



HAL
open science

Devising Configuration Guidance with Process Mining Support

Housseem Chemingui, Camille Salinesi, Inès Gam, Raul Mazo, Henda Ben Ghezala

► **To cite this version:**

Housseem Chemingui, Camille Salinesi, Inès Gam, Raul Mazo, Henda Ben Ghezala. Devising Configuration Guidance with Process Mining Support. 2021. hal-03252153

HAL Id: hal-03252153

<https://hal.science/hal-03252153v1>

Preprint submitted on 7 Jun 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Devising Configuration Guidance with Process Mining Support

Houssem CHEMINGUI ^{1,2}, Camille SALINESI ¹, Inès GAM ^{1,2}, Raúl MAZO ^{3,4}, Henda BEN GHEZALA ²,

1. CRI Laboratory, Paris 1 Panthéon Sorbonne University, 90 rue de Tolbiac, 75013 Paris - France

2. RIADI Laboratory, ENSI Manouba University, 2010 Manouba - Tunisia

3. Lab-STICC, ENSTA Bretagne, 2 rue François Verny, Brest - France

4. GIDITIC, Universidad Eafit, Carrera 49 N° 7 Sur-50, Medellin - Colombia

ABSTRACT

[Context and Motivation] Adopting a Product Line strategy can be a double-edged sword when the configuration process proves complex. In fact, complex configuration processes deprive industry and stakeholders of the benefits of Product Line Engineering; mainly, decreased time to market and increased customer satisfaction. **[Question/Problem]** Unfortunately, this stumbling block is widespread mainly when stakeholders deal with large product line models that contain a plethora of variants. Therefore, the configuration process becomes a monotonous and an error-prone task. Consequently, it is of great interest to industry that product line configuration modules are able to recommend the best configuration alternatives until leading the stakeholders to a satisfying configuration experience. **[Principal idea/Contribution]** From this perspective, this research aims to enhance the new generation of product line configuration modules, maximizing the stakeholders comfort based on previous configuration processes. Therefore, previous configuration processes are logged, mined and interpreted to produce guidance solutions. Afterwards, guidance solutions are devised and orchestrated by the CONFILOG method detailed in this paper. **[Results]** Experiments of CONFILOG show that configuration guidance can be produced with process mining support. Solutions feasibility and distinction is demonstrated through an Empirical Study With Students applied to a bike product line.

KEYWORDS

Product Line, Configuration Process; Complex; Logged; Mined; Devised; CONFILOG, Guidance Solutions; Process Mining

1. Introduction

Product lines present a business strategy that advocates a reuse vision. The purpose of product lines is no longer to develop single products massively and separately but rather to develop, capitalize and reuse a set of products according to the variation factors. Extensive research and industrial experience have widely proven the significant benefits of product lines, among them, increased efforts, reduced time to market and optimized customizations. [Clements et al., 2001]

In Product Line Engineering, literature [Clements et al., 2002] [Pohl et al., 2005] distinguishes two main processes, namely domain engineering and application engineering. In a nutshell, domain engineering consists of specifying variable assets and their reuse context through variability models. However, application engineering involves developing products by reusing the domain. Indeed, variability models are not only used as an abstract representation for documentation or communication purposes but they must support a domain reuse process called the configuration process.

The configuration process reflects the transition from domain engineering to application engineering. In other words, the configuration process consists in specifying a valid product from a variability model in accordance with user desiderata and domain constraints.

In large-size variability models, unguided configuration processes give rise to monotonous experiences and therefore stakeholders dissatisfaction. In fact, configuration decisions become complex to manage; due to the gobs of variants on the one hand and the out of control propagation of constraint interdependencies on the other hand. Furthermore, configuration decisions become cumbersome and ambiguous leading to inappropriate decisions with unsuitable consequences.

The configuration shortcomings were widely revealed in literature, mainly the decision complexity [Noorian et al., 2017] [Zhan et al., 2018] [Hayashi et al., 2017] and the lack of performance [Ochoa et al., 2017] [Sinnhofer et al., 2016] [Zhang et al., 2013]. Unfortunately, the decision complexity and the lack of performance reflect, into the

bargain, the challenging problems for contemporary industrial configurators. From a marketing standpoint, proposing to stakeholders time consuming and indecipherable configuration processes lead to incomplete configuration. Generally an incomplete configuration reflects that the configuration originator voluntarily left the process and probably he turned to look elsewhere for competitors offering more flexible and easy to handle configurators.

Given the above literature observations and industrial challenging problems, the general goal of our research is guided by the following Research Question :

RQ : How to guide and schedule stakeholder decisions throughout the configuration process of product lines?

In response to the above RQ, the Product Line community is ramping up efforts to design solutions aiming to guide the configuration process. In fact, existing solutions addressed the configuration shortcomings since several methods were proposed, mainly, (i) interactive configuration methods [Benavides et al., 2010] [Hadzic et al., 2004], (ii) recommendation methods [Jannack et al., 2011], (ii) a combination of these two methods [Triki et al., 2014] and (iv) heuristic methods [Mazo et al., 2014].

Although existing guidance methods have helped to resolve the scaling issues of the configuration process, the process schedule and the stakeholder understanding still need further consideration.

On the one hand, published solutions consider a static order of the configuration process stages. As far as we know, the dynamic order schedule of the configuration process constitutes a literature gap. However, heuristic methods claim that the process order can be based on the variability model or the domain constraints. Moreover, in the Dopler tool suite [Dhungana et al., 2011] and in the configuration workflow proposed by [Hubaux et al., 2009] the order is considered as static since it presents a series of predefined actions that lack flexibility and customization. Adding to that, these solutions predict values for the model variants that match with a given partial configuration, and not a dynamic series of actions to be scheduled at each configuration stage. For instance, in some situations, a car configurator needs to propose the variability related to the sheet metal and the tapestry before the variability related to the mechanic.

On the other hand, published solutions ignore the guidance goals. From a marketing perspective, it is intolerable to provide the same guidance to stakeholders that have different preferences and/or concerns. Indeed, “standard” guidance saves effort and time, nevertheless it doesn't make stakeholders feel targeted by the recommended choices. In addition, considering the stakeholder behaviors during the configuration seems unconscious. For instance, in some situations, a car configurator needs to propose a rapid configuration process for first time users, and a flexible configuration process for users looking for testing choice combinations.

The revealed shortcomings of the existing guidance methods motivate the idea of a deeper focus, not only, on the recommendation of model variant values but also on the configuration process schedule and the goal-driven guidance.

To overcome the existing solution drawbacks and answer to the industrial call of guidance [Young et al., 2017][Dumitrescu et al., 2013], the main objective of this research is to propose an overall method baptized CONFIOLOG aiming to improve the configuration processes. CONFIOLOG takes advantage of the process mining support in order to meet the guidance expectations. The process mining orientation is motivated by the fact that potential solutions have to schedule processes, capture stakeholder behaviors and consider the guidance goals. Indeed process mining is defined as a set of techniques that enable discovering, monitoring and improving processes through the extraction of knowledge and information from event logs available in the information systems in use nowadays [Aalst, 2011]. From a practical standpoint, there is often a significant gap between what is prescribed or supposedly produced, and what actually happens throughout the process. In this research, the main purpose of adopting a process mining strategy is to extract process traces that are able to highlight different configuration difficulties and obstacles. In fact, configuration process traces can be used to automatically discover process models. Process models allow product line designers to get insights on various perspectives of the configuration process including time, control flow, decisions, bottlenecks, undo choices, etc... Furthermore, process mining proposes simulation techniques [Tiwari et al., 2008] for replaying the process models in order to have a deeper comprehension of stakeholders and capture performance indicators of the configuration process.

This article presents CONFIOLOG, a method for guiding the configuration process. The main contributions presented in this paper are:

- An event log extractor that captures the stakeholders actions during the configuration process.
- Process mining guidelines that aim to discover process models from event logs making guidance and improvement insights of configuration.

- A Goal question metric model to master the different guidance contexts.
- A recommendation algorithm that dynamically schedules the configuration stages and predicts the model variant values.

CONFILOG was evaluated by Empirical Studies With Students. This evaluation was a pilot to make sure of the method power with the purpose of evaluating the performance, completeness and pertinence of the produced guidance. Moreover, the evaluation provided insights to improve the event logs extractor by adding new logging attributes about the configuration stakeholders. Logging the stakeholder profiles needs further consideration since it may extend discovered process models with customization information. On the other hand, CONFILOG uses the fuzzy miner algorithm [Günther, et al., 2007]. However, it will be interesting to discover configuration process models using other process mining algorithms such as the Alpha miner and Heuristic miner [Porouhan et al., 2014].

The remainder of this paper is structured as follows. Section 2 gives an overview of the CONFILOG method underpinnings. Section 3 details the CONFILOG process discovery while section 4 details the confiLog recommendation engine. Section 5 shows the tooling instruments of the CONFILOG method and its evaluation with Empirical Studies With Students carried out following a well-referenced process proposed by [Carver et al., 2010]. Section 6 discusses related works while section 7 concludes the paper and suggests future research directions.

2. The CONFILOG method underpinnings

At first sight, CONFILOG stands for CONFIguration guidance based on process LOGs. Indeed, CONFILOG is a guidance method for complex product line configuration processes. The CONFILOG method's purpose is to guide stakeholders who deal with large-scale product line models, during the configuration process.

Figure 1 shows the CONFILOG method overview. In fact, CONFILOG takes as inputs :

- (i) Guidance goals expressed by stakeholders, such as performance and flexibility (Middle Left figure 1),
- (ii) A Product Line Model from which stakeholders are configuring the product (Middle Left figure 1),
- (iii) A Goal Question Metric (GQM) repository that encompasses all goals, questions and metrics (Down Right Figure 1)
- (iv) A trace repository that capitalizes event logs of previous configuration processes (Top Right Figure 1).

The outputs of CONFILOG are :

- (i) A guidance solution that covers a process order and a variant value returned to the stakeholder (Top figure 1),
- (ii) A process trace of the current configuration capitalized later in the traces repository (Down figure 1).

To achieve that, CONFILOG proposes a process mining strategy aiming to improve the configuration experience (cf. section 3). Furthermore, the method relies on a recommendation engine (cf. section 4) that orchestrates guidance solutions throughout the configuration process.

