ASM Sc. J., 16, 2021

https://doi.org/10.32802/asmscj.2021.767

An Integrated Approach for Discovering Process Models According to Business Process Types

Lamghari Zineb*

LRIT Associated Unit to CNRST (URAC 29), Faculty of Sciences, Rabat IT Center, Mohammed V University in Rabat, Morocco

Process discovery technique aims at automatically generating a process model that accurately describes a Business Process (BP) based on event data. Related discovery algorithms consider recorded events are only resulting from an operational BP type. While the management community defines three BP types, which are: Management, Support and Operational. They distinguish each BP type by different proprieties like the main business process objective as domain knowledge. This puts forward the lack of process discovery technique in obtaining process models according to business process types (Management and Support). In this paper, we demonstrate that business process types can guide the process discovery technique in generating process models. A special interest is given to the use of process mining to deal with this challenge.

Keywords: Process mining; process discovery; business process type; process mining techniques; process model perspectives

I. INTRODUCTION

Business Processes (BPs) are nowadays a crucial element in any organisational structure. They are established to manage and improve the company business. In this context, information systems assure the automation of BPs (Nel and Abdullah, 2017; Anon, 2020), by including Business Process Management (BPM) systems, Enterprise Resource Planning (ERP), Customer Relationship Management (CRM), etc. In this sense, information systems record related event data to the BP execution, to analyse and guide issues concerning the creation of the company value. To achieve these objectives, process mining techniques are created.

Process mining is a scientific discipline that focuses on the analysis of event data logged during the execution of a BP, in order to discover, monitor and enhance BPs. It consists o two major phases (Van der Aalst, 2016a): Pre-processing and Processing. Depending on the process subject, we extract data from Repository. Once the data has been extracted; we proceed to filter it from deficiencies in the log cleaning step. By doing so, we arrive to the process model discovery technique. Next, we apply the conformance checking step, which aims at evaluating how well this discovered process model corresponds to reality. This gives input for changing parameters in the main process, and requires going back to the data cleaning step, to re-apply the discovery technique. After evaluation, we diagnose all obtained results, to provide inputs for business process improvements.

Therefore, event data analysis and the discovery technique are required for obtaining suitable process models that takes into consideration: event logs content, their levels of abstraction (Baier *et al.,* 2014) and the process model perspectives (Mannhardt, 2018). By necessity, a suitable process model reflects the business reality and met requirements.

Process model perspectives (Van der Aalst, 2016b) are: Control-flow (The order in which its activities should be executed), Organisational (The resources required for the execution of a process and how they interact with each other), data-flow (The data objects created and updated

*Corresponding author's e-mail: zineb_lamghari2@um5.ac.ma

during the execution of the process) and time (The time-related aspects of the process).

In this work, we assume that the pre-processing phase works directly on cleaning event logs, while the processing phase focuses on analysing events and constructing process models.

In this sense, depending on the input data and the questions that need to be answered, a suitable process model abstraction can be presented. The model may be too abstract and thus unable to answer relevant questions. The model may also be too detailed, e.g., the required input cannot be obtained, or the model becomes too complex to be fully understood. In this context, many process discovery methods assume that recorded events correspond to meaningful activities in the instances of a process.

However, events may be recorded on different levels of granularity. Some events may refer to activities on a high-level of abstraction. Their execution is easily recognizable for process workers. Other events may be recorded with a lower level of abstraction. Multiple of such low-level events may referee to a recognizable high-level activity. When discovering processes based on those low-level events, the resulting process model can impact process workers' structure. Consequently, the discovered model represents the wrong level of abstraction.

Therefore, the event data Domain Knowledge (DK) is the crucial parameter that can guide the process discovery technique❶, in terms of diagnosing and representing process models according to their DK. Besides, the management community has defined the BP type as the DK that can impact (Harmon, 2019; Prakash *et al.,* 2020) the BP representation and treatment ❷.

Based on these two information ❶ and ❷, we assume the following hypothesis: BP type as DK may influence process model that can be discovered with process discovery techniques, in terms of perspectives priority, i.e., according to DK which perspective will be treated or combined firstly with the control-flow perspective for guiding the process discovery in generating process models.

The BP types (Burattin, 2015) are:

- The Management BP: Describes the process of the product or services realization that is provided by the company to their customers.
- The Operational BP: Defines the organisation strategy.
- The Support BP: presents the process that offers resources to other processes to ensure the smooth running of the company.

Thus, we need an approach that can tackle the treatment of events and discover models according to BP types (Burattin, 2015), towards guiding the process discovery technique in discovering suitable models. This can be achieved using DK as the BP type related to multiple process model perspectives. In this context, we must discuss different related issues that involve the intersection between BP types and process mining. For this purpose, our paper proposes an approach that demonstrates how BP type, as DK, will guide and impact the process discovery process.

In this sense, our paper is organised as follows: Section 2 illustrates scientific papers that deals with domain knowledge in the process mining context. Section 3 presents the still encountered related issues of the intersection between BP types and process mining. The section 4 details our approach that consists of guiding the process discovery technique according to BP types. Also, an illustrative example is shown to simulate the applicability of our proposed approach. Conclusion and further directions are presented in section 5.

II. DOMAIN KNOWLEDGE

To identify publications addressing event and model abstractions methods, activity recognition and event processing, we investigated three platforms: SCOPUS database, Process Mining Wiki and Google scholar. SCOPUS is the largest database of peer-reviewed literature. Process Mining Wiki is a publication platform that promotes research on the topic of process mining and contains publications only on process mining. Moreover, to not miss any paper, we also explored Google Scholar which allows a wide range of academic literature. According to our hypothesis (+), we focus on publications related to knowledge-centric methods. These approaches assume knowledge on event abstraction and the process models generation.

Methods based on Complex Event Processing (CEP) (Cugola & A. Margara, 2012) and activity recognition (Liu *et al.,* 2016) typically assume a stream of events over which queries are evaluated. When a query is matched, a high-level activity is detected. Traditionally, CEP does not consider the notion of process instance (i.e., case) and in case of overlapping queries (e. g., shared functionalities) both high-level activities would be detected. Still, there is some work that uses CEP within a business process context (Bülow *et. al.,* 2014; Hallé & Varvaressos, 2014; Oliveira *et. al.,* 2013; Weidlich *et al.,* 2014). However, none of these works treat events based on domain knowledge as BP types.

