Advances in Science, Technology and Engineering Systems Journal Vol. 5, No. 1, 183-190 (2020)

www.astesj.com

Special Issue on Multidisciplinary Sciences and Engineering

ASTESJ

ISSN: 2415-6698

Configurable Process Model: Discovery Approach from Event Logs

Rabab Sikal[*,1], Hanae Sbai[2], Laila Kjiri[1]

[1]*AlQualsadi research team, ENSIAS, Mohammed V University of Rabat, Morocco*

[2] *FST, University Hassan II of Casablanca, Mohammedia, Morocco*

ARTICLE INFO

Article history:
Received: 30 September, 2019
Accepted: 29 December, 2019
Online: 22 January, 2020

Keywords:
Configurable Process Model
Process Discovery
Variability

ABSTRACT

In the domain of business process management, the configurable process model is widely used to optimize time and cost of business process models design, which is known as the concept of "reuse". Using process mining techniques for process model discovery helps to provide a better view on processes and improve quality of models. The majority of existing configurable model discovery approaches work intensively on control flow discovery as main process perspective without considering other perspectives such as resources and data, and do not propose a detailed discovery of variability elements. In addition, the configurable process model creation is generally done by merging variant models not directly from event logs, which is not the optimal way to get a reliable configurable process model. This paper presents an overview of new multi-perspective variability discovery approach. The approach respects the variability of different process perspectives and allows users to create a configurable process model directly from event logs.

1. Introduction

Process models are designed, managed, configured mainly through methods and tools provided by the Business Process Management (BPM) domain [1,2]. Actually, organizations are increasingly opting for event systems, also called process Aware Information Systems (PAIS) for the purpose of analyzing, supervising and optimizing the organization's processes [3]. Nowadays, rapid increase of business needs and fast changing of the enterprise environment derive the enterprise to face the challenge of saving time, reducing costs and minimizing errors of process management. Therefore, opting for reuse concept is a big requirement for making an optimal and flexible business process design [4,5]. Hence, the importance of taking into account the previous design experiences and do not design processes from scratch. In this sense, different approaches have proposed the reuse concept in process design while flexibility and adaptability are addressed in business process models [6, 7].

The configurable process model is defined as a single model that assembles all process variants in one model. It is also called “customizable process model” which means that this kind of process model regroups options, which can be configured by users to derive desired process variants. It represents commonalities and differences between all process variants, which offers flexibility and enables process design through reuse concept. The variation point is a configurable element of configurable process model. It represents where the variation occurs in the process model and represents all possible design choices. The configuration of the model consists of making choices of options for each configurable element according to specific requirements in order to derive individual and suitable model for the enterprise with minimal design effort. However, despite the diversity of approaches proposing creation and configuration of configurable process models, their management still requires a significant manual work in different steps (e.g. design, configuration and evolution).

Against this background, the techniques of process mining are introduced with the aim to automate the process management and minimize human intervention. The process mining uses data recorded in the event logs during the process execution in order to help organizations for discovering, checking conformance and enhancing their business processes.

Creating process model manually is a hard and redundant task, since the use of similar processes becomes more and more popular. For that reason, configurable process model discovery is used as an alternative for reusable process design. In this context, several

[*]Corresponding Author Rabab Sikal, rababsikal@gmail.com

https://dx.doi.org/10.25046/aj050124

works have proposed approaches for configurable process discovery [8, 9, 10]. However, existing works discover the control-flow as the main process perspective without considering other process perspectives like data and resources. In addition, the proposed variability discovery approaches do not present an explicit and detailed discovery of variability elements, namely variation points and variants. Moreover, existing approaches for configurable model creation use algorithms to construct configurable model by merging similar variant models, which result in large and complex models. In the light of these limitations, we have proposed in [11,12] a multi-perspective configurable process discovery approach with respect to variability of activities and resources. The comparative studies presented in these two works showed the lack of support for variability discovery for various process perspectives. This paper extends the work originally presented in the 2018 IEEE 5th International Congress on Information Science and Technology (CiSt) [11] that we complete by proposing an approach for configurable process model creation directly from event logs. The organization of this paper is as follows: section 2 defines the background of our research field, while section 3 reviews approaches working on configurable process model discovery. In section 4, we present our proposed configurable process discovery approach. Section 5 depicts our multi-perspective discovery framework and presents the algorithms for variability extraction and configurable model creation. Finally, Section 6 presents conclusions and some future directions.

