

# Handling Sudden and Recurrent Changes in Business Process Variability: Change Mining based Approach

Asmae HMAMI<sup>1</sup>, Mounia FREDJ<sup>3</sup>  
AIQualsadi Research Team, ENSIAS, University  
Mohammed V of Rabat, Rabat, Morocco

Hanae SBAI<sup>2</sup>  
Faculty of Sciences and Technology, University Hassan II  
of Casablanca, Mohammedia, Morocco

**Abstract**—Changes are random and unavoidable actions in business processes, and they are frequently overlooked by managers, especially when managers need to deal with a collection of process variants. Because they must manage every single business process variant separately which is a time-consuming task. They exist many approaches to manage a collection of business process and deal with variability. Such as process mining approaches, that can discover configurable business process models, enhancing them and verify conformity automatically. However, those approaches do not cover changes and concept drift that occur over time. This paper presents a novel change mining approach that discovers changes in a collection of event logs and reports them on a change log. This change log can be analyzed to determine whether the changes are sudden or recurrent and recommend afterward some improvement to the configurable process model.

**Keywords**—Component; variability; process variant; configurable process model; process mining; change mining; concept drift

## I. INTRODUCTION

In the last few years, there has been a growing interest in managing changes regarding the actual situation of the world related to the pandemic. Due to the coronavirus situation, many organizations must make changes to their business strategies. Education centers, schools, and universities should make online courses. Medical industries must produce a new cure. Hospitals are obliged to add more resources to support the huge demand. Other companies were required to reduce the number of employees in each space to respect the social distancing... All those changes directly affect existing models that will no longer be adapted to new conditions. In the business process context, changes are a big challenge, when changes happen during the execution stage of the business process several new features could be added and some modification can be made to the elements of the business process, so the behavior of the process is going to follow the new features rather than the existing model[1]. So those changes will reduce the validity of the first proposed models for business processes; as a result, changes must be analyzed and consider, which is an important research area on business processes. Besides, changes were managed manually with patterns and tools like ADePt (The analytical design planning technique) [2]. In order to reduce human intervention, process mining approaches are introduced to add more intelligence and automation, for model construction, validation and enhancement of processes [3]. Moreover, change management in business processes is also improved by using process mining

techniques, which is known as “change mining”. The main idea is to use event logs or change logs associated with a business process and then discover changes observed in traces [4].

However, current research on change mining focuses on discovering changes from a single event log which is associated with a single business process [4][5][6]. But companies use several copies of the same process that are similar to each other, with some differences on specific points. Those points present the variation between processes [7]. Each process variant is a business process that fits most the need that has been previously expressed by managers of this specific business. All those process variants are grouped on one model so-called configurable process model [8].

Several research works have appeared in recent years presenting and documenting configurable process models. However, to the best of our knowledge, few research works are available in the literature that addresses change mining in business process variability.

In this paper, we present a novel approach to perform change mining in a collection of event logs. The collection of event logs represents in our case a data stream that we are going to analyze. Such analyze has already been carried out by machine learning algorithms. The proposed approach in this work is inspired by one of machine learning approaches in order to extract the changed fragments from the list of events.

The remainder of this paper is organized as follows. Section 2: is a background of the most important concept related to our paper. Section 3 is a related work section. Section 4 presents the proposed change mining approach. Section 5 is an implementation and test section. Finally, section 6 concludes the article.

## II. FOUNDATIONS

To perform a change mining in a collection of event logs related to a configurable process model, it is necessary to understand the configurable process model and the variability concept. Then how the changes could impact the model.

### A. Configurable Business Process Model

1) *Definition*: Companies use several copies of a business process such business process models are similar in different fragments with a slight difference in specific points. To represent and regroup all those models in one, the configurable process model concept was proposed [8].

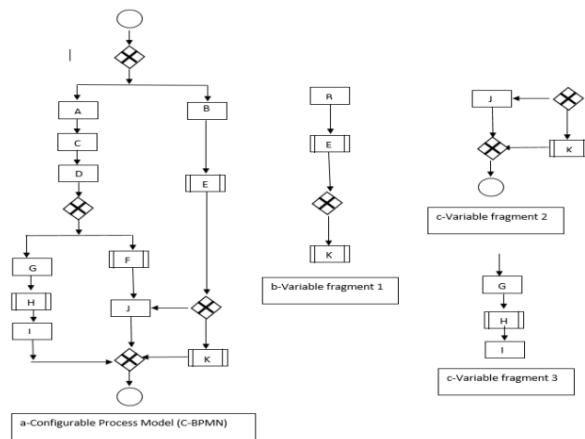


Fig. 1. Example of a Business Process Model and its Variable Fragments

Configurable process models aim to provide generic models integrating possible process variations into one model. Afterward such a model can be configured to a specific solution. This means a configurable model should guide the user to a solution that fits with the user's requirements. [8]

The configurable business process model can be represented by some specific modeling language such as C-EPC [7], C-BPMN [9] The modeling language must provide the needed options that can help to create and drive desired process variants from a configurable process model.

2) *Key elements*: The important concept of a configurable process model is "variability", which is related to two elements: Variation point and Variants.

- The variation points are locations likely to be different in each business process variant [8].
- The variants are the possible values that variation points can have in each process variant [8].

