Science Arts & Métiers (SAM)
is an open access repository that collects the work of Arts et Métiers Institute of Technology researchers and makes it freely available over the web where possible.

This is an author-deposited version published in: https://sam.ensam.eu
Handle ID: .http://hdl.handle.net/10985/12514

To cite this version :

Widad ES-SOUFI, Lionel ROUCOULES, Esma YAHIA - A Process Mining Based Approach to Support Decision Making - In: IFIP 14th International Conference on Product Lifecycle Management, Espagne, 2017-07 - Product Lifecycle Management for Digital Transformation of Industries - 2017

Any correspondence concerning this service should be sent to the repository Administrator : scienceouverte@ensam.eu

A Process Mining Based Approach to Support Decision Making

Widad Es-Soufi[1], Esma Yahia[1], Lionel Roucoules[1]

[1]Arts et Métiers ParisTech, CNRS, LSIS, 2 cours des Arts et Métiers 13617 Aix en Provence, France
{Widad.ES-SOUFI, Esma.YAHIA, Lionel.ROUCOULES}@ensam.eu

Abstract. Currently, organizations tend to reuse their past knowledge to make good decisions quickly and effectively and thus, to improve their business processes performance in terms of time, quality, efficiency, etc. Process mining techniques allow organizations to achieve this objective through process discovery. This paper develops a semi-automated approach that supports decision making by discovering decision rules from the past process executions. It identifies a ranking of the process patterns that satisfy the discovered decision rules and which are the most likely to be executed by a given user in a given context. The approach is applied on a supervision process of the gas network exploitation.

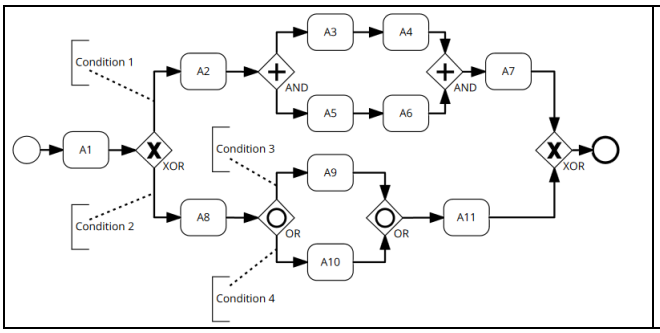
Keywords: Process mining, Decision mining, Process patterns, Decision-making, Business process.

1 Introduction

Business process is defined as a set of activities that take one or more inputs and produce a valuable output that satisfies the customer [1]. In [2], authors define it as a set of activities that are performed in coordination in an organizational and technical environment and provide an output that responds to a business goal. Based on these definitions, the authors of this paper describe the business process, as a set of linked activities that have zero or more inputs, one or more resources and create a high added value output (i.e. product or service) that satisfies the industrial and customers constraints. These linked activities represent the business process flow and are controlled by different process gateways (And, Or, Xor) [3, Sec. 8.3.9] that give rise to several patterns (pattern 1 to 9 in Fig. 1) where each one is a linear end-to-end execution. The "And" gateway, also called parallel gateway, means that all the following activities are going to be executed in several possible orders. The "Or" gateway, also called inclusive gateway, means that one or all the following activities are going to be executed based on some attributes conditions. The "Xor" gateway, also called exclusive gateway, means that only one following activity among others, is going to be executed based on some attributes conditions.

The presence of gateways in business processes results in making several decisions based on some criteria like experience, preference, or industrial constraints [4].

Making the right decisions in business processes is tightly related to business success. Indeed, a research that involved more than a thousand companies, shows an evident correlation between decision effectiveness and business performance [5].



Process patterns:
1- $A_1A_2A_3A_4A_5A_6A_7$
2- $A_1A_2A_3A_5A_4A_6A_7$
3- $A_1A_2A_3A_5A_6A_4A_7$
4- $A_1A_2A_5A_6A_3A_4A_7$
5- $A_1A_2A_5A_3A_6A_4A_7$
6- $A_1A_2A_5A_3A_4A_6A_7$
7- $A_1A_8A_9A_{11}$
8- $A_1A_8A_{10}A_{11}$
9- $A_1A_8A_9A_{10}A_{11}$

Fig. 1. Example of process patterns (expressed in BPMN[1])

In [6], authors explain that the process of decision-making can be broken down into two sub processes: The global and the local decision making. In this research, authors focus on the global decision making and aim at developing a generic approach that assists engineers in managing the business process associated with the life of their products or services. The approach automatically proposes a predicted ranking of the business process patterns, that are the most likely to be executed by a given user in a given context. This comes down to exploring these patterns and the decisions that control them in a complex business process, i.e. where all gateways are present (Fig. 1). Authors assume that this objective can be achieved using process mining techniques.

