ConfiLog Algorithm

Important : ConfiLog Guidance is a set of recommendations on two dimensions including:

- **Variants values assaignement (To select or to deselect a variant)**
- **A configuration process Order (for example Variant 1 before Variant 2 and variant 4 after Variant 3)**

We distinguish two ways of guiding in the ConfiLog algorithm:

- **Static guidance: Static guidance is offered to users in step 0 of the configuration process (no assignment of values to model variants has been made yet). It is a reading of the previous configuration process model and displaying static results (that do not change during the process) that correspond to a type of guidance chosen by a user.**

- **Dynamic Guidance: Dynamic guidance is provided in a calculated way at each configuration step. Depending on the partial trace (assignment of values to the model variants by a user), the ConfiLog algorithm recalculates the guidance.**

The user can express his guiding type preferences :

- **The rapidity of the configuration process,**
- **The flexibility of the configuration process,**
- **The customization of the configuration process.**

Static Guidance (Partial = 0)	**Dynamic Guidance**
The rapidity of the configuration process,	
• Process order and variant values of a fast configuration.	Algorithm
The flexibility of the configuration process,	

• Process order and variant values of a fashionable configuration. • Process order and variant values of a best seller configuration. • Process order and variant values of a Full options configuration. • Process order and variant values of a Min options configuration.	Algorithm
The customization of the configuration process.	
• Process order and variant values of the last configuration of a given user products • Process order and variant values of a Best sellers configuration for a given category (Men / Women) (Summer / Winter) (Students / Children) ...	Algorithm

Dynamic Guidance (Calculated based on a partial configuration)

The recommendation algorithm ConfiLog makes it possible to generate recommendations according to:

- A guiding goal given by the user (rapidity, flexibility or customization)
- A partial configuration

The recommendations for an allowed activity provide predictive information based on the mined traces with their target value (rapidity, Flexibility or Customization) and interpreted in process models.

Par exemple

- the mined trace "x" lasted 500 seconds
- the mined trace "y" has a flexibility impact = 15
- the mined trace "z" has a customization impact = 14

The duration and impact values are retrieved from the process mining and stored in the guidance dataset.

The ConfiLog recommendation service requires the presence of this guidance dataset that contains information about processes that have been executed, mined, and interpreted in process templates stored in xml files.

This algorithm does not take into account process mining. Process mining is considered as a black box that produce the guidance dataset and evaluations such as frequency and performance.

1. Preliminaries

Let's suppose that A is a set of activities. A^* is a set on finite sequence on A.

A mined trace $\sigma \in A^*$ is a sequence of finite activites, given by the function $\sigma:\{1,...,n\} \rightarrow A$, with $|\sigma|=n$ is the size of the sequence.

The sequences are noted as $\sigma=\{a1, a2,...,an\}$, where $\forall 1 \leq i \leq n \ \sigma(i)=ai$.

In the logged traces, we define the member operator to identify whether a given activity is included in the mined trace.

Ps: A mined trace is a configuration process that took place and able to answer a given guidance question.

Définition 1 (Member operator of a mined Trace).

Let's $\sigma:\{1,...,n\} \rightarrow A$ and $\sigma':\{1,...,m\} \rightarrow B$ are traces with $\sigma=\{a1,a2,...,an\}$ et $\sigma' = \{b1,b2,...,bm\}$.

Member: a ∈ σ ⇐⇒ ∃1≤i≤n ai = a

The "member" operator is made to identify the membership of an activity of a partial trace in a mined trace. An activity is the fact of selecting or rejecting a given variant.

A guidance dataset contains process models that have detailed traces of previous configurations.

For each guidance purpose our dataset contains selected traces during the process mining.

- Fast process traces
- Flexible process traces
- Customized process traces

For each (ideally) guiding question, we have responses presented as a set of process steps with variant values.