A variable fragment is a subset of a business process model that captures variability [10]. It contains at least one variation point.

As an example, part (a) of the Fig.1 presents a configurable process model and parts b) c) d) in the same figure are some variable fragments of this specific model.

When the configuration is made based on the model, different possibilities are available, and choices can be made from the list of variants of each variation point, to create the desired business process variant. According to the "hide and block" technique [9], from the list of variants each specific configuration choices to ON (use the variation) OFF (hide the variation) OPT (The choice depend on some condition) the variation, so we have to use none, one or many variants for each variation point [9].

3) *Creation of a configurable process model*: To create a configurable business process model there are two different methods based on whether it will be created from scratch or by using a process mining technique.

a) *Creation from scratch*: Configurable process models can be constructed in different ways. They can be designed

from scratch, but if a collection of existing process models already exists, a configurable process model can be derived by merging the different variants. [11]

Different approaches have been proposed in order to merge existing process models into a configurable process model. [12][13][14][15] the input of almost all those approaches is a collection of business process models of the same family and the output is a configurable process model.

b) *Creation by using process mining*: Another way of obtaining a configurable process model is not by merging process models but by applying process mining techniques on a collection of event logs.

The aim of process mining is to use the recorded data about the previous execution stored in a file called event logs [3] in order to discover new models, enhance business processes or verify conformity [3].

There are four approaches for discovering configurable process models. [16]

- Merging individually discovered process models: the configurable process model is discovered based on merging the discovered process variants which is made individually.
- Merging similar discovered process models from a common model: in this approach, the configurable business process is discovered based on merging discovered process variants which are made from the discovered common model.
- Discovering a single process model then discover configurations: in this approach, a common model is discovered from the collection of event logs and secondly, the configuration is mined from the common model in order to create a configurable process model.
- Discovering process model and configurations at the same time: in this approach, the configurable process model is created directly from the collection of event logs [16].

However, process mining techniques do not take into account changes that may occur during the life cycle of the process. Because business processes are not in steady state, so the configurable process model discovered from event logs, can no longer be adapted to the real situation.

In the next section, we define changes in variability. We start by defining changes and when they will happen, then how do they show up on the event log.

### B. Changes in Business Process Variability

Changes in variability can be classified into two categories predictable changes and unpredictable ones.

1) *Predictable Change*; (Reengineering, redesign, improvement)

Each configurable process model once created may be subject to many changes during its life cycle, due to different circumstances. Those changes can be related to a reengineering or a redesign or an improvement ... [17].

The change in the business process management field has various definitions. We selected below the most relevant ones:

- Change is the fundamental rethinking and radical redesign of business processes to achieve dramatic improvements in critical, contemporary measures of performance, such as cost, quality, service, and speed [17].
- business process change management is a strategy-driven organizational initiative to improve and (re)design business processes to achieve competitive advantage in performance (e.g., quality, responsiveness, cost, flexibility, satisfaction, shareholder value, and other critical process measures) through changes in the relationships between management, information, technology, organizational structure, and people[18].

As mentioned above, those changes are performed and happen during each phase of the life cycle of the business process model, in each phase a specific modification can be made, which is depicted into four phases:

- Phase (a) : process design, which is the first step and in this step, changes can be made as a redesign or reengineer,
- Phase (b): process configuration, in this phase process variant, is created from a configurable process model by choosing for each variation point one or many variants, in this phase, we can add implicitly a variant or a variation point depending on the new requirements,
- Phase (c): process enactment, the process variant is made into production and test in order to verify the compliance with the need, if minor adjustments are required, they will be made. So changes will be made on the model,
- Phase (d): a process diagnosis phase which leads to process adaptations, and in this phase, we can recommend a new model, to design new process models [19].

However, on the one hand, this type of change is not all reported in guidelines, in order to solve a problem quickly, many managers can perform changes without documenting the performed actions, which can lead to some confusion when working on the same business process model. On the other hand, predictable changes are not the only cause of changes. Concept drift can also change the behavior and the structure of a configurable process model.

## 2) Unpredictable change: Concept Drift

The configurable process model can during its life cycle, meet some unpredictable changes that are not made by managers but occur due to some actions made by other users or systems running. Those changes are named concept drift, this type of change will affect the initial concept (which is subject to change). There are four types.

- Sudden change: an unanticipated event that occurs or takes place unexpectedly,

- Recurring change: seasonal changes, that appear many times over time,
- Gradual change: this Change starts with a limited context and increase slowly to be finally applied to the entire stream,
- Incremental change: small different mutations happen to the concept many times until it becomes a new completely different concept [20].

In this paper, we are concerned with the first two types of changes.

## C. Definition of Variability Change in a Collection of Event Logs

1) *Definition*: The presented changes are almost all related to the execution of the business process and are related to the behavior of the business process and how it is executed. And as the execution is recorded on event logs, we will search changes from event logs which is the dynamic aspect of the business process.

In variability context, we are using not only one event log but a collection of event logs.

So, a changed event in a collection of event logs is an event that occur multiple times in the event log, and this event is different from all the possible events from all the possible process variants of the same family.

From this definition, an event (in the context of process variability) can be concerned as a changed one if and only if:

- The event is repeated in many traces of the same process variant (if not those events are concerned as errors).
- The event is not expected. Not only in the process variant where the change happens but also in the other process variants of the same family.

