

# Extended Inductive Miner to Discover Semantic Annotated Process Model: A Variability-Based Approach

**Aicha Khannat\*** 

Department of AlQualsadi Research Team ENSIAS, Mohammed V  
University of Rabat (UM5), Rabat, Morocco  
E-mail: aicha.khannat@um5s.net.ma

**Hanae Sbai** 

Department of Mathematics, Computer Science and Applications  
Laboratory, Hassan II University of Casablanca (UH2C), Mohammedia,  
Morocco  
E-mail: sbai.hanae87@gmail.com

**Laila Kjiri** 

Department of AlQualsadi Research Team ENSIAS, Mohammed V  
University of Rabat (UM5), Rabat, Morocco  
E-mail: lkjiri80@gmail.com

## ABSTRACT

In recent years, the concept of a configurable process model has made many contributions, with the need for generic and reusable models. The high demand for this type of generic model is accompanied by quality requirements, as these models must be as comprehensive as possible to facilitate the task of customization. Many approaches exist to deal with configurable process models, especially in the field of process mining, including its three types of techniques: discovery, conformance, and enhancement. However, there is a lack of semantic representation in the resulting models. In this study, we propose a novel automated approach based on the extension of the Inductive Miner algorithm used to discover business process models. This method discovers a semantically annotated configurable process model using two ontologies: variability ontology and domain ontology-related concepts, which will improve and facilitate the configurable process model customization.

**Keywords:** configurable process model, process mining, semantic, inductive miner, ontologies, variability

**Received:** March 24, 2024  
**Accepted:** June 4, 2024

**Revised:** June 1, 2024  
**Published:** December 30, 2024

\***Corresponding Author:** Aicha Khannat  
 <https://orcid.org/0009-0008-0223-9142>  
**E-mail:** aicha.khannat@um5s.net.ma



All JISTaP content is Open Access, meaning it is accessible online to everyone, without fee and authors' permission. Open Access articles are automatically archived in the Korea Institute of Science and Technology Information (KISTI)'s Open Access repository (AccessON). All JISTaP content is published and distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>). Under this license, the authors retain full ownership of their work, while permitting anyone to use, distribute, and reproduce the content in any medium, as long as the original authors and source are cited. For any reuse, redistribution, or reproduction of a work, users must clarify the license terms under which the work was produced.

## 1. INTRODUCTION

Nowadays, companies try to make their business processes easy to understand and model while growing and opening to the market, so that the process management activity will be done easily and smoothly. Configurable process models (CPM) are useful for large structures that run several activities (Ayora et al., 2013). These models offer the option to regroup multiple variants related to the same business process in one model, which can be customized according to the execution environment properties (Pereira Detro et al., 2020). However, configurable process models present some limits and difficulties such as complexity, redundancy, and misunderstanding (Assy et al., 2014).

Process mining techniques have been very helpful in discovering business process models and configurable process models that represent the real execution of processes (Buijs et al., 2013). Process mining is divided into three types of techniques: discovery, conformance checking, and enhancement. Process model discovery explores event logs to create a clear process model based on the order of activities present in event logs (Ozates et al., 2024). Conformance checking uses the event log to detect any deviation during the process execution. Enhancement aims to improve the process model based on the data present in the event log (Ozates et al., 2024).

In spite of the number of contributions in the process mining field, most of the works have been focusing primarily on the syntactic analysis of process execution traces, which generate multiple issues related to the size and complexity level of the model (Barbieri et al., 2023). And, usually, only business analysts and domain experts can handle analysis regarding the resulting process model, due to the absence of domain knowledge in the process model (Berti et al., 2024). To remedy these limitations, semantics were introduced as a support tool to improve process mining results (Barbieri et al., 2023; Pereira Detro et al., 2020; Rebmann & van der Aa, 2022). For configurable process models, the lack of semantics complicates the task of configuration and impacts on the quality of process model variants (El Faquih et al., 2015; Fang et al., 2023).

As part of this work, we present a new approach to mine configurable process models that are enriched semantically with variability concepts and domain knowledge, starting with a collection of event logs. We aim to capitalize on the semantic analysis of event logs to discover a configurable process model that integrates semantics.

For this purpose, we will use two ontologies, an ontol-

ogy related to variability concepts and an ontology related to domain knowledge. The main idea is to improve comprehension and quality of the process model discovered. The configurable process model enriched semantically will be simpler to configure. Instead of discovering the model and then checking it against ontologies to validate it, ontologies will be used during the discovery phase to provide additional information, related to variability and process domain, on the process model discovered. This will reduce the cost of conformance and enhancement of these models.

The remainder of this paper is organized as follows. In Section 2, we include a background of the most important concepts. Section 3 gives an overview of the existing work related to this research area. Section 4 describes our objectives and contribution. In Section 5, we present a global overview of our approach. Section 6 provides details about the discovery component. Finally, in Section 7 we conclude the paper with an outlook for perspectives.

## 2. BACKGROUND

### 2.1. Configurable Process Models

Configurable process models regroup multiple execution cases of the same business process into one single process model. This type of model presents multiple advantages such as reusability, flexibility, and the ability to fit individual needs without performing any process modeling activities (Fang et al., 2023). Configurable process models have been used in different application areas, especially healthcare (Pereira Detro et al., 2020), industry (La Rosa et al., 2011), and e-learning (Azouzi et al., 2017).

Otherwise, configurable process models can be adapted to the execution context through the configuration decisions made at design time (Assy et al., 2014). The configuration allows users to disable all unnecessary process parts in order to derive the appropriate model (Gottschalk et al., 2009).

Configurable process models are characterized by the support of variability via two key elements: variation points and variants (El Faquih & Fredj, 2017). The variation points represent the decision points that link a fragment of common parts along with a fragment of variable parts in the model, and the variants are the multiple alternatives relating to the variation point. The choice of variants is made based on the rules related to the execution context and variation point. During the configuration process, variant activities can be set as ON (Included), OFF (Excluded), or OPT (Optionally included).

A variety of languages integrate variability in order to represent configurable process models: configurable event-driven process chains (C-EPC) (Rosemann & van der Aalst, 2007), configurable integrated EPC (C-iEPC) (La Rosa et al., 2007), configurable BPMN (C-BPMN) (El Faquih & Fredj, 2017), variant rich BPMN (vrBPMN) (Schnieders & Puhmann, 2007), and configurable yet another workflow language (C-YAWL) (La Rosa & Gottschalk, 2009). Fig. 1 (La Rosa, 2009) represents an example of the configurable process model and its variants.

### 2.2. Discovery of Configurable Process Models

Configurable process models can be built in two different ways:

- Approaches based on merging variants: The configurable process model is built by merging a collection of existing process models, and the configuration of the resulting model corresponds to the original models (Buijs et al., 2013). Two approaches are included in this category; the difference between them is how to get the variants. Approach (1) discovers the variants individually and then merges them into one consolidated configurable process model. Method (2) merges the collection of event logs, then applies process mining techniques to obtain the common process model, which does not integrate variability,

then works on the customization of this model, and after that merges the process variants to get the final configurable process model.