This paper is organized as follows. In Section 2, a literature review on decision and trace variants mining are discussed. The proposed approach is presented in Section 3 and then illustrated in a case study in Section 4. Finally, the discussion of future work concludes the paper.

2 Literature Review on Decision and Trace Variants Mining

Process mining is a research field that supports process understanding and improvement, it helps to automatically extract the hidden useful knowledge from the recorded event logs generated by information systems. Three types of applications in process mining are distinguished: *discovery*, *conformance*, and *enhancement* [7]. In this paper, authors focus on the *discovery* application, namely, the *decision mining* and the *trace variants mining*. A brief summary is provided of each.

Decision mining is a data-aware form of the process discovery application since it enriches process models with meaningful data. It aims at capturing the decision rules that control how the business processes flow (e.g. conditions 1,2,3,4 in Fig. 1). In [8], authors define it as the process in which data dependencies, that affect the routing of each activity in the business process, are detected. It analyses the data flow to find the

[1] http://www.bpmn.org/

rules that explain the rationale behind selecting an activity among others when the process flow splits [9].

While executing a business process, one may adopt the same logic several times (e.g. always executing pattern 1 in Fig. 1, rather than patterns 2 to 6, if condition 1 is enabled). This results in the existence of similar traces in the recorded event log. *Trace variants mining* aims at identifying the trace variants and their duplicates (e.g. patterns 1 to 9 in Fig. 1). Each trace variant refers to a process pattern that is a linear end-to-end process execution where only the activities execution order is taken into account [6].

2.1. Decision Mining

The starting point of the most common decision mining techniques is a recorded event log (i.e. past executions traces) and its corresponding petri net[2] model that describes the concurrency and synchronisation of the traces activities. To automatically generate a petri net model from an event log, different algorithms were proposed. The *alpha algorithm, alpha++ algorithm, ILP miner, genetic miner*, among others, are presented in [10], and the *inductive visual miner* that was recently proposed in [11].

Many research works contribute to decision mining development. In [8], authors propose an algorithm, called *Decision point analysis*, which allows one to detect decision points that depict choice splits within a process. Then for each decision point, an exclusive decision rule (Xor rule) in the form "*v op c*", where "*v*" is a variable, "*op*" is a comparison operator and "*c*" is a constant, allowing one activity among others to be executed is detected. The *decision point analysis* is implemented as a plug-in for the ProM[3] framework. In [12], authors propose a technique that improves the *decision point analysis* by allowing one to discover complex decision rules for the Xor gateway, based on invariants discovery, that takes into account more than one variable, i.e. in the form "v_1 *op c*" or "v_1 *op* v_2", where v_1 and v_2 are variables. This technique is implemented as a tool named *Branch Miner*[4]. In [13], authors propose a technique that embeds decision rules into process models by transforming the Xor gateway into a rule-based Xor gateway that automatically determines the optimal alternative in terms of performance (cost, time) during runtime. This technique is still not yet implemented. In [14], authors propose an approach to derive decision models from process models using enhanced decision mining. The decision rules are discovered using the *decision point analysis* algorithm [8], and then enhanced by taking into account the predictions of process performance measures (time, risk score) related to different decision outcomes. This approach is not yet implemented. In [15], authors propose a method that extends the *Decision point analysis* [8] which allows only single values to be analysed. The proposed method takes into account time series data (i.e. sequence of data points listed in time order) and allows one to generate complex decision rules with more than one variable. The method is implemented but not publicly shared. In [16], authors propose a process mining based technique that allows one to identify the most performant

[2] https://en.wikipedia.org/wiki/Petri_net
[3] http://www.promtools.org/
[4] http://sep.cs.ut.ee/Main/BranchMiner

process path by mining decision rules based on the relationships between the context (i.e. situation in which the past decisions have taken place), path decisions and process performance (i.e. time, cost, quality). The approach is not yet implemented.