To illustrate variability changes and the importance of detecting those changes let us take look at the example in the next sub-section.

2) *Example of change in a collection of event logs*: From a configurable business process model, many process variants are driven and during their execution, events go through activities of the business process to compose a trace. From each complete execution, those traces are recorded on event logs. This event log is the input of process mining techniques. However, the recorded traces do not fit all the normal behavior that is described in the process model. If these unexpected behaviors are not detected and labeled, they will lead to errors when performing process mining.

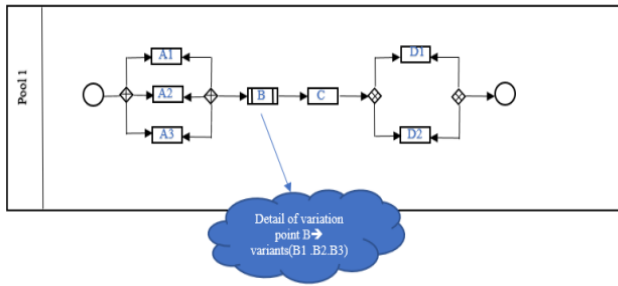


Fig. 2. Example of a Configurable Process Model.

The Fig.2 is an example of a configurable process model. This model can generate three process variants. The recorded traces can have the following variability changes.

- <A1B3C1D2D1> 1- Normal trace
- <A2ZC1D2D1> 2-Trace with change on the variation.
- <A3B3ND2D1> 3-Change on the variable fragment.
- <A1B3A1B3C1D2D1> 4- Change in the order of execution.

- Normal trace: is our reference and we consider it is normal because it follows the structure and the behavior of the predefined configurable process model.
- Trace with change in the variation: in this trace, the variant is different from the list of variants of this specific variation point.
- Change on the variable fragment (activity): the directed connected activities to the variants have changed.
- Change on the variable fragment (execution): the sequence flow has changed due to the change in the order of the execution of activities.

To highlight the importance of detecting changes before applying a process mining technique on the event log, we have applied a process mining algorithm on event logs that contained the list of changes presented in the example. The obtained result is presented in Tab.1, which presents the discovered business process model obtained after applying Alpha algorithm [21] exiting in prom as plugin [22].

As we can observe in Tab.1 the discovered model contains activities that have been added or removed due to changes, and models are no more the same as it is in the predefined one. We can easily recognize that without hiding those changes and identifying them, the process mining algorithm will lead to confusion.

TABLE I. EXAMPLE OF PROCESS MINING IN LOGS WITH CHANGES

Applied changes	Trace Example	Discovered model
No changes	<A1B3C1D2D1> <A2B3C1D1D2> <A3B3C1D2D1>	
Change in variants	<A1B3C1D2D1> <A2NBC1D1D2> <A3NBC1D2D1> <A1NBC1D2D1> <A2B3C1D2D1>	
Change in activities on the variable fragment	<A1B3C1D2D1> <A2B3NND1D2> <A3B3NND2D1> <A1B3C1D2D1> <A2B3C1D1D2> <A3B3NND2D1>	
Change in sequence of the variable fragment	<A1B3A1B3C1D2D1> <A2B3A2B3C1D1D2> <A3B3A3B3C1D2D1>	

### III. RELATED WORK

Managing changes has been widely discussed in the field of business process management, many works have proposed approaches to deal manually with changes such as AdePt [2].

In recent years change mining, which is an automatic approach to discover changes has emerged and those approaches were widely used to detect changes in business processes. In Previous work, a comparative study has been conducted [23]. This study shows that almost all selected papers in this comparative study had dealt with changes in a single business process [4][5][6]. And only a few papers have proposed approaches to deal with changes in a collection of business processes [24][25]. Those approaches are limited to propose some rules [24] related specially to the configuration and how process variants could be created from the configurable process model based on the observed behaviors [25]. However, they do not detect changes over time which are known as concept drift.

Recently work on concept drift in the business process have taken some importance and novel approaches have proposed [26][27][28] to detect concept drifts that occur during the execution phase of the business process. However, none of them have dealt with the variability concept.

So, our goal in this paper is to overcome the limitation of change mining approaches related to variability by proposing an approach that can detect changes in variability. The proposed approach is based on mining change in data recorded on event logs of a collection of process variants. Afterward, the detected changes will be reported in change log of variability that can be used in future work to recommend improvement to the configurable process model.

#### IV. PROPOSED CHANGE MINING APPROACH

As it is presented in our previous work, the proposed approach is based on many steps that are presented in the figure below Fig.3.

##### A. Steps of the Proposed Change Mining Approach

- Preparing event log of variable fragments: In this step, we prepare the event log of variable fragments using the merging and the filtering approach presented in our previous work [29]. This step's starting point is a variability specification file, a collection of event logs, and the output is the event log of variable fragments.
- Applying Change mining algorithm: this step is based on mining change from the event log of variable fragments and creating the change log. The detail of this algorithm is presented in this paper.
- Analyzing and recommending operations: in this phase, some metrics are used to identify the most significant changes to recommend as a future evolution of the configurable business process model.

