On the Use of Process Mining and Machine Learning to Support Decision Making in Systems Design

Widad Es-Soufi(✉), Esma Yahia, and Lionel Roucoules

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. Research on process mining and machine learning techniques has recently received a significant amount of attention by product development and management communities. Indeed, these techniques allow both an automatic process and activity discovery and thus are high added value services that help reusing knowledge to support decision-making. This paper proposes a double layer framework aiming to identify the most significant process patterns to be executed depending on the design context. Simultaneously, it proposes the most significant parameters for each activity of the considered process pattern. The framework is applied on a specific design example and is partially implemented.

Keywords: Collaborative design process · Process mining · Supervised classification · Process patterns · Decision-making

1 Introduction

The product design is a purposeful, constrained and decision making process [1]. Indeed, companies aim at satisfying their clients by proposing innovative products. Thus, a multidisciplinary group of engineers work in collaboration in order to propose the products that best meet the clients' requirements. However, current practises and software do not allow one to document the decisions that were taken as well as the rejected design choices. Hence, engineers lose a considerable amount of time while retrieving the design decisions when changes are occurring [2].

Our research is interested in product design that is a process in which a high added value output (i.e. product) is produced. To solve the problem of decisions retrieval, authors have already proposed in [3, 4] a collaborative design process meta-model to capitalize the design rationale. Authors proposed to gather and archive all the design process executions' traces, that conforms to a trace meta-model, in a trace base. The trace meta-model describes the design process knowledge that needs to be captured and saved for a further analysis. In this paper, authors propose a generic method that couples mining and learning techniques in order to assist engineers in their decision-making processes. Indeed, through a process traces analysis, the most suitable design activities to be executed are identified. Then, for each activity, the most convenient design choices are identified. Authors assume that this objective can be achieved using Process mining

R. Harik et al. (Eds.): PLM 2016, IFIP AICT 492, pp. 56–66, 2016.
DOI: 10.1007/978-3-319-54660-5_6

(PM) and machine learning (ML) techniques. Indeed, PM allows one to explore the design process from past executions by generating the different process patterns based on the occurrence of activities in the trace data base. Whereas, ML techniques allow one to extract knowledge from data that is collected in the trace data base and thus, help predicting future data according to the new design context.

This paper is organized as follows. In Sect. 2, the relevant research tackling the process mining for process discovery, as well as the machine learning for activity parameters prediction are introduced. Related works are presented in Sect. 3. In Sect. 4, the decision-making support technique is introduced and then tested on a case study in Sect. 5. Finally, future work is discussed and the paper is concluded.

2 Literature Review on Process Mining and Machine Learning

PM is a research field that supports process understanding and improvements, it derives from the field of data mining that is fully concentrated on data and thus cannot provide a complete description of the end-to-end process [5]. This discipline helps to automatically extract the hidden useful knowledge from the recorded event logs generated by information systems. In [6], the author distinguishes three types of applications in process mining: *discovery*, *conformance* and *enhancement*. In this present work, authors are more interested in the *discovery* application of PM.

Process Discovery allows one to automatically generate the process model from the event log by analysing the observed behaviour from the recorded events (i.e. trace), and without using any a-priori information out of the event log. Through this application, PM allows one to extract the patterns which frequently occur in a design process. The pattern concept was first introduced in [7] as an entity that describes both a frequent problem and the solution that was considered to resolve it. In the context of software design, different specific patterns among component, composite, etc. were defined in order to describe the frequent solutions that resolve software design problems [8]. In [9], authors reuse these patterns to build product models during the product information system specification. In our context, the pattern concept is slightly different since it refers to a possible end-to-end design process execution that encapsulates the process information (i.e. who did what, when, where, why and how defined in [4]).

ML consists in "*building computer programs able to construct new knowledge or to improve already possessed knowledge by using input information*" [10]. The starting point in machine learning is a data set that consists of a set of data records (also called instance, observation or case). An instance is described by a n-dimensional attribute vector $X = (X_1, X_2, \ldots, X_n)$ and has a target attribute Y called the class or label. Most learning problems fall into one of four categories: supervised, unsupervised, semi supervised or reinforcement learning. A brief summary is provided of each.