In [9], authors introduce a technique that takes the process petri net model, the process past executions log and the alignment result (indicating whether the model and the log conform to each other) as inputs, and produces a petri net model with the discovered inclusive/exclusive decision rules. It is implemented as a *data flow discovery* plug-in for the ProM framework. Another variant of this plug-in that needs only the event log and the related petri net as inputs is implemented as well. In [17], authors propose a technique that aims at discovering inclusive/exclusive decision rules even if they overlap due to incomplete process execution data. This technique is implemented in the *multi-perspective explorer* plug-in [18] of the ProM framework. In [19], authors propose an approach to explore inclusive decision rules using the *Decision point analysis* [8]. The approach consists in manually modifying the petri net model by transforming the "Or" gateway into an "And" gateway followed by a "Xor" gateway in each of its outgoing arcs.

2.2. Trace Variants Mining

Different researches were interested in trace variants mining. In [20], authors propose an approach based on trace clustering, that groups the similar traces into homogeneous subsets based on several perspectives. In [21], authors propose a *Pattern abstraction* plug-in, developed in ProM, that allows one to explore the common low-level patterns of execution, in an event log. These low-level patterns can be merged to generate the process most frequent patterns which can be exported in one single CSV file. The *Explore Event Log (Trace Variants/Searchable/Sortable)* visualizer[5], developed in ProM, sorts the different trace variants as well as the number and names of duplicate traces. These variants can be exported in separate CSV files, where each file contains the trace variant, i.e. process pattern, as well as the related duplicate traces.

2.3. Discussion

In this paper, authors attempt to discover the decision rules related to both exclusive (Xor) and inclusive (Or) gateways, as well as the different activities execution order. Regarding decision mining, the algorithm that generates the petri net model should be selected first. Authors reject the algorithms presented in [10] and select the *inductive visual miner* [11] as the petri net model generator. Indeed, experience has shown that only the *inductive visual miner* allows the inclusive gateways to be identified by the decision mining algorithm. This latter should afterward be selected.

The research works presented in [8], [12]–[16] attempt to discover exclusive decision rules considering only the exclusive (Xor) gateway. The work presented in [19] considers the inclusive and exclusive decision rules discovery, but the technique needs a manual modification of the petri net model which is not practical when

[5] https://fmannhardt.de/blog/process-mining-tools

dealing with complex processes. Therefore, authors assume that these works are not relevant for the proposition and consider the works presented in [9] and [17] which allow the discovery of both inclusive and exclusive decision rules. Moreover, authors assume that the *data flow discovery* plug-in [9] is more relevant since the experience has shown that the other one [17] could not correctly explore the decision rule related to the variables whose values do not frequently change in the event log.

Regarding *trace variants mining*, authors do not consider the approach presented in [20] as relevant for the proposition since the objective is to discover the patterns that are exactly similar, i.e. patterns with the same activities that are performed in the same order. The work presented in [21] and the *Explore Event Log* visualizer are considered as relevant for the proposition. Since none of the proposed techniques allow one to export a CSV file that contains only the trace variants and their frequency, authors assume that exploring trace variants using the *Explore Event Log* visualizer is more relevant because the discovered patterns can be exported in separate CSV files, which facilitates the postprocessing that needs to be made.

3 Decision and Trace Variants Mining Based Approach

The approach presented in Fig. 2 is the global workflow of the proposal and enables the achievement of the current research objective through seven steps.

