and the authors have contributed to the improvement, but the problem is that there is not yet an approach that formalizes the transformation in a general way regardless of the type of sensor data element. Our approach is based on a model-driven architecture (MDA), which allows us to generate source-to-target data transformations. The main objective is to establish the MDA approach via transformation rules based on machine learning techniques.

Keywords: Process mining (PM), raw sensor log, event log, Model Driven architecture (MDA).

1. Introduction

The emergence of new communication networks and the Internet of Things (IoT) has created a whole new concept in which sensors are linked into our environment, providing researchers and experts with massive volumes of data [1]. In this context, according to the work, the number of objects connected to the Internet of Things, for example, connected cars, industrial machines, meters, consumer products, etc., is expected to grow by an average of 21% between 2016 and 2022, or 18 billion units globally in the range [2]. This IoT device has profoundly changed personal lives and the world of work. They have introduced advanced sensors, enabling the creation of new models. Moreover, Internet of Things (IoT) devices are capable of generating massive amounts of data. It is crucial to implement automated data handling processes to keep up with this abundance of information, as it would simply be impossible to manage and analyze such quantities of data manually given the daily time constraints.

In addition, IoT devices need to be connected in a way that suits enterprise operations. If business processes help companies achieve their objectives, IoT capabilities can help organizations establish business processes-aware IoT by linking the physical and digital worlds. However, business processes have evolved due to the explosion of the Internet of Things (IoT) and real-time data collection. But it doesn't stop there. By enabling companies to extract valuable information from this real-time data, process mining has revolutionized process management. It's crucial to underline that the effective application of process mining methods requires clearly organized data, in particular an event log. These event logs provide a precise record of every activity and interaction in a specific process. As a result, process mining goes beyond the simple collection of real-time data to provide significant operational insights.

Furthermore, to generate business processes via IoT devices, it is crucial to transform sensor logs into event logs to apply PM techniques. Several transformation approaches have been proposed [3], [4], and [5]. Several transformation approaches have been proposed. In this context, we have proposed in [6] and [7] a large comparative study. Among these difficulties is that the study requires detailed browsing behaviors of different target groups. Among these difficulties is that the study needs detailed navigation behavior of different target groups. Moreover, the study of existing approaches also shows that these approaches do not allow the generation of all types of sensor logs. This means that only one type of sensor log can be reviewed. Similarly, another important requirement is to save time and effort and reduce errors by automating the modification of models as much as possible. Our approach is based on MDA, which allows us to generate source-to-target data transformations to make current models more reusable. In this context, model transformation is an essential part of MDA. Thus, in this article, the main objective is to establish the MDA approach, which has the need to formally express the transformations between models, thus making them productive, and to formalize not only the languages in which these are described but also the meta-models describing these languages. Then, using the MDA approach via transformation rules based on machine learning techniques that we have studied in existing approaches, this transition is carried out with a level of PIM2PSM transformations to generate a target meta model, which is an event log. This paper is organized as follows: The first section provides a brief description of the sensors and sensor logs monitored by the MDA approach. Some related works are presented in the second section with a comparative study that summarizes the weak points of the existing approaches. Our approach is described in the third part with a presentation of the proposed meta model. The fourth section shows the applicability of our approach through an illustrative example. The last section concludes the paper and presents future work

2. BACKGROUND REVIEW AND FOUNDATIONS

A. *Process mining (PM)*

Process mining combines data science and process science to improve the study of operational processes using event logs (Figure 1). Process mining begins by collecting data about processes as they occur. Process mining's purpose is to extract an explicit process model from event logs, i.e., the task of creating a process model from a log of events that is compatible with the observed dynamic behavior. Using the event log, various process mining types can be performed in order to generate valuable process-related insights. There are three types of process mining. [8] (figure 1).

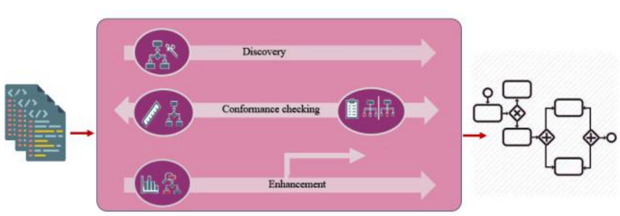

Figure 1:Axes of Process Mining

The first axe is process discovery. A discovery technique may be used to extract process models that reflect process behavior from an event log [4]. Many algorithms focus on deducing the sequence of process activities from the data. The second type of process mining is conformance [9]. As part of the conformity check, this step seeks to assess the extent to which the data relating to the events corresponds to a given reference model, as well as detect inconsistencies between event log behavior and the process model [8]. The third type of process mining is enhancement. Using process data, these techniques can help enhance and expand an existing process model. Model repair is an upgrade type that enables the alteration of a process model based on event logs.