Supervised learning: this kind of machine learning is a two steps process. The learning step uses a labelled data set, where the label Y of each attributes vector X is known, to build and evaluate a classification model (i.e. classifier). The Prediction step uses the already constructed classification model to predict class labels for given real-world data

[11]. Supervised learning has two tasks: *supervised classification* where the label Y is a discrete set (True/False), and *regression* where the label Y is a continuous number.

Unsupervised learning: in this type of machine learning, labels in the data set are unknown. The task consists in exploring the data set and identifying data groups by the exploratory analysis and then gathering them in groups called clusters [12].

Semi supervised learning: this technique makes use of both supervised and unsupervised learning and has two tasks. First, the *semi supervised classification* uses a data set that contains both labelled and unlabelled data, the goal is to train a classifier from both of them. Second, the *constrained clustering* uses a training data that consists of unlabelled data as well as a-priori information about clusters such as must-link and cannot-link constraints [13].

Reinforcement learning: this technique combines the field of programming and supervised learning. The goal is to develop learners or software agents that learn from their own experience and from the feedback of environment which may be expressed by a reward or punishment [14].

ML in product design is defined as the learning methods that can be applied to the acquisition of knowledge [15]. In [16], authors introduce a method that allows one to automate the generation of the key design parameters, they use a ML genetic algorithm [17] to select the optimal design among a set of trial designs and thus produce designs with higher performance gains. In [18], authors show that the commercial success of a product depends not only on its functionality but also on its physical appearance that may meet or not the consumer's expectations. Authors use ML techniques to predict the consumer response to any product form. In [19], authors introduce a hybrid algorithm linking ML and search techniques. It uses the capitalized expert knowledge from past optimizations and allows one to select the project scenarios in the early product design phase. In [20], authors aim to adapt CAD models for numerical simulations by simplifying or removing some features of the designed product, this is called the CAD models preparation process. Authors use ML based techniques to predict the features that can be removed or simplified in a new CAD model based on the past expertise knowledge. In [21], authors propose a method that couples the fuzzy theory with the ML techniques to approximate the product design duration as it is constrained by several random factors and is not a forecast problem.

3 Literature Review on Coupling PM and ML

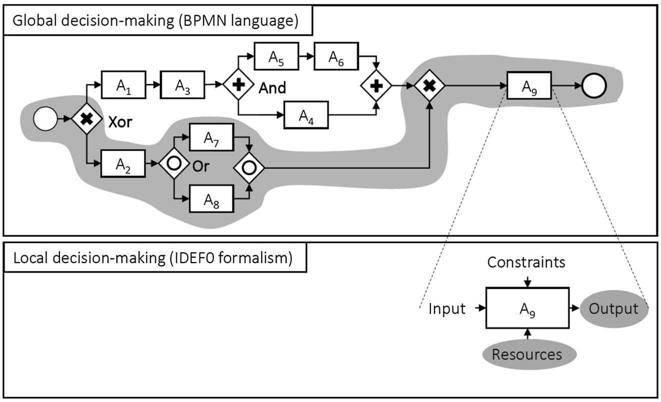
Many research works link both PM and ML techniques. In [22], authors propose a tool that supports collaborative writing of an electronic document. The tool uses ML techniques for extracting document changes as well as PM techniques for extracting event logs, that capture the user's behaviour, and then generating the process model. In [23], authors use the Case Data Extraction mining plugin (PM technique) to extract the case data of the dyeing log, and the association rule mining (ML technique) to find relationships between the extracted data. Our work is closely related to [24], where the authors propose a method for discovering business rules. The PM is first used to analyse the information in event log by identifying the category of users that performs the activity and then ML is used to analyse the context information existing in the event log to discover the set of possible activity parameters' values.