2. Background

This section introduces three basic concepts used in this paper: process discovery, configurable process model, and variability. Then, we present briefly the four configurable process model discovery approaches described in [8].

2.1. Process discovery

Process mining is a set of techniques applied to extract data recorded in event logs. These data concern all process actions captured during process execution and are used to discover, monitor, and improve processes [1]. There are three main areas of process mining [1]:

- Process Discovery: the discovery algorithm takes an event log in input and produces a process model in output without using any additional knowledge.
- Conformance checking: verifies if an existing or discovered process model fits to its event log, or vice versa.
- Enhancement: extends and enrich existing process model, already discovered, by using information recorded in event logs.

In our approach, we focus on process discovery. There are two main kinds of process discovery:

Process discovery from one event log: it is the classical type of discovery. It allows for extracting one process model for each event log. However, this kind generates redundant processes [13,14].

Process discovery of a collection of event logs: this concerns the discovery of configurable process model. It requires firstly regrouping all event log that can belong to the same family and then applying techniques to discover the configurable process [8,9].

In Our work, we are interested in the configurable process model discovery from a collection of event logs.

2.2. Configurable process model & variability

The configurable process model represents shared/non configurable and unshared/configurable parts by all process variants in one global model. The configuration of configurable parts depends on the needs and the various constraints specific to the organization [15, 16]. Indeed, modeling all process variants and updating common process items cause redundancies and errors. Hence, the choice of configurable processes, generally presented by the merge of multiple process variants into a single process, is very useful to facilitate reuse and manage variability [17, 18]. Different extensions of process modeling languages have been developed for configurable process models representation, namely C-BPMN, C-EPC [15], C-YAWL [16] and configurable process tree.

The configuration consists of deriving individual process models corresponding to the different process variants from the configurable model. This operation is called individualization. The individualization is about blocking a given path of the model, so it cannot be taken or hiding activities they it can be skipped during the process execution. Designing a configurable process model consists in defining all the different variants of a given business process first and then integrating all of them into a single configurable model.

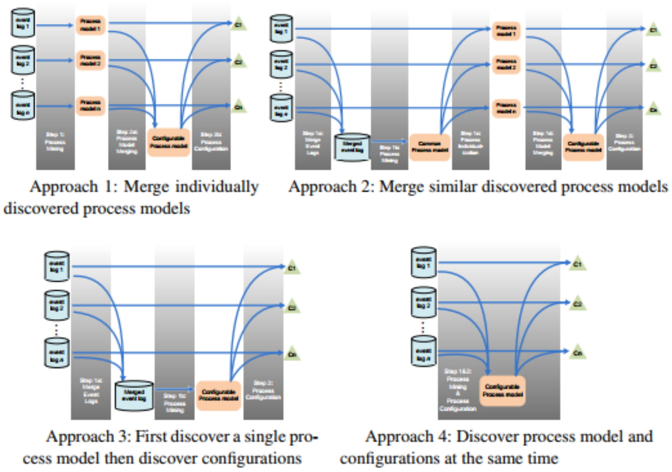
The variability is a key concept for configurable process model creation. It is represented by two main elements, namely, variation point and variants [13]. The variability defines and manages variable elements of business process [17]. Therefore, we define a process model that supports organizational, behavioral, functional and informational variability, as a multi-perspective configurable process model [1]. This can make the configurable process more explicit and valuable.

2.3. Approaches for configurable process model discovery

Process mining techniques for automated process discovery use data recorded in the event logs to represent the process behavior through a process model [6, 1]. Indeed, applying mining techniques for configurable process discovery is very useful, given the time saved and the effort reduced compared to conventional methods. In the literature, different approaches of configurable process discovery have been proposed. Buijs et al. [8] proposed four configurable process model discovery approaches, presented as follow:

- **Approach 1:** it is an approach initially proposed by Gottshalk [14]. The configurable process model discovered with this approach is the result of merging process models discovered from each event log.
- **Approach 2:** with the aim to improve the approach 1, the approach 2 merges all event logs and uses them to discover a common process model. Then, for every event log an

individual model is generated. Finally, the construction of the configurable model is done by merging the individual models.