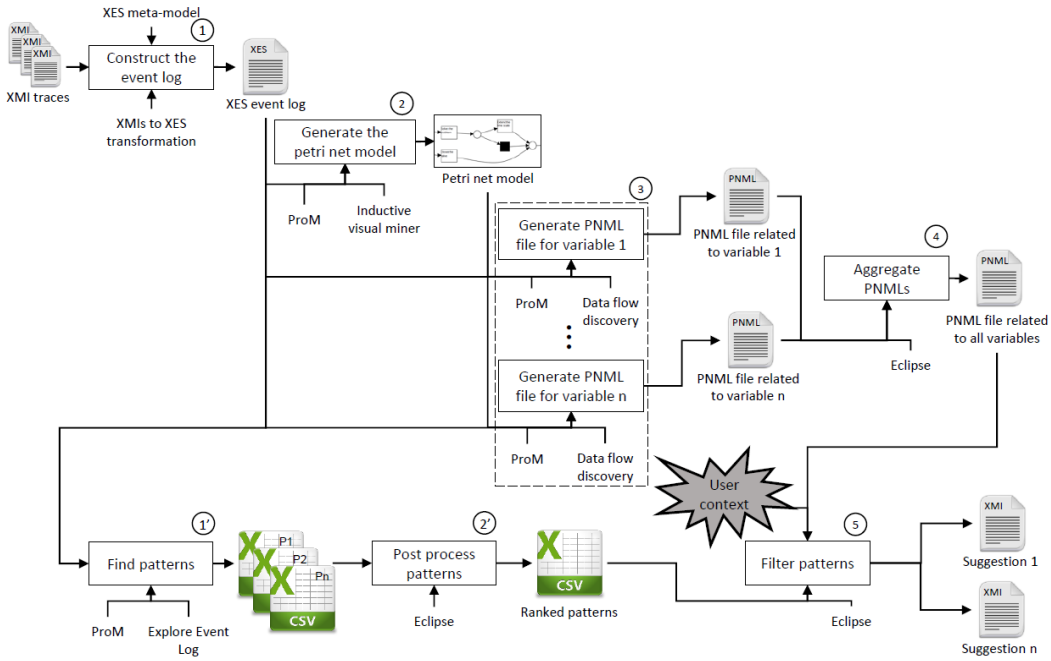


Fig. 2. Overview of the proposal (expressed in IDEF0[6])

[6] https://en.wikipedia.org/wiki/IDEF0

The first step of the approach concerns the construction of the event log from the past process executions. These latter represent the process traces generated with respect to the trace metamodel depicted in [22,23] and expressed in XMI (XML Metadata Interchange) format. These traces should be automatically merged into a single XES[7] (eXtensible Event Stream) event log in order to be processed in ProM, the framework in which the selected decision mining technique is developed. This automatic merge is implemented using ATL[8] (Atlas Transformation Language).

The second step concerns the generation of the petri net model from the event log. To this end, the *inductive visual miner* is used. Having both the event log and its corresponding petri net model, the decision mining practically starts using the *data flow discovery* plug-in as discussed in Section 2.

The third step aims at deriving the decision rules related to all the variables in the event log and exporting them in a single PNML[9] (Petri Net Markup Language) file. PNML is a XML based standardized interchange format for Petri nets that allows the decision rules to be expressed as guards, this means that the transition from a place (i.e. activity) to another can fire only if the guard, and thus the rule, evaluates to true. For instance, condition 1 in Fig. 1 is the decision rule that enables the transition from A_1 to A_2. The experience has shown that when all the variables in the event log are considered in the decision mining, some decision rules related to some of these variables may not be derived as expected, the origin of this problem is not yet clear. Therefore, to avoid this situation and be sure to have a correct decision rule, authors propose to execute the *data flow discovery* plug-in for each variable, this results in as much decision rules as variables (step 3 in Fig. 2).

The PNML files, that are generated in step 3, should be automatically merged into one single PNML file that contains the complete decision rules, i.e. related to all the event log's variables (step 4 in Fig. 2). This automatic merge is implemented using the Java programming language.

In parallel with decision mining (finding the Or and XOR rules), the trace variants mining can be performed in order to find the end-to-end processes (e.g. patterns 1 to 9 in Fig. 1). The *Explore Event Log* visualizer, as discussed in Section 2, is used to explore patterns in an event log. The detected patterns are then exported in CSV files where each file contains one pattern and its duplicates (step 1' of Fig. 2). To fit our objective, the patterns files need to be automatically post processed. This consists in computing the occurrence frequency of each pattern and removing its duplicates and then creating a file that contains a ranking of the different, non-duplicate, patterns based on their occurrence frequency (step 2' in Fig. 2). This post processing is implemented using the Java programming language.

During a new process execution, the ranked patterns file is automatically filtered to fit both the discovered decision rules and the user's context (user's name, date, process type, etc.). In other words, the patterns that do not satisfy the decision rules and those that are, for example, performed by another user than the one that is currently performing the process are removed. As a result, a ranking of suggestions (i.e. patterns that are the most suitable for the current user's context) are proposed to

[7] http://www.xes-standard.org/
[8] https://en.wikipedia.org/wiki/ATLAS_Transformation_Language
[9] http://www.pnml.org/

the user (step 5 in Fig. 2). The selected pattern is, then, captured and stored in order to enrich the event log.