The studied works use ML techniques to extract the knowledge from data and find relationships between them. Afterwards, they use PM techniques to identify the process model from the recorded event log. The work reported in this paper adopts the same logic and, in addition, presents a double process analysis. Indeed, authors start by discovering the process' possible patterns using PM. Then, for each pattern, authors predict the activity possible parameters using the supervised classification (ML technique). The most suitable pattern and activity parameters are then proposed to the decision-maker according to the design context.

4 Double Layer Framework for Decision-Making Support

In the context of product design, the proposal distinguishes two types of decision-making (Fig. 1): The global decision-making (expressed in BPMN [25]) and the local decision-making (expressed in IDEF0[1]).

Global decision-making: when working or reworking on a product design, the engineer has the choice to perform some activities before others. The different gateways (And, Or, Xor) [25, Sect. 8.3.9] are used to control how the process flows and, thus, give rise to several possible executions (Fig. 1). The global decision-making consists in automatically proposing a ranking of the most significant patterns (i.e. set of ordered activities), that match the design context.



Possible Executions

1	A_1	A_3	A_5	A_6	A_4	A_9
2	A_1	A_3	A_4	A_5	A_6	A_9
3	A_1	A_3	A_5	A_4	A_6	A_9
4	A_2	A_7	A_8	A_9		
5	A_2	A_8	A_7	A_9		

Fig. 1. Global and local decision-making (in grey, the decisions that have been taken)

Local decision-making: when dealing with the product design process, two main types of activities are distinguished: modelling and decision activities. On the one hand, the

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

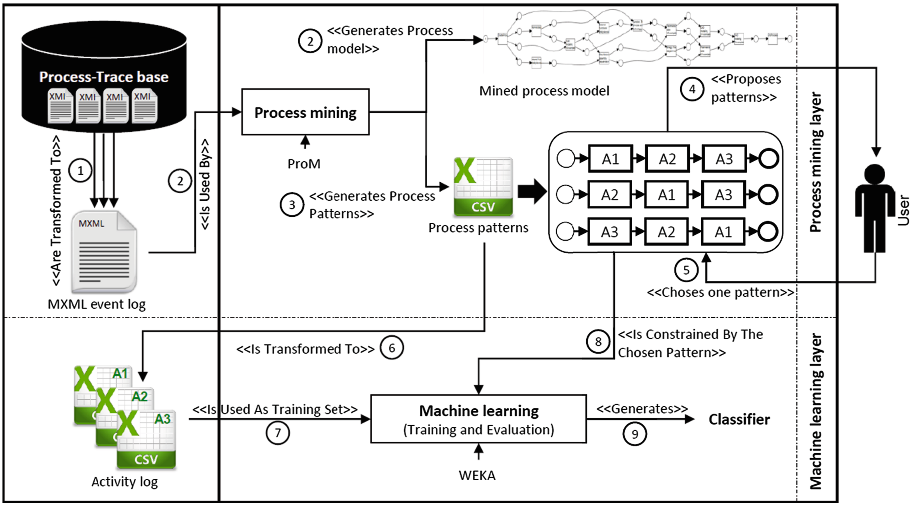


Fig. 2. Overview of the proposal

modelling activity transforms an input into output, by taking in consideration the imposed design constraints, and by using some resources (human, documentary, software, etc.). On the other hand, the *decision activity* consists in choosing one or several solutions among all the design alternatives by taking into account the constraints and using some resources. The local decision-making consists in predicting the most relevant resources for the modelling and decision activities. In addition, it consists in predicting the output (i.e. Decision) in the case of the decision activity (Fig. 1).

The double layer framework shown in Fig. 2 represents the proposal of this research. In the process mining layer, the goal is to extract the most frequent patterns of the product design process. In the machine learning layer, the goal is to predict the most frequent parameters, given the context, for each activity of the selected pattern.