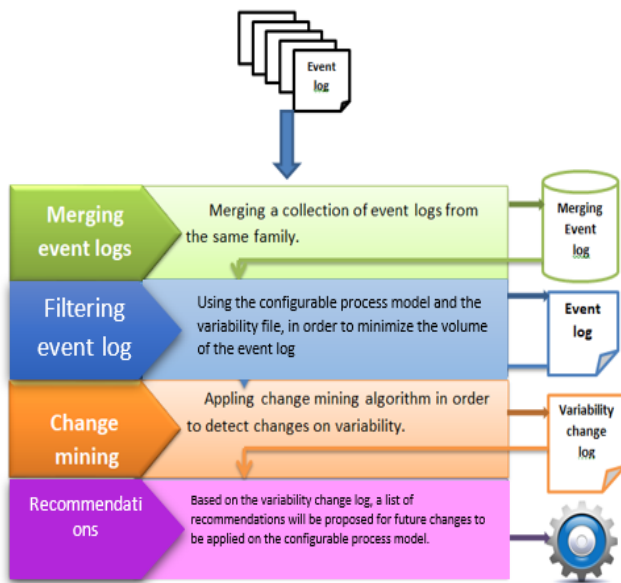


Fig. 3. Change Mining Framework.

##### B. Change Mining Algorithm

The proposed change mining algorithm is performed on the event log of variable fragments which is the output of merging and filtering algorithms proposed in the previous work [29]. So in this work, the input is an event log of variable fragments that contains events of variable fragments. The proposed algorithm is based on six steps: Classification, Initialization, Projection, Evaluation, Aggregation, Storage, Fig. 4.

Those steps are inspired by an existing algorithm to detect concept drift in a data stream called STAGGER. This algorithm is one of the first ones used in machine learning to overcome the problem of concept drift [30]. It is based on attributing three discrete attributes with three possible values each, for example

- $size \in \{small, medium, large\}$
- $color \in \{green, blue, red\}$
- $shape \in \{triangle, circle, rectangle\}$  [31].

In order to project the STAGGER algorithm description into our case, let first assume that an event log is an ensemble A that contains vectors T, each vector is a trace of the event log. Each trace is a collection of events grouping a certain number of activities. So  $A = \sum (T)$ .

The event log of variable fragments is ensemble E contains many events log of each process variant of A without the common elements. If B is an ensemble of the common element the ensemble E is  $E = \sum (A \cap B)$ .

And finally, the variable fragment is a vector F of three elements.

So, the three discrete attributes are process variant, fragments, and fragment's elements. In the STAGGER algorithm for each attribute, they are three values however in our case the number of possible values is a number greater than one.

- The number of possible values for process variants depends on the number of event logs in the collection.
- The number of values for fragments depends on the number of variation points in the configurable process model.
- Values for fragment's elements will be a vector of three components that present the previous, the variation and the next elements. The possible parameter for each value of the attribute fragment's elements are previous (start point, activity, or list of activity) a variation (activity) next (activity, list of activity or end point).

Thus, our discrete attribute will have the following values

- process variant  $\in \{\text{all possible process variants}\}$ ,
- fragments  $\in \{\text{all possible fragments}\}$ ,
- fragment's element  $\in \{\text{previous, variation, next}\}$ ,

And for example, a fragment will have the following syntax  $\langle PV1, F3, [A,M1,B] \rangle$ .

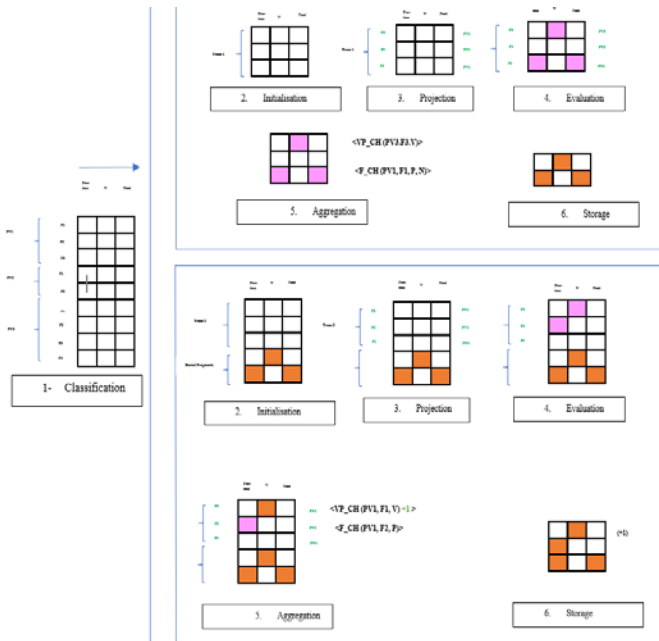


Fig. 4. Behavior of the Change Mining Algorithm.

Those discreet attributes with their values are the keys elements of each step of the proposed algorithm, in each step; we will have some tasks to complete as they are described on Fig.4, in order to get after many iterations through the event log of variable fragments, a change log.

Tasks in each step are as follows.