Approach 1: Merge individually discovered process models

Approach 2: Merge similar discovered process models

Approach 3: First discover a single process model then discover configurations

Approach 4: Discover process model and configurations at the same time

Figure 1: the four approaches of configurable process model discovery [8]

- **Approach 3:** it suggests the merging of different event logs into one merged event log. Then, the configurable is discovered. It captures the behavior of all model that describes the behavior of these event logs.

- **Approach 4:** it allows for the discovery of the process model and its configurations at the same time [8].

In this paper, our discovery approach belongs to the approach 4. It deals with redundancies and brings more flexibility in using discovery techniques to construct processes that capture variability.

3. Related works

In this section, we present several existing approaches for configurable process model discovery. Then, these approaches are evaluated according to four criteria. Finally, we discuss results and limitations.

3.1. Configurable process model discovery works

Several BPM studies are interested in using the paradigm of the "design by reuse" for the construction of a configurable process model. Some of them have proposed to construct the configurable process model by merging all process variants models into one model. Others have proposed to create a configurable process model using mining techniques on a collection of event logs. The author of [9] uses trace-clustering method for configurable process model discovery from collection of event logs. In [10], the author discovers configurable process fragments to avoid complex and large models. The author of [14] presents an approach using process mining and analysis techniques to merge two business process models into a single model for further process optimization. The approach in [16] merges the models of process variants to create configurable process model based on log files from various systems. The work provides suggestions for common and individual configurations. In [19], the author proposes two algorithms, one to compute merged models. The other, to extract digests from a merged model. The work [20] proposes an algorithm for constructing a configurable process by merging the process model of each variant, the process model generated by the algorithm is pre-annotated for the configuration step. The study of [21] splits the event logs in a cluster and for each cluster, A process model can be discovered. In case of large configurable process model, the model is reduced into a sub process model. Each sub-process model is configured independently to improve performance and to reduce complexity.

3.2. Comparative study

To summarize the previous section, the table 1 presents the principal points related to our approach and reached by every work. The approaches are evaluated according to four criteria defined as follows:

- **Variability discovery:** it indicates if the approach discovers explicitly the elements of variability (e.g. variation point, variants and variables).

- **Perspective discovery**: it presents the perspectives discovered by the approach. The main process perspectives are: control flow (C.F), resource (R), data (D) and configuration (C).

- **Configurable Model construction**: it indicates if the approach constructs configurable model.

- **Discovery approach for configurable model construction:** it indicates which approach, from the four approaches proposed by [8], is used for the construction of configurable model.

Table 1: comparative study

Works	Variability discovery	Perspective discovery				Configurable Process model construction	Discovery approach for configurable model construction			
		C.F	R	D	C		App 1	App 2	App 3	App 4
[9]		+	-	-	+	+	-	-	+	-
[10]	+	+	-	-	+	+	-	+	-	-
[14] [16] [19]	-	+	-	-	-	+	+	-	-	-
[20]	-	+	-	-	-	+	-	+	-	-
[21]	-	+	-	-	+	+	-	-	+	-

Table 1 is a summary of the evaluation criteria developed by each of the approaches presented in this section.

Variability discovery: The discovery of variability elements, namely variation point and variants, is present in few works. The works [10] discovers variability for configurable fragment of the process. The other studies [9, 14, 16, 19 20, 21] don't focus on variability elements in their discovery approaches.

Perspectives discovery: The studies [9, 10, 14, 16, 19, 20, 21] are limited to the discovery of control flow as a main process perspective, and discovery of its configurations. Thus, we notice the absence of support for discovery of other process perspective like resource and data.

Configurable process model construction: all the presented approaches propose the construction of configurable process model using different modeling language, like C-BPMN, C-EPC or process tree.

Discovery approach for configurable model construction: different works adopted different approaches for model construction. The works [14, 16, 19] construct configurable model based on approach 1. The [10, 20] adopt the approach 2 while [9,21] use approach 3.

3.3. Discussion

The analysis of the presented approaches shows that the majority of them do not present an explicit discovery of elements of variability and still limited to control flow discovery as a main perspective. While, other perspectives like resource still neglected and not integrated in the discovery approach. In addition, the construction of configurable process model is generally based on approach 1, 2 or 3 using merging techniques. We notice the absence of approaches for configurable model discovery based on the approach 4.