*1) **Event log :***

The event log standard known as XES is based on XML. For the exchange of events log data between applications, it offers a structured format. Its organization is based on a well-defined flow of activities and events. It contains a log as a collection of activities with the name traces. A trace is a chronological list of events [10]. They are each represented by an XML element. Time stamps, activities, creators, descriptions, and other information are stored in the attributes.

For more details, a unique identification for each case named Case id, the activities that each case contained named 'Activity', a reference to when each activity was executed named 'Timestamp'. Aside from this, an event log can additionally include details about the type of event named transaction type, the resource connected with the event named resource, and other details about the activity or scenario. [11]

*2) **Sensor log:***

Sensor logs are the output of a device that detects and responds to some type of input from the physical environment. This output can be used to provide information to another system or to guide a process [3]. The events contained in sensor logs are detailed enough, but the actions in mining models should be more abstract. For example, the time series, as a special case of sensor logs, is defined as follows: A data point is a measurement taken by a particular sensor at a given time that records the result of the measurement. Such data can play an important role in improving management.

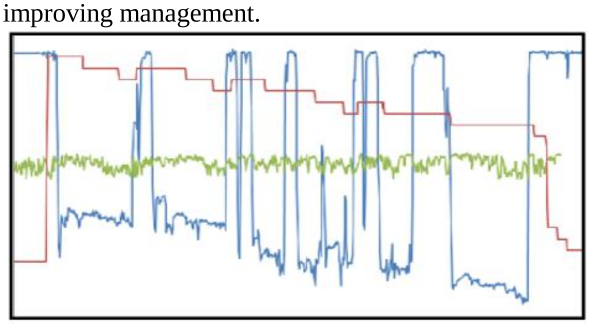

Figure 2: Example of sensor data[4]

For instance, [4] provided an illustration of a sensor log that reflects the measurements of three sensors through time in the form of a chronological series.

Machine learning (ML) is a technique for teaching computers to process data more efficiently. Machine learning's goal is to learn from data and retrieve useful facts.

3) *Machine learning*

To solve data challenges, machine learning employs a variety of algorithms. Data scientists want to emphasize that there is no one-size-fits-all algorithm that is best for solving an issue [12]. The type of algorithm used is determined by the type of problem to be solved.

Supervised machine learning is the search for algorithms that reason from externally given cases to generate broad hypotheses that subsequently predict future instances. In other words, the purpose of supervised learning is to create a compact model of class label distribution in terms of predictor characteristics. A few samples of supervised learning algorithms are linear regression, logistic regression, artificial neural networks, and support vector machines.

Linear Regression	In order for the algorithm to anticipate new inputs, linear regression involves fitting a continuous linear function through the data.
Logistic Regression	Is a known statistical method for determining the contribution of several factors to a pair of outcomes.
Artificial Neural Networks	To achieve good performance, neural networks use extensive interconnections of ”neurons,” which are basic processing components.
SVMs (Support Vector Machines)	It is used for classification and employs kernel approaches to handle the more challenging scenario of non-linearly separable patterns.
k Nearest Neighbors	Is one of the machine learning methods that is regarded as being the simplest. The algorithm uses the outputs of its nearest neighbors in the training set to estimate the output of any new input after memorizing the training set.

Unsupervised machine learning: Unlike supervised learning, the algorithms are left to uncover and show the fascinating structure of the data on their own. Unsupervised learning algorithms learn a limited number of features from the data. When new data is introduced, it employs previously learned characteristics to identify the data’s class. Unsupervised learning algorithms include, but are not limited to, K-means clustering and dimensionality reduction algorithms.

K-Means Clustering	This algorithm automatically creates k unique clusters. The variables in the data are grouped together based on relationships between them in this kind of unsupervised learning.
Dimensionality Reduction Methods	One example is the Principle Component Analysis Algorithm (PCA), whose objective is to minimize the projection error by shortening the distance between each feature and a particular projection line.

4) *Model Driven Architecture (MDA)*

A technique for application modeling and generation that has gained a lot of interest recently is called model-driven architecture (MDA). Many organizations are now considering the concepts of MDA [13] and [14], which were promoted by the Object Management Group (OMG), as a way to plan out and manage their application solutions.