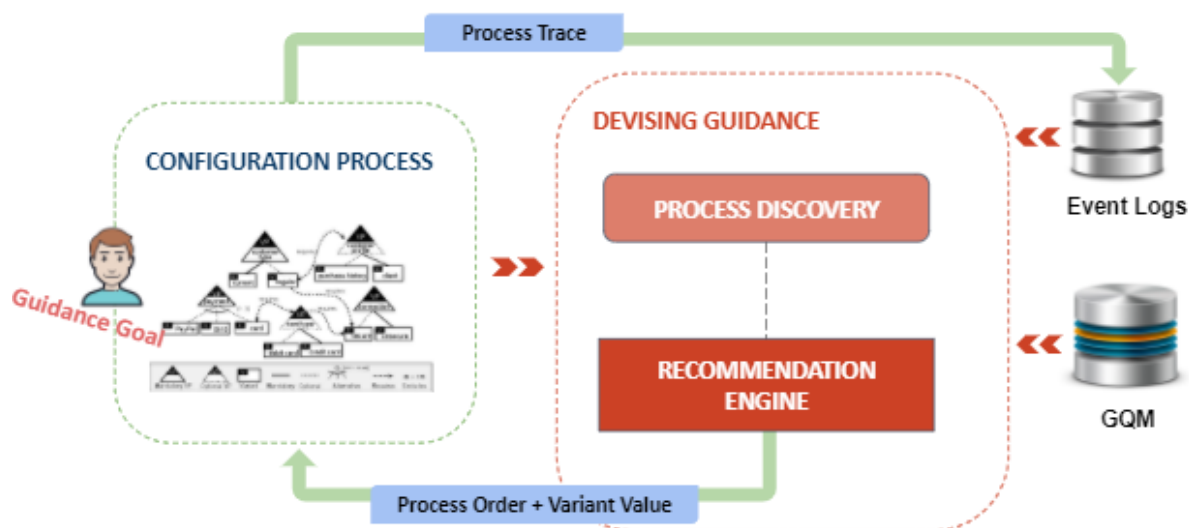


Figure 2. Overview of the CONFILOG Method

In this respect, the configuration process is conducted gradually, through a series of stakeholder interactions. Whenever variants are configured, CONFILOG proposes new recommendations related to variants that are still waiting to be configured. The Product Line Model variants must strictly comply with the domain constraints specified in the design phase. At each configuration stage, the CONFILOG method provides a set of generated recommendations that are based on the current partial configuration and a guidance goal expressed by the stakeholder. On the other hand, guidance goals are outlined in a GQM repository considered as a dictionary of the recommendation. In practice, for each goal, the CONFILOG algorithm queries process models, discovered beforehand, in order to provide answers for the guidance questions. Moreover, metrics are used to evaluate guidance answers coming from process mining treatments.

Usually, CONFILOG cares about the stakeholder freedom. Indeed, the recommendation engine promotes the recommended value of the variant without imposing it. Following the recommended order of configuration steps remains optional. Only the variants waiting to be configured are the aim of the recommendation, however, variants included in the partial configuration are not concerned. Stakeholders feel free to adapt a recommendation or change the type of guidance at any time during the configuration process. Selected variants are therefore systematically propagated and their domain consistency is checked. Checking domain consistency of the configuration guarantees that next recommendations are based only on a correct partial configuration. This reasoning is repeated in loop until reaching a configuration deemed complete by the stakeholder. At each configuration stage, the stakeholder feels free to undo his choices. Undo choices are taken into account by the CONFILOG algorithm which reacts dynamically by indicating which variants remain compatible to reproduce other guidance alternatives.

The originality of the method lies on the fact that (i) produced guidance solutions are consistent and complete since they were based on traces of only coherent processes (ii) stakeholders feel free to undo choices and not taking into account recommended solutions (iii) guidance solutions are bidimensional considering both the order of process stages and a variant value at each configuration stage (iv) stakeholders can express their guidance goal such (ie. performance or customization) (v) mechanisms are generic to different contexts of configuration (ie COTS and ERPs) [Chemingui et al., 2020]. (cf. section 6)

The next two sections explain with more details each brick of the CONFILOG method and how they operate to produce configuration guidance.

3. The CONFILOG Process Discovery

To emerge a guided enactment of the configuration process based on a process mining strategy, it is straightforward to identify which data is needed to achieve a given goal? Identifying pertinent data to analyse set forth a very high impact on the discovered models and their interpretation. In the current work, the goal is to boost performance, flexibility and customization indicators of the configuration process.

The configuration process is a set of decision stages calling the shots. For each configuration stage, the CONFILOG traces repository encompasses the following attributes:

- The unique identifier of the referring configuration process "ProcessID",
- The configuration stakeholder "Originator",
- The name of the configured variant "VariantName",
- The type of the configured variant "VariantType". Variants can be mandatory, optional or grouped variants containing cardinalities,
- The type of the configured variant "VariantValue", if it was selected or rejected for example,
- The type of the decision "DecisionType". Decisions are made manually by the user or automated after constraint propagations,
- The step of the decision "DecisionStep" on the whole process,
- A timestamp of the decision "Timestamp" containing the date, and milliseconds timing,
- A guidance goal "GuidanceGoal" expressed by the user. Configurations can be processed without a guidance goal.

To achieve guidance goals, CONFILOG needs to make sure that right answers are made to the right questions using the right metrics. The GQM (Goal Question Metric) method [Basili et al., 1994] is a simple way to make sure that the metrics are closely tied to the goals. As a matter of fact, the GQM was proposed as a framework for defining and

interpreting software measurement. Later, the GQM was expanded with many other concepts and became a result of many years of practical experience and academic research [Van Solingen et al., 2002][Berander et al., 2006].

Table 1 shows the three basic levels of the GQM model: Goals, Questions and Metrics. First, the left upper level presents goals (the conceptual level) that have to be defined. Goals have to be operationalized by generating questions that help to understand how to meet them. Second, questions (the operational level) characterize the object of measurement in a context of a quantified issue. Finally, the lower level presents the required metrics (the quantitative and qualitative level) to evaluate answers. In most cases, the metrics are used to evaluate more than an answer for a given guidance question.

To proceed with the GQM approach, stakeholders goals were identified beforehand. Afterwards, questions were carefully addressed to meet each goal. A plethora of questions was raised. Hence, CONFILOG categorized them by combining indistinguishable goals that may affect other goals. For each guidance question, the answer is evaluated using at least one metric. Table 1 outlines, the CONFILOG GQM Model :

Table 1. Goal Question Metric Model of the CONFILOG Method

Goals
<ul style="list-style-type: none"> - G1: Reduce the required time to reach a valid configuration from a process perspective - G2: Maximize the flexibility and efficiency of the configuration from a process perspective - G3: Maximize the customization of the configuration from a process perspective
Questions
<ul style="list-style-type: none"> - G1.Q1 : Which variants should be configured first? - G1.Q2 : What is the average time required to reach a complete and a valid configuration process? - G1.Q3 : What are the processes that don't involve undo choices? - G1.Q4 : What are the processes that were executed before/after the average time? - G2.Q1 : Was the user lost during the configuration process? - G2.Q2 : What are the variants that cause backtracks? - G2.Q3 : What processes have been executed with a minimum/maximum of manual choices? - G2.Q4 : What are the processes that were executed with a minimum/maximum of propagated choices? - G2.Q5 : What are the safest/doubtful paths in a process? - G3.Q1 : What are the processes that were made by serious users? - G3.Q2 : What is the trace of a given user? - G3.Q3 : Which users have similar intentions? - G3.Q4* : What are the processes that include best-sellers/worst-sellers for each user category (Gender, age, Country, etc.) or for a given period (summer, holiday, etc.) - G3.Q5* : What are the processes that include fashionable configurations? - G3.Q6 : What are the processes that include full/minimum options of the configuration?
Metrics
<ul style="list-style-type: none"> - M1 : Number of variants involved in the configuration process. - M2 : Provide a not a valid configuration. - M3 : Time required to complete n configuration steps. - M4 : Tolerated time after average time. - M5 : Manual or automatic assignment of the variant value. - M6 : Number of undo choices. - M7 : Frequency time of switching from one variant to another. - M8 : Number of processes that start with a given variant. - M9 : Contextual information (Period, events, location, etc.) - M10 : Contextual information (Fashion time interval)

* Question that needs contextual information

Given that product line configurations are usually complex and produce unstructured logs, the CONFILOG method makes use of the fuzzy mining technique [Günther, et al., 2007] to answer the guidance questions. Fuzzy Miner is a process mining algorithm addressing the problems of large numbers of activities and highly unstructured behavior. Discovered fuzzy models are simple to manage in an interactive manner. In fact, the Fuzzy miner uses significance and correlation indicators to interactively simplify the process model at different desired levels of abstraction.

Compared to the Alpha miner and Heuristic miner [Porouhan et al., 2014], the fuzzy miner can leave out less important configuration activities or hide them in clusters. Fuzzy models of the configuration process can be used to animate the event log on top of the created model to get a feeling for the dynamic process behavior. The model can be adjusted to particular situations and question answers. While the fine-grained configurability of the algorithm, finding the “right” parameter settings (frequency, performance, start event, end event, activity filters) can sometimes be time-consuming.

In the CONFILOG context, filtering activities is powerful for many reasons such as removing some configuration process instances (cases), adding events, removing events or modifying events. In some cases, CONFILOG needs to filter event logs based on “artificial” start and end events for instance to monitor a part of the process in different situations. Similarly, filtering event logs can be very helpful to select only those process instances that start and/or end with particular activities in the first place or in a particular time frame. To answer the aforementioned guidance questions CONFILOG queries the fuzzy model using the following directives that are useful if there are hundreds of different events and answering to a given guidance question focuses only on particular aspects that occur in the configuration model.

- Remap Element Log filter : This directive combines several activities into one.
- Duplicate Task filter : This directive removes direct repetitions of events with the same name.
- Attribute value filter : This directive keeps only events that have a given attribute value.
- Process Instance Length filter : This directive specifies which process instances to keep based on a threshold on the number of events.
- Process Instance Frequency filter: This directive specifies how many process execution variants are in the event log. After, this treatment allows focusing on the most frequent (e.g., 80%) execution variants only.
- Enhanced Event Log filter: This directive filters both events and process instances based on an activity-based frequency percentage threshold. This is useful if there are hundreds of different events to, for example, focus only on activities that occur in most of the cases.

4. The CONFILOG Recommendation engine

At each configuration stage, the CONFILOG recommendation engine generates recommendations based on (i) a guidance goal selected by the stakeholder (Performance, Flexibility or customization), (ii) a partial configuration and (iii) a set of enabled activities. The recommendation engine provides predictive information, interpreted from discovered fuzzy models containing indicators values such as duration process, number of actions etc. The CONFILOG recommendation engine requires guidance datasets containing guidance indicators values.

Figure (2) details the steps of the ConfiLog recommendation process, namely (i) Configuration process conduct, (ii) Mining answers abstraction and filtering, (iii) Mining answers weighting and (iv) Impact decision Calculation.

The ConfiLog method divides the product line model to be configured, by the stakeholders, to variants and variability points. Each variability point covers variants on which the stakeholder performs configuration activities. For instance, the variability point “Car Color” can contain the “Black”, “Gray”, “White” or “Red” variants.