The limitations we conclude are as follow:

- The need for detailed discovery of variability elements, e.g. variation points, variation point types and variants. Whereas, discovering an explicit and detailed variability can be used to build configurable process models as well as its configurations. It can also be archived for potential process changes or improvements.
- Lack of multi-perspective discovery of process elements, namely data and resources. Instead of remaining focused on the analysis of the control flow, the extension of the configurable model with different process perspectives is of great importance. It helps analysts to manage the evolution of business process and to improve decision-making.
- The construction of the configurable model is based on merging individual models. Unlike, an approach that discovers configurable model directly from the event logs without merging individual models may provide a better model structure and better configuration options.

In the light of the presented limitations, we propose an approach with a detailed discovery of variability elements for different perspectives (control flow, resource). In order to enhance the variability discovery for other perspectives, we generate variability specification files for detailed variability. This can ensure traceability and optimize the process of changing or updating business process. In addition, our approach adopts the approach 4 proposed in [8] for optimal creation of configurable process model.

4. Preliminaries

In this section, we present formal definitions of basic concepts related to this work.

4.1. Event logs

Event logs are defined as files that store process data collected during process execution. The process mining techniques use data recorded in event logs for discovering process models, checking conformance between process model and its event log, detecting execution deviations or errors and observing social behaviors.

The table 2 illustrates an example of event log for a "purchase online" process. To buy an article the customer starts with "creating a personal account online (a)". After, the customer "choose products to buy (b)" and then "chose the payment method (c)" it can be "payment by card (d)", "payment by PayPal (e) " or "bitcoin payment (f)". Thereafter, the customer "confirms the payment (g)". If the payment is ok, "delivery service is activated (h)". If not, the customer must verify the payment data.

IS: Information System

Table 2: event log of purchase online process

Case ID	Activity	Resource	Date	Time	…
1	a	IS	23-05-2017	13:00:00	…
1	b	Jane	23-05-2017	13:10:00	…
2	a	IS	23-05-2017	14:00:00	…
1	c	Jane	25-05-2017	11:20:00	…
2	b	Ellen	25-05-2017	12:00:00	…
1	d	IS	25-05-2017	13:40:00	…
3	a	IS	25-05-2017	14:05:00	…
3	b	Adele	25-05-2017	15:00:00	…
1	g	Jane	26-05-2017	08:45:00	…
2	c	Ellen	26-05-2017	09:36:00	…
3	c	Adele	26-05-2017	11:00:00	…
2	e	IS	26-05-2017	11:15:00	…
3	f	Adele	26-05-2017	11:20:00	…
2	g	Ellen	26-05-2017	11:40:00	…
1	h	IS	27-05-2017	09:00:00	…
3	g	Adele	27-05-2017	09:30:00	…
3	h	IS	27-05-2017	10:00:00	…
2	h	IS	27-05-2017	11:20:00	

The data recorded in event logs present the execution history of one business process within an organization. A log case represents one process instance execution. The log represented in Table 2, records three different executions of the same process. Each process execution is called process instance, and referenced by an ID. The event log contains additional attributes, such as resource that executes the activity, date and time of the activity execution.

Definition 1: (Trace, Event log). Let A be a set of activities in some universe of activities. A trace $\sigma \in A*$ is a sequence of activities. An event log $L \in T(A*)$ is a multi-set of traces, i.e., an event log.

For instance, < a; b; c; d; g; h > is a trace that belongs to the event log in Table 2.

4.2. Log based relation

A log file is a set of traces. A trace can be defined as a sequence of events ordered chronologically and executed correctly. The execution order of activities in a process instance is of great importance. It helps to define dependency between activities and to capture all possible patterns encoded in the event log [1]. Based on the activities execution orders in traces, four ordering relations can derived from an event log: >L, →L, ||L and #L [22].

Definition 2: (Log-based ordering relations [22]) Let $L \in M (A*)$ be an event log over A. Let a; b∈A. The four ordering relations are precedence ($>_L$), causality ($\rightarrow_L$), parallelism (||) and choice ($\#_L$), defined as follows:

Precedence: $a >_L b$ if $\exists \sigma =< a1; a2; :::an >\in L$ and $1 \leq i \leq n - 1$ such that $ai = a \wedge ai+1 = b$.

Causality: $a \rightarrow_L b$ if $a >_L b \wedge b >_L a$.