There are also proposals for supervised event abstraction that are more closely related to the field of process mining. We can cite (Tax *et al.,* 2016) that assume the existence of a labelled training set of traces. This approach is limited to processes without concurrent high-level activities. Also, conditional random fields are used to infer the correct mapping. On the other hand, George *et al.* (2016) assume a labelled training set of events organised in traces. They apply frequent sequence mining and learn constraints from the events. However, the approach does not deal with noise in the event data and treat them without taking into consideration BP objective or type.

Moreover, Senderovich *et al.* (2016) determine an optimal mapping between sensor data of real-time locating system and activities, based on finding an optimal mapping using integer linear programming. In addition, Ferreira *et al.* (2013) assume a complete process model of the high-level activities. They use hierarchical Markov models together with an expectation maximization method to find the mapping between low-level events and the high-level activities in the process model; this recognition does not take into consideration the management view. Later, the work of Ferreira *et al.* (2014) proposes a different greedy approach that can map an existing high-level model with events to produce process models with more or less details. Further, Fazzinga *et al.* (2015) reports that existing methods is applied only on short traces with minimum of 30 events.

The methods developed by (Baier *et. al.,* 2015; Baier, 2015; Baier *et al.,* 2014) assume the knowledge of a single high-level model for the overall process. The goal is to automatically discover the relation between events and activities. Therefore, these methods are mainly targeting the situation where the process is assumed to be well known. The proposed methods use clustering methods and heuristics miner algorithms and answer to the challenge of event logs from processes with concurrent high-level activities and noise. A later proposal uses constraint programming approach (Baier *et al.,* 2015) that only considers the control-flow perspective.

Also, Begicheva *et al.* (2017) demonstrate a method that requires a mapping from low-level activities to high-level activities as input. The method abstracts low-level events by directly replacing the low-level events with the corresponding high-level activities. However, it works only for non-cyclic high-level process models. Here, high-level activities may not be repeated, and it does not consider noise in the low-level event log. Also, the produced process model does not treat the BP objective.

Beyond, Mannhardt (2018) presents a Guided Process Discovery (GDP) method (Plug-in in the prom tool) that uses domain knowledge encoded in multi-perspective activity patterns (Folino *et al.,* 2015) to address the granularity challenge and handle noise. The method lifts low-level events to high-level activities and discovers hierarchical multi-perspective process models. This method gives rentable results in term of event level abstraction (Bose & Van der Aalst, 2009). Besides, this method does not take into consideration the BP type from the configuration phase. The Limit consists on the difficulty of obtaining suitable activity patterns for abstraction, in situations with little domain knowledge. Also, this method does not take into consideration BP type impact on the process model representation.

To conclude, none of the related work tackles the treatment of events according to BP types, towards guiding the discovery process to generate suitable models, using domain knowledge as BP type related to events abstraction and multiple process model perspectives.

In this context, it is required to discuss the still encountered issues that involve the intersection between BP types and process mining.

III. BRIDGING THE GAP BETWEEN PROCESS MINING AND BP TYPES

In this section, we present the still encountered issues related to the intersection between BP types and process mining, which are: Event data quality, Management view

(Configuration phase, BP status, and BP execution), Levels of representing process models or their abstraction that includes process model perspectives and the clustering technique.

A. Event Logs Quality

The quality is achieved by filtering out noise (incorrectly logged), incomplete (missing events), chaotic (arbitrary executed) and infrequent behaviours from event data. These deficiencies are denoted, in Figure 1, respectively as (Domain knowledge N, (Domain knowledge I, etc).

In the literature, we find many research papers (Conforti *et. al.,* 2016; Suriadi *et al.,* 2017) dealing with this topic. This issue has received a significant focus from the research community, where all related ambiguities have been resolved. Here, we propose to use the BP type as an additional filtering parameter, to refine the main BP context.

B. Management View

Information systems record event data. Their configuration is one of the most prominent tasks related to event data preparation.

Generally, a BP is defined with specific parameters as DK (BP type), BP status (informs about the BP life-cycle) and other proprieties that depend on the BP objective and the organisation requirements. These parameters describe the configuration file content, to record behaviours and avoid errors that may be provided during the BP execution. The point has not matched, in this regard, is the definition of the BP type, its particularity and how it may impact the process discovery technique and the process model perspectives, in the term of representation. Available research (Mannhardt, 2018; Conforti *et. al.,* 2016; Boubaker *et al.,* 2016) consider recorded events that are resulting from an operational BP type. These studies do not mention the two other BP types (Support and Management), while the BP type can influence the representation of process models that can be discovered with process discovery techniques. In this respect, the DK or BP type is implicitly defined in the configuration file, it has not been declared as attribute in the event object (a part of the configuration file). This provides another untreated point to consider. Thus, the propriety we propose to use is to declare BP type as an attribute that will determine the BP domain knowledge.

C. Clustering Techniques

The clustering technique is treated in the processing phase as mentioned in (De Souza Oliveiraand Queiroz, 2015). We do not observe it in the preprocessing phase. Introducing this technique into the preprocessing phase can refine the final process model construction.

Moreover, this technique based on DK as BP type definition that groups a set of activity instances with the same context (BP type), to determine a clustered process model. In this sense, for event logs sequences, we can determine the set of similar data (cluster).

D. Levels of Abstraction

The process discovery technique generates models using the 2-D method (Van der Aalst, 2016a), where the process model can be viewed in different levels of abstraction. As mentioned in the Mannhardt (2018) events of low-level are mapped to activity instances of high-level, based on specific DK. The correspondence between these two levels is treated and provided adequate results (Cook *et. al.,* 2013; Mannhardt, 2018; Baier *et. al.,* 2016; Di Ciccio *et al.,* 2020). For instance, one high-level activity instance may result in multiple low-level events being recorded, and vice versa, one such low-level event may relate to multiple high-level activity instances.