A configuration activity a corresponds to the fact of selecting or rejecting a given variant. Before starting the configuration, the stakeholder expresses his guiding goal. The guidance goal τ can be selected by the stakeholder from three main goals. $\text{Enum } \tau \{ \text{Performance, Flexibility, Personalization} \}$. Incrementally, the stakeholder performs a configuration activity a , the partial configuration $\rho \in A^*$ is updated with new traces. In parallel, the domain constraints are propagated automatically. Consequently, new enabled activities E appear and new recommendations are initiated. A recommendation is initiated by a recommendation request r . Formally, a recommendation request r is defined by a tuple $r(\tau, \rho, E)$ with a guidance goal τ , a partial trace $\rho \in A^*$ and a set of enabled activities (the rest of the variants of the model to be configured which go with the partial trace) $E \subseteq A$. The enabled activities E constitute the parameter of abstraction and filtering of the guidance answers.

Gradually according to the configuration progress, some activities become allowed E and others become censored as they violate domain constraints. The recommendations are calculated only for enabled activities E by considering mining answers σ . A mining answer $\sigma \in A^*$ is a finite sequence of activities A , given by a function $\sigma: \{1, \dots, n\} \rightarrow A$ and $\sigma': \{1, \dots, m\} \rightarrow B$. The mining answers meet a given guidance goal with a contribution $\tau(\sigma)$ as shown in the examples below. The contributions of the mining answers are indicators identified through the configuration process mining and recorded in a guidance dataset D . The fuzzy process configurations model can be

converted to guidance datasets containing, for instance, the following information in relation with goal indicators, configuration variants order and values :

- Dataset 1 : Performance - Contribution [15] : [Variant1 - Selected, Variant3 - Deselected, Variant5 - Selected]
- Dataset 2 : Flexibility - Contribution [77] : [Variant1 - Selected, Variant5 - Selected, Variant6 - Selected]
- Dataset 3 : Customization - Contribution [18] : [Variant4 - Selected, Variant7 - Selected, Variant6 - Deselected]

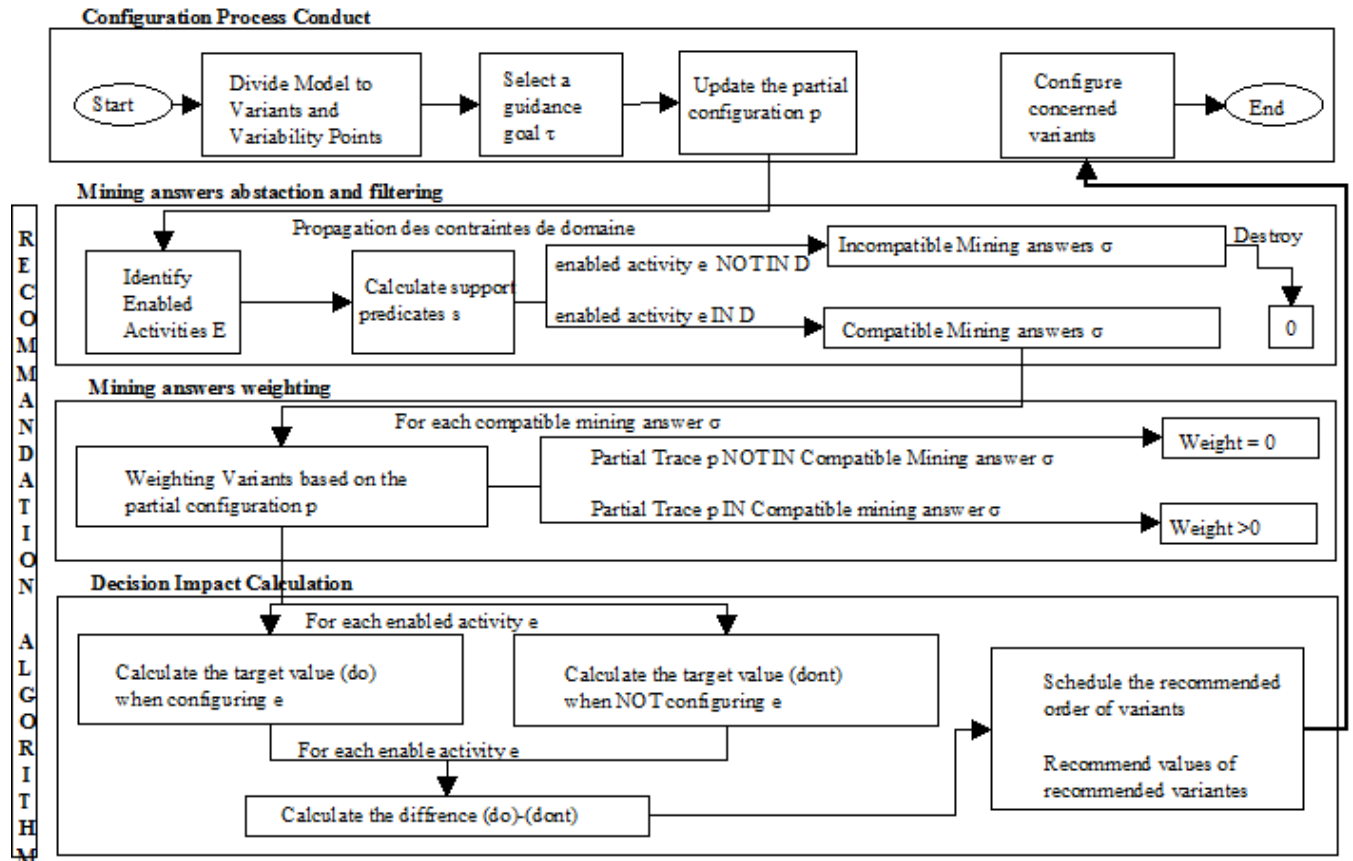


Figure 2 - The ConfiLog recommendation Process

For each guidance goal, the dataset contains selected mining answers to answer the guidance questions and represent the Fast, flexible and personalized process instances. Only the mining answers deemed compatible with the recommendation request should be retained. The relevance of a mining answer to produce a recommendation is determined based on the support of this answer to the recommendation being considered. The mining answers which do not support (i) the guiding goal τ , (ii) the partial trace p and (iii) the enabled activities E are destroyed. To do this, a predicate $s(\tau, \rho, \sigma, e)$ is applied to filter out unnecessary answers. The member operator (\in) identifies the membership of an activity in a given mining answer: $a \in \sigma \Leftrightarrow \exists 1 \leq i \leq n \text{ ai} = a$

To classify the degree of relevance of the mining answers retained for the recommendation, they are weighted. Mining answers with a high degree of correlation with the partial trace should be weighted higher than those with low or no correlation. A weighting function (ω) is defined to express the importance of each mining answer σ for the recommendation of an enabled activity e . The value 1 indicates that the mining answer σ and the partial trace p fully match. The value 0 indicates that the mining answer σ and the partial trace p do not match at all. The weight of the mining answer $\omega(\tau, \rho, \sigma)$ is calculated as a function of dividing the number of activities of the partial trace included in the mining answer by the number of activities of the trace partial. $\rho : \omega(\tau, \rho, \sigma) = |\rho \cap \sigma| \text{ DIV } |\rho|$ is a mining answer weighting.

The mining answer weights are used to calculate the impact of each future configuration decision made by the stakeholder. For each enabled activity e , the algorithm of the ConfiLog method calculates its decision impact based on two factors: (i) An expected target value when configuring (do) a given enabled activity and (ii) An expected target value when NOT configuring ($dont$) and enabled activity.

The (do) is calculated as a function of sum Σ the weights of the compatible mining answers $\omega(\tau, \rho, \sigma)$ multiplied by their contribution $\tau(\sigma)$; all divided by the sum Σ of the weights of the mining answers $\omega(\tau, \rho, \sigma)$.

$$do(e, \tau, \rho, D) = \frac{\sum_{\sigma \in Ds(\tau, \rho, e)} [\omega(\tau, \rho, \sigma) * \tau(\sigma)]}{\sum_{\sigma \in Ds(\tau, \rho, e)} \omega(\tau, \rho, \sigma)}$$

Likewise, the ($dont$) is calculated in the same way, however, it considers the mining answers that do not support the enabled activity.

$$dont(e, \tau, \rho, D) = \frac{\sum_{e' \in E \setminus \{e\}} \sum_{\sigma \in Ds(\tau, \rho, e') \setminus Ds(\tau, \rho, e)} [\omega(\tau, \rho, \sigma) * \tau(\sigma)]}{\sum_{e' \in E \setminus \{e\}} \sum_{\sigma \in Ds(\tau, \rho, e') \setminus Ds(\tau, \rho, e)} \omega(\tau, \rho, \sigma)}$$

The calculations of (do) and ($dont$) are iterative and must be recalculated at each configuration stage, at each time the partial configuration p is updated. For each configuration stage, the ConfiLog algorithm schedules the activities to recommend based on the subtraction of the values [$do - dont$]. Indeed, the activities with the highest subtraction result are recommended first if the algorithm seeks to **minimize** time. However, if the algorithm seeks to **maximize** flexibility or customization, activities with the lowest subtraction result are recommended first.

5. The CONFILOG recommendation engine through a Running example

A basic smartphone product line can be simplified as shown in figure (3). Indeed, each phone must contain a battery, a Screen, a Network sensor and a SIMReader. The ScreenColor, CameraHD, DoubleSIM reader and WaterResistance are optional. Phones containing a CameraHD must have a ScreenColor.

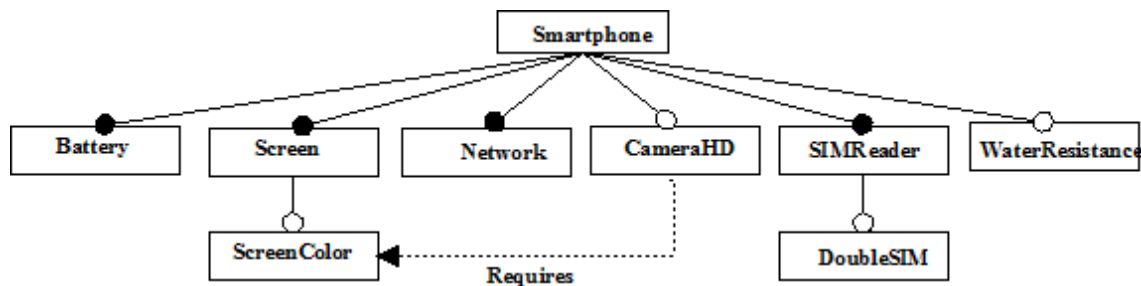


Figure 3. A basic smartphone product Line Model

The smartphone product line model is designed using the FODA notation [Kang et al., 1990]. The FODA notation dont need further division of models, all constituents present variants.