4.1 Process Mining Layer

The starting point of PM is some information about the past executions of design processes in the form of event logs that could be expressed in XES format (eXtensible Event Stream) or MXML format (Mining eXtensible Markup Language). In this paper, authors use MXML as it is an XML-based syntax format and has a meta model [26]. In the process mining layer (Fig. 2), authors start by transforming the process executions' traces contained in the Process-Trace base [3], where each trace is a single past execution of the process and is expressed in XMI format (XML Metadata Interchange). Indeed, the trace is considered as an instance of the trace meta-model proposed in [4], that is implemented in the Eclipse environment[2], and whose instantiation generates a

[2] https://eclipse.org/.

XMI trace. The transformation consists in a translation of all the traces into a MXML event log. This latter stores the information related to several executions of the process, each execution refers to a case (i.e. trace) and contains ordered events where each event refers to an activity of the process [26]. Event logs should conform to the MXML meta model but may store additional information. In our context, an event log contains the basic elements described in the MXML meta model [26] as well as the concepts of the process meta model identified in [4].

There are many mining software tools that help to discover event logs, authors can cite, inter alia, Mylnvenio[3], ProM[4] and Disco[5]. In this present work authors use ProM as it is a free and open source process mining tool that supports the development of PM plugins. ProM allows the *process discovery* by generating process models from the input event log. In addition, it allows the process patterns discovery by identifying the most frequent patterns in a process through the pattern abstraction visualization [27]. Mined patterns can be filtered depending on certain parameters including their apparition frequency in the process and their size (i.e. number of contained activities) and they can be exported in CSV format. Hence, during a new execution of the process, authors propose the mined patterns classified according to their frequency. The user chooses one of them and its choice constrains the machine learning layer.

4.2 Machine Learning Layer

In this layer, authors aim at predicting the resources to be used and/or the decision to be considered for each activity of the chosen process pattern. In this context, the most relevant learning technique, among those presented in Sect. 2, is the supervised classification, since its objective is to learn from known values to predict new ones. There are several machine learning tools that support supervised classification, authors chose Weka[6] since it is a free and open source framework, simple to use and well documented. As it is considered, each time, only one activity of the process pattern while applying the supervised classification algorithms with Weka, a training set for this considered activity is constructed in order to start the supervised classification process (Fig. 3). Thus, the different executions of each activity are extracted from the CSV file of the mined process patterns and saved separately in an activity log in CSV format. The training set is constructed as following: X is the input attributes vector that describes the properties of the activity and Y is the output variable. The training set can be pre-processed in case some attributes of the vector X need to be eliminated [28].

The next step is to select a supervised classification algorithm (i.e. classifier) among naïve Bayes, decision tree, SVM, neural nets, etc. and apply it on the training set. After being trained, the selected algorithms are evaluated by using a testing set, where Y is unknown for each attributes vector X. The most accurate classifier will be then used to

[3] https://www.my-invenio.com/.

[4] http://www.promtools.org/.

[5] https://fluxicon.com/disco/.

[6] http://www.cs.waikato.ac.nz/ml/weka/downloading.html.

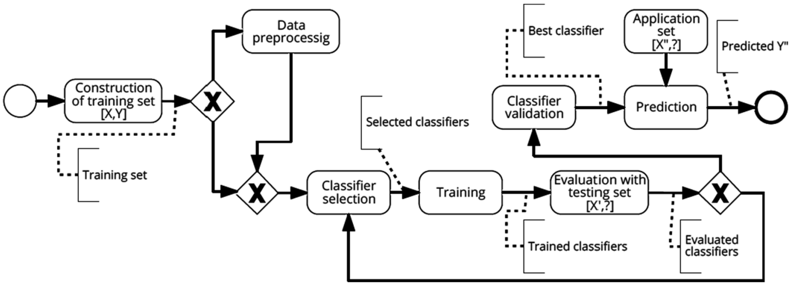


Fig. 3. Process of the supervised classification

predict the desired variable during a new activity execution. The evaluation of supervised classification algorithms is not addressed in this paper since authors assume that the decision tree algorithm is the most suitable in our context.