- Direct mining approaches: The discovery of a configurable process model is done automatically by applying process mining techniques (Buijs et al., 2013). There are two approaches: Approach (1) merges the collection of event logs and applies process mining techniques to discover the CPM that will be configured to obtain process variants. Approach (2) discovers the CPM and its configuration in one step. These approaches are not yet implemented concretely.

Several algorithms are present in the field of process model discovery: Alpha Miner (van der Aalst et al., 2004), Genetic Miner (van der Aalst et al., 2005), Heuristic Miner (Weijters et al., 2006), Fuzzy Miner (Günther & van der Aalst, 2007), and Inductive Miner (IM) (Leemans et al., 2013). Each algorithm has its strengths and weaknesses. Table 1 demonstrates a comparison between these algorithms. Following this comparison, IM is the most suitable algorithm to deal with any potential noise or incompleteness in the event logs and can replay the event logs. It constructs a process tree using the method of splitting the event log recursively into sub-event logs (blocs) and determining the relation between blocs until a base case is found (Leemans et al., 2013). There are four types

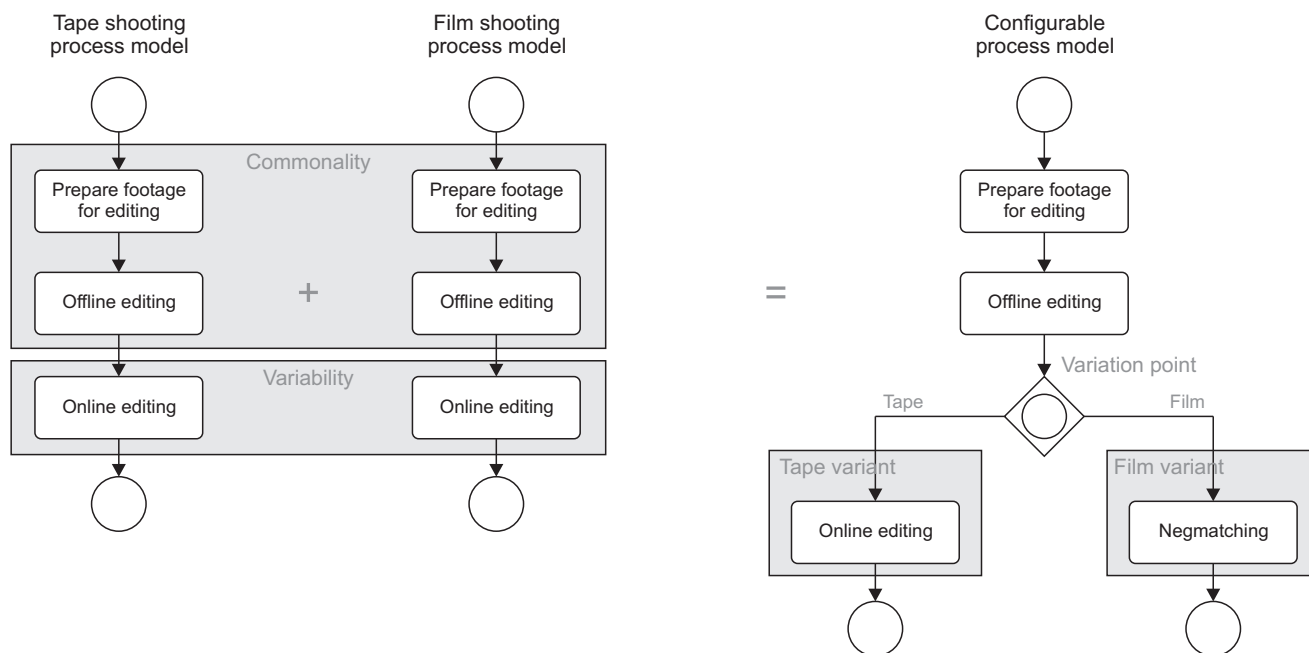


Fig. 1. Example of the configurable process model. Adapted from La Rosa (2009).

of splits in the event log: Sequential, Parallel, Concurrent, and Loop.

The configurable process models constructed using the different approaches need material support or a human background during the configuration stage. This was considered a weakness of process models discovered, and to overcome it, many works propose integrating semantic technologies.

### 2.3. Semantics in Configurable Process Models

Configurable process models are designed to be changed and customized depending on the execution conditions of the business environment. However, their ability to represent multiple business process cases impacts other properties, and engenders several challenges due to the size of the model, its complexity, and the time plus resources needed to evolve and keep up this model (Detro et al., 2017; El Faquih et al., 2015).

Present research proves that the integration of semantic features into process models can improve model quality and solve multiple misunderstanding issues. Accordingly, the use of semantics in process modeling becomes a necessity, as they bring more clarity and ease during process model manipulation.

The integration of semantics in the configurable process model is inspired from their integration in the busi-

ness process model. It corresponds to the combination between business process and semantic web technologies. A semantic process model is made up of a range of activities, including additional information related to the functional, behavioral, organizational, operational, and non-functional aspects of these activities. These aspects are described in a formal representation that is machine-readable and machine understandable. The concept of ontologies serves to illustrate these data in process models (Lautenbacher et al., 2009).

An ontology represents a precise and formal specification of a shared conceptualization related to a specific subject area (Gruber, 1993). It defines concepts of a domain and the relationships between them.

Fig. 2 (La Rosa, 2009) illustrates an example of the BPMN configurable process model, annotated semantically using BPM ontology concepts.

The next section is dedicated to the presentation and discussion of the existing works in relation to our study.

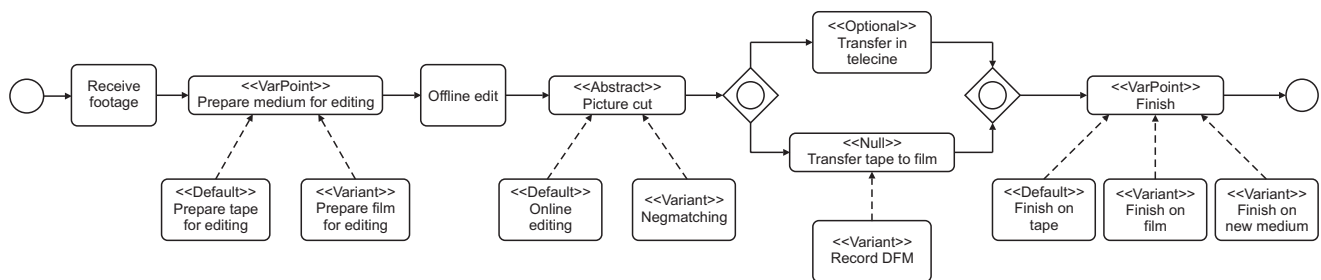
### 3. RELATED WORKS

Over the last few years, the implementation of business process models has been studied, through both theoretical and practical methods. Thus, various studies have identified the business process weakness in dealing with different execution environment conditions. For that reason, configurable process models were introduced with one principal advantage of integrating all the behaviors of all process variants into one model (Gottschalk et al., 2007). This type provides the possibility of extracting the variant models through two techniques, extension and restriction (La Rosa et al., 2017).