For the configuration of a smartphone from the above model, the ConfiLog method automatically propagates the mandatory variants [Battery, Screen, Network, SIMReader] and gives the choice to the stakeholder for the selection or the rejection of the following variants: [ScreenColor, CameraHD, DoubleSIM, WaterResistance].

The remainder of this section considers that the stakeholder wants a rapid configuration process τ = Performance. After process mining, the guidance dataset D is fed by a process model selecting only the process instances that answer the guidance questions of Goal 1 :

- Dataset 1 : Performance - Contribution [100 seconds] : [ScreenInColors-selected, CameraHD-deselected, Water Resistance-selected]
- Dataset 2 : Performance - Contribution [200 seconds] : [ScreenInColors-deselected, CameraHD-selected, Water Resistance-selected, DoubleSIM-deselected]
- Dataset 3 : Performance - Contribution [150 seconds] : [ScreenInColors-selected, Water Resistance-deselected, DoubleSIM-selected]

The stakeholder follows a set of configuration process steps until they finish configuring the smartphone suitable to their preferences.

In the first configuration stage, the stakeholder selects the `ScreenColor` variant. Then, the partial configuration variable p and the enabled activities variable E are updated as follows: $P = [ScreenColor-selected]$; $E = [CameraHD, DoubleSIM, WaterResist]$.

The calculation of the predicates according to the enabled activities is detailed in table (1). Indeed, the table contains three columns, a column for each enabled activity and three lines, a line for each mining answer recorded in the guidance dataset. The “T” means that there was, indeed, a decision to select or reject that enabled activity in the given mining answer. The “F” means that no decision on enabled activity was made in the given mining answer.

Table 1. Predicates of the first configuration stage

	CameraHD	WaterResistance	DoubleSIM
Dataset 1	T	T	F
Dataset 2	T	T	T
Dataset 3	F	T	T

In this case, all the processes will be weighted because they contain at least an enabled activity T. The process weighting calculation is carried out as follows in table 2:

Table 2. Useful Datasets Weighting

Dataset	Weighting Formula	Result
Dataset 1	$\frac{ [ScreenColor-selected] \cap [ScreenColor-selected, CameraHD-deselected, WaterResist-selected] }{ [ScreenColor-selected] }$	1
Dataset 2	$\frac{ [ScreenColor-selected] \cap [ScreenColor-deselected, CameraHD-selected, WaterResist-selected, DoubleSIM-deselected] }{ [ScreenColor-selected] }$	0
Dataset 3	$\frac{ [ScreenColor-selected] \cap [ScreenColor-deselected, CameraHD-selected, WaterResist-selected, DoubleSIM-deselected] }{ [ScreenColor-selected] }$	1

The expected target value when configuring each enabled activity of stage 1 is calculated as follows in table 3:

Table 3. (Do) Calculation of Enabled Activities in stage 1

Enabled Activity	(Do) Formula	Result
CameraHD	$(1 * 100 \text{ seconds}) + (0 * 200 \text{ seconds})$	100
WaterResistance	$(1 * 100 \text{ seconds}) + (0 * 200 \text{ seconds}) + (1 * 150 \text{ seconds})$	250
DoubleSIM	$(0 * 200 \text{ seconds}) + (1 * 150 \text{ seconds})$	150

As shown in table 1, the dataset 3 don't contain any configuration activity about the `CameraHD` variant. Similarly, the dataset 1 don't contain any configuration activity about the `DoubleSIM` variant. For that, these datasets will be considered for the (dont) calculation of the `CameraHD` and `DoubleSIM` enabled activities as follows in table 4 :

Table 4. (Dont) Calculation of Enabled Activities in stage 1

Enabled Activity	(Dont) Formula	Result
CameraHD	$(1 * 150 \text{ seconds})$	150
WaterResistance	0	0
DoubleSIM	$(1 * 100 \text{ seconds})$	100

The difference between the functions [do - dont] for the CameraHD, WaterResistance, DoubleSIM is respectively : 50, 250 and 50.

The ConfiLog algorithm recommends variants with the **highest** subtraction result [do - dont] first because it tries to **minimize** the duration. Indeed, for the stage 1, t the recommended process order is : [Select : WaterResistance < Deselect : CameraHD ≤ Select: DoubleSIM]

For the configuration stage 2, the stakeholder don't take into account the guidance provided by ConfiLog and rejects the CameraHD variant. Then the partial configuration variable p and the enabled activities variable E are updated as follows: p = [ScreenInColors-selected, CameraHD-deselected]; E = [Water Resistance, DoubleSIM].

In this case, the three datasets must be taken into account because they contain at least an enabled activity. The process weighting calculation is carried out as follows in table 5 :

Table 5. Useful datasets Weighting

Dataset	Weighting Formula	Result
Dataset 1	$\frac{ [ScreenColor-selected, CameraHD-deselected] \cap [ScreenColor-selected, CameraHD-deselected, WaterResist-] }{ [ScreenColor-selected, CameraHD-deselected] }$	1
Dataset 2	$\frac{ [ScreenColor-selected, CameraHD-deselected] \cap [ScreenColor-deselected, CameraHD-selected, WaterResist-DoubleSIM-deselected] }{ [ScreenColor-selected, CameraHD-deselected] }$	0
Dataset 3	$\frac{ [ScreenColor-selected, CameraHD-deselected] \cap [ScreenColor-selected, WaterResist-deselected, DoubleSIM-deselected] }{ [ScreenColor-selected, CameraHD-deselected] }$	0,5

The expected target value when configuring each enabled activity of stage 2 is calculated as follows in table 6:

Table 6. (Do) Calculation of Enabled Activities in stage 2

Enabled Activity	(Do) Formula	Result
WaterResistance	$(1 * 100 \text{ secondes}) + (0 * 200 \text{ secondes}) + (0.5 * 150 \text{ secondes})$	175
DoubleSIM	$(0 * 200 \text{ secondes}) + (0.5 * 150 \text{ secondes})$	75

The expected target value when NOT configuring each enabled activity of stage 2 is calculated as follows in table 7:

Table 7. (Dont) Calculation of Enabled Activities in stage 2

Enabled Activity	(Do) Formula	Result
WaterResistance	$(1 * 100 \text{ secondes}) + (0 * 200 \text{ secondes}) + (0.5 * 150 \text{ secondes})$	0
DoubleSIM	$(0 * 200 \text{ secondes}) + (0.5 * 150 \text{ secondes})$	100

The difference between the functions [do - dont] for the WaterResistance and DoubleSIM is respectively : 175 and 25.

For the stage 2, the recommended process order for this configuration step is : [Select : WaterResistance < Select: DoubleSIM]. The stakeholder selects the WaterResistance variant. For the configuration stage 3, there is only one variant to recommend (DoubleSIM). The stakeholder configures it and finalizes the configuration process.

6. Experiments planning

To evaluate the CONFLOG method, Empirical Studies With Students (ESWS) [Booth et al., 2003] were conducted. As a matter of fact, ESWS help researchers to gain insight into methods. Like other empirical studies, ESWS can be useful to industry and research communities especially when they are carried out in an adequate manner, meet appropriate objectives and do not overestimate the results. Therefore, ESWS are used to obtain

preliminary evidence and to confirm or disprove a research hypothesis. In this research, ESWS aim to answer the following Experiment Research Questions (ERQ) [Booth et al., 2003]:

- **ERQ1:** *Experts are studying student's actions during the configuration of a bike product line, because they are trying to find out how the configuration process becomes an error prone task, in order to help readers to better understand how useful to provide guidance.*
- **ERQ2:** *Experts are studying student's actions during the configuration of a bike product line because they trying to find out how it's possible to mine and analyze the process shortcomings, in order to help readers to better understand how to design a configurator that can assist the configuration process.*

This experiment involves two main Processes, aiming to collect and analyse expectations, traces of configuration of a bike product line and feedbacks from students :

- **(P1)** Who didn't use a particular method for the guide the configuration process.
- **(P2)** who use the ConfiLog method to guide the configuration process.

The ConfiLog method was implemented under a product line configurator ^{1/2} that will be used in (P1) and (P2).

(P1) feeds the ConfiLog event log repository with traces of the configuration process that will be reused in mining and recommendation calculation. Adding to that, (P1) addresses the cold start problem of the recommendation. However, (P2) measures the level of student satisfaction and evaluates the qualities, failures and deficiencies of the ConfiLog guidance.

To achieve (P1) and (P2), this experiment was made with the second-year miashs license students at Panthéon Sorbonne Paris 1 University in France and the first-year applied license students in computer science at ISET Nabeul in Tunisia. Students were divided into pair groups. All the pairs, having contributed to this experiment, took part in (P1) and also in (P2). Each pair represents a "Baseline" student and an "expert" student. The "Baseline" student configures the bike according to his preferences without using any particular guidance method (P1), first, and using ConfiLog guidance (P2), second. However, the "Expert" student explains to his colleague the particularly technical choices (for example the inch unit, the different types of handlebars, the difference between aluminum and carbon, etc.) during (P1) and also during (P2).

The identification of "Baseline" students and "Expert" students was made by means of a pre-questionnaire³. The distribution of students in pairs aims to make the students work together on a configuration with additional tasks detailed in the experimental protocol below.

Figure (4) describes the experimental protocol and the phases carried out during the present experiment. Supervisors had to perform the following tasks:

- Contextualization: Explain to the different students the experiment context,
- The creation of student groups based on the pre-questionnaire renderings,
- Collecting bike configuration traces,
- The application of the ConfiLog method only for the configuration with guidance (P2) and
- The evaluation of the renderings of the configuration with guidance (P1) and the configuration without guidance (P2). The configuration renderings present the students' expectations, their configuration traces and their feedback.

However, students were asked to complete the following tasks:

- The answer to the pre-questionnaire allows them to identify their knowledge about bicycles and to clarify their expectations. Based on the pre-questionnaire renderings, the experimental supervisors formed the groups in pairs, namely: A "Baseline" student and an "Expert" student.
- The configuration of a bicycle in pairs. The "Baseline" student makes configuration decisions which are monitored and supervised by the "Expert" student.
- The answer to the post-questionnaire⁴ once the configuration is finalized. Questionnaire renderings and student actions throughout the bike configuration are capitalized to evaluate the ConfiLog method later.

¹ <https://github.com/HoussemCHEMINGUI/ConfigLogFFV>

² <https://www.youtube.com/watch?v=Ds69w86GCz0>

³ <https://docs.google.com/forms/d/e/1FAIpQLSd0BSe205rJuBa5Nt--i6rqKbJbN-d7rU0zhXjJJ6APkx0jA/viewform>

⁴ https://docs.google.com/forms/d/e/1FAIpQLSeobHu9Ba_geejniieSGvcSynYDZPtZfUPSvmyRRhe4CG0do90/viewform

Both the pre questionnaire and the post questionnaire were designed using a Likert scale [Likert, 1932]. This psychometric scaling method contains a set of items in a five-point format: (1) strongly disagree, (2) somewhat disagree, (3) neither agree nor disagree, (4) slightly agree, and (5) strongly agree. The pre questionnaire is used to identify the background and general profile of the students in order to distinguish any differences between them concerning their basic skills. This will confirm their knowledge about the configuration and their familiarization. A post questionnaire measuring the satisfaction level of all groups was provided. The post questionnaire similarly adopts the Likert scale to assess the experience of students who have used the bike configurator.