Parallelism: $a||_L b$ if $a >_L \wedge b >_L a$.

Choice: $a\#_L b$ if $a >_L b \wedge b >_L a$.

Based on the four log-based ordering relations, the discovered process model (figure 2) describes the behavior observed in the event log (table 2). The generation of the model is done by a discovery algorithm (e.g. [9,22]), and its representation by BPMN modeling language.

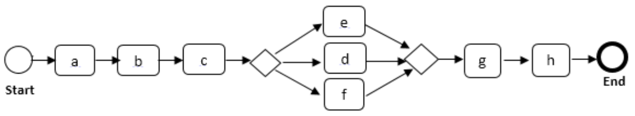


Figure 2: The process model corresponding to the event log in table 2

Event logs can be stored and exchanged using different forms of data source. MXML (Mining eXtensible Markup Language) is a standard notation for storing process attributes such as timestamps, resources and transaction types [23]. XES (eXtensible Event Stream) [24] is the MXML successor created to extend MXML.

4.3. Event log pre-processing

Logs are widely available in many applications, but the purpose of their creation and their level of details varies. To construct a configurable process model with meaningful behavioral patterns, the event logs must be pre-processed before using mining techniques.

- The elimination of confidential data is required before any data processing.
- The balance in the level of details in event logs is recommended. The generated process model has not to be highly detailed.
- The use of the same ontological concept for different sources of event logs.

5. Configurable process discovery approach

The construction of the process model can be done by merging models of process variants, which is complicated and error-prone especially when the number of variants is quite high. The approach we propose builds configurable process model from event log without merging exiting process models. It is based on event logs because of several reasons:

- Event logs are commonly available in Process Aware Information Systems (PAIS), such as : ERP, CRM and workflow management systems
- Business process models do not always exist. Therefore, techniques of merging cannot be applied.
- Event logs record process execution data exactly as it was executed in reality. The information recorded in the event logs is very useful for the business process design or configuration. For example, a priori process model does not present information like activity execution frequency, execution errors, and social behavior between users or services.

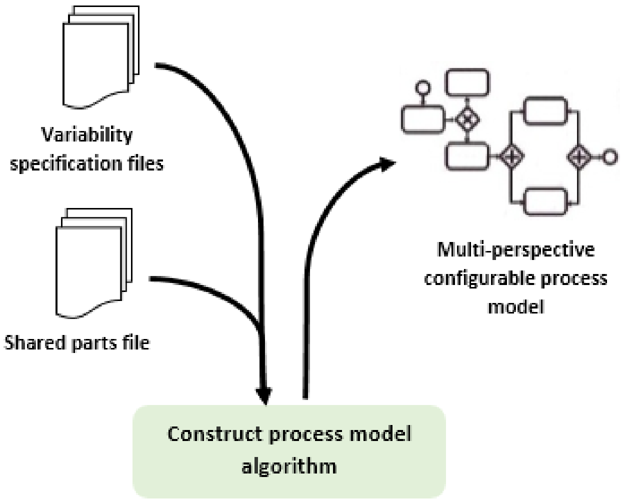

Variability specification files
Shared parts file
Multi-perspective configurable process model
Construct process model algorithm

Figure 3: Construct configurable process model approach

In our approach (Figure 3), we discover configurable process model from event logs using two important files created directly from a collection of event logs: variability specification file and specification file of shared parts. The purpose behind the generation of these variability specification files is firstly to discover, in detailed manner, the variability of activity and resource and secondly to keep a record of variability for any forward configuration or update of the business process. The information recorded in the variability specification file represent variation points and variants of the configurable process while the specification file of shared parts represents the non-variable parts of the configurable process.

5.1. Framework architecture

The contribution presented in [11] proposes a variability discovery approach for business processes taking into account the variability of resource perspective. In this paper, we shed light on our framework and its components. As well as, we describe the role of each component and we explain the interdependence between components. The discovery approach comes after, to create the configurable process model directly from event logs without using merging techniques.