In parallel with this, a study by Hepp et al. (2005) showed that business processes need to adopt a machine-readable representation and proposed to combine business process management and semantic web technology under one consolidated technology, semantic business

**Table 1.** Comparison of process mining algorithms

Process mining algorithm	Ability to deal with noisy logs	Ability to deal with incomplete logs	Ability to replay the log
Alpha Algorithm	No	No	No
Heuristic Miner	Yes	Partially	No
Fuzzy Miner	Yes	Yes	No
Genetic Miner	Yes	Yes	No
Inductive Miner	Yes	Yes	Yes



**Fig. 2.** Example of the semantic configurable process model. Adapted from La Rosa (2009). DFM, digital film master.

**Table 2.** Comparison study of process mining applications

Work	Process mining algorithm	Input	Type of the resulting model	Is the approach automatic or manual?	Does the resulting model support variability?	Does the resulting model support semantics?	Type of ontology if used	The objective behind using semantics
Pereira Detro et al. (2020)	Heuristic Miner Decision Mining	Event log	Process variants	Automatic	No	Yes	Domain ontology Business ontology	Providing suggestions during customization
Nykänen et al. (2015)	Heuristic Miner	Event log	Process model	Manual	No	Yes	Domain ontology	Analysing the process models in different abstraction levels
Cairns et al. (2014)	Heuristic Miner Semantic LTL checker	Semantically annotated event log	Process model	Semi-automatic	No	Yes	Domain ontology	Discovering a less complex process model Process analysis at a conceptual level
Fang et al. (2023)	A new algorithm proposed based on trace clusters	Semantically annotated event log	Context tree corresponding to the process variant	Automatic	No	Yes	-	Formalization of the behavior semantic of event log Discovering new process variants that cannot be discovered using existing methods
Rebmann and van der Aa (2022)	A method based on the direct causal relations between events	Semantically annotated event log	Directly follows graph corresponding to the event log	Automatic	No	Yes	-	Enabling new analysis opportunities that consider the meaning of events
Khodayari Tehaninejad and Khoshnevis (2022)	Alpha Algorithm	Collection of event logs	Configurable process model	Automatic	Yes	No	-	-
Ramos-Gutiérrez et al. (2021)	Inductive Miner	Configuration log	Configuration workflow	Automatic	Yes	No	-	-
Assy et al. (2014)	Alpha Algorithm	Collection of event logs	Configurable process fragment	Automatic	Yes	No	-	-

LTL, linear temporal logic.

process management, to provide a uniformed representation of business processes on a semantic level.

This evolution has a major impact on process mining results, since the first applications of process mining techniques were based on syntactic analysis, and the models discovered were too complicated and large to be maintained. Many studies have used semantics in different ways to improve the process mining results. Table 2 (Assy et al., 2014; Cairns et al., 2014; Fang et al., 2023; Khodayari Tehraninejad & Khoshnevis, 2022; Nykänen et al., 2015; Pereira Detro et al., 2020; Ramos-Gutiérrez et al., 2021; Rebmann & van der Aa, 2022) presents a comparison of existing works in the field of process model discovery. The comparison study was done using the following criteria:

- Process mining algorithm: Indicates the algorithm used to discover the process model.
- Input: The inputs used for process discovery application (one event log, collection of event logs, semantically enriched logs).
- Type of the resulting model: The output of process discovery application (business process model, configurable process model, variant model).
- Automatic/manual: Indicates if the process discovery is done automatically or manually.
- Does the resulting model support variability?
- Does the resulting model support semantics?
- Type of ontology if used: When the resulting model supports semantics, this criterion identifies which type of ontology is used (domain ontology or business ontology).
- The objective behind using semantics: Indicates for which purpose semantics were employed in the approach.

According to this overview, we can observe that there are two categories of works: the first category (Assy et al., 2014; Khodayari Tehraninejad & Khoshnevis, 2022; Ramos-Gutiérrez et al., 2021) includes works that discover configurable process models or fragments through the application of process mining algorithms on collections of event logs. The resulting model supports variability but lacks semantics. The second category (Cairns et al., 2014; Nykänen et al., 2015; Rebmann & van der Aa, 2022) includes works that discover business process models without variability, but integrate semantics to improve event log data quality or improve the resulting model quality. Some other works (Fang et al., 2023; Pereira Detro et al., 2020) use semantics to derive process variants from an

existing configurable process.

Following this comparison, we concluded that most of the existing approaches do not combine variability and semantics in process model discovery. No one of the existing works tries to integrate semantics in configurable process model discovery, despite the importance of this element to improve model quality and facilitate its verification and enhancement. Semantics can be also used as a support tool during process model configuration and analysis.

#### 4. OBJECTIVES & CONTRIBUTION

The objective of this work is to integrate semantics in configurable process model discovery. The use of semantics has produced significant results for business process models. By analogy, we intend to apply the same for CPM, especially since configurable process models are too large and complex and require more effort and resources to analyze, configure, and enhance. Therefore, the usage of semantics will provide rich support during the lifecycle management of CPM.

In this paper, we aim to integrate semantics using ontologies (CPM ontology and domain ontology). The CPM ontology is used to pre-process the event logs and mark variability elements in the resulting merged event log. The domain ontology is used during the process mining application, and related annotations will appear in the resulting configurable process model.

Given that IM is the most used algorithm in process mining practices (Amoury & Bertet, 2023) and can deal with noisy, incomplete traces, we choose to extend it to enrich the resulting configurable process models with semantics.

#### 5. APPROACH OVERVIEW

In our approach, configurable process mining has been realized as a multi-step mechanism, schematized in the below Fig. 3. Below are the three main steps of the proposed approach.

- Preparing event logs: As a first step, we prepare the merged event log and insert annotations as presented in our previous work (Khannat et al., 2021). We start with a collection of event logs, variability specification files, and CPM ontology, and as a result, we generate a merged event log enriched semantically with CPM annotations.
- Discovering CPM fragments: In the second step,