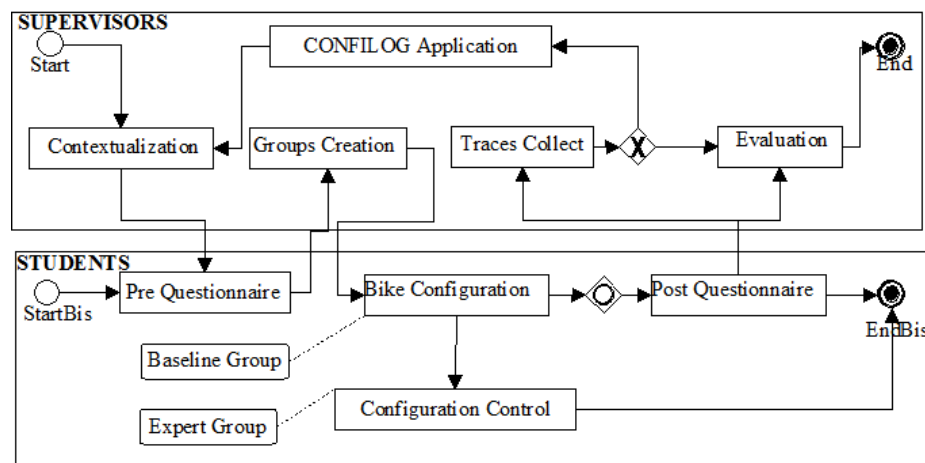


Figure 4. Experimental Protocol and phases

The experiments were carried out in classrooms, a familiar environment for the students, with the aim of minimizing external and environmental factors that may threaten the validity of the results. Students performed treatments without being influenced by a colleague with inferior or superior knowledge. Furthermore, students had limited access to the internet in order to minimize distractions (instant messaging, email, etc.). Students were free to undo their configuration decisions already made.

The Baseline students were invited to configure a bike according to the instructions of the contextualization. The model of bikes to configure is detailed in Figure (5). The specification of the bike product line imposes that the student makes decisions on variants associated with variability points. For example, for the variability point `Frame Type` with a `[1..1]` cardinality, the student is invited to select only one variant among several, in particular, the `Diamond frame`, the `Step-Through frame`, or the `Triathlon frame`. Furthermore, for the `handlebar` variability point, the student is invited to select only one variant among several, including the `Drop`, `Pursuit` or `Riser handlebars`. Decisions on these variants are exclusive, only one variant is allowed. However, for other variability points with a `[0..*]` cardinality, the student is allowed to select one or more variants. For example for the `Accessories` variability point the student can select or refuse a bike with `Mudguards`, a `Kask Road`, an `Air Pump`, a `bike cover`, a `water bottle`, a `big pannier` or a `small pannier`. Indeed, the product line model of bikes, distinguishes mandatory and optional variability points. Mandatory variability points are in solid line triangles, like the `fork`, however, optional variability points are in dotted line triangles like `Theft Preventions`. Finally, the variants of the Bike product line model are interdependent according to domain constraints. For example, a student who has a `Budget`, less than 500 coins deprives him to configure a bike with accessories. This kind of dependency is specified via the constraint type `Excludes`. In addition to that, a student who has selected a bike with `light` must necessarily select an `energy source`, which can be `Dynamo`, a `Battery` or even a `solar panel`. This kind of dependency is specified via the `requires` constraints.

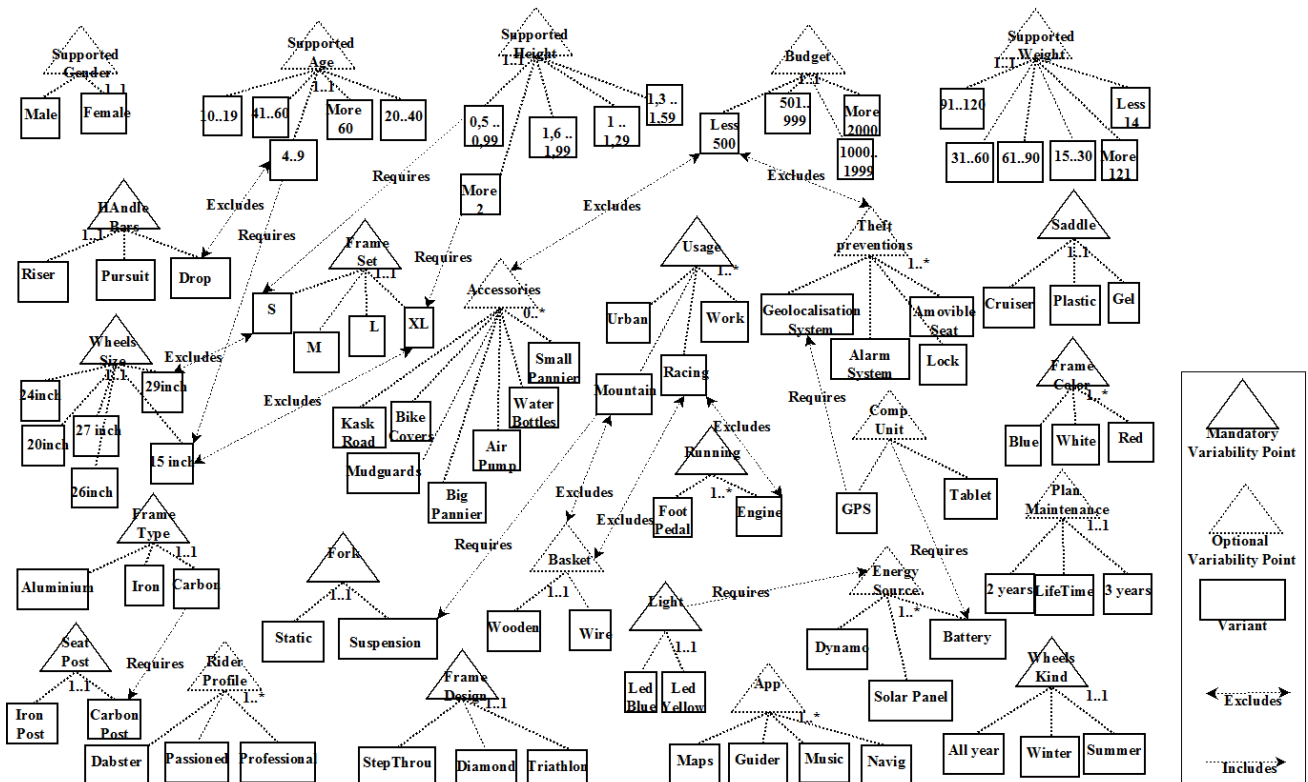


Figure 5. The bike product line model

The bike product line model is designed with the OVM notation [Pohl et al., 2005]. Indeed, it contains 116 variants including 17 mandatory variants, 14 optional and 85 grouped variants with different cardinalities. Moreover, 16 domain constraints are specified including 8 exclusion constraints (Excludes) and 8 inclusion constraints (Requires). The number of variants to configure is not revealed to the students but it is equal to the number of optional variants grouped in cardinality. The students were only called upon to gradually build a configuration of a bike, which goes with their preference, through a set of iterations.

The multiplication of the possible solutions for each Variability Point (VP) is equivalent to the set of possible solutions from the model. The number of possible combinations (C) of the variants associated with a given (VP) is given by the formula below. However, only discernible variants were considered. In addition, the repetition of the selected variants, their permutations and their order of selection are not taken into account.

$$VP1 \text{ Max} \\ VPn \text{ Min} \prod \sum C_n^p = \frac{n!}{p!(n-p)!}$$

VP1 : First Variability Point

VPn : Last Variability Point

Min : Minimal Cardinality

Max : Maximal cardinality

C : Possible Combinations

p : Number of variants to select

n : Number of variants of a variability point

Actually, the bike product line model leads to more than a billion possible configurations, however, they are not all valid, hence the need for configuration guidance. The unguided configuration does not propagate decisions linked to a given domain constraint. Unguided configuration lets the stakeholder configure the entire line and they check the

consistency of the configuration a posteriori. However, performing valid configurations in a solution space that leads to over a billion possible configurations seems absurd. All in all, it was decided to help the students who participated in the experiment by providing them with knowledge-based configurations. Indeed, the ConfiLog configurator automatically propagated student decisions. The decision's propagation reduces the solutions space and allows coherent decisions to be made useful to produce guidance with the ConfiLog method in (P2).

From the traces collected in (P1) and the mining carried out by the ConfiLog method and its recommendation engine, guidance was produced to guide the students in (i) the sequence of the configuration process, (ii) the assignment values to the variants of the bike product line model according to the stakeholders respective goals.

7. Evaluation

The evaluation details the backgrounds of the students and their expectations regarding the configuration of a bicycle. Then, the results of the two experimentation processes are compared and analyzed in order to justify the added value of the ConfiLog method guidance. Finally, student feedback is assessed as well as execution time and upscaling measures.

The students, who participated in this experiment, were asked about their knowledge and expectations regarding bikes and online configurations.

First of all, 58.3% of the participants are male students and 41.7% are female students. They have heterogeneous knowledge of bikes. Indeed, 61.1% claim that they are familiar with the particularities, the inclusive and exclusive decisions that bikes can impose; for example, energy is mandatory for lighting and several energy sources can be used such as batteries, dynamos, solar panels, etc. The rest of the students admit that they don't have clear ideas about that. Furthermore, 27.3% expressed their familiarity with the configuration process, 12.2% were neutral and 60.4% think that they have minimal knowledge of how it actually works.

In turn, student expectations of the ease of the configuration were different. Indeed, 59.7% expect that the bike configuration will be easy versus 40.3% who expect a confusing configuration. However, the students' expectations related to (i) the configuration rapidity, (ii) the flexibility of decisions and (iii) their assistance in making simple and consistent decisions, were firm and decisive. In fact, 62.5% of students expect that the bike configuration process will not take more than 10 minutes. 93.5%, moreover, expect to be able to make decisions and change them later. Adding to that, 51.1% think that the configurator will assist their decisions in order to not get lost. Nevertheless, 77% of students say they already have an idea about the suitable bike.

The expectations of the students support the hypothesis of a need for guidance assumed by the ConfiLog method, moreover, their backgrounds were exploited for the formation of the groups.