- **Classification:** labeling a list of fragments that will be used to select valid and invalid fragments when we loop on the event log of variable fragments. In this step, each attribute's values will be assigned, and a list of valid fragments will be created. Fig.5 presents the algorithm for this phase.
- **Initialization:** initialization of the fragment pool with fragments of the trace (i). These fragments will be formatted in the form of the chosen attribute Fig .6.
- **Projection:** projection of the selected trace's fragments on the list of valid fragments created on the classification step. This projection finds the most likely fragments to the selected fragment. Fig .6.
- **Evaluation:** in this step, if the projection detects changes in one of trace's fragments, we will identify the specific element concerned by the change. If change is in previous or next elements, we have a position change, which is fragment change. If the change is on the variation, it is a change in variation points. Fig .6.
- **Aggregation:** names the change by its appropriate target (Change on fragments, change on variants, or change on variation point) and gives a count number to the target name to put it on a specific change type group. Fig .6.
- **Storage:** deletes fragments that did not meet a change from the pool and store the changed one on an XML file. Fig .6.

```

Step 1: classification
1-while (variability_file) do
2-For (i=1; i<=nombre_previous; i++)
3-    For (j=1; j<= nombre_variants; j++)
4-        For (k=1 ; k<= nombre_next; k++)
5-            Var_new= Previous[i]. variants[j].next[k]
6-            Store (Var_new)
7-        End for
8-    End for
9-End for
10-End while
    
```

Fig. 5. Classification Algorithm.

```

Steps 2, 3, 4, 5, 6:
1. While (tarce_event log of variable fragments) do
2. //Step 2: initialization
3. Array M_trace //while M_trace is a matrix variable fragments
4. Array F //while F is a vector for each fragment.
5. For (each fragment in trace)
6.     F[process_variant]=trace_id;
7.     F[fragments]=fragment_id;
8.     F[Fragment_elements] = [fragment[0], fragment[1], fragment[2]] ;
9.     M_tarce[]=F ;
10. End for
11. //Step 3: Projection
12. For(F[i] of the new trace of M_trace)
13.     If(F[i] is not stored_fragments)
14.         F[change][i]="change";
15. //Step 4: Evaluation
16.     F_near=Equidistance(stored_fragments,F[i]);
17.     Variant=Compare_diff_variant(F_near,F[i]);
18.     Previous=Compare_diff_previous(F_near,F[i]);
19.     Next=Compare_diff_next(F_near,F[i]);
20. //Step 5: Aggregation
21.     F[targed][i]=[previous, variant,next];
22.     End if
23. End for
24. //Step 6: Storage
25. For(F[i] of the new trace of the M_tarce)
26.     If(F[change][i]=="null")
27.         Discard(F[i])
28.     Else
29.         Store(F[targed][i])
30.     End for
31. End while
    
```

Fig. 6. Initialization, Projection, Evaluation, Aggregation, Storage Algorithms.

The first algorithm in Fig.5 is for the initialization step. The second sub-code in the Fig .6 is for steps that we will loop on through the event log of variable fragments

At the end of the proposed algorithm, we will have a change log as an output. This change log is formatted as XML format and contains a chronologically sorted list of detected changes. Fig .7 is an example of the obtained change log.

```

<?xml version="1.0" encoding="UTF-8" ?>
<change>
  <id>23</id>
  <date>2019-01-01T16:03:01.012345</date>
  <tarce_id>56</tarce_id>
  <id_pv>2</heading>
  <id_variation_point>5</id_variation_point>
  <type>next_fragment_change</type>
  <change_true>Activity_B</change_true>
  <change_old>Activity_W</change_old>
</change>
<change>
  <id>23</id>
  <date>2019-01-01T16:04:08.0325468</date>
  <tarce_id>20</tarce_id>
  <id_pv>3</heading>
  <id_variation_point>4</id_variation_point>
  <type>variant_change</type>
  <change_true>Activity_M</change_true>
  <change_old>Activity_Z</change_old>
</change>
    
```

Fig. 7. Sub-Part of a Change Log obtained with the Proposed Algorithm.

The generated change logs should be detailed enough and accurate enough, to provide the information required for performing a future analyzes.

It must answer the following question

- 1) When did the change happens?
- 2) Which trace is concerned by this change?
- 3) Which business process variant is concerned by this change?
- 4) Which variability is concerned?
- 5) It is a fragment change or variants change?
- 6) Which is the name of the concerned element by this change?
- 7) What is the new element in this change?

In order to answer this list of questions each detected change is stored in the xml file with the following attribute:

- 1) Date of change.
- 2) Id of trace.
- 3) Id of process variant
- 4) Variation point
- 5) Type of change (Fragments or variants).
- 6) Name of the changed element
- 7) Name of the new value due to the change.

## V. IMPLEMENTATION, PROTOTYPE, AND TESTS

### A. Implementation and Prototype

The proposed approach is implemented as a new function on the toolset, the “random configurable process model generator” [32].

This toolset is a set of functions that provide the ability to generate random business process models with their process variants and simulate their execution and get event logs. Implementing the change mining algorithm in this tool will facilitate to test the implemented algorithm because all the required inputs are available in the same tool.

The algorithm takes as input only the event log of variable fragments. Because we implemented the algorithm in the same environment where previous algorithms have already been implemented, which are filtering and merging ones.

However, it is possible to run all three functions as one, if we have all the required input which are the variability specification file and a collection of event logs from the interface shows in Fig.8.