5 Case Study: Application on an Electric Torch Design Process

The example considered in this study is an electric torch design process (Fig. 4). First, the engineer has the choice to begin the process with one of the fourth activities: "External functional analysis", "Internal functional analysis", "Nomenclature" or

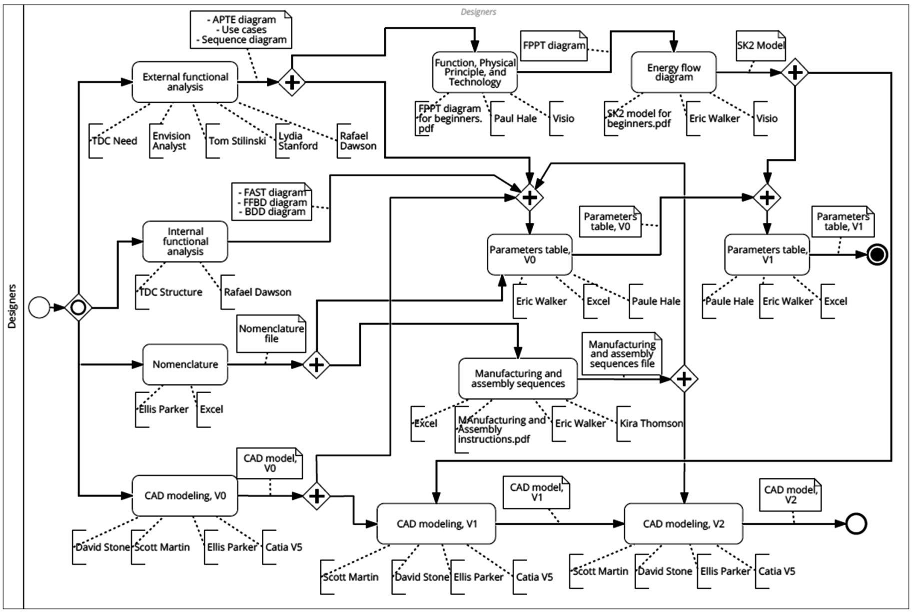


Fig. 4. Electric torch design process in BPMN formalism

"CAD modelling V0". Second, after performing the previous fourth activities, he can either perform the "Function, Physical principle and technology" activity or the "Manufacturing and assembly sequences" activity. After that, he has the possibility to choose to begin with creating the "Parameters table V0" or the "Energy flow diagram". Finally, he has the choice to perform the "Parameters table V1" or "CAD modelling V1". Note that there are several different possible executions.

Process mining phase: in order to derive the process patterns to be followed, ProM is used to analyse the past executions (in MXML format) and then extracts the most frequent patterns (Fig. 5). In this example, the first pattern proposed by ProM contains four activities (i.e. pattern alphabet), and in 90% of the cases, the "Parameters table V1" activity is followed by "CAD modelling V1" activity which is followed by the "CAD modelling V2" activity that concludes the process. The patterns proposed by ProM are classified by their apparition frequency and can be exported in CSV format (Fig. 6). The engineer chooses, then, the most suitable pattern; where the first is the most frequently used in this design context.

Pattern Alphabet	Pattern Sequence Set	Alphabet Count	Instance Count (%)
CAD modeling V2-complete CAD modeling V1-complete Parameters table V1-complete EndProcess-complete	Parameters table V1-complete CAD modeling V1-complete CAD modeling V2-complete EndProcess-complete	9	90
CAD modeling V2-complete CAD modeling V1-complete Parameters table V1-complete Parameters table V0-complete Energy flow diagram-complete Manufacturing and assembly sequences-complete EndProcess-complete Function, Physical principle and technology-complete	Function, Physical principle and technology-complete Manufacturing and assembly sequences-complete Energy flow diagram-complete Parameters table V0-complete Parameters table V1-complete CAD modeling V1-complete CAD modeling V2-complete EndProcess-complete	7	70