A total of 66 student pairs participated in the unguided configuration (P1) and 113 pairs participated in the guided configuration (P2). During the unguided configuration (P1), to configure 66 bikes, 19118 events were produced leading to only 36 full bike configurations. However, in the guided configuration (P2), to configure 113 bikes, 11107 events were produced leading to 100 full bike configurations. With the guidance provided to students, the average time for a configuration process is reduced by 81% ; 2.7 minutes instead of 21.9 minutes. Also, the median duration of a configuration process moves from 255 seconds to 115 seconds, a reduction of 45%.

In addition, it turns out that the guidance “forges” the decisions to be made. In fact, during (P1) 251 different decisions were made to produce 66 configurations. On the other hand, during (P2), 246 different decisions were made to produce 113 configurations. The minimum number of events makes it possible to identify complete configurations. In (P1), the same decision is made on average 76.17 throughout the event log. However, in (P2) the same decision is made on average 45.15 of the decisions in the entire event log. The decrease in this percentage reflects that the guidance provided is volatile and does not reuse standard guidance stereotypes. The median frequency confirms the adaptability and diversity of the guidance provided.

Adding to that, the guidance provided to the 113 pair students was based on the goals. 71% of students chose to perform rapid configurations. Typically, rapid configurations focus on minimal manual choices by students and which take minimal time to achieve a finite and valid configuration. 19% of students chose to perform flexible configurations. Typically, flexible configurations focus on manual choices made by students and which do not induce backtracks to achieve a finite and valid configuration. Finally, 10% of students chose to perform custom configurations. Usually custom configurations focus, first, on model variables related to the student's profile, and

second, on the student's history if they have. The configuration process model of (P1) and (P2) are attached in the links at the bottom of the page⁵.

After configuring the bike, the students were invited to a post-questionnaire summarizing their feedback. Student feedback aims to draw positive and negative lessons from the guidance provided. 11.54% of students confirm that the configurator crashed at least once during (P1), however, during (P2) all students confirm that the configurator didn't crash. Technically, this performance is reflected in the reduction of undo choices. Therefore, the configurator takes much less time for driving and propagating student decisions. The experience feedback shows that 26.92% of the students in (P1) did not encounter any difficulties in building a coherent configuration. In return, 73.45% of the students of (P2) did not find any problem. 93.8% of students in (P2) found that the guidance was relevant to meeting their guidance goal. The post-questionnaire also includes questions related to the feelings and motivation of the students throughout the configuration of the bike. Indeed 84.62% of the students of (P1) find that the decision-making was hyper monotonous. However, 43.36% of (P2) students find that the bike configuration process was attractive. Optimizing this component requires extensive considerations of interactivity and ergonomics of the configuration interfaces. Human-Machine Interface drivers [Wang et al., 2018] can be adopted to an optimal decision-making process.

The ConfiLog method still requires revisions, especially once incorporated into a commercial context, where the functional and non-functional requirements of configurators are vital. Indeed, 18.58% of (P2) students expect even more guidance.

After the first ConfiLog configurator runs, an application bottleneck related to the time required for the guidance calculation was identified. The left part of Figure (6) confronts graphically two curves of the time required for the execution of the same configuration decisions. The lower curve shows the execution time required when configuring in (P1), however, the upper curve shows the times required when configuring in (P2). These measurements are performed using a laptop computer with the following configuration:

- 64-bit Windows 7 Professional,
- An Intel (R) Core™ i5-4200M 2.50 GHz processor and
- A RAM memory of 8.00 GB.

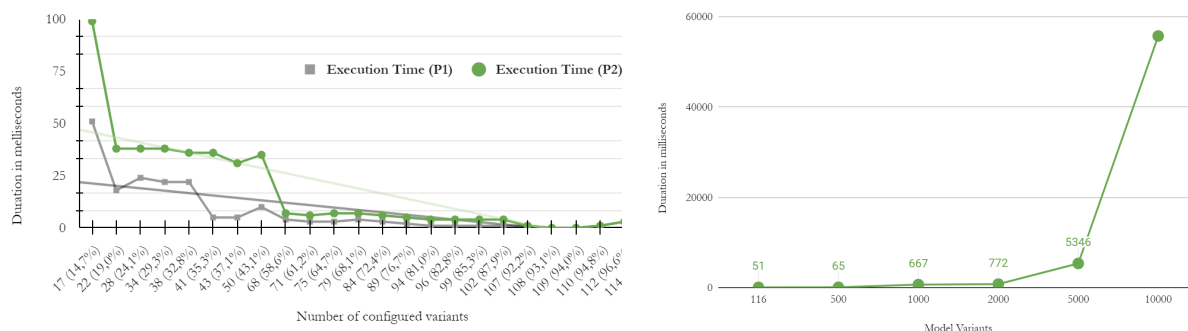


Figure 6. Execution time measurements

The graph shows that the execution times required during (P1) and those required during (P2) are similar, however, (P2) takes an additional time reflecting the time required for the calculation of the recommendation. As decisions are made, ConfiLog's recommendation engine queries the datasets, calculates process instance weights and decisions impacts in accordance with the guidance goal expressed by the stakeholder.

The first configuration stage takes the longest execution time. Technically, the first configuration stage leads to consistency calculations related to all the model variants. The execution time of stage 1 in (P2) doesn't exceed 99 milliseconds so remains reasonable. However, the execution time for the last configuration step was one millisecond. This reflects the fact that as the configuration process progresses: (i) the partial configuration is updated, (ii) the solution

⁵ Link : <https://www.dropbox.com/sh/06n9dc59txi55u0/AADOXdMmd9BBpP29FMdxNH5pa?dl=0>

space is reduced and (iii) the datasets violating the partial configuration are ignored. Moreover, a free fall of the curve is observed in stage 2 as the configurator has gone from 17 to 4 constraint propagations.

Since ConfiLog relies on both knowledge-based recommendation and process mining, execution time primarily depends on two criteria including, (i) the product line size and (ii) the trace repository size. However, for ConfiLog's performance at scale-up, the present experiment does not have process traces for the different models compared. The right part of figure (6) shows the evolution of the execution time when the number of models variants increases. The curve shows that execution time remains reasonable, (less than one second), even for variability models containing 3000 variants. However, the execution time increases exponentially from 5,000 variants. Exceeding the threshold of models containing 3000 variants, the performance decreases.

8. Related Works and discussions

The shortcomings of the product line configuration process have been addressed by the software engineering community [Chemingui et al., 2019] [Noorian et al., 2017] [Sayyad et al., 2013] which has proposed a variety of solutions such as, interactive configuration methods [Benavides et al., 2010], [Hadzic et al., 2004], recommendations based methods [Jannach et al., 2011], heuristic methods [Mazo et al., 2014] and Mining based methods [Varela-Vaca et al., 2019].

In interactive methods, product line specifications are mapped into an executable language, for instance, finite domain constraint programs [Salinesi et al., 2011] and propositional formulas [Batory et al., 2005]. The goal of this transformation is to automatically propagate user decisions and provide a valid solution when there is only one possible choice. Our work with companies [Dumitrescu et al., 2013] revealed that this is very efficient to help stakeholders to identify inconsistent decisions, but that the probability of inconsistent decisions was so high that this approach is virtually impossible to use when stakeholders don't already have a deep expert knowledge of the systems to configure.

In recommendation-based methods, configurations are mainly based on "content-based" filtering [Mooney et al., 2000], and/or "collaborative" filtering [Linden et al, 2003]. For instance, [Zhang, 2013] uses data mining to recover complex feature correlations and association rules in order to make recommendations. Other works, such as [Triki et al., 2014] propose an interactive configuration by organizing the configuration process in a series of stages where decisions are made using recommendations based on the similarity between former decisions and all other decision processes that were made before. This approach has 2 important drawbacks. First, it suffers from the cold-start issue, which is typical of all collaborative filtering strategies. Second, it creates cognitive reinforcement : stakeholders receive recommendations that correspond to their way of working, rather than suggesting other potentially more adequate or efficient approaches.

In heuristic-based methods, stakeholders are guided throughout the configuration by using heuristics that aim to reduce the number of configuration actions and/or minimize impact of decisions [Mazo et al., 2014]. Heuristics are intended to help stakeholders to specify the characteristics of their products step-by-step according to their requirements. The main interest of this research is to improve the configuration performance by avoiding inefficient decisions.. However, other configuration issues such as the guidance customization seem conscient but not overcome. This approach does not recommend a process but it recommends features of a product using heuristics rather than similarity with past configurations.

Mining-based configuration methods group data mining based methods and process mining based methods. The process mining is at the crossroads of (i) the data approach including structuring, centralization and visualization and (ii) the process approach including the modeling and execution of the event sequences.

Configuration methods based on data mining [Chastel et al., 2015] [Zhang, 2013] [Witten and Frank, 2002] use data mining to improve future configurations. Usually, this kind of method captures configuration traces in order to reuse them by identifying correlations of complex variants and association rules to produce recommendations. Using artificial intelligence algorithms, these methods suggest the next action to stakeholders when they build a configuration. To do so, configuration traces are categorized using clustering metrics to determine which cluster is closest to a given configuration. However, these methods also provide stakeholders with suggestions on a single action to take, not a series of actions to take. In addition, the suggested action is based on its frequency in the log and does not take into account any guidance goal. In configuration methods based on process mining, [Varela-Vaca et al., 2019] recommend a configuration "workflow", in other words, the order and sequence in which the various configuration items are presented to stakeholders. To do so, configuration actions are captured and subsequently reused to determine the configuration workflow that better matches with the capitalized traces. However, the workflow proposed in the next configurations

considers only the frequency of the product line variants, which means that the sequence order to be proposed is standard and does not take into account the heterogeneous preferences of stakeholders.

Other methods propose to deal with complex configurations and provide intelligent configuration processes, the notions of views and workflows have been defined [Hubaux et al., 2009]. This kind of guidance is provided especially to handle collaborative configurations that involve different users with different concerns. Nevertheless, this method presents limits since it doesn't consider user intentions and doesn't focus on process execution behaviors. In fact, proposed guidance solutions are standard; and do not take into account the variety of support needs and situations faced by the configuration collaborators. Moreover, it provides limited choices that make users face inflexible processes; that may suggest variants values but never a configuration process path and order.

Revealed limits motivated the idea of adopting a process mining strategy that aims to analyze the whole process of executions. The purpose is to guide future configurations by suggesting, at each stage, decision instances on the one hand, and decisions order, on the other hand. All guidance solutions take into account a guidance goal, a process state and a process model of previous configurations.