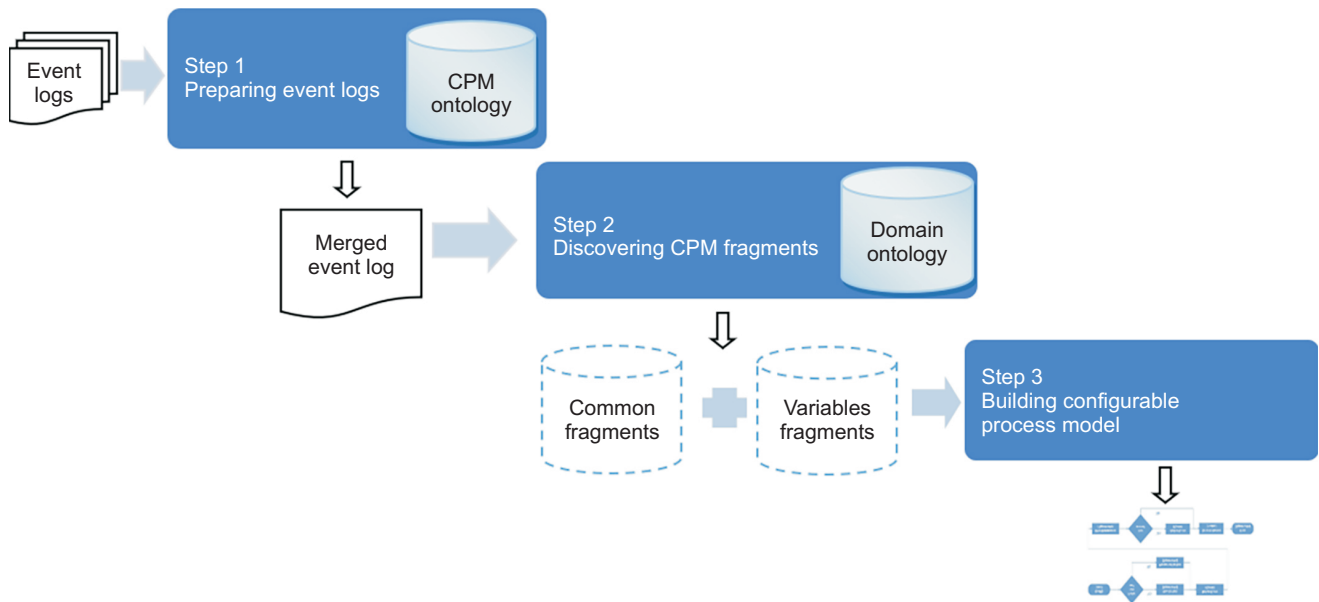


Fig. 3. Semantic configurable process mining approach. CPM, configurable process model.

having the merged event log enriched semantically, we work on the mining of configurable process model fragments. The detail of this step is presented in this paper.

- Building the configurable process model: Finally, in the last step, we consolidate the discovered fragments into the configurable process model.

## 6. CPM FRAGMENTS DISCOVERY APPROACH

The configurable process model discovery approach is performed on a collection of event logs that is merged and enriched semantically with CPM ontology. The approach proposed is based on multiple steps, as presented in Fig. 3. These steps are described in the next sections.

First, we merge the collection of event logs and annotate the resulting log using the CPM ontology.

We consider that the process we are dealing with is composed of the three types of activities that characterize the configurable process model: i) atomic activity, ii) complex activity, and iii) configurable activity. An atomic activity is a simple activity that performs an atomic task that cannot be divided into sub-activities. A configurable activity includes variability and can be configured during execution. A complex activity consists of more than one atomic activity and configurable activity (Sharma et al., 2014).

In the second step, we will split the merged event log into multiple sub-logs that correspond to CPM fragments,

depending on the activity types defined above, and classify them into two categories (with/without variability). Therefore, we will apply mining techniques to discover each CPM fragment separately.

Finally, we will link fragments and integrate activities to obtain the configurable process model.

Input: Collection of event logs+variability specification file+domain ontology+CPM ontology

Output: Configurable process model fragments

1. Pre-processing event logs (merging+annotation with CPM ontology)
2. Annotation of merged event log with taxonomy classes
3. Split merged event log on sub-logs
4. Classify sub-logs in two categories (with variability/without variability)
5. Mining of CPM fragments (common & variant)

### 6.1. Event Logs Pre-Processing

This step was the subject of our previous work (Khannat et al., 2021). Before starting this step, we assume that the collection of event logs used as input belongs to the same process execution in different conditions, and that the logs are clear from noise, complete, and homogeneous. Thus, we will discover one configurable process model that will be meaningful regarding variability and is less complex.

The first task is performed to merge the collection of event logs into one log. For this we use the technique in-

roduced in Claes and Poels (2014) that aims to create one consolidated event log starting from two event logs, based on the configured merging rules. After this, we annotate the merged log with CPM ontology concepts based on the variability specification file.

- Definition 1 - Event log: An Event log  $E$  is defined as a set of traces (or cases)  $E = \sum T$ , and each trace is a sequence of events of activities  $T = \sum a$ .
- Definition 2 - Merged event log: Merged event log  $mE$  is defined as a set of event logs corresponding to the execution of the same process  $mE = \sum E$ , and each event log  $E = \sum T (i, j)$  is a set of traces with two indexes, where index  $i$ : refers to trace id and index  $j$ : refers to event log id.
- Definition 3 - Variable traces: Variable traces set  $VarTrace(E)$  is defined as a set of traces that contains at least one configurable activity and which will be annotated with #Variable annotation.
- Definition 4 - Variable activities: Variable activities set  $VarActivity(E)$  is defined as a set of variation points that will be annotated with #VariationPoint annotation.
- Definition 5 - Activity variants: Activity variants set  $Var(ai)$  contains the variants of the activity  $ai$  that will be annotated with #Variant annotation.

We test with the sample event logs below:

$E1 = \{T1 \parallel T2 \parallel T3\}$ : composed of three traces

$E2 = \{T1 \parallel T3 \parallel T4\}$ : composed of three traces

$E3 = \{T2 \parallel T4\}$ : composed of two traces

$T1 = \{a1 \parallel a2 \parallel a4 \parallel a5 \parallel a7 \parallel a8\}$

$T2 = \{a1 \parallel a3 \parallel a4 \parallel a6 \parallel a7 \parallel a8\}$

$T3 = \{a1 \parallel a3 \parallel a4 \parallel a5 \parallel a7 \parallel a8\}$

$T4 = \{a1 \parallel a2 \parallel a4 \parallel a6 \parallel a7 \parallel a8\}$

The merged event log is  $E = \{T11 \parallel T21 \parallel T31 \parallel T12 \parallel T32 \parallel T42 \parallel T23 \parallel T43\}$

In terms of activities:  $E = \{\{a1 \parallel a2 \parallel a4 \parallel a5 \parallel a7 \parallel a8\} \parallel \{a1 \parallel a3 \parallel a4 \parallel a6 \parallel a7 \parallel a8\} \parallel \{a1 \parallel a3 \parallel a4 \parallel a5 \parallel a7 \parallel a8\} \parallel \{a1 \parallel a2 \parallel a4 \parallel a6 \parallel a7 \parallel a8\} \parallel \{a1 \parallel a3 \parallel a4 \parallel a6 \parallel a7 \parallel a8\} \parallel \{a1 \parallel a2 \parallel a4 \parallel a6 \parallel a7 \parallel a8\}\}$

$VarTrace(E) = \{T1 \parallel T2 \parallel T3 \parallel T4\}$

$VarActivity(E) = \{a1 \parallel a4\}$

$Var(a1) = \{a2 \parallel a3\}$ ,  $Var(a4) = \{a5 \parallel a6\}$

### 6.2. Semantic Annotation of Merged Event Log

The objective of this step is to enrich the merged event log with meta-data related to domain knowledge. For this purpose, we are using a domain ontology that includes a taxonomy classifying process tasks in a hierarchical structure, as shown in Fig. 4 (Paquette, 2010).