Fig. 5. Frequent patterns (only two are presented) in the electric torch design process

case	event	startTime	completeTime	Input	Documentary reso	Software resou	Human resource	Output	Internal Control
1	CreateProcess	1/2/2016 9:00	1/2/2016 9:00						
	External functional analysis	1/2/2016 10:15	1/2/2016 10:15			TDC Need	Tom Stilinski	APTE diagram	
	Internal functional analysis	1/2/2016 11:15	1/2/2016 11:15			TDC Structure	Rafael Dawson	FAST diagram	
	Nomenclature	1/2/2016 12:15	1/2/2016 12:15			Excel	Ellis Parker	Nomenclature file	
	CAD modeling V0	1/2/2016 13:15	1/2/2016 13:15			Catia V5	Ellis Parker	CAD model V0	
	Function, Physical principle and te	1/2/2016 14:15	1/2/2016 14:15		FPPT diagram for b	Visio	Paul Hale	FPPT diagram	APTE diagram
	Manufacturing and assembly sequ	1/2/2016 15:15	1/2/2016 15:15		Manufacturing and	Excel	Eric Walker	Manufacturing a	Nomenclature file
	Energy flow diagram	1/2/2016 16:15	1/2/2016 16:15		SK2 model for beg	Visio	Eric Walker	Sk2 model	FPPT diagram
	Parameters table V0	1/2/2016 17:15	1/2/2016 17:15		Nomenclature file	Excel	Paul Hale	Parameters table	Manufacturing and assembly sequence file
	Parameters table V1	1/2/2016 18:15	1/2/2016 18:15		Nomenclature file	Excel	Paul Hale	Parameters table	Parameters table V0
	CAD modeling V1	1/2/2016 19:15	1/2/2016 19:15	CAD model	Nomenclature file	Catia V5	Ellis Parker	CAD model V1	Sk2 model
	CAD modeling V2	1/2/2016 20:15	1/2/2016 20:15	CAD model	Nomenclature file	Catia V5	Ellis Parker	CAD model V2	Manufacturing and assembly sequence file
	EndProcess	1/2/2016 20:30	1/2/2016 20:30						
2	CreateProcess	1/2/2016 9:00	1/2/2016 9:00						
	Internal functional analysis	1/2/2016 10:15	1/2/2016 10:15			TDC Structure	Rafael Dawson	FAST diagram	
	External functional analysis	1/2/2016 11:15	1/2/2016 11:15			TDC Need	Tom Stilinski	APTE diagram	
	Nomenclature	1/2/2016 12:15	1/2/2016 12:15			Excel	Ellis Parker	Nomenclature file	
	CAD modeling V0	1/2/2016 13:15	1/2/2016 13:15			Catia V5	Ellis Parker	CAD model V0	
	Function, Physical principle and te	1/2/2016 14:15	1/2/2016 14:15		FPPT diagram for b	Visio	Paul Hale	FPPT diagram	APTE diagram
	Manufacturing and assembly sequ	1/2/2016 15:15	1/2/2016 15:15		Manufacturing and	Excel	Eric Walker	Manufacturing a	Nomenclature file
	Energy flow diagram	1/2/2016 16:15	1/2/2016 16:15		SK2 model for beg	Visio	Eric Walker	Sk2 model	FPPT diagram
	Parameters table V0	1/2/2016 17:15	1/2/2016 17:15		Nomenclature file	Excel	Paul Hale	Parameters table	Manufacturing and assembly sequence file
	Parameters table V1	1/2/2016 18:15	1/2/2016 18:15		Nomenclature file	Excel	Paul Hale	Parameters table	Parameters table V0
	CAD modeling V1	1/2/2016 19:15	1/2/2016 19:15	CAD model	Nomenclature file	Catia V5	Ellis Parker	CAD model V1	Sk2 model
	CAD modeling V2	1/2/2016 20:15	1/2/2016 20:15	CAD model	Nomenclature file	Catia V5	Ellis Parker	CAD model V2	Manufacturing and assembly sequence file
	EndProcess	1/2/2016 20:30	1/2/2016 20:30						