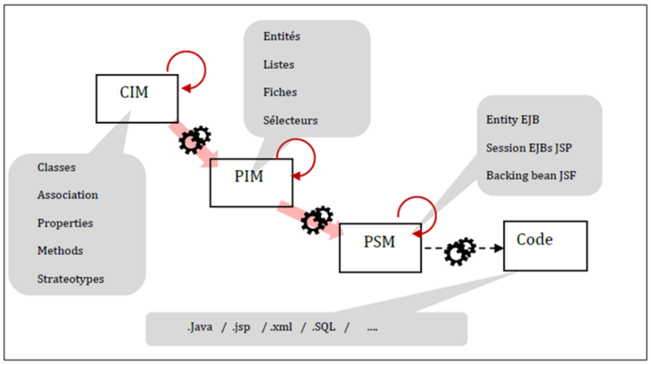

Figure 3: Model driven Architecture

This strategy is based on the development of source models and their transformation through various levels of abstraction until the source code is produced automatically. The MDA also includes the definition of several standards [15]. Standardized models and subsequent changes among them are the foundation of model-driven architecture (MDA), which is founded on these ideas. According to MDA, links and rules are represented by a collection of six connected models. It offers a set of guidelines for structuring these model specifications.

- CIM is independent of any computer system. It is the business model or the model of the application domain. The CIM allows the vision of the system in the environment where it will operate, but without entering into the details of the structure of the system or its implementation. It helps to represent what the system will have to do exactly.
- PIM: It is unrelated to any technical platform, such as EJB, CORBA,.NET, and so on, and does not include information about the technologies that will be used to deploy the application. It is a computer model that represents a partial view of a CIM.
- PSM: It is dependent on the technical platform specified by the architect. The PSM primarily serves as a foundation for the generation of executable code for the technical platforms. The PSM describes how the system will use these platforms. There are several levels of PSM. The first, from the transformation of a PIM, is represented by a platform-specific UML schema.

3. State of Art

A. *Related works*

In the context of business process management, and in particular in an intelligent environment, there are a limited number of works that generate output event logs for any input sensor log. Many methods and approaches have been proposed. for business process management through machine learning techniques to transform inputs into a structured format. [3] [4] [5] [16] In order to automatically discover the process through PM techniques [3], first we will introduce the criteria on which our comparison study is based, and then we will show and debate the comparative study results. The authors of [16] proposed a method for discovering process models from an email log. The method is capable of processing sent and received emails using various process models. The method is able to process emails exchanged in a variety of processing models. Hierarchical clustering and K-means algorithms are used. Emails are grouped based on their subject matter, then based on the process instance they are part of, and finally based on email activity using the hierarchical and k-means clustering algorithms.

The study [5] offered a method to cover the creation of an event log for use by process mining tools by extracting information from complicated semi-unstructured databases. The authors of [4], [5], [16], and [17] look at a method for converting sensor data into an event log that can be fed into any process mining algorithm. The mapping of sensor measurements to human activities and the clustering of activities in the same context are the two key issues this study addresses in the application of process mining techniques to sensor data. from a new point. A strategy for discovery processes from raw sensor data was given in [3]. This study proposes a framework for extracting event location sensor data to identify processes and activities. The framework is adaptable enough to be used with any data set, even unprocessed sensor data.

B. *Criteria for comparison*

The works included in the comparative study are evaluated using seven criteria, which are listed in the following order:

Types of coverage Domain: determines the method used to prepare event logs before applying process mining techniques. This criterion is used to evaluate the application of machine learning when preparing event logs before applying process mining techniques.

Axes of PM: It indicates the category of process mining techniques (discovery, conformance, or enhancement) studied in the work. The objective is to identify the most interesting area for research.

Log Input: Are the data derived from different sources in a smart environment or in another one that is populated with unstructured data?

Log output: identify the data generated after applying the approaches associated with machine learning algorithms.

Language or Technologies: indicates the algorithm used to transform raw, unstructured data from the sensor log into event logs with a good level of structuring capable of coping with these traditional PM assumptions

Table 1: comparative study (Part 1)

Approach	Type of coverage Domain	Axe of PM	Input log					Language & Technologies	Output
			Loca-tion data	Time Serie .S	Text data	Device	Time		
[3]	Transformation	Discovery	+	-	-	+	+	Correlation	Event log
[4]	Transformation	Discovery	-	+	-	+	+	Segmentation & Clustering	Event Log
									Event Log
[5]	Transformation	Discovery	-	-	+	+	+	Domain-Specific Language	Event Log
[14]	Transformation	Discovery	-	-	+	+	+	Preprocessing Clustering k-means clustering R statistical language hierarchical	Event Log

Table 2: comparative study (Part 2)

[13]	Transfor-mation	Discovery	-	-	+	+	+	***	Event log
[1]	Transfor-mation	Multi perspective conformance checking and alignment of event logs to process models	+	-	-	+	+	Hierarchy Clustering	Event log
[15]	Defining Meta Model	Discovery	-	-	+	-	+	***	Event log
[16]	Transfor-mation	Discovery	-	-	+	-	+	Classification	Event log
[17]	Transfor-mation	Discovery	-	-	+	-	+	Classification	Event Log

By analyzing these approaches, we find that only one particular category of sensor log is covered by each method. All of these methods use machine learning strategies to support a structured event log, for example, segmentation [4], clustering [16], [15], and classification [18], [19]. For the work [4], the input data is of the time series type. From another point of view, [3] and [1] have treated location data. and there are also approaches that choose to generate event logs via text data [19], [18], [17].