4 Case Study: Supervision of Gas Network Exploitation

Systems supervision is a decision-based activity carried out by a supervisor to survey the progress of an industrial process. It is a business process that produces an action, depending on both the supervision result and the set-point (i.e. target value for the supervised system), that resolves systems malfunction. The authors of this paper present a supervision case study where the supervisor of an industrial process should take, in the shortest time, the right decision in case an alarm is received. The challenge here is to provide this supervisor with a ranking of the process patterns that are the most likely to be executed in his context. The proposed approach is verified under a specific supervision process related to gas network exploitation.

The process starts by receiving the malfunction alarm. The Chief Operating Officer (COO) has, then, to choose the process that best resolves the problem in this context. This latter can be described by the field sensors values, season, supervisor's name, etc. The first step of the proposed approach is to transform the already captured sixty traces of this supervision process into a single XES event log (step 1 in Fig. 2) and then generate its corresponding petri net model (step 2 in Fig. 2). Then, from the event log and the petri net model, generate the decision rules for each variable and export them in PNML files (step 3 in Fig. 2). In this process, the decision variables are: Pressure, season, network status, flow rate, human resource (the decision rule related to the pressure variable is depicted in Fig. 3). These PNML files are then merged into one single PNML file that contains the complete decision rules related to all the decision variables (step 4 in Fig. 2).

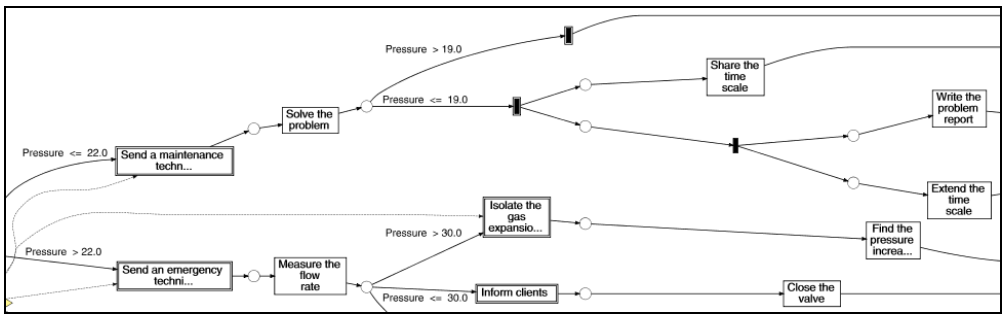
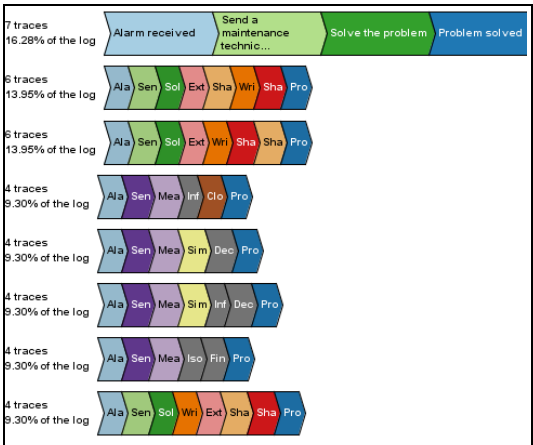


Fig. 3. Discovered decision rules for the pressure variable

In this process, based on both pressure value and season, the COO decides whether to send an emergency or a maintenance technician. If the emergency technician is sent (i.e. the decision rule: *((pressure>22millibars)and(season≠fall))* evaluates to true), he has then to decide which action should be performed based on the measured flow rate. If the decision rule *((pressure≤22millibars)and(season=fall))* evaluates to true, then the maintenance technician is sent. Moreover, if the rule *(pressure<19millibars)* evaluates to true, then in addition to sending the maintenance technician, the

supervisor should extend the time scale then share it and write the problem then share it. In this last case, the inclusive logic is transformed into a parallel logic, and thus the activities may be executed in different possible orders.

In parallel with the decision rules mining, the step 1' and 2' in Fig. 2 are performed; the patterns that are contained in the event log are discovered (Fig. 4.a) then exported in CSV files and finally post processed by removing each pattern's duplicates and computing their occurrence frequency. If we consider all the possible process patterns and the different rules, it is possible to construct the BPMN process depicted in Fig. 5.