The point not yet treated is the BP type (Domain knowledge_L in Figure 1) impact on the process model abstraction, by necessity process model perspectives' priority, i.e., according to the BP type as DK, which process model perspective will be combined with the control-flow perspective (see Figure 1).

E. Synthesis

According to this discussion, we have detected some untreated stages related to BP types (Cf. Figure 2). These questionable stages emerged new issues, relatively to process mining. We have also explored an additional treated phase, which is the configuration phase. Therefore, our contribution will focus on three phases (configuration, pre-processing and processing). In this sense, DK is our guided approach parameter. Thus, we will define DK as (Cf. Figures 2 and 3):

- BP type presents event object in the configuration phase (Domain knowledge_G).
- BP type presents an additional filter to define the main cluster for pre-processing phase (Domain knowledge_C).
- BP type as BP type objective and related to process model perspectives (Domain knowledge_G). Here, we propose to use the bptype as an additional filtering parameter, to refine the main bpcontext.

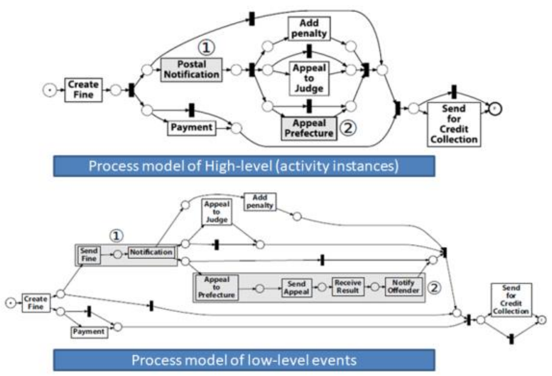


Figure 1. Mapping event logs to activity instances (Mannhardt, 2018)

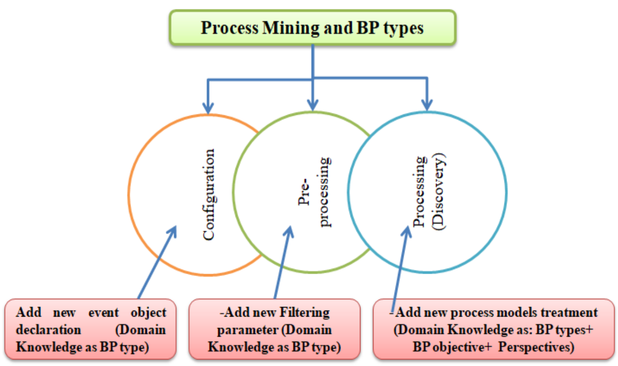


Figure 2. Our contribution aspects

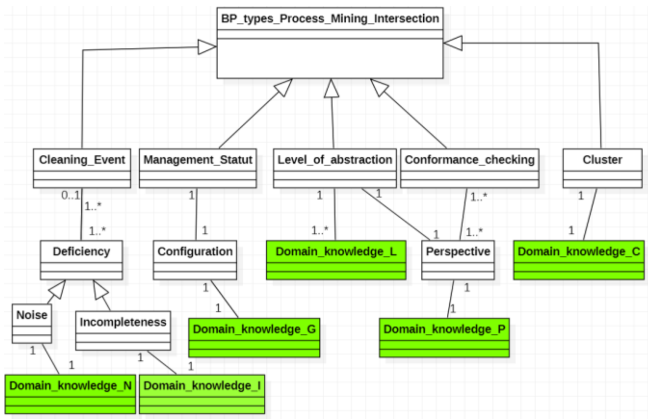


Figure 3. Related issues to the intersection between BP types and process mining

IV. OUR INTEGRATED GUIDED PROCESS DISCOVERY APPROACH

Our approach aims at guiding the process discovery technique in representing process models according to different BP types. It consists of three phases: configuration, pre-processing, and processing (Cf. Figure 4).

Our approach process starts by configuring the information system, where all BPs will be executed and recorded according to the event object parameters. Once, the selected BP is executed, we verify if the configuration elements are successfully achieved. If this later is well done, we proceed to the preprocessing phase, which aims at cleaning event data from deficiencies. If not, we loop back to the configuration phase. Finally, we pass to the processing phase, to obtain process model representation.

Throughout our approach phases:

We add a new event object in the configuration phase 1. After executing a BP and collecting recorded event logs, we arrive to the second operation named 2 that uses an additional filtering attribute, which is the BP type of the pre-processing phase 3. Finally, we apply 4 the processing phase techniques for mining process models according to BP types, by necessity the BP type objective and the process model perspectives (control-flow, time, organisational).

We define different DK, and we obtain different outputs (1-main frame based on the BP type, 2-main cluster based on the BP type, 3-clustered process model based on the BP type).

A. Configuration Phase

The configuration phase (Cf. Figure 4-①) takes as an input a BP model. It gives idea and insights about the BP type. Indeed, the configuration phase takes into consideration the BP type as DK.

The reason to choose the BP type as DK within the configuration phase is:

- The use of management view from early stage.
- The bp type impacts on the next two phases.
- Preparing events in the configuration phase as an early basic filter.

The configuration process consists of:

- Determining the bp domain knowledge as the operational bp, the management bp, and the support bp.
- define the event object content.
- Declare the bp type as attribute in the event object.
- Activate the recording option.