Overall, a common limitation of most work is that the models are not formalized enough to see how their transformation would affect the standard mapping and the domain expert designing the output.

Moreover, as it is noted in the input element, each of these approaches has treated the input from a particular point of view. So, our approach proposes first a meta model, which makes it possible to define a sensor log for any category whose objective is to propose an MDA.

After the text edit has been completed, the paper is ready for the template. Duplicate the template file by using the Save As command, and use the naming convention prescribed by your conference for the name of your paper. In this newly created file, highlight all of the contents and import your prepared text file. You are now ready to style your paper; use the scroll down window on the left of the MS Word Formatting toolbar.

4. Our Approach: Applying process mining to sensor data in smart environment: Model driven architecture

In order to formalize the mapping to an event log, we propose a framework based on the model-driven architecture. In this approach, we propose a raw sensor log metamodel based on embedded sensors. The code is then generated using M2T (model-to-text) transformations to convert the data and build the meta-model event log that already exists in the literature [20]. Figure 4 highlights the many steps that characterize our method.

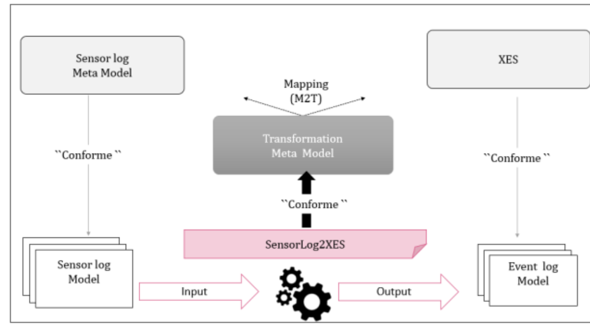

Figure 4. Our Approach for Generating event log

The input format for the above table is as follows:

A. Meta Model Sensor Log

Given the absence of a meta model associated with a log sensor in the literature, we have proposed a meta model that contains the following elements:

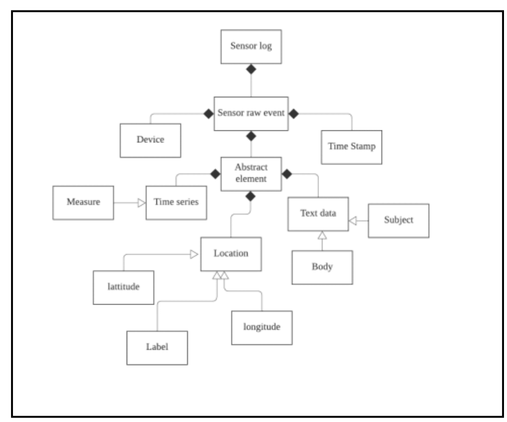

Figure 5. Sensor log meta model

- Time stamp: the time when the event occurred.
- Device: This term refers to the event's resource. This might refer to a physician, a healthcare practitioner, or a medical gadget. The way to obtain the sensor events produced by different sensors becomes more and more complex. These sensors tend to produce massive, distributed, heterogeneous, and complex logs in the new environments of the Internet of Things without a structured schema. This brings an additional level of complexity to what is defined as a sensor raw log, namely: Time series
- Time series: Sensor logs could well be thought of as a time series of sensor readings. A given data point is a measurement collected by a specified sensor at a certain point in time and recorded as the measurement's value.

- Text data: Each row represents one email in the input data, which is a csv file. The columns reflect email information such as sender, receiver, topic, content, timestamp, and so on.
- Location: Motion sensors in smart homes and smart industries provide input. This type of event data may or may not include a unique identifier that may be used to determine who caused the event. It may also include a geographical description, such as the locations where the activities were carried out, as well as their frequency and linkages.

B. M2T transformation:

We need to be able to determine the equivalent elements of the source and tar-get meta-models in order to complete this transformation. The mapping rules between sensor raw log and event log meta-model elements are exposed in the following table (see Table 3).