Using the knowledge shared about the process, we try to link each event in the merged log to one class of the taxonomy. Through this method, we can determine if the event reports to a complex activity or atomic activity. For example, if our process is composed of the below activities:

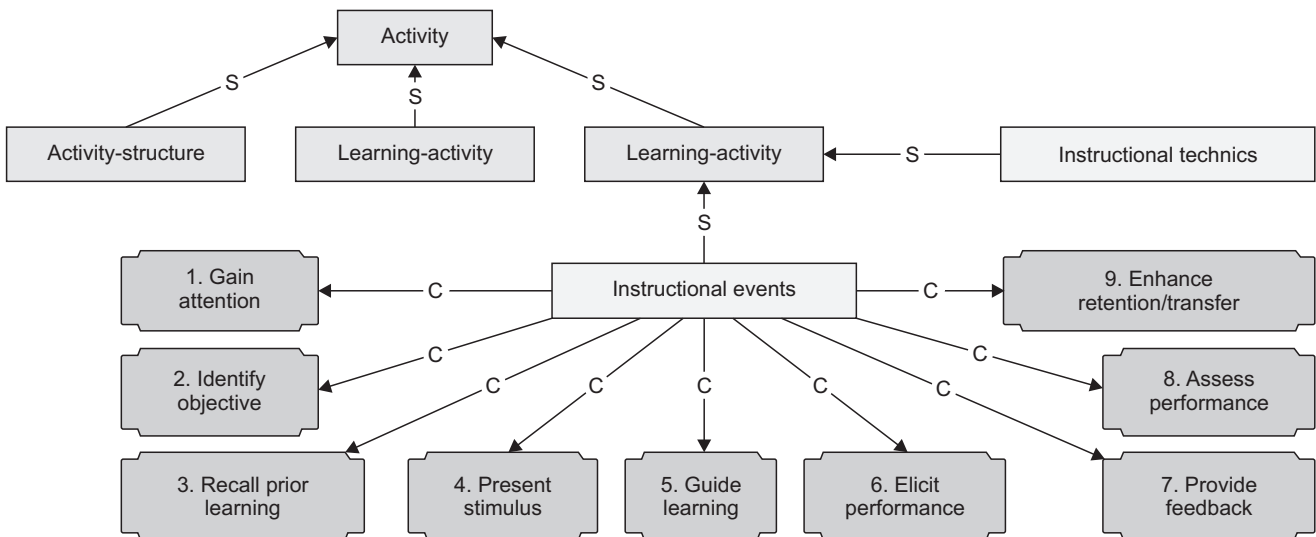


Fig. 4. Taxonomy of class activity in ontology learning design. Adapted from Paquette (2010). S, SubClassOf; C, Class.



**Table 3.** Semantically annotated event log

Case ID	Task name	Originator	Timestamp	Semantic annotation
1	a1	User1	20-07-2004 14:00:00	<<CA1>>
1	a2	User1	20-07-2004 14:05:01	<<CA1>>
2	a1	User2	20-07-2004 14:06:00	<<CA1>>
2	a3	User2	20-07-2004 14:20:00	<<CA1>>
2	a4	System	20-07-2004 15:00:00	<<CA2>>
3	a1	User3	20-07-2004 15:01:00	<<CA1>>
1	a4	System	20-07-2004 15:13:00	<<CA2>>
2	a5	User2	20-07-2004 15:27:00	<<CA2>>
3	a3	User3	20-07-2004 16:00:00	<<CA1>>
3	a4	System	20-07-2004 16:09:00	<<CA2>>
2	a7	SYSTEM	20-07-2004 16:35:00	<<CA3>>
2	a8	SYSTEM	20-07-2004 16:38:00	<<CA3>>
1	a6	SYSTEM	20-07-2004 17:00:00	<<CA2>>
1	a7	SYSTEM	20-07-2004 17:14:00	<<CA3>>
1	a8	SYSTEM	20-07-2004 17:22:00	<<CA3>>
3	a5	User3	20-07-2004 18:00:00	<<CA2>>
3	a7	SYSTEM	20-07-2004 18:21:00	<<CA3>>
3	a8	SYSTEM	20-07-2004 18:30:00	<<CA3>>

CA1: {a1 || a2 || a3}

CA2: {a4 || a5 || a6}

CA3: {a7 || a8}

(CA<sub>i</sub>) refers to complex activity, and (ai) refers to atomic activity.

Table 3 demonstrates a fragment of the event log enriched semantically with domain ontology; the event log captures atomic activities executed during the process. For each activity (ai) in event log, we use the column “Semantic Annotation” to reference related complex activity (CA<sub>i</sub>).

### 6.3. Split Merged Event Log

To discover a less complex model, we chose to split the merged event log into multiple sub-logs, where each sub-log corresponds to one complex activity. The idea here is to obtain the sub-logs that will serve as an input for the process mining algorithm to discover configurable process model fragments. Fig. 5 demonstrates the algorithm proposed for this step.

This algorithm builds a set of sub-logs based on the merged event log and a list of complex activities present in the log.

**Algorithm 1** Algorithm for splitting merged event log

```

1: input: Merged Event Log  $L \subseteq \mathcal{E}$ 
2: output: Set of sub-logs  $S(L) = \{l_i; i \in N\}$ 
   {Create sub-logs corresponding to complex activities}
3:  $i = 1;$ 
4: for  $C \in \text{Comp}_L$  do {for each complex activity of L}
5:    $l_i = \text{Clone}(L, i)$  {create a copy of L with identifier i}
6:   for  $\mathcal{E} \in L$  then
7:     if  $\text{GetSemanticAnnotation}(\mathcal{E}) \neq C$  then
8:        $\text{Cleanlog}(l_i, \mathcal{E})$ 
9:     end if
10:  end for
11:   $i = i + 1;$ 
12: end for

```

**Fig. 5.** Algorithm for splitting merged event log.

For each complex activity, it clones the merged event log and goes through the elimination technique to keep only events related to this complex activity in the copy created. Once we deduced a sub-log for each complex activity of the process, we used the algorithm in Fig. 6 to classify them into two categories: Category 1: sub-logs

that contain variability (at least, they contain one variable activity) and Category 2: sub-logs without variability (all activities are common).

Regarding our example, the split event logs are below:

$$\begin{aligned} E(\text{CA1}) &= \{\{a1 \parallel a2\} \parallel \{a1 \parallel a3\} \parallel \{a1 \parallel a3\} \parallel \{a1 \parallel a2\} \\ &\parallel \{a1 \parallel a3\} \parallel \{a1 \parallel a2\} \parallel \{a1 \parallel a3\} \parallel \{a1 \parallel a2\}\} \\ E(\text{CA2}) &= \{\{a4 \parallel a5\} \parallel \{a4 \parallel a6\} \parallel \{a4 \parallel a5\} \parallel \{a4 \parallel a5\} \\ &\parallel \{a4 \parallel a5\} \parallel \{a4 \parallel a6\} \parallel \{a4 \parallel a6\} \parallel \{a4 \parallel a6\}\} \\ E(\text{CA3}) &= \{\{a7 \parallel a8\} \parallel \{a7 \parallel a8\} \parallel \{a7 \parallel a8\} \parallel \{a7 \parallel a8\} \\ &\parallel \{a7 \parallel a8\} \parallel \{a7 \parallel a8\} \parallel \{a7 \parallel a8\} \parallel \{a7 \parallel a8\}\} \\ \text{Variable\_logs} &= \{E(\text{CA1}) \parallel E(\text{CA2})\} \\ \text{Common\_logs} &= \{E(\text{CA3})\} \end{aligned}$$

#### 6.4. Mining CPM Fragments

After classifying the sub-logs into two categories, we will apply the process mining algorithm to discover configurable process model fragments. We are expecting to get for each sub-log a fragment of the configurable process model.