Table 8 compares related works using the following criteria :

- **Guidance consistency:** Even in large and complex models with interdependent variants, the way of perceiving the configuration process must ensure guidance consistency. In fact, decisions have to be propagated and will never lead to recommendations that violate the domain constraints.
- **Guidance flexibility :** Throughout the configuration process, the stakeholder feels free to undo choices and modify previous decisions already made.
- **Guidance dimension :** The guidance of the configuration process must consider the configuration process order on the one hand and the variant value at each configuration stage on the other side.
- **Guidance independence :** In some cases, the stakeholder is implicitly guided by the conceptual hierarchy of the product line model such as the FODA notation [Kang et al., 1990]. Indeed, the variants positioned in the upper level of the model (roots) are configured first. Then, the stakeholder proceeds to configure the variants positioned in the lower level (sheets until leaves). Other implicit forms of guidance can lead the stakeholder to configure the root and its leaves and then move on to the next root and its leaves. The produced guidance must ignore the product line model hierarchy and consider orthogonal variability such as the OVM notation [Pohl et al., 2005].
- **User intention :** Faced to a panoply of user intentions, different ways of guiding the stakeholder prove to be important. distinguishing user intentions leads to a better guidance customization.
- **Guidance Portability :** Other configurable software such as ERP, and COTS aim to boost and improve their configuration process. Portable guidance that covers other configuration fields are welcome.

Table (8) Related works comparison

Related Works	Comparison criterion					
	Consistency	Flexibility	Dimension	Dependency	User intention	Portability
COLOSSI [Varela-Vaca et al., 2019]	N-A	Strict	Stepped Values	Process mining Analysis	Ignored	Limited
WeaFQAs [Horcas et al., 2018]	Priori	Strict	One shot values	PLM Syntactic	Ignored	Generic
SATVaEA[Xiang et al., 2018]	Priori	Strict	One shot values	Systematic reasoning	Ignored	Limited
QcCP[Noorian et al., 2017]	Priori	Flexible	One shot values	PLM Syntactic	Considered	Limited
Agile Tames Product Line Variability[Hayashi et al., 2017]	Priori	Strict	Stepped values	N-A	Ignored	Generic
Extended FeatureIDE [Pereira et al., 2017]	Priori-Posteriori	Flexible	Stepped values	Systematic reasoning	Considered	Generic
ZEN-FIX[Lu et al., 2016]	Priori	Strict	One shot values	Systematic reasoning	Ignored	Limited
Extended pure::variants [Sinnhofer et al., 2016]	Posteriori	Flexible	Stepped values	N-A	Considered	Generic

Extended VariaMos[Mazo et al., 2014]	Priori	Flexible	One shot Values	PLM Syntactic	Ignored	Limited
Recodyn [Triki et al., 2014]	Posteriori	Strict	Stepped Values	Systematic reasoning	Ignored	Limited
VITAL[Zhang 2013]	Posteriori	Strict	One shot Values	Correlation analysis	Ignored	Limited
IBEA Configurations [Sayyad et al., 2013]	Priori	Strict	One shot values	PLM Syntactic	Ignored	Limited
SPAF[Guana et al., 2013]	Posteriori	Strict	One shot values	Systematic reasoning	Ignored	Limited
CO-OVM[Dumitrescu et al., 2013]	Posteriori	Flexible	Stepped values	PLM Syntactic	Ignored	Limited
FAMILIAR [Acher et al., 2013]	Posteriori	Strict	One shot values	Systematic reasoning	Ignored	Limited
GEARS [Krueger et al., 2013]	Priori	Strict	One shot values	Systematic reasoning	Ignored	Limited
SpineFM[Urli et al., 2014]	Posteriori	Flexible	One shot values	Data Mining analysis	Ignored	Limited
PLiC[Elsner et al., 2010]	Priori	Strict	One shot values	PLM Syntactic	Ignored	Limited
SPLOT[Mendonca et al., 2008]	Priori	Strict	One shot Values	PLM Syntactic	Ignored	Limited
Genarch [Cirilo et al., 2008]	Posteriori	Strict	One Shot Values	Systematic reasoning	Ignored	Limited
VISIT-FC[Botterweck et al., 2007]	Posteriori	Strict	One shot values	PLM Syntactic	Ignored	Generic
COVAMOF [Sinnema et al., 2006]	Priori	Strict	Views	PLM Syntactic	Ignored	Generic
ReMAP [Schmid et al., 2006]	Posteriori	Strict	Ont shot values	Decision Models	Ignored	Generic
CONFILOG	Priori	Flexible	Process Order & variant values	Context-dependent	Goal-based guidance	All Configurable software

To do so, the CONFILOG method processes the configuration using integer constraints [Salinesi et al., 2010] which conducts the process with a zero margin of error. Event logs from which the process mining is conducted are consistent. Consequently, the CONFILOG recommendation engine never provides recommendations that don't meet the domain constraints. Once the stakeholder undo choices, the CONFILOG method recalculates the consistency of the configuration and the relevance of the recommendation.

At each configuration stage, CONFILOG calculates a new order for the rest of the variants not configured yet. The dynamic variant ordering takes into account the partial configuration and the guidance goal expressed by the stakeholder. Besides the guidance order, CONFILOG provides recommendations for variant values. The recommendation algorithm retrieves information to recommend the selection, rejection or ignorance of a given variant.

Furthermore, unconscious influence by the PLM hierarchy is taken into account by CONFILOG so that the recommendation of the order of the variants is orthogonal and insensitive to the tree structure of the model.

CONFILOG distinguishes three main guidance goals : performance, flexibility and customization of the configuration process. For each guidance goal, guidance questions are asked. The answer to these questions is given through a process mining strategy.

- **Configuration process performance** : Performance is a very important factor to which competitive configurators must respond. To succumb to this, CONFILOG recommends values and a variant order which accelerate the configuration process. For that, configuration process models that don't contain undo choices and backtracks are reused.
- **Configuration process flexibility** :Flexibility must be ensured in a configuration context. Indeed, the stakeholder is faced with a wide range of decisions leading to a monotonous experience. As a solution, the recommendations avoid using traces where there were “problems”. In addition, the process will be animated by several configuration suggestions; for example, recommending fashionable products or which can be reached in a limited number of steps.

- **Configuration process customization** : customization is a very important aspect of product line engineering. To reinforce this aspect, the CONFIOLOG method aims to recommend variants taking into account the history of the stakeholder in question, his previous intentions, "best sellers" etc.

The CONFIOLOG method is applicable to any product line model independently of its variability notation. Experiments demonstrated the portability of the CONFIOLOG method to support all configuration processes based on large and complex models. Ongoing works are focusing on applying the CONFIOLOG method to other configurable software such as ERP and COTS. [Chemingui et al., 2020]

9. Conclusions and Future Works

This paper aims to answer the following research question: **How to guide and schedule stakeholder decisions throughout the configuration process of product lines?** In response to this research question, a new method called ConfiLog is proposed to guide the product line configuration process. ConfiLog supports stakeholders in making decisions during the configuration process and provides guidance directed by goals. The ConfiLog guidance is dynamic. Indeed, the guidance is bidimensional; the order of configuration process decisions and the decisions instances at each configuration stage. Based on process traces and interactively with the stakeholder, ConfiLog generates knowledge and process mining recommendations. ConfiLog reasons on a subset of variants (called partial configuration) and a guidance goal expressed by the stakeholder. Guidance is calculated by mining traces of previous line configuration processes and a recommendation algorithm that enables interactive guidance orchestration. ConfiLog considers insufficiencies of existing guidance solutions for the software configuration especially (i) the configuration consistency, (ii) the stakeholder freedom, (iii) the dual dimension of guidance (Decision orders and their instances), (iv) the independence of the guidance to the tree structure of the product line model, (v) the goals consideration, (vi) the incrementality and completeness of the solution and (vii) the portability of guidance.

The ConfiLog method evaluation was carried out based on an empirical study with students. In the first experiment process, students were invited to configure a bike product line without ConfiLog guidance, however, in the second experiment process, students were invited to configure a bike using the ConfiLog guidance. The evaluation carried out is based on a comparison of the configuration traces of students in the first and the second process. In addition, stakeholder feedback was considered.

Several scientific questions arise concerning the consequences and the disadvantages of ConfiLog for the practitioners. Generally, product line models evolve over time [Dhungana et al., 2010], consequently, the requirements, the implementations, the tests, etc evolve too. Indeed, new variants and / or domain constraints can be incorporated into the new product line model. Therefore, ConfiLog guidance guidelines could not avoid the discrepancy between what is capitalized in the tracks and what will contain the new product. Technically, guidance based on the previous version of the line becomes incompatible with the new specifications of the line model. On the other hand, the guidance produced by the ConfiLog method is always directed by a single goal expressed by the stakeholder before starting the configuration process. However, this deprives the stakeholder from changing ideas during the configuration. For example, a user who goes to an online car store and wants to perform a flexible configuration. After suggesting many decisions, the stakeholder decides to change the guidance goal to configure rapidly the remaining variants. In addition, sometimes, stakeholders are sure which guidance goal suits their requirements or whether a combination of goals seems interesting to them. In perspective, the consideration of situational methods [Kraiem et al., 2010], intentional methods [Rolland et al., 2010] and / or contextual methods [Frederiksen et al., 2008] in the ConfiLog method seems able to enhance the guidance quality. In addition, the integration of these methods increases the freedom of the stakeholders. Indeed, stakeholders will not never be burdened by an only guidance goal so they can switch from a guidance goal to another.

Acknowledgements

This work is supported by the French–Tunisian CMCU project 16G/1416 called CONFIGURE.

REFERENCES

[Aalst, 2011] Aalst, W. (2011). Discovery, Conformance and Enhancement of Business Processes.