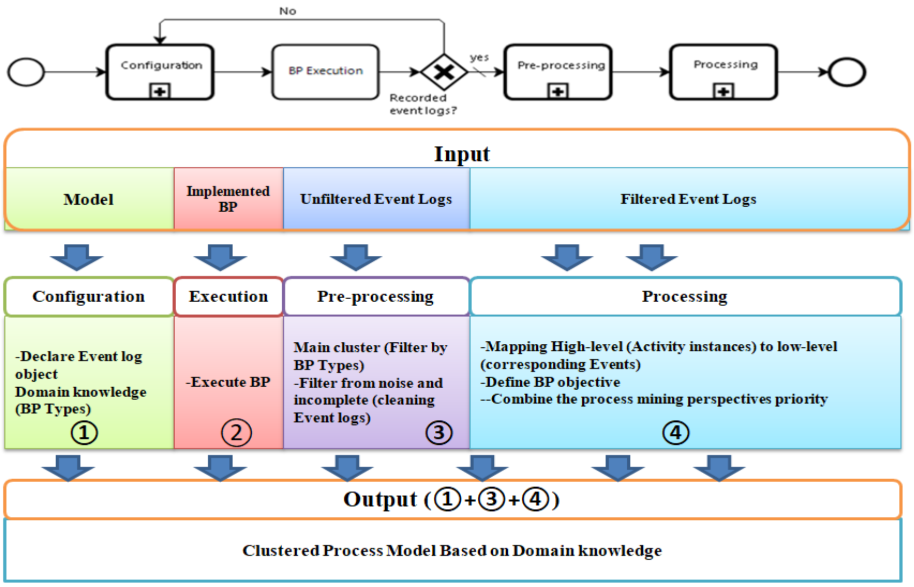


Figure 4. Integrated Guided Process Discovery approach phases

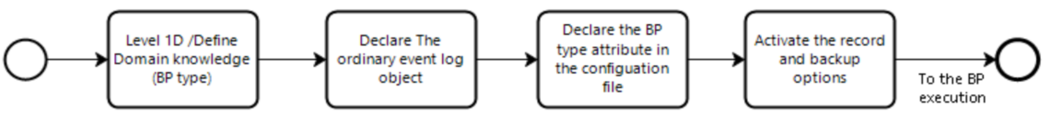


Figure 5. The configuration sub-process

```
example_log_1 %>% #configuring the information with BP types
eventlog(
case_id= "case_id",
activity_id= "activity",
lifecycle_id= "status",
timestamp= "timestamp",
resource_id= "resource",
bp_type= "bp_type"
  )
```

Figure 6. The event object with the BP type attribute

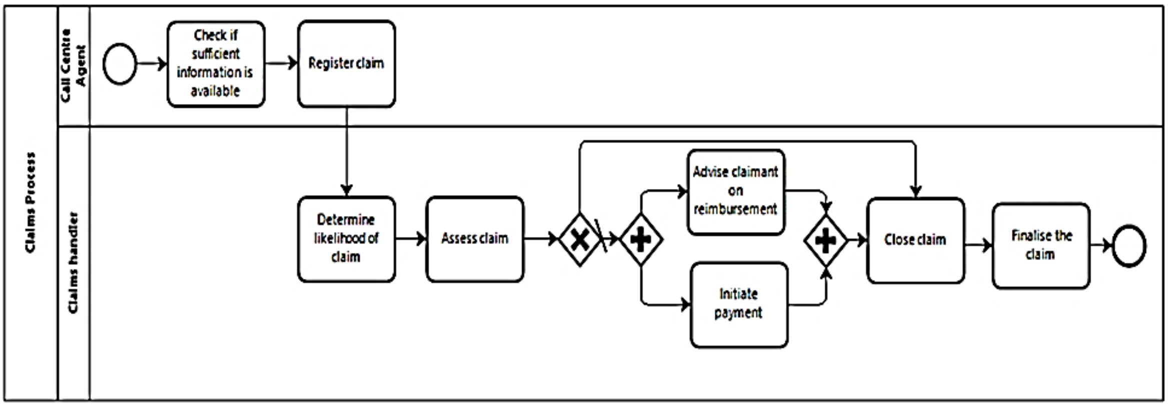


Figure 7. Claims handler Business Process

B. Execution Phase

During the execution phase (Cf. Figure 4-②), we record BP event logs. For instance, the underlined part of the Figure 5 means that the event 1 with the activity 'check if sufficient' was executed by the call centre agent resource in 2008-07-27, where the BP status is running, and the BP type is operational. In this context, the minimum information required of a standard format for an event log is: The case ID is a unique identifier for a process instance, the information stated in chronological order according to events. Also, additional information is possible, we find attributes such as timestamp (the time when the activity took place), resources (who performed the activity), transaction type and costs associated with the event.

The event object, mentioned in Figure 6, is about a running BP of claims handler (Cf. Figure 7). It makes sure that claims are handled efficiently and that payment for valid claims is made. Also, this process consists of making decisions on the extent and validity of a claim, and the checking for any potentially fraudulent activity.

C. Pre-processing Phase

The pre-processing process (Cf. Figures 8 and 4-③) starts by extracting events in the adequate form, to apply the filtering operation. This operation cleans events from deficiencies: noise, incompleteness, and infrequent behaviours.

After obtaining a cleaned event logs, we proceed to the pre-processing phase, to define the main cluster using BP type.

1. Extract Event logs

After the handler claims BP execution, we obtain the data illustrated in Figure 7. The extracted data can differ; it is dependent on the systems which are supporting processes, where some data entails plain text and other data might entail complete databases. Therefore, data should be pre-processed to form proper event logs (event refers to a case, an activity, and a point in time. An event log can be seen as a collection of cases and a case can be seen as a trace/sequence of events). In our illustrative example, we use the prom tool, this is required at least the Xes format.

2. Filter from noisy and incomplete data

Log cleaning is important for the translation into standard formats; is the selection of suitable data, as we have an overwhelming amount of data, only the event logs which are applicable for the relevant processes need to be selected. By the aid of filtering and querying the data minimum information is acquired which is needed within a standard format of an event log. The activity contains process steps that took place such as pay invoice and receive order. The case ID; the information stated in chronological order according to the events, a case ID is a unique identifier for a process instance, which for example could be an invoice or an order. Also, additional information is possible, these are called standard attributes such as timestamp (the time when the activity took place), resources (who performed the activity), transaction type and costs associated with the event.

3. Define the main cluster (Filter by BP type)

Normally, we define clusters in the processing phase. The particularity in this work is to introduce the clustering technique in the pre-processing phase. This is done by filtering events using the BP type parameter. It can define the main cluster, by necessity the main frame of event data. The reason for introducing a main cluster:

- This is made more details about events and process model semantic context.
- Combining filters of noise, incompleteness and BP type provides an advanced pre-processing technique.