Variable fragments will be discovered from variable sub-logs, and common fragments will be discovered from common sub-logs.

Knowing that the common logs did not contain any variable activity, we will directly apply IM to discover the corresponding fragments, and for variable sub logs, we will apply a customized algorithm *Inductive\_Miner\_Var* that includes variability in the process tree discovered.

##### 6.4.1. Mining CPM Common Fragments

For the common fragments we directly apply the basic

Algorithm 2	Algorithm for classifying sub-logs
1:	<b>input:</b> Set of sub-logs $S(L) = \{l_i; i \in N\}$
2:	<b>output:</b> Variable_logs, Common_logs
	{Build Variable_logs and Common_logs sets}
3:	$i = 1;$
4:	<b>for</b> $l_i \in S(L)$ <b>do</b> {for each sub-log}
5:	var_flag=true;
6:	<b>while</b> var_flag <b>do</b>
7:	<b>for</b> $\mathcal{E} \in l_i$ <b>then</b>
8:	<b>if</b> GetCpmAnnotation( $\mathcal{E}$ ) <b>is not NULL</b> <b>then</b>
9:	Variable_logs = Variable_logs $\cup$ $l_i$
10:	var_flag=false;
11:	<b>end if</b>
12:	<b>end for</b>
13:	<b>end while</b>
14:	<b>end for</b>
15:	Common_logs = $S(L) \setminus$ Variable_logs;

Fig. 6. Algorithm for classifying sub-logs.

version of IM (Pohl, 2019), which is based on four main functions: cut detection, log splitting, base cases, and fall through. IM takes as input an event log and builds the appropriate directly-follows graph (dfg). IM looks for a cut in dfg, following this order of priority:

- 1) XOR Cut
- 2) Sequence Cut
- 3) Parallel Cut
- 4) Loop Cut

Depending on the cut found, one of the different split functions below is applied to split the initial log into one or more sub-logs.

- 1) XOR Split
- 2) Sequence Split
- 3) Concurrent Split
- 4) Loop Split

After this step, the algorithm is recursively applied to each sub-log until a base case, i.e. a dfg with a single activity (Leemans, 2017), is detected. In case no base case is found and no more cuts are detected, a fall through is applied (i.e. a process tree is discovered) (Leemans, 2017). Fig. 7 shows the common fragment mined from sub-log  $E(\text{CA3})$ .

##### 6.4.2. Mining CPM Variable Fragments

For the variable fragments, we conceived the new algorithm, *Inductive\_Miner\_Var*, that represents an extension of the IM supporting variability notions. This algorithm (Fig. 8) is also based on the four main functions of an IM, with further processing to include variability as metadata in the resulting process tree. When a cut is detected, we search for single activities in the cut partitions. Then, we check if this activity is a variant, according to the annotations present in the log; if yes, we mark the activity with the label  $\langle\langle$ Variant $\rangle\rangle$  and mark the direct relation connected to this activity with one of the labels  $\langle\langle$ Vp 1..1 $\rangle\rangle$ ,

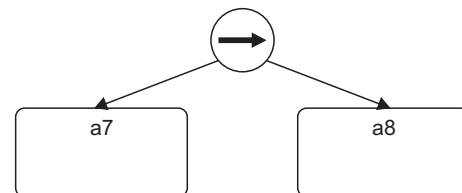


Fig. 7. Process tree discovered from sub-log  $E(\text{CA3})$ .

<<Vp 0..n>>, or <<Vp 0..1>>, depending on the type of the variation point, i.e. Alternatif or Optional or Optional-Alternatif.

Fig. 9 illustrates the resulting variable fragment related to the sub-log E(CA1).

```

Algorithm 3 Algorithm for mining variable fragments
1: input: Set of sub-logs  $Var(S(L))=\{l_i: i \in N\}$ 
2: output: Set of process trees  $(Q_1, \dots, Q_n)$ 
3:  $R = NULL$  {Relation between cuts}
4:  $i = 0$ 
5: While not ( $Var(S(L)).isEmpty()$ ) do
6:  $BC = findbaseCaseLM(l_i)$ 
7: if ( $i > 0$  &  $BC$  is not  $NULL$ ) then
   {Delete sub-log processed from the list}
8:  $Var(S(L)) = Var(S(L)) \setminus \{l_i\}$ 
9: if ( $BC \neq \tau$ ) then {Base Case is a single activity}
10:  $An = GetCPMAnotation(BC)$ 
11: if ( $An = CPMO\#default$ ) then
   {Annotate the activity BC with CPM annotation}
12:  $Annotate(BC, \ll Default \gg)$ 
13: else if ( $An = CPMO\#variant$ ) then
    $Annotate(BC, \ll Variant \gg)$ 
14: else if ( $An = CPMO\#alternative$ ) then
   {Annotate relation linked to BC with CPM annotation}
15:  $AnnotateRelation(BC, \ll Vp 1..1 \gg)$ 
16: else if ( $An = CPMO\#optional$ ) then
    $AnnotateRelation(BC, \ll Vp 0..n \gg)$ 
17: else if ( $An = CPMO\#optionalAlternative$ ) then
    $AnnotateRelation(BC, \ll Vp 0..1 \gg)$ 
18: end if
19: end if
20: end if
21: end if
22: end if
23: end if
24: end if
25: end if
26: end if
27:  $i = i + 1$ 
28: else
   {Calculate Directly-following graph resulting of the sub-log}
29:  $DFG = ComputeDFG(l_i)$ 
30:  $(R, P_1, P_2, \dots, P_k) = FindCut(l_i)$  {Find a cut partition}
31: if  $R$  is not  $NULL$  then
   {Apply split to sub-log using relation R}
32:  $E_1, \dots, E_k = Split(R, l_i)$ 
33:  $Var(S(L)) = Var(S(L)) \cup \{E_1, \dots, E_k\}$  {Add sub-logs to the list}
34: else
35: return FallThrough ( $l_i$ )
36: end if
37: end if
38: end While
    
```

Fig. 8. Algorithm for mining variable fragments.

### 6.5. Consolidation of CPM

The last step is designed for the consolidation of the configurable process model by linking the discovered common and variant fragments. During this step, we annotate common fragments with the label <<Common>> and variable fragments with the label <<VAR>>. Fig. 10 shows the resulting process tree of our example after linking variables and common fragments.

## 7. CONCLUSION

Seen from an analysis point of view, configurable process models are too large, complex, and difficult to evolve efficiently. Therefore, representing semantics that reference concepts of a formal ontology in those models is much needed.

The semantics of annotations will provide support during the configuration and the extension of the configurable process model. Going further, these labels can be used for the validation of derived variants. In this regard, we focus on discovering configurable process models enriched semantically. Semantics are incorporated using annotations related to variability and domain concepts.

In the present paper, we proposed a novel approach to discover a configurable process model annotated semantically, starting with a collection of event logs and two ontologies: domain ontology and business process ontology.