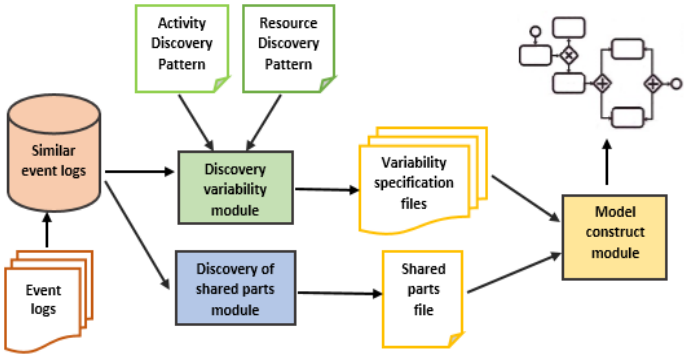

Activity Discovery Pattern
Resource Discovery Pattern
Similar event logs
Discovery variability module
Variability specification files
Model construct module
Event logs
Discovery of shared parts module
Shared parts file

Figure 4: Architecture of the proposed framework [11]

The figure 4 presents the four components of the proposed framework:

- **Similar event logs:** It is a storage module of similar event logs collected and sorted by using existing techniques of clustering [9, 25, 26]. It takes as input a set of event logs and generates a set of pre-processed event logs.
- **Discovery variability module:** this module is important to discover the variability of different process perspectives namely activity, resource and data. It takes as input algorithms for activity and resource variability discovery and generates variability specification files for both activity and resource.
- **Discovery of shared parts module:** this module discovers the common parts shared by all process variants. It takes as input a set of similar event logs and generates specification file of shared parts of the process.
- **Model construction module:** this module constructs the configurable process model using the specification files of variability and shared parts.

The variability discovery module allows discovering the configurable fragments of the business process. It uses two algorithms to discover variability elements in detail. The first discovers the variability of activities [11] and the second discovers the variability of resources [23]. As output, the module generates i) a variability specification file, with variation points, variants, and variation point types for both activity and resource.

The shared parts discovery module discovers the common activities between all process variants. It generates ii) a specification file of common parts. This module allows discovering the non-configurable fragments of the business process.

Both variability discovery module and shared parts discovery module are essential for building the configurable model. The algorithm for configurable model creation uses i) and ii) for the configurable process model construction.

5.2. Discovery of variability

The discovery algorithm that we have introduced in [11] aims to provide an explicit variability discovery of activities. It discovers elements of variability such as variation points, type of variation point and variants. The algorithm uses discovery rules for each variability element [11]. These rules are used to define relations between activities and to construct the control flow process model.

Algorithm 1 : discovery of activity variability

```
Begin
Select the activity column
//Step 1: Discovery of variation point
For each activity do //From all activities of the process
For the direct activity successor do
Select successor1
// application of variation point discovery rule VP_rule
For each other_successor of the selected activity do
  If successor1≠other_successor then
        a variation point exists
        save successor1
        save other_successor
  End if
//Step 2: discover of variants; application of var_rule
  For each successor do
   Define a variant
//Step3:name variation point as successor name
Name variation point=concatenation of all its
successors+vp as prefix
//Step4:define variation point type by application of
TVP_Parallel rule
   If (successor1 is followed by other_successor)
    and (other_successor is followed by successor1)
   Then the point of variation is optional
    End if
   // application of TVP_Choice rule
   If (successor1 is not followed by other_successor)
   and (other_successor is not followed by successor1)
   Then the point of variation is alternative
   End if
    End for
End for
End for
End
```

The discovery algorithm proceeds in different steps:

The algorithm discovers variation points based on the definition of activity successors from event logs. Then, it discovers the variants of variation points by defining the different successors identified in the previous step. After that, we concatenate the variants names to create the variation point name with add of the prefix “vp”. Finally, the algorithm discovers the type of variation point. For that, the log-based ordering relations [22] are used to discover relations between activities (choice or parallel).

The application of the discovery algorithm for activity variability generates variability specification file.

Definition 3: (variability specification file). Let A be a set of all activities of an event log L and $a \in A$. Let VP, V be a two sets of finite activities, with $VP \subset A$, $V \subset A$ and $VP \sqcup V \subseteq A$

Let o, a be the optional and alternative type of variation point as defined in [11], and let T= {t_o, t_a} be the type given to each variation point, respectively. Configurable fragment is defined as:

$\mathrm{ConfigFrag}(VP,T,V)= (a \in VP, t\in T , (a_1,...a_n) \in V \ \ n\in IN)$

Variability specification file is a set of configurable fragments. It is defined as: $\sum \mathrm{ConfigFrag}(VP,T,V)$

This file contains details about variability elements of control-flow perspective. Different algorithms in our approach use this file for the discovery of resource variability and shared parts of the process variants.