In our example, we define the main cluster (events of the same BP type) by filtering events with the operational BP type. It assures that we will work on one BP type and, by necessity, one BP objective. It also presents an additional filter that avoids probable ignored errors during the filtering step (Cf. Table 1).

Table 1. Cleaned and filtered event logs fragment

Case_ID	**Activity**	**Timestamp**	**Status**	**resource**	**BP_type**	**MAIN CLUSTER**
1	check if sufficient information is available	2008-07-27T01:00:00.000+02:00	start	Call Centre Agent	**Operational**	
	idem	2008-09-27T01:00:00.000+02:00	complete	idem		
	register claim	2008-09-17T01:00:00.000+02:00	start	idem		
	idem	2009-05-15T01:00:00.000+02:00	complete	idem		
	determine likelihood of claim	2009-06-15T01:00:00.000+02:00	start	Claims handler		
	idem	2009-06-28T01:00:00.000+02:00	complete	idem		
	assess claim	2010-04-18T01:00:00.000+02:00	start	idem		
	idem	2010-05-18T01:00:00.000+02:00	complete	idem		
	initiate payment	2010-08-18T01:00:00.000+02:00	start	idem		
	idem	2010-11-18T01:00:00.000+02:00	complete	idem		
...						

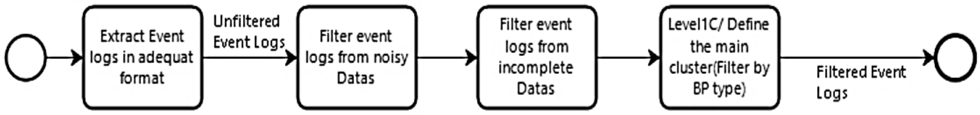
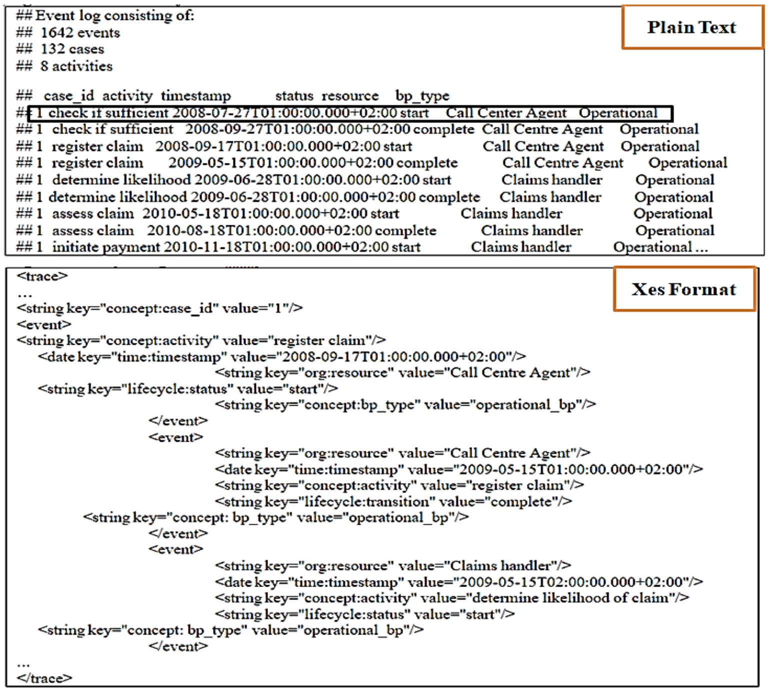


Figure 8. The pre-processing sub-process



Plain Text

```
## Event log consisting of:
##  1642 events
##  132 cases
##  8 activities

##   case_id  activity  timestamp          status  resource    bp_type
## 1  check if sufficient 2008-07-27T01:00:00.000+02:00 start    Call Center Agent   Operational
## 1   check if sufficient   2008-09-27T01:00:00.000+02:00 complete  Call Centre Agent    Operational
## 1   register claim   2008-09-17T01:00:00.000+02:00 start             Call Centre Agent    Operational
## 1   register claim      2009-05-15T01:00:00.000+02:00 complete        Call Centre Agent     Operational
## 1   determine likelihood 2009-06-28T01:00:00.000+02:00 start          Claims handler       Operational
## 1  determine likelihood 2009-06-28T01:00:00.000+02:00 complete    Claims handler       Operational
## 1   assess claim   2010-05-18T01:00:00.000+02:00 start              Claims handler         Operational
## 1   assess claim    2010-08-18T01:00:00.000+02:00 complete         Claims handler         Operational
## 1   initiate payment 2010-11-18T01:00:00.000+02:00 start          Claims handler       Operational ...
```

Xes Format

```
<trace>
...
<string key="concept:case_id" value="1"/>
<event>
<string key="concept:activity" value="register claim"/>
    <date key="time:timestamp" value="2008-09-17T01:00:00.000+02:00"/>
                        <string key="org:resource" value="Call Centre Agent"/>
    <string key="lifecycle:status" value="start"/>
                        <string key="concept:bp_type" value="operational_bp"/>
                 </event>
                 <event>
                        <string key="org:resource" value="Call Centre Agent"/>
                        <date key="time:timestamp" value="2009-05-15T01:00:00.000+02:00"/>
                        <string key="concept:activity" value="register claim"/>
                        <string key="lifecycle:transition" value="complete"/>
          <string key="concept: bp_type" value="operational_bp"/>
                 </event>
                 <event>
                        <string key="org:resource" value="Claims handler"/>
                        <date key="time:timestamp" value="2009-05-15T02:00:00.000+02:00"/>
                        <string key="concept:activity" value="determine likelihood of claim"/>
                        <string key="lifecycle:status" value="start"/>
    <string key="concept: bp_type" value="operational_bp"/>
                 </event>
...
</trace>
```