The mapping rules are defined in the second phase. These models can be built using the following rules:

Table 3: Identification Of Corresponding Elements Of Sensor log Meta Model element and event log Meta Model element

Rule	**Sensor log Meta Model element**	**Sensor log Meta Model element**	**Transformation Rules name**
Rule 1	AbstractElement	EventLog	AbstractElement2EventLog
Rule 2	TextData	Instances	TextData2Instances
Rule 3	Body&subject	Activity	Body&subject2Activity
Rule 4	TimeSeries	Instances	TimeSeries2 Instances
Rule 5	DiversityMeasures	Activity	DiversityMeasures2Activity
Rule 6	LocationLog	Instances	LocationData2Instances
Rule 7	PositionGeographies&Label	Activity	PositionGeographies&Label2Activity
***	TimeStamp	TimeStamp	***
***	Device	Device	***

The first step is to identify the symbols that we will use later, as well as their meaning, namely:

Idi: The Body of each i email

Awi: Words associated with an email i

Si: The subject of each email i

{m1,...,mn}: Measurement set

Fij: Frequency associated of a i th email and j th word

C: Model cluster

D : Geographical distance matrix of all clusters.

Li : Label associeted of each cluster i

Ai : Context information of location data .

Longi/Lati : The weight of word tj in file Idi

1) *Rule 1: AbstractElement2EventLog*

The abstract element can be represented in different formats, namely text data, time series, and location, and the list of events in the log is generated by applying three rules: the instances of activities are obtained by applying the rule TextData2 Instances for each TextData element of this composite, and if the abstract element is in time series format, we apply the rule TimeSeries2 Instances. And the last element is applying the rule LocationLog2Instances to each location log element of this composite.

Algorithm 1 AbstractElement2EventLog	
1:	**Input**: AbstractElement This rule creates the event Log which is conform to event log meta model from a Sensor raw event of sensor log meta model .
2:	The name of event is the name of the Sensor raw event Parameters of a sensor raw event are time resource and abstract element. If an abstract element is TextData, apply TextData2 instances.
3:	If the abstract element is TimeSerie apply TimeSerie2 instances.
4:	If the abstract element is location log, apply LocationLog2 instances.
5:	**Output:** Event Log

Rule 2: TextData2Instances.

Text data is transformed into an activity instance by applying the "subject&body2activity" rule to each body and subject element associated with the text data.

Algorithm 2 :Text Data2 Instances	
1:	**Input :** TextData
2:	This rule creates the instances from a text data meta model
3:	The name of event is the name of the Sensor raw event
4:	Parameters of a Sensor raw event are time resource and abstract element
5:	if abstract element is Text Data apply Body&subject2Activit
6:	**Output :** Instances

Rule 3: Body&subject2activities.

This rule allows the two elements body and subject to be combined to form an activity:

Algorithm3:Body&subjet2activities

Input : Body
for each Idi
  Separate each IDi to words AWi using white space as a delimiter
end for
return tokens AWi
for each AWi **in** Idi
  Remove stop words ,numbers , punctuation and lowering capital letters
return vectors{AWi}
for each IDi
  for each awj **in** IDi
    Appling numerical Term frequency Inverse Document : Fij<= Tf Idf (AWi)
  end for
end for
return formulate term document matrix
for each Si attach to Idi
 for each Idi
  for each awj in Idi
   Cutting word SWi by separated words using white space as delimiter process
  end
 end
   If
    SWi similaire a AWi **then** Fij<= 2* Fij
   End if
   Extract the verb-noun pairs are likely to be candidates of being activities
end
Initialize: each email as an individual clusters Ci <= IDi
Repeat
  /*Compute the proximity matrix */
  Merge(Ci , Cj) if Ci similar to Cj
  update the proximity matrix

```
until only a single cluster remains
 return {P C1={ID1,ID2,…,IDk},.., PCn={
IDi,IDj,…,IDm }}
for each PCi
        /*Calculate distance between  two different
        emails j and k in the same process model
        cluster Ci  :*/
         Distance (Eij , Eik )
        Applying hierarchical clustering
end
return { IC1={ID1,ID2,…,IDk}, ...,ICn={
IDi,IDj,…,IDm }  }
assumption: an email can contain 0, 1 or more
activities.
for each IDi in ICi

        Extracted feature of each IDi
        Applying classification to obtain pertinence
        sentence
        for each ACi
        /* choose the top N verb-noun pairs
        mentioned in the activity cluster*/
        Activity <= verb-noun( IDi)
        end for

end for
return {ID1={A1,A2,…,Ak},...,
IDn={Ai,Aj,…,Am }  }
Output:Activity
```

Rule 4: TimeSeries2 Instances

A time series is transformed into an instance by applying the "Measure2Activity" rule for each diversity measure .