5.3. Discovery of shared parts

To define shared parts of the process we implement algorithm based on the theory of sets in mathematics. It is defined as follow:

Let ET, EV and EC to be three sets of process activities. Where ET is the set of all process activities present in the event logs, EV is the set of activities present in the specification file of variability and EC the set common activities, we have: $\{E\,T = EV \sqcup EC \} \Leftrightarrow \{EV \subset ET , EC \subset ET , EV\cap EC =\emptyset\} \Leftrightarrow\{EC = ET\backslash EV\}$

The algorithm uses the variability specification files and similar event logs. Which means that, the similarity between two activities name is maximum between their syntactic similarity and their linguistic similarity.

Algorithm 2 : Discovery of shared activities

```
init E(C) = ∅
begin
    for each x ∈ E(T) do
        if x not element of E(V) then
        write x in E(C)
        else continue
        end if
    end for
  return E(C)
end
```

The algorithm parses the variability specification file of activities to discover the activities in common with all process variants, and then generates, as output, a specification file of common activities.

5.4. Creation of configurable process model

The discovery approach we propose constructs a configurable process model from event logs using the two files generated by the presented algorithm. The algorithm extracts the configurable fragment of the process model from the variability specification file of activities and extract the non-configurable fragment of the process model from the specification file of shared activities.

Algorithm 3 : Discovery of configurable model

```
Begin
Init Graph G
While E(C) ≠ ∅ do
    for x ∈ E(C) do
        if x ∈ E(V) then
        G← create node x
        define type of connector node
        G ←create children nodes of  x successors
        else
         G ←create node
        end if
    end for
  return graph G
end while
end
```

The algorithm parses the two generated specification files to construct the configurable model. It starts by selecting an element of shared parts and parses the variability specification file. If the element exists in the variability specification file, then the current element is a variation point, and the algorithm creates the corresponding node and its variants. If not, the activity is common of all process variants and the node is created.

6. Conclusion & future work

Many approaches were interested in control flow discovery, but few ones have been proposed for explicit variability discovery of different process perspectives. Given the importance of the variability for process reuse in business process management, the aim of our work is to propose a discovery approach of configurable process model from event logs, which provides a detailed information about variability elements in business process. In addition, the approach proposes a discovery algorithm for variability of other perspectives like activities and resources.

In the future work, we intend to work on the implementation of our approach with test results performed to evaluate its feasibility through different experiments. In addition, we will show the practical usefulness of our approach and publish a paper about the integration of resource variability in the discovered configurable process model.

References

[1] W.M.P. van der Aalst, Process mining: data science in action, 2nd edition Springer, 2016.

[2] M. Weske, Business Process process Management management concepts, languages, architectures. 2nd edition, Springer, 2005.

[3] M. Dumas, WM.P. Aalst, A.H.M. Hofstede, Process-Aware Information Systems Bridging People and software through process technology. John Wiley & sons, 2005.

[4] J. Becker, M. Kugeler, M. Rosemann, Process management: a guide for the design of business processes, Springer Berlin, Heidelberg, New York, 2003.

[5] Wil M. P. van der Aalst. "Workflow verification: Finding control-flow errors using petri-net-based techniques". In *Business Process Management, Models, Techniques, and Empirical Studies*, Springer-Verlag, 1806, 161-183, London, UK, 2000.

[6] R. Rychkova, S. Nurcan, "Towards adaptability and control for knowledge-intensive business processes: Declarative configurable process specifications", 1-10, IEEE Computer Society, 2011 https://doi.org/10.1109/HICSS.2011.452

[7] L. Seinturier, P. Merle, R Rouvoy, D. Romero, V. Schiavoni, J. B. Stefani. "A component-based middleware", 42(5), 559-583, May 2012.

[8] J.C.A.M. Buijs, B.F. van Dongen, W.M.P. van der Aalst., "Mining configurable process models from collections of event logs". In: Daniel F., Wang J., Weber B. (Ed.), Business Process Management. Lecture Notes in Computer Science, 8094, 33-48, Springer, Berlin, Heidelberg, 2013. https://doi.org/10.1007/978-3-642-40176-3_5

[9] Y.P.J.M. Van Oirschot, Using trace clustering for configurable process discovery explained by event log data, master thesis, University of technology Eindhoven, Netherlands, 2014.