Our toolset gives to the user the ability to choose between having an event log with or without changes. The user can also choose the type of change to apply from the interface presented in Fig. 9. This will help us to test our change mining algorithm on event logs with different types of changes.

To test the proposed algorithm, we use a running example of collections of event logs based on three business process models. The three models are generated randomly using the toolset.

- Model 1: contains two variation points. The first variation point has 3 variants and the second has 4 variants Fig.10.
- Model 2: contains four variation points. The first variation point has 3 variants and the second has four variants the third has three variants and the fourth has four variants Fig.11.
- Model 3: contains six variation points. The first variation point has 3 variants and the second has four variants, the third has three variants, the fourth has four variants the fifth has three variants and finally the sixth has four variants Fig.12.

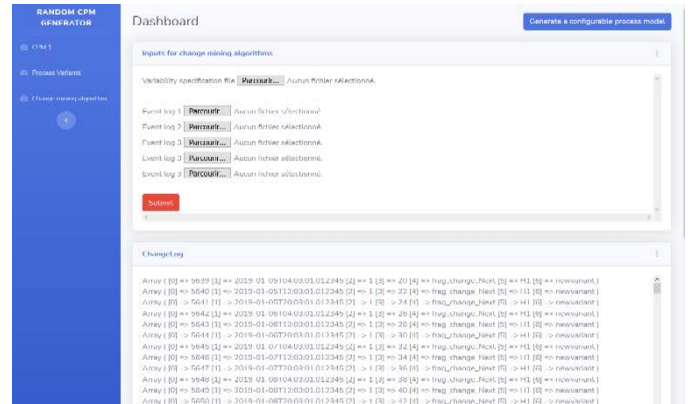


Fig. 8. Interface for Collecting Inputs and Performing a Change Mining.

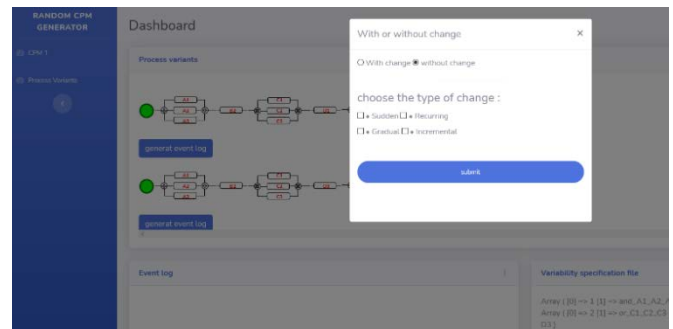


Fig. 9. Interface for Choosing the Type of Change to Add Randomly to the Event Log.

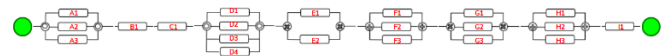


Fig. 10. Configurable Process Model 1.



Fig. 11. Configurable Process Model 2.

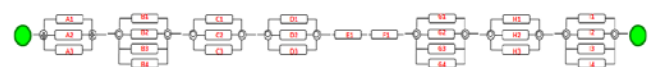


Fig. 12. Configurable Process Model 3.

From the three generated business process model, we will generate a collection of process variants each collection. The toolset will simulate the execution of those process variants and generate their event logs. In addition, we will apply randomly on the obtained event logs a different type of drift to get an event logs with changes.

### B. Tests and Results

The three collections of event logs are shown below.

- Collection 1: contains three event logs of three process variants of a configurable business process model 1. Fig.10. Each event log contains 100 traces.
- Collection 2: contains six event logs of six process variants of a configurable business process model 2 Fig.11. Each event log contains 100 traces.
- Collection 3: contains twelve event logs of twelve process variants of a configurable business process model 3 Fig.12. Each event log contains 100 traces.

When generating event logs, we will apply random changes to each event log.

For each collection, we perform a change mining in order to detect the applied changed on the event log and generate the change log. First, we apply the merging and filtering algorithms to get the event log of variable fragments. Fig.13, part (a) is a subpart of the event log of variable fragment and we highlight changed fragment with a green shape. Second, we apply the change mining algorithm. Finally, we export the results of the mining as an XML file, which is the change log. An example of the obtained change log is presented in Fig 13 part (b).