(a)

		Variables			
case	event	Human resource	Season	Pressure	Frequency
P12	Alarm received	John	Fall	14	6
	Send a maintenance technician	John	Fall	14	
	Solve the problem	John	Fall	14	
	Extend the time scale	John	Fall	14	
	Write the problem report	John	Fall	14	
	Share the problem report	John	Fall	14	
	Share the time scale	John	Fall	14	
	Problem solved	John	Fall	14	
case	event	Human resource	Season	Pressure	Frequency
P13	Alarm received	John	Fall	10	4
	Send a maintenance technician	John	Fall	10	
	Solve the problem	John	Fall	10	
	Extend the time scale	John	Fall	10	
	Write the problem report	John	Fall	10	
	Share the time scale	John	Fall	10	
	Share the problem report	John	Fall	10	
	Problem solved	John	Fall	10	

(b)

Fig. 4. (a) Discovered patterns, (b) proposed patterns

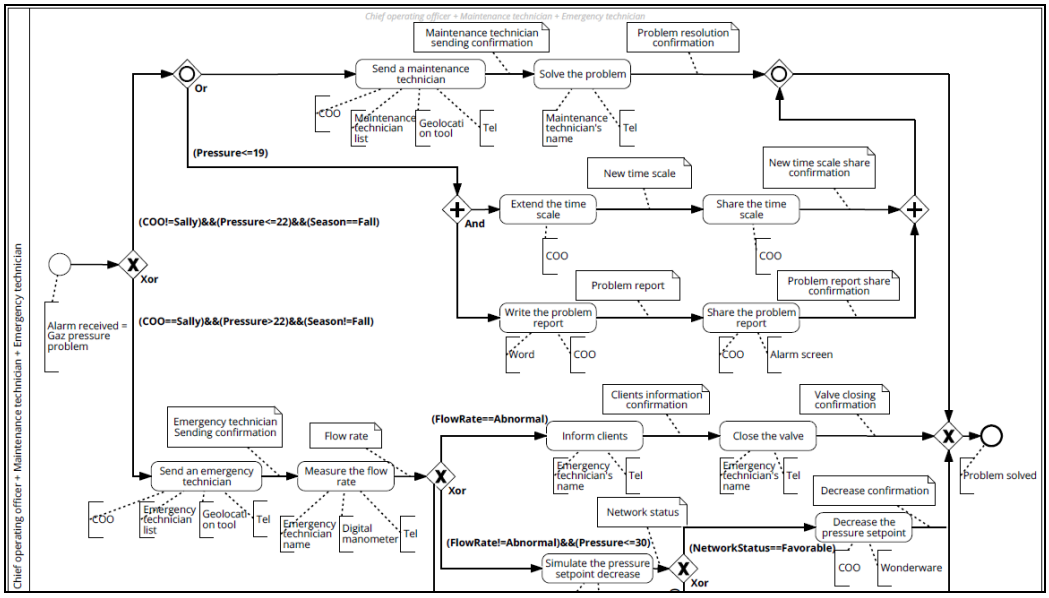


Fig. 5. Part of the resulting supervision process with the different rules (expressed in BPMN)

These patterns (Fig. 4.a) are, then, filtered based on the current context and the decision rules that are generated (step 4 in Fig. 2). For instance, if the alarm is received in the fall by John, and the pressure of the supervised network equals to 18 millibars which is less than both 22 and 19 millibars (Fig. 5), the approach proposes two possible patterns to solve the problem (Fig. 4.b), where the first one "P12" is the most frequently used in this context.

5 Conclusion and Future Work

The objective of this paper is to support engineers in their decision-making processes by proposing the most relevant process patterns to be executed given the context. Through the proposed approach, the past execution traces are first analysed and the decision rules that control the process are mined. Then, the patterns and their occurrence frequency are discovered, postprocessed and filtered based on the discovered decision rules and the user context parameters. A ranking of the most likely patterns to be executed are then proposed. This approach illustrates the feasibility of the assumption about using process mining techniques to support decision making in complex processes that are controlled by inclusive, exclusive and parallel gateways. Future work consists in fully automating the approach and integrating it in the process visualizer tool presented in [22]. It also consists in evaluating this approach, using real-world design and supervision processes, with respect to some performance indicators such as execution time, quality of the proposed decisions, changes propagation, etc.

Acknowledgments. This research takes part of a national collaborative project (Gontrand) that aims at supervising a smart gas grid. Authors would like to thank the companies REGAZ, GDS and GRDF for their collaboration.