Algorithm 4 TimeSeries2 Instances

```
Input: TimeSeries
This rule creates the event of event log meta model
from a sensor raw event of sensor loT meta model
The name of event is the name of the Sensor raw event
Parameters of a Sensor raw event are time resource and
abstract element
if abstract element is time Series apply
Measure2Activity
Output: Instances
```

Rule 5:DiversityMeasures2Activity

The rule "DiversityMeasures2Activity" allows the creation of an activity from Diversity Measures as described in the following

Algorithm 5 DiversityMeasures2Activity

```
Input: {m1,…,mn}
Create the proposed detailed segmentation of size wi to
include the segments
 ( w1,...,wk ).  With k<n.
Each wi corresponds to a set of measures {mi}i<k
for each segment wi do
        Feature selection and calculation of wi
          Characterization of each segment wi
          /*Labeling such that similar segments receive
the same label */
        if  wi similar to wj  then    label (wi)=label (wj)
end for
return  set of labelled segments
Lj  <=  Label (wj)
Initialisation : Let C1,….Ck  be the initial cluster
centers
for each point Oi in Ci do
            Calculate the distance between Li and the Oi.
            Assign each  Li that is closest to the other
centroids  Oi.
            Cnew <= newcluster(Li)
Repeat
            Update its center by averaging all of the
            points oj that have been assigned to it.
Until convergence
return   {C1, . . . , Ck }
Identify to each group an activity according to the
characteristics of the center of gravity of each cluster
Ai <=Activity (Ci)
Output : Activity
```

Rule 6: LocationLog2Instances

A location log is transformed into an instance, and the list of activities is obtained by applying the rule 'PositionGeographies&Label2activity' for each couple position and label of the composite one.

Algorithm 6 LocationLog2Instances

```
Input : Location Log
This rule creates the event of event log meta model
from a Sensor raw event of sensor log meta model
The name of event is the name of the Sensor raw
event
Parameters of a Sensor raw event are time resource
and abstract element
if abstract element is LocationLog  apply
PositionGeographies&Label2Activity
Output : Instances
```

Rule 7 : PositionGeographies&Label2activity.

The rule "PositionGeographies&Label2activity" allows the creation of an activity from PositionGeographies and Label .

Algorithm7 PositionGeographies&Label2activity

```
Input: locationData
C <= [ ]
for each locationData  ai  do
        cnew <= newCluster(lati, longi)
        add cnew to C
end for
D <= ( )
for each ci,cj in  C:
      merge clusters that are close to each other and
represent the same label
      if label(ci) = label(cj ) then
            C ← C \ {ci, cj} ∪ {ci ∪ cj}
            recompute centroid of the new cluster {ci ∪ cj}
            D ← update geographical distances matrix

      end if
end for
Identify to each group an activity according to the
characteristics of the center of gravity of each cluster
Ai <=Activity (Ci)
Output:Activity
```

5. Evaluation

For our case study, we use one of the elements of the sensor log for sensors that measure the temperature in the context of climate adaptation in different cities. For the 16 automatic adaptations of the movement of the intelligent windows, the latter makes it possible to react according to the environmental conditions surrounding the installation. Our objective is to apply the proposed rules for generating a model of an XES event log. At the CIM level, one of the input elements of the sensor log is a time series. This type contains:

- Case ID: a unique number for each line.
- Date and Time: timestamp of the measurement.
- Temperature: temperature measurement at the time stamp.
- Humidite (%): air humidity measurement at the time stamp

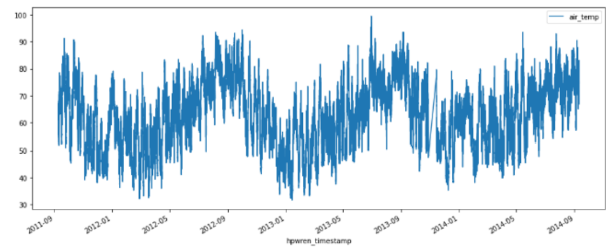

Figure 6. Temporal Variation of Temperature Measurements from Sensor Data in the Context of Climate Adaptation