Figure 9. Event logs Plain text and Xes formats of the claims handler BP

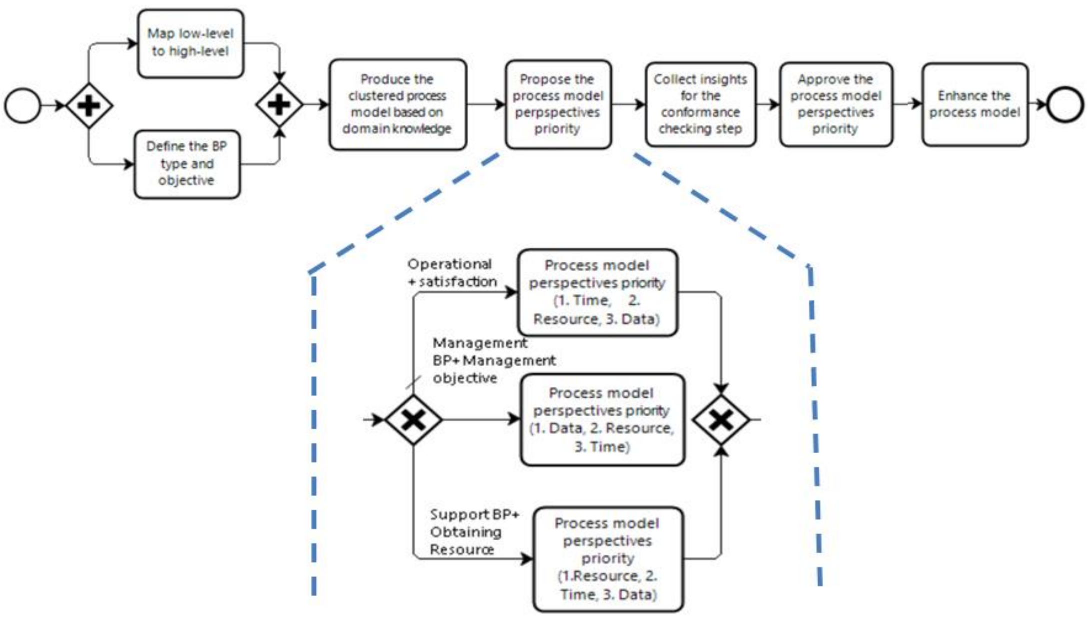
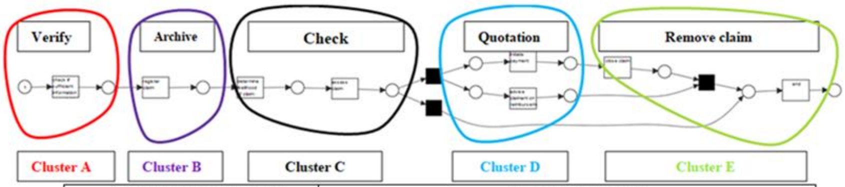


Figure 10. The processing sub-process



High-level activity instances		Low-level events			
hl activity	hl instance	Id	L1 activity	time	resource
Verify	Ins1	Id1	check if sufficient information is available	x	xx
Archive	Ins2	Id2	register claim	y	yy
Check	Ins3	Id3	determine likelihood of claim	z	zz
Check	Ins3	Id4	assess claim	n	nn
Quotation	Ins4	Id5	initiate payment	t	tt
Quotation	Ins4	Id6	Advise claimant for reimbursement	k	kk
Remove claim	Ins5	Id7	Close claim	p	pp
Remove claim	Ins5	Id8	end	j	jj

Figure 11. Clustered process model based on domain knowledge

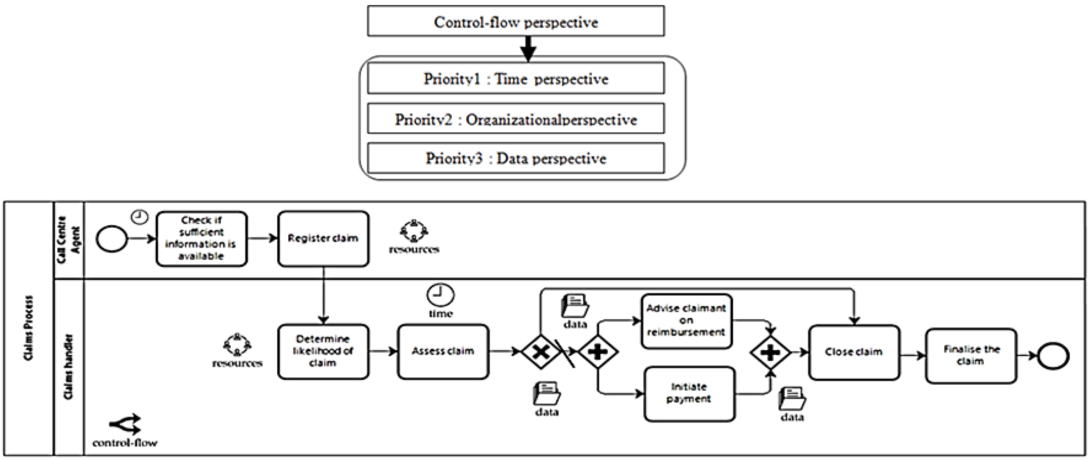
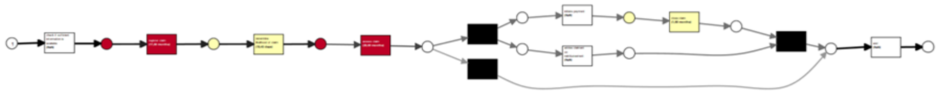
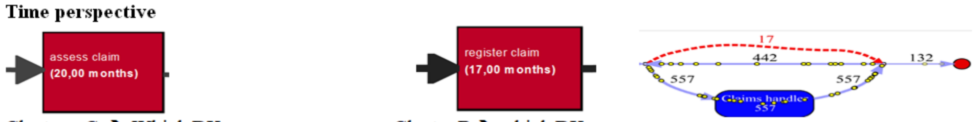


Figure 12. Example of process model perspectives order

Processing phase:
Verify the priority of process mining perspective by applying the replay plug-in in the conformance checking step



Time perspective

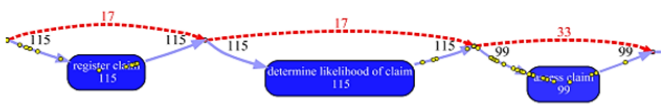


Cluster C ➔ Which DK

Cluster B➔ which DK

Organizational perspective: claim handler and call centre Agent resources

Data flow perspective:



data flowing between these resources causes 3 queues (17, 17, 33)

Figure 13. Approving the process model perspectives priority

D. Processing Phase

After declaring the BP type attribute in the configuration phase and defining the main cluster based on the BP type in the pre-processing phase, we arrive to the processing phase(Cf. Figure 4-④). Here, the domain can be expressed as a function that uses:

1)set of multi-perspective activity pattern, which is mapping between low-level events and instantiations of high-level. This is inspired from the GDP method of (Mannhardt, 2018). 2) BP objective, BP types and process model perspectives.