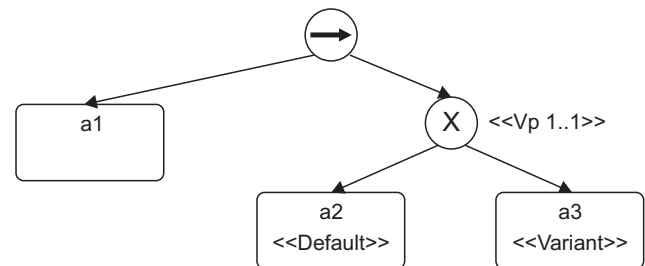


Fig. 9. Process tree discovered from sub-log E(CA1).

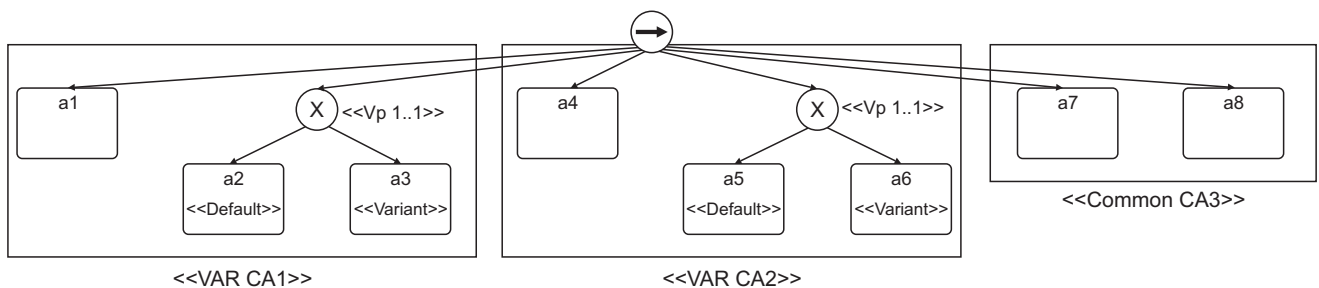


Fig. 10. Semantic configurable process tree.

The method consists of discovering common fragments and variable fragments of configurable process models separately, and after that linking them. For variable fragments, we proposed an extension of the IM algorithm to generate variability annotations in the discovered process trees.

The proposed approach shows the theoretical approach related to the creation of annotated models. However, the implementation of the test environment and testing on real logs should be the subject of future research. Moreover, we plan to provide a model evaluation of the proposed method. Also, as future work, we aim to extend this approach to include the discovery of variants/configurations at the same time of the configurable process model discovery through the configuration matrix, and contribute to the enhancement of existing configurable process models using ontologies.

## CONFLICTS OF INTEREST

No potential conflict of interest relevant to this article was reported.

## REFERENCES

- Amoury, S., & Bertet, K. (2023). [Process mining for the improvement of a serious game]. France. <https://univ-rochelle.hal.science/hal-04117522/>
- Assy, N., Gaaloul, W., & Defude, B. (2014, May 22-24). Mining configurable process fragments for business process design. In M. C. Tremblay, D. VanderMeer, M. Rothenberger, A. Gupta, & V. Yoon (Eds.), *Advancing the impact of design science: Moving from theory to practice* (pp. 209-224). Springer. [https://doi.org/10.1007/978-3-319-06701-8\\_14](https://doi.org/10.1007/978-3-319-06701-8_14)
- Ayora, C., Torres, V., Weber, B., Reichert, M., & Pelechano, V. (2013, June 17-18). Enhancing modeling and change support for process families through change patterns. In S. Nurcan, H. A. Proper, P. Soffer, J. Krogstie, R. Schmidt, T. Halpin, & I. Bider (Eds.), *Enterprise, business-process and information systems modeling* (pp. 246-260). Springer. [https://doi.org/10.1007/978-3-642-38484-4\\_18](https://doi.org/10.1007/978-3-642-38484-4_18)
- Azouzi, S., Ghannouchi, S. A., & Brahmi, Z. (2017, December 18-20). Towards supporting modeling variability in e-learning application: A case study. In S. J. Horng (Ed.), *18th International conference on parallel and distributed computing, applications and technologies* (pp. 488-494). IEEE Computer Society Conference Publishing Services. <https://doi.org/10.1109/PDCAT.2017.00083>
- Barbieri, L., Madeira, E., Stroeh, K., & van der Aalst, W. (2023). A natural language querying interface for process mining. *Journal of Intelligent Information Systems*, 61(1), 113-142. <https://doi.org/10.1007/s10844-022-00759-9>
- Berti, A., Kourani, H., Hafke, H., Li, C. Y., & Schuster, D. (2024). *Evaluating large language models in process mining: Capabilities, benchmarks, evaluation strategies, and future challenges*. <https://arxiv.org/html/2403.06749v1>
- Buijs, J. C. A. M., van Dongen, B. F., & van der Aalst, W. M. P. (2013, August 26-30). Mining configurable process models from collections of event logs. In F. Daniel, J. Wang, & B. Weber (Eds.), *Business process management* (pp. 33-48). Springer. [https://doi.org/10.1007/978-3-642-40176-3\\_5](https://doi.org/10.1007/978-3-642-40176-3_5)
- Cairns, A. H., Ondo, J. A., Gueni, B., Fhima, M., Schwarfeld, M., Joubert, C., & Khelifa, N. (2014). *Using semantic lifting for improving educational process models discovery and analysis*. Paper presented at the 4th International Symposium on Data-Driven Process Discovery and Analysis, Milan, Italy.
- Claes, J., & Poels, G. (2014). Merging event logs for process mining: A rule based merging method and rule suggestion algorithm. *Expert Systems with Applications*, 41(16), 7291-7306. <https://doi.org/10.1016/j.eswa.2014.06.012>
- Detro, S. P., Portela, E., Rocha, E. L., Panetto, H., & Lezoche, M. (2017). Configuring process variants through semantic reasoning in systems engineering. *Insight*, 20(4), 36-39. <https://doi.org/10.1002/inst.12179>
- El Faquih, L., & Fredj, M. (2017). Ontology-based framework for quality in configurable process models. *Journal of Electronic Commerce in Organizations (JECO)*, 15(2), 48-60. <https://doi.org/10.4018/JECO.2017040104>
- El Faquih, L., Sbaï, H., & Fredj, M. (2015). *Configurable process models: A semantic validation*. Paper presented at 2015 10th International Conference on Intelligent Systems: Theories and Applications, Rabat, Morocco. <https://doi.org/10.1109/SITA.2015.7358436>
- Fang, H., Liu, W., Wang, W., & Zhang, S. (2023). Discovery of process variants based on trace context tree. *Connection Science*, 35(1), 2190499. <https://doi.org/10.1080/09540091.2023.2194578>
- Gottschalk, F., van der Aalst, W. M. P., & Jansen-Vullers, M. H. (2007). Configurable process models — A foundational approach. In J. Becker & P. Delfmann (Eds.), *Reference modeling: Efficient information systems design through reuse of information models* (pp. 59-77). Physica-Verlag HD.
- Gottschalk, F., Wagemakers, T. A. C., Jansen-Vullers, M. H., van der Aalst, W. M. P., & La Rosa, M. (2009, June 8-12). Configurable process models: Experiences from a municipality case study. In P. Eck, J. Gordijn, & R. Wieringa (Eds.), *Advanced information systems engineering* (pp. 486-500). Springer. [https://doi.org/10.1007/978-3-642-02144-2\\_38](https://doi.org/10.1007/978-3-642-02144-2_38)

- Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2), 199-220. <https://doi.org/10.1006/knac.1993.1008>
- Günther, C. W., & van der Aalst, W. M. P. (2007, September 24-28). Fuzzy mining – Adaptive process simplification based on multi-perspective metrics. In G. Alonso, P. Dadam, & M. Rosemann (Eds.), *Business process management* (pp. 328-348). Springer. [https://doi.org/10.1007/978-3-540-75183-0\\_24](https://doi.org/10.1007/978-3-540-75183-0_24)
- Hepp, M., Leymann, F., Domingue, J., Wahler, A., & Fensel, D. (2005). *Semantic business process management: A vision towards using semantic web services for business process management*. Paper presented at IEEE International Conference on e-Business Engineering (ICEBE'05), Beijing, China.
- Khannat, A., Sbaï, H., & Kjiri, L. (2021). Configurable process mining: Semantic variability in event logs. In J. Filipe, M. Smialek, A. Brodsky, & S. Hammoudi (Eds.), *Proceedings of the 23rd international conference on enterprise information systems - (Volume 1)* (pp. 768-775). SciTePress. <https://doi.org/10.5220/0010484207680775>
- Khodayari Tehraninejad, D., & Khoshnevis, S. (2022). Proxima: Process mining for extracting configurable process models using software product line concepts. *Journal of Computing and Security*, 9(1), 1-29. <https://doi.org/10.22108/jcs.2021.130065.1080>
- La Rosa, M. (2009). *Managing variability in process-aware information systems* (doctoral dissertation). Queensland University of Technology, Brisbane, Australia.
- La Rosa, M., & Gottschalk, F. (2009, September 8). Synergia – Comprehensive tool support for configurable process models. In A. K. A. de Medeiros & B. Weber (Eds.), *Business process management demonstration track 2009*. CEUR.
- La Rosa, M., Aalst, W. M. P. V. D., Dumas, M., & Milani, F. P. (2017). Business process variability modeling: A survey. *ACM Computing Surveys*, 50(1), Article 2. <https://doi.org/10.1145/3041957>
- La Rosa, M., Dumas, M., ter Hofstede, A. H. M., & Mendling, J. (2011). Configurable multi-perspective business process models. *Information Systems*, 36(2), 313-340. <https://doi.org/10.1016/j.is.2010.07.001>
- La Rosa, M., Dumas, M., ter Hofstede, A. H., Mendling, J., & Gottschalk, F. (2007). *Beyond control-flow: Extending business process configuration to resources and objects*. <https://eprints.qut.edu.au/215913/>
- Lautenbacher, F., Bauer, B., & Forg, S. (2009). *Process mining for semantic business process modeling*. Paper presented at 2009 13th Enterprise Distributed Object Computing Conference Workshops, Auckland, New Zealand.
- Leemans, S. J. J. (2017). *Robust process mining with guarantees*. (doctoral dissertation). Eindhoven University of Technology, Eindhoven, Netherlands.
- Leemans, S. J. J., Fahland, D., & van der Aalst, W. M. P. (2013, June 24-28). Discovering block-structured process models from event logs - A constructive approach. In J. M. Colom & J. Desel (Eds.), *Application and theory of Petri nets and concurrency* (pp. 311-329). Springer. [https://doi.org/10.1007/978-3-642-38697-8\\_17](https://doi.org/10.1007/978-3-642-38697-8_17)
- Nykänen, O., Rivero-Rodriguez, A., Pileggi, P., Ranta, P. A., Kailanto, M., & Koro, J. (2015). *Associating event logs with ontologies for semantic process mining and analysis*. Paper presented at Proceedings of the 19th International Academic Mindtrek Conference, Tampere, Finland.
- Ozates, A., Akin, E., Bilgic, A., Akca, E., & Tastimur, C. (2024). *Analysis of barrel electroplating line with process mining and petri-net model*. Paper presented at 2024 International Conference on Computer, Electrical & Communication Engineering, Kolkata, India.
- Paquette, G. (2010). Ontology-based educational modelling - making IMS-LD visual. *Technology, Instruction, Cognition and Learning*, 7, 263-296.
- Pereira Detro, S., Santos, E. A. P., Panetto, H., Loures, E. D., Lezoche, M., & Cabral Moro Barra, C. (2020). Applying process mining and semantic reasoning for process model customisation in healthcare. *Enterprise Information Systems*, 14(7), 983-1009. <https://doi.org/10.1080/17517575.2019.1632382>
- Pohl, T. (2019). *An inductive miner implementation for the PM4PY framework*. <https://www.pads.rwth-aachen.de/cms/pads/studium/abgeschlossene-abschlussarbeiten/2019/~fiwjs/an-inductive-miner-implementation-for-th/>
- Ramos-Gutiérrez, B., Varela-Vaca, Á. J., Galindo, J. A., Gómez-López, M. T., & Benavides, D. (2021). Discovering configuration workflows from existing logs using process mining. *Empirical Software Engineering*, 26(1), 11. <https://doi.org/10.1007/s10664-020-09911-x>
- Rebmann, A., & van der Aa, H. (2022). Enabling semantics-aware process mining through the automatic annotation of event logs. *Information Systems*, 110, 102111. <https://doi.org/10.1016/j.is.2022.102111>
- Rosemann, M., & van der Aalst, W. M. P. (2007). A configurable reference modelling language. *Information Systems*, 32(1), 1-23. <https://doi.org/10.1016/j.is.2005.05.003>
- Schnieders, A., & Puhlmann, F. (2007). Variability modeling and product derivation in e-business process families. In W. Abramowicz & H. C. Mayr (Eds.), *Technologies for Business Information Systems* (pp. 63-74). Springer.
- Sharma, D. K., Hitesh, & Rao, V. (2014). *Configurable business process modeling notation*. Paper presented at 2014 IEEE International Advance Computing Conference, Gurgaon,

India.

- van der Aalst, W. M. P., de Medeiros, A. K. A., & Weijters, A. J. M. M. (2005, June 20-25). Genetic process mining. In G. Ciardo & P. Darondeau (Eds.), *Applications and theory of Petri nets 2005* (pp. 48-69). Springer. [https://doi.org/10.1007/11494744\\_5](https://doi.org/10.1007/11494744_5)
- van der Aalst, W., Weijters, T., & Maruster, L. (2004). Workflow

- mining: Discovering process models from event logs. *IEEE Transactions on Knowledge and Data Engineering*, 16(9), 1128-1142. <https://doi.org/10.1109/TKDE.2004.47>
- Weijters, A. J. M. M., van der Aalst, W. M. P., & Alves de Medeiros A. K. (2006). Process mining with the Heuristics-Miner algorithm. *BETA publicatie : Working papers*, 166.