Time series are divided into separate windows to transform sensor data into a set of measurements. These small windows are then labeled. To determine which actions were performed during these observations, the data segments collected in the sensor log are grouped together on the basis of the properties of each segment. To determine which actions were performed during these observations, the data segments collected in the sensor log are grouped together on the basis of each segment's properties. And for each class, we assign the associated activity. We used our recommendations to extract the event log. At the CIM layer, we get an event log that conforms to the XES.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<log xmlns="http://www.xes-standard.org/">
  <trace>
    <string key="Case ID">1</string>
    <event>
      <string key="Timestamp">2023-09-19 10:00:00</string>
      <float key="Temperature (°C) ">23.5</float>
      <string key="Activity">Démarrage du système</string>
  </event>
  </trace>
  <trace>
    <string key="ID Case">2</string>
    <event>
      <string key="Timestamp">2023-09-19 10:15:00</string>
      <float key="Temperature (°C) ">24.0</float>
      <string key="Activity">Connexion d'Utilisateur</string>
  </event>
  </trace>
</log>
```

Figure 7. Sample event output showing temperature Measurements from Sensor Data in the Context of Climate Adaptation to XES format

As perspective and to illustrate the modeling example, figure14 shows the modeling process using BPMN2.0.

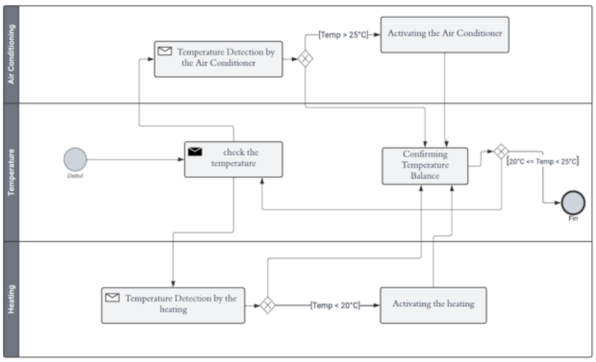

Figure 8. The process model for an temperature Measurements system

This model represents an indoor temperature management process based on measured values. It uses an exclusive gateway to make decisions based on the detected temperature range and act accordingly, activating either heating or cooling or doing nothing if the temperature is already within a comfortable range. The process ends once one of these actions has been taken, confirming that the temperature is balanced.

6. CONCLUSION

With the evolution of the Internet of Things and the strong presence of sensors in various environments, we are encouraged to integrate product data for the automatic discovery of business processes and to exploit the process information by providing a better understanding via the use of process mining. However, because the data in our resource becomes obviously voluminous and unstructured, it is not compatible with process mining. To solve this problem, we propose an MDA approach for the generation of a vent log from the sensor log, whose objective is to automatically discover a business process. We proposed a set of mapping rules for applying the M2T transformation of the sensor log model into the event log model. Our future work will give a semantic representation to describe the concepts related to the IoT and the elements of an executable business process described in BPMN. We use standard semantic technologies, in particular ontologies.

ACKNOWLEDGMENT

I would like to express my sincere gratitude to the Journal of Intelligent Information Systems for their effort and the opportunity provided. I also wish to extend my deep appreciation to Mrs. Sbai for her invaluable guidance throughout this research, and to Mr. Kabil for his unwavering support and valuable advice.

REFERENCES

[1] O. Vermesan and P. Friess, Eds., *Internet of Things Applications - From Research and Innovation to Market Deployment*. Taylor & Francis, 2014. doi: 10.1201/9781003338628.

[2] B. b. Gupta and M. Quamara, ‘An overview of Internet of Things (IoT): Architectural aspects, challenges, and protocols’, *Concurr. Comput. Pract. Exp.*, vol. 32, no. 21, p. e4946, 2020, doi: 10.1002/cpe.4946.

[3] A. Koschmider, D. Janssen, and F. Mannhardt, ‘Framework for Process Discovery from Sensor Data’.

[4] M. L. van Eck, N. Sidorova, and W. M. P. van der Aalst, ‘Enabling process mining on sensor data from smart products’, in *2016 IEEE Tenth International Conference on Research Challenges in Information Science (RCIS)*, Jun. 2016, pp. 1–12. doi: 10.1109/RCIS.2016.7549355.

[5] Á. Valencia Parra, B. Ramos Gutiérrez, Á. J. Varela Vaca, M. T. Gómez López, and A. García Bernal, ‘Enabling Process Mining in Aircraft Manufactures: Extracting Event Logs and Discovering Processes from Complex Data’, 2019, Accessed: Oct. 12, 2023. [Online]. Available: https://idus.us.es/handle/11441/133387

[6] E. Iman, M. D. Laanaoui, and H. Sbai, ‘Applying Process Mining to Sensor Data in Smart Environment: A Comparative Study’, in *Innovations in Smart Cities Applications Volume 6*, M. Ben Ahmed, A. A. Boudhir, D. Santos, R. Dionisio, and N. Benaya, Eds., in Lecture Notes in Networks and Systems. Cham: Springer International Publishing, 2023, pp. 511–522. doi: 10.1007/978-3-031-26852-6_47.