- [Acher et al., 2013] Acher, M., Collet, P., Lahire, P., & France, R. B. (2013). Familiar: A domain-specific language for large scale management of feature models. *Science of Computer Programming*, 78(6), 657-681.
- [Batory et al., 2005] Batory D. (2005) Feature Models, Grammars, and Propositional Formulas. In: Obbink H., Pohl K. (eds) *Software Product Lines. SPLC 2005. Lecture Notes in Computer Science*, vol 3714. Springer, Berlin, Heidelberg.
- [Basili et al., 1994] Basili, V. R., Caldiera, G., & Rombach, H. D. (1994). The Goal Question Metric Approach, Chapter in *Encyclopedia of Software Engineering*. Wiley.
- [Benavides et al., 2010] Benavides, D., Segura, S., & Ruiz-Cortés, A. (2010). Automated analysis of feature models 20 years later: A literature review. *Information systems*, 35(6), 615-636.
- [Berander et al., 2006] Berander, P., & Jönsson, P. (2006, September). A goal question metric based approach for efficient measurement framework definition. In *Proceedings of the 2006 ACM/IEEE international symposium on Empirical software engineering* (pp. 316-325).
- [Booth et al., 2003] Booth, W. C., Booth, W. C., Colomb, G. G., Colomb, G. G., Williams, J. M., & Williams, J. M. (2003). *The craft of research*. University of Chicago press.
- [Botterweck et al., 2007] Botterweck, G., Nestor, D., Preußner, A., Cawley, C., & Thiel, S. (2007). Towards supporting feature configuration by interactive visualisation.
- [Carver et al., 2010] Carver, J. C., Jaccheri, L., Morasca, S., & Shull, F. (2010). A checklist for integrating student empirical studies with research and teaching goals. *Empirical Software Engineering*, 15(1), 35-59.
- [Chastel et al., 2015] Chastel, F., Blay-Fornarino, M., Pereira, C. D. C., & Urli, S. (2015). Apprendre des actions utilisateur pour guider le processus de configuration dans les lignes de produits logiciels.
- [Chemingui et al., 2019] Chemingui, H., Gam, I., Mazo, R., Salinesi, C., & Ghezala, H. B. (2019). Product Line Configuration Meets Process Mining. *Procedia Computer Science*, 164, 199-210.
- [Chemingui et al., 2020] Chemingui, H., Gam, I., Mazo, R., Ghezala, H. B., & Salinesi, C. (2020). A Unified Vision of Configurable Software. In *INFORSID* (pp. 93-98).
- [Cirilo et al., 2008] Cirilo, E., Kulesza, U., & de Lucena, C. J. P. (2008). A product derivation tool based on model-driven techniques and annotations. *J. UCS*, 14(8), 1344-1367.
- [Clements et al., 2011] Clements, P., Northrop, L. *Software Product Lines: Practices and Patterns*, Addison-Wesley, 2001. doi:10.1111/j.1365-2648.2011.05843.
- [Dhungana et al., 2011] D. Dhungana, P. Grunbacher, and R. Rabiser, "The DÖPLER meta- tool for decision-oriented variability modeling: A multiple case study," *Automated Software Engineering*, vol. 18, no. 1, pp. 77–114, 2011.
- [Dumitrescu et al., 2013] Dumitrescu, C., Mazo, R., Salinesi, C., & Dauron, A. (2013, August). Bridging the gap between product lines and systems engineering: an experience in variability management for automotive model based systems engineering. In *Proceedings of the 17th International Software Product Line Conference* (pp. 254-263). ACM.
- [Elsner et al., 2010] Elsner, C., Ulbrich, P., Lohmann, D., & Schröder-Preikschat, W. (2010, September). Consistent product line configuration across file type and product line boundaries. In *International Conference on Software Product Lines* (pp. 181-195). Springer, Berlin, Heidelberg.
- [Guana et al., 2013] Guana, V., & Correal, D. (2013). Improving software product line configuration: A quality attribute-driven approach. *Information and Software Technology*, 55(3), 541-562.
- [Günther et al., 2007] Günther, C. W., & Van Der Aalst, W. M. (2007, September). Fuzzy mining—adaptive process simplification based on multi-perspective metrics. In *International conference on business process management* (pp. 328-343). Springer, Berlin, Heidelberg.
- [Hadzic et al., 2004] Hadzic T., Andersen H. R. An introduction to solving interactive configuration problems. Technical Report TR-2004-49, The IT University of Copenhagen, 2004.
- [Hayashi et al., 2017] Hayashi, K., Aoyama, M., & Kobata, K. (2017, September). Agile Tames Product Line Variability: An Agile Development Method for Multiple Product Lines of Automotive Software Systems. In *Proceedings of the 21st International Systems and Software Product Line Conference-Volume A* (pp. 180-189). ACM.
- [Hubaux et al., 2009] Hubaux, A., Classen, A., & Heymans, P. (2009, August). Formal modelling of feature configuration workflows. In *Proceedings of the 13th International Software Product Line Conference* (pp. 221-230). Carnegie Mellon University.
- [Horcas et al., 2018] Horcas, J. M., Pinto, M., & Fuentes, L. (2018). Variability models for generating efficient configurations of functional quality attributes. *Information and Software Technology*, 95, 147-164.
- [Jannack et al., 2011] Jannack, D., Zanker, M., Felfering A., & Friedrich G. (2011) *Recommender Systems, an Introduction*. Cambridge University Press. New York, USA, 335 pages.
- [Kang et al., 1990] Kang, K. C., Cohen, S. G., Hess, J. A., Novak, W. E., & Peterson, A. S. (1990). Feature-oriented domain analysis (FODA) feasibility study (No. CMU/SEI-90-TR-21). Carnegie-Mellon Univ Pittsburgh Pa Software Engineering Inst.
- [Krueger et al., 2013] Krueger, C., & Clements, P. (2013, August). Systems and software product line engineering with BigLever software gears. In *Proceedings of the 17th International Software Product Line Conference co-located workshops* (pp. 136-140).
- [Likert, 1932] Likert, R. (1932). A technique for the measurement of attitudes. *Archives of psychology*.
- [Linden et al., 2003] Linden, G., Smith, B., & York, J. (2003). Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing*, (1), 76-80.
- [Lu et al., 2016] Lu, H., Yue, T., Ali, S., & Zhang, L. (2016, April). Nonconformity resolving recommendations for product line configuration. In *2016 IEEE International Conference on Software Testing, Verification and Validation (ICST)* (pp. 57-68). IEEE.

- [Mazo et al., 2014] Mazo, R., Dumitrescu, C., Salinesi, C., & Diaz, D. (2014). Recommendation heuristics for improving product line configuration processes. In *Recommendation Systems in Software Engineering* (pp. 511-537). Springer, Berlin, Heidelberg.
- [Mendonca et al., 2008] Mendonça, M., Bartolomei, T. T., & Cowan, D. (2008, March). Decision-making coordination in collaborative product configuration. In *Proceedings of the 2008 ACM symposium on Applied computing* (pp. 108-113). ACM.
- [Mooney et al., 2000] Mooney, R. J., & Roy, L. (2000, June). Content-based book recommending using learning for text categorization. In *Proceedings of the fifth ACM conference on Digital libraries* (pp. 195-204). ACM.
- [Noorian et al., 2017] Noorian, M., Bagheri, E., & Du, W. (2017). Toward automated quality-centric product line configuration using intentional variability. *Journal of Software: Evolution and Process*, 29(9), e1870.
- [Ochoa et al., 2017] Ochoa, L., Pereira, J. A., González-Rojas, O., Castro, H., & Saake, G. (2017, February). A survey on scalability and performance concerns in extended product lines configuration. In *Proceedings of the Eleventh International Workshop on Variability Modelling of Software-intensive Systems* (pp. 5-12). ACM.
- [Pereira et al., 2017] Pereira, J. A., Matuszyk, P., Krieter, S., Spiliopoulou, M., & Saake, G. (2017). A feature-based personalized recommender system for product-line configuration. *ACM SIGPLAN Notices*, 52(3), 120-131.
- [Pohl et al., 2005] Pohl, K., Böckle, G., & van Der Linden, F. J. (2005). *Software product line engineering: foundations, principles and techniques*. Springer Science & Business Media.
- [Porouhan et al., 2014] Porouhan, P., Jongsawat, N., & Premchaiswadi, W. (2014, November). Process and deviation exploration through Alpha-algorithm and Heuristic miner techniques. In *2014 Twelfth International Conference on ICT and Knowledge Engineering* (pp. 83-89). IEEE.
- [Salinesi et al., 2011] Salinesi, C., Mazo, R., Djebbi, O., Diaz, D., & Lora-Michiels, A. (2011, May). Constraints: The core of product line engineering. In *2011 FIFTH INTERNATIONAL CONFERENCE ON RESEARCH CHALLENGES IN INFORMATION SCIENCE* (pp. 1-10). IEEE.
- [Salinesi et al., 2010] Salinesi, C., Mazo, R., Diaz, D., & Djebbi, O. (2010, September). Using integer constraint solving in reuse based requirements engineering. In *2010 18th IEEE International Requirements Engineering Conference* (pp. 243-251). IEEE.
- [Sayyad et al., 2013] Sayyad, A. S., Ingram, J., Menzies, T., & Ammar, H. (2013, May). Optimum feature selection in software product lines: Let your model and values guide your search. In *Proceedings of the 1st International Workshop on Combining Modelling and Search-Based Software Engineering* (pp. 22-27). IEEE Press.
- [Schmid et al., 2006] Schmid, K., Krennrich, K., & Eisenbarth, M. (2006, August). Requirements management for product lines: extending professional tools. In *10th International Software Product Line Conference (SPLC'06)* (pp. 10-pp). IEEE.
- [Sinnema et al., 2006] Sinnema, M., Deelstra, S., & Hoekstra, P. (2006, June). The COVAMOF derivation process. In *International Conference on Software Reuse* (pp. 101-114). Springer, Berlin, Heidelberg.
- [Sinnhofer et al., 2016] Sinnhofer, A. D., Pühringer, P., Potzmader, K., Orthacker, C., Steger, C., & Kreiner, C. J. (2016, June). A framework for process driven software configuration. In *BMSD 2016* (pp. 1-8).
- [Tiwari et al., 2008] Tiwari, A., Turner, C. J., & Majeed, B. (2008). A review of business process mining: state-of-the-art and future trends. *Business Process Management Journal*.
- [Triki et al., 2014] Triki, R., Mazo, R., & Salinesi, C. (2014). Combining configuration and recommendation to enable an interactive guidance of product line configuration.
- [Urli et al., 2014] Urli, S., Blay-Fornarino, M., & Collet, P. (2014, September). Handling complex configurations in software product lines: a tooling approach. In *Proceedings of the 18th International Software Product Line Conference-Volume 1* (pp. 112-121).
- [Van Solingen et al., 2002] Van Solingen, R., Basili, V., Caldiera, G., & Rombach, H. D. (2002). Goal question metric (gqm) approach. *Encyclopedia of software engineering*.
- [Varela-Vaca et al., 2019] Varela-Vaca, Á. J., Galindo, J. A., Ramos-Gutiérrez, B., Gómez-López, M. T., & Benavides, D. (2019, September). Process mining to unleash variability management: discovering configuration workflows using logs. In *Proceedings of the 23rd International Systems and Software Product Line Conference-Volume A* (pp. 265-276).
- [Xiang et al., 2018] Xiang, Y., Zhou, Y., Zheng, Z., & Li, M. (2018). Configuring software product lines by combining many-objective optimization and SAT solvers. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 26(4), 14.
- [Young et al., 2017] Young, B., Cheatwood, J., Peterson, T., Flores, R., & Clements, P. (2017, September). Product Line Engineering Meets Model Based Engineering in the Defense and Automotive Industries. In *Proceedings of the 21st International Systems and Software Product Line Conference-Volume A* (pp. 175-179). ACM.
- [Zhan et al., 2018] Zhan Z., Zhan Y., Huang M., Liu Y. (2018) Product Configuration based on Feature Model. In: Xhafa F., Patnaik S., Zomaya A. (eds) *Advances in Intelligent Systems*.
- [Zhang et al., 2013] Zhang, B. (2013). Mining Complex Feature Correlations from Large Software Product Line Configurations.