We made the same actions on the three collections and in each collection, we were able to generate change log and detect all most all applied changes.

```
5643,2019-01-06T15:03:01.012345,Cl,3,387
5643,2019-01-06T16:03:01.012345,newvariant,3,387
5643,2019-01-06T17:03:01.012345,El,3,387
5644,2019-01-06T22:03:01.012345,A3,3,388
5644,2019-01-06T23:03:01.012345,B1,3,388
5644,2019-01-07T00:03:01.012345,Cl,3,389
5644,2019-01-07T01:03:01.012345,newvariant,3,389
5644,2019-01-07T02:03:01.012345,E2,3,389
5645,2019-01-07T07:03:01.012345,A3,3,390
5645,2019-01-07T08:03:01.012345,B1,3,390
5645,2019-01-07T09:03:01.012345,Cl,3,391
5645,2019-01-07T10:03:01.012345,newvariant,3,391
5645,2019-01-07T11:03:01.012345,El,3,391
5646,2019-01-07T16:03:01.012345,A3,3,392
5646,2019-01-07T17:03:01.012345,B1,3,392
5646,2019-01-07T18:03:01.012345,Cl,3,393
5646,2019-01-07T19:03:01.012345,newvariant,3,393
5646,2019-01-07T20:03:01.012345,El,3,393
5647,2019-01-08T01:03:01.012345,A3,3,394
5647,2019-01-08T02:03:01.012345,B1,3,394
5647,2019-01-08T03:03:01.012345,Cl,3,395
5647,2019-01-08T04:03:01.012345,newvariant,3,395
5647,2019-01-08T05:03:01.012345,El,3,395
5648,2019-01-08T10:03:01.012345,A3,3,396
5648,2019-01-08T11:03:01.012345,B1,3,396
5648,2019-01-08T12:03:01.012345,Cl,3,397
5648,2019-01-08T13:03:01.012345,newvariant,3,397
5648,2019-01-08T14:03:01.012345,El,3,397
```

```
<?xml version="1.0" encoding="UTF-8"?>
<change>
  <id>1</id>
  <date>2019-01-05T10:03:01.012345</date>
  <trace>5639</trace>
  <pv>2</pv>
  <fragment_id>179</fragment_id>
  <change_type>change_variations</change_type>
  <old_element>D1</old_element>
  <new_element>newvariant</new_element>
</change>
<change>
  <id>2</id>
  <date>2019-01-05T19:03:01.012345</date>
  <trace>5640</trace>
  <pv>2</pv>
  <fragment_id>181</fragment_id>
  <change_type>change_variations</change_type>
  <old_element>D1</old_element>
  <new_element>newvariant</new_element>
</change>
<change>
  <id>3</id>
  <date>2019-01-06T04:03:01.012345</date>
  <trace>5641</trace>
  <pv>2</pv>
  <fragment_id>183</fragment_id>
  <change_type>change_variations</change_type>
  <old_element>D1</old_element>
  <new_element>newvariant</new_element>
</change>
</change>
```

(a) The input: event log of variable fragments. (b) The output: change log.

Fig. 13. Input and Output of the Change Mining Algorithm.

## VI. CONCLUSIONS

This paper presents a novel approach to perform a change mining in a collection of event logs based on a modified STAGGER algorithm.

Our approach is based on detecting sudden and recurrent change by using steps of STAGGER algorithm and storing detected changes in an XML file so-called change log.

The proposed approach is implemented on the toolset “random configurable process model generator”, and it shows its ability to detected drift on synthetic event logs.

In this work, we are concerned only by sudden and recurrent changes. However, more improvement can be applied to detect the other types of changes.

We also aim to test our proposed approach on a real collection of event logs and add more change mining perspectives especially data and resources.

As future work, we aim to create from the generated change log, a recommendation system that proposes a new configurable process model based on the detected changes. Also, as perspective, we intend to make our approach suitable with the situation where the configurable process model is not discovered.

## REFERENCES

- [1] Song, W., & Jacobsen, H. A. (2016). Static and dynamic process change. *IEEE Transactions on Services Computing*, 11(1), 215-231.
- [2] Austin, S., Baldwin, A., Li, B., & Waskett, P. (2000). Analytical design planning technique (ADePT): a dependency structure matrix tool to schedule the building design process. *Construction Management & Economics*, 18(2), 173-182.
- [3] Van Der Aalst, W., Adriansyah, A., De Medeiros, A. K. A., Arcieri, F., Baier, T., Blickle, T., ... & Wynn, M. (2011, August). Process mining manifesto. In *International Conference on Business Process Management* (pp. 169-194). Springer, Berlin, Heidelberg.
- [4] Hompes, B., Buijs, J. C., van der Aalst, W. M., Dixit, P. M., & Buurman, H. (2015). Detecting Change in Processes Using Comparative Trace Clustering. *SIMPDA*, 2015, 95-108.
- [5] Kaes, G., & Rinderle-Ma, S. (2017, June). On the similarity of process change operations. In *International Conference on Advanced Information Systems Engineering* (pp. 348-363). Springer, Cham.
- [6] van der Aalst, W. M. (2015, August). Change Point Detection and Dealing with Gradual and Multi-order Dynamics in Process Mining. In *Perspectives in Business Informatics Research: 14th International Conference, BIR 2015, Tartu, Estonia, August 26-28, 2015, Proceedings* (Vol. 229, p. 161). Springer.
- [7] van der Aalst, W. M., Dreiling, A., Gottschalk, F., Rosemann, M., & Jansen-Vullers, M. H. (2005, September). Configurable process models as a basis for reference modeling. In *International Conference on Business Process Management* (pp. 512-518). Springer, Berlin, Heidelberg.
- [8] Gottschalk, F., Van der Aalst, W. M., & Jansen-Vullers, M. H. (2007). Configurable process models—a foundational approach. In *Reference Modeling* (pp. 59-77). Physica-Verlag HD.
- [9] Zhang, H., Han, W., & Ouyang, C. (2014, September). Extending BPMN for Configurable Process Modeling. In *INSEPE CE* (pp. 317-330).
- [10] Mancioffi, M., Danylevych, O., Karastoyanova, D., & Leymann, F. (2011, August). Towards classification criteria for process fragmentation techniques. In *International Conference on Business Process Management* (pp. 1-12). Springer, Berlin, Heidelberg.
- [11] Buijs, J. C., van Dongen, B. F., & van der Aalst, W. M. (2013). Mining configurable process models from collections of event logs. In *Business process management* (pp. 33-48). Springer, Berlin, Heidelberg.
- [12] Gottschalk, F., van der Aalst, W.M.P., Jansen-Vullers, M.H.: Merging Event-driven Process Chains. In: Meersman, R., Tari, Z. (eds.) OTM 2008, Part I. LNCS, vol. 5331, pp. 418–426. Springer, Heidelberg (2008)
- [13] La Rosa, M., Dumas, M., Uba, R., Dijkman, R.: Business Process Model Merging: An Ap-proach to Business Process Consolidation. *ACM Transactions on Software Engineering and Methodology* 22(2) (2012)