During the processing phase, the same BP of claims handler will be treated as an illustrative example. Since the Prom tool requires at least the Petri net notation for detecting deviations, we represent the same BP of the Figure 7 with Petri net notation (see Figure 11).

The processing phase consists of (Cf. Figure 10):

- Defining the high-level activity instances and the low-level of its corresponding events based on the recent GDP method (Mannhardt, 2018); mapping the events logs as low-level to its corresponding activity instances as high-level. For instance, the check activity instance contains two sub-events (cf. Figure 11). To achieve this representation, we use the iDHM and MPE plug-ins of prom (process mining tool).
- Using the clustering technique, to get a clustered process model, i.e., clustered process model based on domain knowledge. For instance, in Figure 11, we suppose that we have detected different sub-clusters in our process model. So, in the cluster **A** is expressed by the "check" group. Here, we can determine as set of instances "ins2" described by the recorded events "assess claim and determine likelihood of claim". This is an ascending classification (cluster→activity instance→events. Indeed, we have defined clusters' titles semantically (Staab & Studer, 2010).
- Defining the process model perspectives priority: the BP type gives idea on the main BP objective and gives insights on the process model perspectives order (in which order perspectives will be combined with the control-flow) within this logic:

  The BP type + The BP objective→ the order of representing process model details. For instance, Figure 12 mentioned the representation of our process model perspectives order. This order is applied on the BP of claims handler. It is an operational BP with the objective of client satisfaction.

  - Operational BP + satisfaction ➔(1. Time, 2. Resource, 3. Data).
    - Satisfy the client means that a process respects time limitation. If not, we must verify why resource did not respect the time condition in order to the data-flow into this BP.
  - Management BP + management method ➔(1. Data, 2. Resource, 3. Time).
    - Achieve the management method objective in respect to the data-flow used, into this BP, by resources in time condition.
  - Support BP + resource support ➔(1. Resource, 2. Time, 3. Data-flow).
    - Obtain support resources in respect to the time indication and the data-flow of this BP.
- The perspectives order can be verified in the configuration phase, and it can provide more insights of the probably detected deviations. For instance, in the Figure 13, we have detected firstly the time deviation. Then, by defining on which activity the deviation is provided, we can define the cluster and the concerned domain knowledge. The resource responsible for this deviation and respectively the data-flow contributed to this later. This can acquire the enhancement insights for improving the process model.
- Enhancing the process model.

After comparing the process mining project phases (Van Eck *et al.,* 2015) to our proposed approach, we reveal out with these advantages:

- BP type as domain knowledge is used into three phases.
- Clustering technique is applied from the pre-processing phase.
- Filtering insights are mentioned from the configuration phase.
- Conformance checking indications are appeared from the discovery step.

- Filtering operation is applied from the configuration phase (by BP types).
- Combining the management view with the process model perspectives (process model perspectives priority).
- New process model representation according to BP types.

E. Synthesis

Our approach is applied on three phases: configuration, pre-processing and processing. Throughout these phases, we have presented a guided process discovery approach by BP types.

The general idea consists of taking into consideration the impact of BP types on process model representation. We have treated the still encountered related issues of the intersection between BP types and process mining: Management view, Clustering technique and Process model. The management view takes into consideration the impact of the BP type on event logs recognition. The clustering technique uses BP type as context, to group a set of commune activity instances. The process model perspectives propose a flow of representation guided by BP types. In this sense, the conformance checking technique can approve the application order of process model perspectives according to BP types.

During this approach, the definition of DK differs from the configuration phase to the processing phase.

In the first and second phases, the DK has been defined as BP type, while in the third phase, the DK has been considered as the correspondence of different levels (events and its corresponding activity instances). Also, we have defined DK as a complex function of BP type, BP type objective and process model perspectives.

Then, we used the clustering technique to obtain a clustered process model based on domain knowledge. This aims at presenting an understandable, and easy to manipulate model to the end user, with management view.

By mapping the obtained clustered process model, the BP type objective, and the process model perspectives, we can obtain insights on process model perspectives priority. These perspectives will be combined and treated, by order, with the control-flow perspective. This results a suitable process model according to event logs levels and its perspectives.

V. CONCLUSION

In this paper, we present an approach dealing with the process discovery technique according to BP types. Indeed, we aim to guide the process mining discovery technique, in order to generate suitable process model for each BP type.

For this purpose, we investigate the still encountered issues related to the intersection between BP types and process mining. We observe four mains required objectives: the management view, process model perspectives and the clustering technique. In this context, we match each challenge with a specific phase of our proposed approach. Consequently, our approach is applied on three phases: configuration, pre-processing and processing.

In this respect, the configuration phase declares BP type as event object, to define selected event data by BP type. Then, the pre-processing phase use a new filter, which aims to refine event data frame by BP type. Last, the processing phase treats event logs, using the correspondence between BP types and process model perspectives' priority, to represent process models according to BP types. This helps in acquiring insights on which order perspectives could be combined to the control-flow perspective.

As further research, we plan to develop a full plug-in that can be implemented into the Prom tool, for applying our proposed guided process discovery approach according to BP types and improvement metrics.

VI. ACKNOWLEDGEMENT

This work was supported by the National Center for Scientific and Technical Research (CNRST) in Rabat, Morocco.

VII. REFERENCES

Anon 2020, '3-Satisfiability data mining technique with hybrid clonal selection algorithm for autistic spectrum disorder screening', ASM Science Journal.