References

[1] M. Hammer and J. Champy, *Reengineering the Corporation: A Manifesto for Business Revolution*. Harper Business, 1993.
[2] M. Weske, *Business Process Management: Concepts, Languages, Architectures*. Springer Berlin Heidelberg, 2012.
[3] O. M. G., "Business Process Model and Notation (BPMN) Version 2.0," Jan. 2011.
[4] K. Iino and Y. Hatamura, *Decision-Making in Engineering Design: Theory and Practice*. Springer London, 2006.
[5] M. Blenko, M. Mankins, and P. Rogers, *Decision Insights: The five steps to better decisions*. Bain & Company, 2013.
[6] W. Es-Soufi, E. Yahia, and L. Roucoules, "On the use of Process Mining and Machine Learning to support decision making in systems design," *Product Lifecycle Management for Digital Transformation of Industries,* 2017.
[7] W. Van Der Aalst, "Process mining: overview and opportunities," *ACM Transactions on Management Information Systems (TMIS)*, vol. 3, no. 2, p. 7, 2012.
[8] A. Rozinat and W. M. P. Aalst, *Decision mining in business processes*. BPM Center Report BPM-06-10, BPMcenter.org, 2006.

[9] M. de Leoni and W. M. P. van der Aalst, "Data-aware Process Mining: Discovering Decisions in Processes Using Alignments," in *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, Coimbra, Portugal, 2013, pp. 1454–1461.

[10] B. F. Van Dongen, A. A. De Medeiros, and L. Wen, "Process mining: Overview and outlook of petri net discovery algorithms," in *Transactions on Petri Nets and Other Models of Concurrency II*, Springer, 2009, pp. 225–242.

[11] S. J. Leemans, D. Fahland, and W. M. van der Aalst, "Exploring processes and deviations," in *International Conference on Business Process Management*, 2014, pp. 304–316.

[12] M. De Leoni, M. Dumas, and L. García-Bañuelos, "Discovering branching conditions from business process execution logs," in *International Conference on Fundamental Approaches to Software Engineering*, 2013, pp. 114–129.

[13] S. Catalkaya, D. Knuplesch, C. Chiao, and M. Reichert, "Enriching business process models with decision rules," in *International Conference on Business Process Management*, 2013, pp. 198–211.

[14] E. Bazhenova and M. Weske, "Deriving decision models from process models by enhanced decision mining," in *International Conference on Business Process Management*, 2015, pp. 444–457.

[15] R. Dunkl, S. Rinderle-Ma, W. Grossmann, and K. A. Fröschl, "A method for analyzing time series data in process mining: application and extension of decision point analysis," in *Forum at the Conference on Advanced Information Systems Engineering (CAiSE)*, 2014, pp. 68–84.

[16] J. Ghattas, P. Soffer, and M. Peleg, "Improving business process decision making based on past experience," *Decision Support Systems*, vol. 59, pp. 93–107, 2014.

[17] F. Mannhardt, M. de Leoni, H. A. Reijers, and W. M. van der Aalst, "Decision mining revisited-discovering overlapping rules," in *International Conference on Advanced Information Systems Engineering*, 2016, pp. 377–392.

[18] F. Mannhardt, M. de Leoni, and H. A. Reijers, "The Multi-perspective Process Explorer.," in *BPM (Demos)*, 2015, pp. 130–134.

[19] R. Sarno, P. L. I. Sari, H. Ginardi, D. Sunaryono, and I. Mukhlash, "Decision mining for multi choice workflow patterns," in *2013 International Conference on Computer, Control, Informatics and Its Applications (IC3INA)*, 2013, pp. 337–342.

[20] M. Song, C. W. Günther, and W. M. P. van der Aalst, "Trace Clustering in Process Mining," in *Business Process Management Workshops*, 2009, pp. 109–120.

[21] R. J. C. Bose and W. M. van der Aalst, "Abstractions in process mining: A taxonomy of patterns," in *Business Process Management*, Springer, 2009, pp. 159–175.

[22] L. Roucoules, E. Yahia, W. Es-Soufi, and S. Tichkiewitch, "Engineering design memory for design rationale and change management toward innovation," *CIRP Annals - Manufacturing Technology*, vol. 65, no. 1, pp. 193–196, 2016.

[23] W. Es-Soufi, E. Yahia, and L. Roucoules, "Collaborative Design and Supervision Processes Meta-Model for Rationale Capitalization," in *Advances on Mechanics, Design Engineering and Manufacturing*, 2017, pp. 1123–1130.