[10] N. Assy, Automated support for configurable process models, PhD thesis, University Paris-SACLAY, France, 2015.

[11] R. Sikal, H. Sbai, L.Kjiri, "Configurable process mining: variability Discovery Approach", the 5th International Congress on Information Science and Technology,(IEEE CIST'18), Morocco, October 2018. https://doi.org/10.1109/CIST.2018.8596526

[12] R. Sikal, H. Sbai, L. Kjiri, "Promoting resource discovery in business process variability", proceedings of the 2nd international conference on Networking, Information System & Security, 48, 1-7. 2019. https://doi.org/10.1145/3320326.3320380

[13] F. Milani, M. Dumas, N. Ahmed, R. Matulevičius, "Decomposition driven consolidation of process models". In proceedings of the International Conference on Advanced Information Systems Engineering (CAiSE), 7908, 193-207. Springer, Berlin, Heidelberg, 2013.

[14] F. Gottshalk, W.M.P. van der Aalst, M.H. Jansen-Vullers, "Mining reference process models and their configurations". In R. Meersman, Z. Tari, P. Herrero (Ed.), On the Move to Meaningful Internet Systems: OTM 2008 Workshops, 5333, 263-272. Springer berlin Heidelberg, 2008.

[15] M. Rosamann, W.M.P. Van der Aalst, "A configurable reference modeling language". Information Systems, 32(1), 1-23, 2007. https://doi.org/10.1016/j.is.2005.05.003

[16] F. Gottshalk, W. M. P. van der Aalst, M. Jansen-Vullers, M. La Rosa, "Configurable Workflow Models", International Journal of Cooperative Information Systems, 17(2), 177-221. https://doi.org/10.1142/S0218843008001798

[17] H. Sbai, M. Fredj, L. Kjiri, "A pattern based methodology for evolution management in business process reuse". International Journal of Computer Science Issues, 11(1), 1-10, 2014.

[18] H. Sbai, L. El Faquih, M. Fredj, "A novel tool for configurable process evolution and service derivation". International Journal of Enterprise Information Systems (IJEIS), 15(2), 1-18, 2019.

[19] F. Gottshalk, W.M. P. van der Aalst, M. H. Jansen-Vullers, "Merging event-driven process chains". In R. Meersman, Z. Tari, P. Herrero (Ed.), On the Move to Meaningful Internet Systems: OTM 2008 Workshops, 5331, 418-426. Springer, berlin, Heidelberg, 2008. https://doi.org/10.1007/978-3-540-88871-0_28

[20] W. Derguech, S. Bhiri, E. Curry, "Designing business capability-aware configurable process models". Information Systems, 72, 77-94, 2017. https://doi.org/10.1016/j.is.2017.10.001

[21] M. Arriagada-Benítez, M. Sepúlveda, J. Munoz-Gama, J.C. A. M. Buijs, "Strategies to automatically derive a process model from a configurable process model based on event data". Applied sciences, 2017. https://doi.org/10.3390/app7101023

[22] W.M.P. van der Aalst, T. Weijters, L. Maruster. Workflow mining: Discovering process models from event logs. *16 (9).* IEEE Transactions on Knowledge and Data Engineering, 2004. https://doi.org/10.1109/TKDE.2004.47

[23] B. F. Van Dongen. "A meta model for process mining data". In *Proceedings of the CAiSE WORKSHOPS*, 309-320, 2005.

[24] H.M.W. Verbeek, J.C.A.M. Buijs, B.F. van Dongen, W.M.P. van der Aalst. "XES, XESame, and ProM 6". In P. Soffer and E. Proper, editors, *Information Systems Evolution*, vol.72 of *Lecture Notes in Business Information Processing*, 60-75. Springer-Verlag, Berlin, 2010.

[25] M. De Leoni, W.M.P. van der Aalst, "A general process mining framework for correlating, predicting and clustering dynamic behavior based on event logs". In Information Systems, 56, 235-257. Oxford, UK, 2015. https://doi.org/10.1016/j.is.2015.07.003

[26] A. Appice, D. Malerba, A co-training strategy for multiple view clustering in process mining. IEEE Transaction on Services Computing, 9, 832-845, 2016. https://doi.org/10.1109/TSC.2015.2430327