Baier, T, Mendling, J & Weske, M 2014, 'Bridging abstraction layers in process mining', Information Systems, vol. 46, pp. 123-139.

Baier, T 2015, 'Matching events and activities', PhD thesis, University Potsdam.

Baier, T, Di Ciccio, C, Mendling, J & Weske, M 2018, 'Matching events and activities by integrating behavioral aspects and label analysis', Software & Systems Modeling, vol. 17, no. 2, pp. 573-598.

Begicheva, AK & Lomazova, IA 2017, 'Discovering high-level process models from event logs', Моделирование и анализ информационных систем, Systems, vol. 24, no. 2, pp. 125-140.

Boubaker, S, Mammar, A, Graiet, M & Gaaloul, W 2016, 'A formal guidance approach for correct process configuration', in International Conference on Service-Oriented Computing.

Bülow, S, Backmann, M, Herzberg, N, Hille, T, Meyer, A, Ulm, B, Wong, TY & Weske, M 2013, 'Monitoring of business processes with complex event processing', in International Conference on Business Process Management.

Burattin, A 2015, 'Introduction to business processes, BPM, and BPM systems', in Process Mining Techniques in Business Environments, pp. 11-21.

Conforti, R, La Rosa, M & ter Hofstede, AH 2016, 'Filtering out infrequent behavior from business process event logs', IEEE Transactions on Knowledge and Data Engineering, vol. 29, no. 2, pp. 300-314.

Cook, DJ, Krishnan, NC & Rashidi, P 2013, 'Activity discovery and activity recognition: A new partnership, IEEE transactions on cybernetics, vol. 43, no. 3, pp. 820-828.

Cugola, G & Margara, A 2012, 'Processing flows of information: From data stream to complex event processing', ACM Computing Surveys (CSUR), vol. 44, no. 3, pp. 15.

De Souza Oliveira, M & Queiroz, S, 2020, 'Unsupervised Feature Selection Methodology for Clustering in High Dimensionality Datasets', Revista de Informática Teórica e Aplicada, vol. 27, no. 2, pp. 30-41.

Di Ciccio, C, Maggi, FM, Montali, M. & Mendling, J 2018, 'On the relevance of a business constraint to an event log', Information Systems, vol. 78, pp. 144-161.

Fazzinga, B, Flesca, S, Furfaro, F, Masciari, E & Pontieri, L 2015, 'A probabilistic unified framework for event abstraction and process detection from log data', in OTM Confederated International Conferences On the Move to Meaningful Internet Systems.

Ferreira, DR, Szimanski, F & Ralha, CG 2013, 'Mining the low-level behaviour of agents in high-level business processes, International Journal of Business Process Integration and Management', vol. 6, no. 2, pp. 146-166.

Ferreira, DR, Szimanski, F & Ralha, CG 2014, 'Improving process models by mining mappings of low-level events to high-level activities', Journal of Intelligent Information Systems, vol. 43, no. 2, pp. 379-407.

Folino, F, Guarascio, M & Pontieri, L, 2015, 'Mining multi-variant process models from low-level logs', in International Conference on Business Information Systems, pp. 165-177.

Hallé, S & Varvaressos, S, 2014, 'A formalization of complex event stream processing', in 2014 IEEE 18th International Enterprise Distributed Object Computing Conference.

George, L, Cadonna, B & Weidlich, M 2016, 'IL-miner: instance-level discovery of complex event patterns' in Proceedings of the VLDB Endowment.

Harmon, P 2019, 'Business process change: a business process management guide for managers and process professionals', Morgan Kaufmann.

Liu, Y, Nie, L, Liu, L & Rosenblum, DS 2016, 'From action to activity: sensor-based activity recognition', Neurocomputing, vol. 181, pp. 108-115.

Nel, P & Abdullah, MA 2017, 'Mediating effects of business success on the relation between business issues and business prospects amongst immigrant entrepreneurs in Malaysia', Contemporary Management Research, vol. 13, no. 1, pp. 3-18.

Oliveira, CAL, Silva, NC, Sabat, C L & Lima, RMF 2013, 'Reducing the gap between business and information systems through complex event processing', Computing and Informatics, vol. 32, no. 2, pp. 225-250.

Mannhardt, F 2018, Multi-perspective process mining. In CEUR Workshop Proceedings.

Prakash, S, Joshi, S, Bhatia, T, Sharma, S, Samadhiya, D, Shah, RR, Kaiwartya, O& Prasad, M 2020, 'Characteristic of enterprise collaboration system and its implementation issues in business management', International Journal of Business Intelligence and Data Mining, vol. 16, no. 1, pp. 49-65.

Senderovich, A, Rogge-Solti, A, Gal, A, Mendling, J & Mandelbaum, A 2016, 'The ROAD from sensor data to process instances via interaction mining', in International Conference on Advanced Information Systems Engineering.

Staab, S & Studer, R, Eds 2010, 'Handbook on ontologies' Springer Science & Business Media.

Suriadi, S, Andrews, R, ter Hofstede, AH & Wynn, MT 2017, 'Event log imperfection patterns for process mining: Towards a systematic approach to cleaning event logs', Information Systems, vol. 64, pp. 132-150.

Tax, N, Sidorova, N, Haakma, R & van der Aalst, WMP, 2016, 'Event abstraction for process mining using supervised learning techniques', in Proceedings of SAI Intelligent Systems Conference.

Tax, N, Sidorova, N, Haakma, R & van der Aalst, WMP 2017, 'Mining process model descriptions of daily life through event abstraction', Intelligent systems and applications, vol. 751.

Van der Aalst, W, 2016a, 'Data science in action: Process discovery', Springer, Berlin, Heidelberg.

Van der Aalst, W, 2016b, 'Data science in action: Mining additional perspectives, Springer, Berlin, Heidelberg.

Weidlich, M, Ziekow, H, Gal, A, Mendling, J & Weske, M 2014, 'Optimizing event pattern matching using business process models', IEEE Transactions on Knowledge and Data Engineering, vol. 26, no. 11, pp. 2759-2773.