[7] I. Elkodssi, M. D. Laanaoui, and H. Sbai, ‘Toward a New Framework for Process-Aware IoT Discovery’, in *Innovations in Smart Cities Applications Volume 6*, M. Ben Ahmed, A. A. Boudhir, D. Santos, R. Dionisio, and N. Benaya, Eds., in Lecture Notes in Networks and Systems. Cham: Springer International Publishing, 2023, pp. 793–803. doi: 10.1007/978-3-031-26852-6_73.

[8] M. de Leoni, W. M. P. van der Aalst, and M. Dees, ‘A general process mining framework for correlating, predicting and clustering dynamic behavior based on event logs’, *Inf. Syst.*, vol. 56, pp. 235–257, Mar. 2016, doi: 10.1016/j.is.2015.07.003.

[9] J. Wang, R. K. Wong, J. Ding, Q. Guo, and L. Wen, ‘Efficient Selection of Process Mining Algorithms’, *IEEE Trans. Serv. Comput.*, vol. 6, no. 4, pp. 484–496, Oct. 2013, doi: 10.1109/TSC.2012.20.

[10] N. Martin, M. Swennen, B. Depaire, M. Jans, A. Caris, and K. Vanhoof, ‘Batch processing: definition and event log identification’, 2015, Accessed: Oct. 12, 2023. [Online].

Available: https://documentserver.uhasselt.be//handle/1942/20099

[11] E. Rojas, J. Munoz-Gama, M. Sepúlveda, and D. Capurro, 'Process mining in healthcare: A literature review', *J. Biomed. Inform.*, vol. 61, pp. 224–236, Jun. 2016, doi: 10.1016/j.jbi.2016.04.007.

[12] P. V. Klaine, M. A. Imran, O. Onireti, and R. D. Souza, 'A Survey of Machine Learning Techniques Applied to Self-Organizing Cellular Networks', *IEEE Commun. Surv. Tutor.*, vol. 19, no. 4, pp. 2392–2431, 2017, doi: 10.1109/COMST.2017.2727878.

[13] A. Karkouch, H. Mousannif, H. A. Moatassime, and T. Noel, 'A model-driven architecture-based data quality management framework for the internet of Things', in *2016 2nd International Conference on Cloud Computing Technologies and Applications (CloudTech)*, May 2016, pp. 252–259. doi: 10.1109/CloudTech.2016.7847707.

[14] C. Dickerson and D. N. Mavris, *Architecture and Principles of Systems Engineering*. CRC Press, 2016.

[15] Z. Valero-Ramon, C. Fernandez-Llatas, B. Valdivieso, and V. Traver, 'Dynamic Models Supporting Personalised Chronic Disease Management through Healthcare Sensors with Interactive Process Mining', *Sensors*, vol. 20, no. 18, Art. no. 18, Jan. 2020, doi: 10.3390/s20185330.

[16] D. Jlailaty, D. Grigori, and K. Belhajjame, 'A framework for mining process models from emails logs'. arXiv, Sep. 20, 2016. doi: 10.48550/arXiv.1609.06127.

[17] M. ELLEUCH, N. ASSY, N. LAGA, W. GAALOUL, O. ALAOUI ISMAILI, and B. Benatallah, 'A Meta Model for Mining Processes from Email Data', in *2020 IEEE International Conference on Services Computing (SCC)*, Nov. 2020, pp. 152–161. doi: 10.1109/SCC49832.2020.00028.

[18] Y. Rashnavadi, S. Behzadifard, R. Farzadnia, and S. Zamani, 'Business Process Discovery from Emails: Text Classification and Process Mining - A Case Study of Procurement Process'. Rochester, NY, May 01, 2020. doi: 10.2139/ssrn.3671233.

[19] Y. Rashnavadi, S. Behzadifard, R. Farzadnia, and S. Zamani, 'Discovering Business Processes from Email Logs Using fastText and Process Mining'. Preprints, Sep. 03, 2021. doi: 10.20944/preprints202005.0007.v2.

[20] P. Lam, H. Ahn, M. Park, K.-S. Kim, and K. Kim, *A Proportional Process Mining System*. 2019.

Author 1. EL KODSSI Iman

Iman, E. K., & Hanae, S. (2023, May). A study of extending BPMN for IoT-aware process modeling. In *Proceedings of the 6th International Conference on Networking, Intelligent Systems & Security* (pp. 1-4).

Iman, E., Laanaoui, M. D., & Sbai, H. (2022, October). Applying Process Mining to Sensor Data in Smart Environment: A Comparative Study. In *The Proceedings of the International Conference on Smart City Applications* (pp. 511-522). Cham: Springer International Publishing.

Author 2. SBAI Hanae

Sikal, R., Sbai, H., & Kjiri, L. (2018, October). Configurable process mining: variability Discovery Approach. In 2018 IEEE 5th international congress on information science and technology (CiSt) (pp. 137-142). IEEE.

Sbai, H., Fredj, M., & Kjiri, L. (2014). A pattern based methodology for evolution management in business process reuse. arXiv preprint arXiv:1403.6305

Author 3. Kabil Mustapha and a short biography

Leroux, J., & Schmitz, S. (2015, July). Demystifying reachability in vector addition systems. In 2015 30th Annual ACM/IEEE Symposium on Logic in Computer Science (pp. 56-67). IEEE.

Kabil, M. (1992). Enveloppe injective de graphes et de systèmes de transitions et idéaux de mots (Doctoral dissertation, Lyon 1