Fig. 6. Exported process patterns (only two are presented)

Machine learning phase: for each activity of the chosen process pattern, authors aim to propose the parameters (resources and/or decision) to be considered while performing the activity. Weka is used since it provides many supervised classification algorithms which can be trained by the training set that refers to the activity log. In this study, authors use the J48 decision tree classifier which is an implementation of the C4.5 algorithm that generates decision trees [29]. After being trained and tested, the algorithm allows predicting the desired activity parameters' value, in Fig. 7 an example of the software resource that is used to perform the "External functional analysis" activity is presented. In this example, the prediction takes into account the human resource that performs the activity and the start time of the execution of the activity. For example, the J48 algorithm is 77.6% sure that Tom Stilinski uses TDC Need to perform the "External functional analysis" activity at 9.25 AM o'clock.

Instance	Human Resource	Start Time	Software
1	Tom Stilinski	09:25	?
2	Lydia Stanford	10:25	?
3	Rafael Dawson	14:15	?
4	Lydia Stanford	17:30	?
5	Rafael Dawson	07:45	?

➡

```
=== Predictions on test set ===

 inst#     actual  predicted error prediction
     1        1:?  1:TDCNeed       0.776
     2        1:?  1:TDCNeed       0.742
     3        1:? 2:EnvisionAnalyst       0.997
     4        1:?  1:TDCNeed       0.742
     5        1:? 2:EnvisionAnalyst       0.997
```

Fig. 7. The software resource prediction for the "External functional analysis" activity

6 Conclusion and Future Work

The objective of this research is to propose a decision-making support technique that aids engineers, during the design or redesign of a product. Through the proposed technique, the past design executions are analysed and the most relevant design patterns to be followed are proposed. In addition, for each activity of the considered design process, the most suitable resources and/or decision are proposed. The proposed technique illustrates the feasibility of the assumption about the use of PM and ML in the decision-making support. It has been illustrated on a case study where only *And* and *Or* (with true conditions) gateways are considered. Future work consists in addressing the design processes containing the *Xor/Or* gateways that give rise to several patterns, where only the one that satisfies the condition must be executed. The objective is to automatically propose, in a new design context, the process pattern in which all the *Xor/Or* gateways' conditions are satisfied. Future work also consists in studying the impact of the proposal on design processes with respect to some performance indicators such as development time, 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. Gero, J.S.: Creativity, emergence and evolution in design. Knowl. Based Syst. **9**(7), 435–448 (1996)
2. Ullman, D.G.: The Mechanical Design Process. McGraw-Hill Higher Education, New York (2003)
3. Roucoules, L., Yahia, E., Es-Soufi, W., Tichkiewitch, S.: Engineering design memory for design rationale and change management toward innovation. CIRP Ann. Manuf. Tech. **65** (1), 193–196 (2016)
4. Es-Soufi, W., Yahia, E., Roucoules, L.: Collaborative design and supervision processes meta-model for rationale capitalization. In: Eynard, B., Nigrelli, V., Oliveri, S.M., Peris-Fajarnes, G., Rizzuti, S. (eds.) Advances on Mechanics, Design Engineering and Manufacturing, pp. 1123–1130. Springer, Heidelberg (2017)
5. van der Aalst, W.M.P.: Process Mining: Discovery, Conformance and Enhancement of Business Processes. Springer, Heidelberg (2011)
6. van der Aalst, W.M.P.: Process mining: overview and opportunities. ACM Trans. Manag. Inf. Syst. (TMIS) **3**(2), 7 (2012)
7. Alexander, C., Ishikawa, S., Silverstein, M.: A Pattern Language: Towns, Buildings, Construction. Oxford University Press, New York (1977)
8. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design Patterns: Elements of Reusable Object-Oriented Software. Pearson Education, Upper Saddle River (1994)
9. Gzara, L., Rieu, D., Tollenaere, M.: Product information systems engineering: an approach for building product models by reuse of patterns. Robot. Comput. Integr. Manuf. **19**(3), 239–261 (2003)
10. Kodratoff, Y., Michalski, R.S., Carbonell, J.G., Mitchell, T.M.: Machine Learning: An Artificial Intelligence Approach. Morgan Kaufmann, San Mateo (1990)
11. Han, J., Kamber, M., Pei, J.: Data Mining: Concepts and Techniques. Elsevier Science, Amsterdam (2011)
12. Albalate, A., Minker, W.: Semi-supervised and Unsupervised Machine Learning: Novel Strategies. Wiley, New York (2013)
13. Zhu, X., Goldberg, A.B.: Introduction to semi-supervised learning. Synth. Lect. Artif. Intell. Mach. Learn. **3**(1), 1–130 (2009)
14. Kulkarni, P.: Reinforcement and systemic machine learning for decision making. Wiley, New York (2012)
15. Sim, S.K., Duffy, A.H.B.: A foundation for machine learning in design. AI EDAM **12**(02), 193–209 (1998)
16. Tong, S.S., Powell, D., Cornett, D.: Engineous: a unified method for design automation, optimization, and integration. In: Artificial Intelligence in Engineering Design, vol. III, pp. 235–254 (1992)
17. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning, 1st edn. Addison-Wesley Longman Publishing Co., Inc., Boston (1989)
18. Chen, H.-Y., Chang, H.-C.: Consumers' perception-oriented product form design using multiple regression analysis and backpropagation neural network. Artif. Intell. Eng. Des. Anal. Manuf. **30**(01), 64–77 (2016)
19. Pitiot, P., Coudert, T., Geneste, L., Baron, C.: Hybridation of Bayesian networks and evolutionary algorithms for multi-objective optimization in an integrated product design and project management context. Eng. Appl. Artif. Intell. **23**(5), 830–843 (2010)
20. Danglade, F., Pernot, J.-P., Véron, P.: On the use of machine learning to defeature CAD models for simulation. Comput. Aided Des. Appl. **11**(3), 358–368 (2014)