- [14] Schunselaar, D.M.M., Verbeek, E., van der Aalst, W.M.P., Raijers, H.A.: Creating Sound and Reversible Configurable Process Models Using CoSeNets. In: Abramowicz, W., Kriksciuniene, D., Sakalauskas, V. (eds.) BIS 2012. LNBIP, vol. 117, pp. 24–35. Springer, Heidelberg(2012)
- [15] Gottschalk, F., van der Aalst, W.M.P., Jansen-Vullers, M.H.: Mining Reference Process Models and their Configurations. In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM-WS 2008.LNCS, vol. 5333, pp. 263–272. Springer, Heidelberg (2008);
- [16] Buijs, J.C.A.M., La Rosa, M., Reijers, H.A., Dongen, B.F., van der Aalst, W.M.P.: Improving Business Process Models using Observed Behavior. In: Proceedings of the Second International Symposium on Data-Driven Process Discovery and Analysis. LNBIP, Springer (toappear, 2013).
- [17] Kettinger, W. J., & Grover, V. (1995). Toward a theory of business process change management. *Journal of Management Information Systems*, 12(1), 9-30.].
- [18] Kettinger, W.J.; Guha, S.; and Teng, J.T. The process reengineering life cycle methodology: a case study. In V. Grover and W.J. Kettinger (eds.), *Business Process Change: Reengineering Concepts, Methods and TechnolOgies*. Hanisburg, PA: Idea Group Publishing, 1995
- [19] Gottschalk, F. (2009). Configurable process models.
- [20] Kadam, S. (2019). A Survey on Classification of Concept Drift with Stream Data.
- [21] W. van der Aalst, T. Weijters, and L. Maruster, "Workflow mining:Discovering process models from event logs,"*IEEE Transactions onKnowledge and Data Engineering*, vol. 16, no. 9, pp. 1128–1142, 2004.
- [22] Van der Aalst, W. M., van Dongen, B. F., Günther, C. W., Rozinat, A., Verbeek, E., & Weijters, T. (2009). ProM: The process mining toolkit. *BPM (Demos)*, 489(31), 2.
- [23] Hmami, A., Sbai, H., & Fredj, M. (2019, April). Change Mining in Business Process Variability: A Comparative Study. In 2019 4th World Conference on Complex Systems (WCCS) (pp. 1-5). IEEE.
- [24] Assy, N., & Gaaloul, W. (2014, November). Configuration rule mining for variability analysis in configurable process models. In *International Conference on Service-Oriented Computing* (pp. 1-15). Springer, Berlin, Heidelberg.
- [25] Arriagada-Benítez, M., Sepúlveda, M., Munoz-Gama, J., & Buijs, J. C. (2017). Strategies to automatically derive a process model from a configurable process model based on event data. *Applied sciences*, 7(10), 1023.
- [26] Baier, L., Reimold, J., & Kühl, N. (2020, June). Handling Concept Drift for Predictions in Business Process Mining. In *2020 IEEE 22nd Conference on Business Informatics (CBI)* (Vol. 1, pp. 76-83). IEEE.
- [27] Bose, R. J. C., van der Aalst, W. M., Žliobaitė, I., & Pechenizkiy, M. (2011, June). Handling concept drift in process mining. In *International Conference on Advanced Information Systems Engineering* (pp. 391-405). Springer, Berlin, Heidelberg.
- [28] Yeshchenko, A., Di Ciccio, C., Mendling, J., & Polyvyanyy, A. (2021). Visual Drift Detection for Sequence Data Analysis of Business Processes. *IEEE Transactions on Visualization and Computer Graphics*.
- [29] Hmami, A., Sbai, H., & Fredj, M. (2020, March). A new Framework to improve Change Mining in Configurable Process. In *Proceedings of the 3rd International Conference on Networking, Information Systems & Security* (pp. 1-6).
- [30] Ceravolo, P., Tavares, G. M., Junior, S. B., & Damiani, E. (2020). Evaluation goals for online process mining: a concept drift perspective. *IEEE Transactions on Services Computing*.
- [31] Schlimmer, J. C., & Granger, R. H. (1986, August). Beyond Incremental Processing: Tracking Concept Drift. In *AAAI* (pp. 502-507).
- [32] Hmami, A., Sbai, H., & Fredj, M. (2020, March). Enhancing change mining from a collection of event logs: Merging and Filtering approaches; *Journal of Physics: Conference Series*, Volume 1743, 2021 - IOPscience.