21. Yan, H.S., Xu, D.: An approach to estimating product design time based on fuzzy ν-support vector machine. IEEE Trans. Neural Netw. **18**(3), 721–731 (2007)
22. Reimann, P., Calvo, R., Yacef, K., Southavilay, V.: Comprehensive computational support for collaborative learning from writing. In: International Conference on Computers in Education (ICCE), Putrajaya, Malaysia (2010)
23. Saravanan, M., Rama Sree, R.: Application of mining algorithms using ProM and Weka tools. IJCST **2**(3), 331–337 (2011)
24. Crerie, R., Baião, F.A., Santoro, F.M.: Discovering business rules through process mining. In: Halpin, T., Krogstie, J., Nurcan, S., Proper, E., Schmidt, R., Soffer, P., Ukor, R. (eds.) BPMDS/EMMSAD -2009. LNBIP, vol. 29, pp. 136–148. Springer, Heidelberg (2009). doi:10.1007/978-3-642-01862-6_12
25. Object Management Group: Business Process Model and Notation (BPMN) Version 2.0, January 2011
26. van Dongen, B.F., van der Aalst, W.M.P.: A meta model for process mining data. EMOI-INTEROP **160**, 30 (2005)
27. Jagadeesh Chandra Bose, R.P., van der Aalst, W.M.P.: Abstractions in process mining: a taxonomy of patterns. In: Dayal, U., Eder, J., Koehler, J., Reijers, H.A. (eds.) BPM 2009. LNCS, vol. 5701, pp. 159–175. Springer, Heidelberg (2009). doi:10.1007/978-3-642-03848-8_12
28. Witten, I.H., Frank, E., Trigg, L., Hall, M., Holmes, G., Cunningham, S.J.: Weka: practical machine learning tools and techniques with Java implementations (1999)
29. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers Inc., San Francisco (1993)