- Fast processes = Answers to questions of rapidity
    - Processes that were executed rapidly <mean duration
    - Processes that were executed without backtracks?
- Flexible processes = Answers to questions of flexibility
    - Processes that lead to a valid product (safest ones)
    - Processes that include fashionable products
    - Processes that include the best seller products
    - Processes that include full option products
    - Processes that include min option products
    - Processes that are executed with a maximum number of manual (user) choices
    - Processes that are executed with a maximum number of automatic propagations
- Customized processes = Answers to questions of rapiditycustomization
    - Processes of the specific user based on his history
    - Processes that include the best seller products for men
    - Processes that include the best seller products for women
    - Processes that include the best seller products for age children
    - Processes that include the best seller products for adults
    - Processes that include the best seller products in summer
    - Processes that include the best seller products in winter

---

Definition 2 (Guidance Dataset).

Let's suppose A a set of activities. A guidance dataset D ∈ β(A*) is a multi-set of mined traces (each answering a given guiding question) referring to A activities.

Remember that each recommendation contains predictive information based on the user's goal. For the moment, we suppose that this objective can be captured by a user input that introduces his choice on a given type of guidance (rapidity, Flexibility, Customization).

In return, the guidance dataset contains traces that are mined and interpreted (answering given guidance questions). Each mined trace responds to a given target function.

Definition 3 (Target function).

Let's Suppose that A is a set of activities and D ∈ β(A*) is a guidance Dataset on A.

A target function :

τ:D→R is a function that attaches a target value for each trace in the Guidance Dataset.

At the same time during a configuration, τ can be parameterized by a user = {Rapidity, Flexibility, Customization}

---

2. Recommandations

A recommendation is initiated by a recommendation request, which consists of:

- Guidance Goal,
- Partial trace
- a set of allowed activities (the rest of the variants of the model to be configured).

Formally, we define a recommendation request as follows:

Définition 4 (Recommendation Request).

Let's suppose that A un ensemble d'activités eti a set of activities and $\rho \in A^*$ is a partial trace. Let's suppose that $E \subseteq A$ is a set of allowed activities.

We call $r = (\tau, \rho, E)$ a recommendation request.

For each allowed activity, we determine the expectations of:

- The target value expected during this activity **(do)**,
- The target value expected for alternative activities, ie, other permitted activities **(dont)**.

Functions **do** and **dont** are explained in Definition 10 and 11.

A recommendation result is an order of recommendations containing variant values.

Définition 5 (Recommendations).

$(e, do(e,\tau,\rho,D), dont(e,\tau,\rho,D)) \in E \times R \times R$ is a recommendation.

We use R to designate the universe of recommendations.

A recommendation result $R=<(e1,do(e1,\tau,\rho,D),dont(e1,\tau,\rho,D)), (e2,do(e2,\tau,\rho,D),dont(e2,\tau,\rho,D)), \ldots, (en,do(en,\tau,\rho,D),dont(en,\tau,\rho,D))>$ is a serie of recommendation, $R \in R^*$ et $\forall 1 \leq i < j \leq n$ $ei \neq ej$.

The order of the recommendations remains abstract. In the next section, we describe how recommendations are generated by a recommendation service based on the guidance dataset.

3. Traces abstraction

When generating recommendations, only mined traces that are relevant to determine the predictive information of an allowed activity should be retained.

From these mined traces, those with a high degree of matching with the partial trace should be weighted higher than those with little or no matching.

To determine which traces are relevant to provide recommendations, our recommendation service provides an overall abstraction

---

Définition 6 (Abstraction of mined traces).

$\sigma s = \{a \mid a \in \sigma\}$ means the overall abstraction of σ

In the next section, we explain how we determine which traces are relevant to obtain predictive information of an allowed activity.

In the weighting section, we will explain how to calculate the dataset process weight.

4. Support predicate

The relevance of a mined trace to produce a recommendation is based on support. Typically, they are the traces that support the allowed activity

We define an ensemble predicate**.**

A mined trace σ support support an allowed activity e whenever the activity e was observed in the guidance dataset.

Definition 7 (Function of activities Support).

We use the predicate $s(\tau,\rho,\sigma,e)$ to indicate that this trace σ support the execution of e after a partial trace ρ.

The predicate is defined by

$$ss(\tau,\rho,\sigma,e) \Leftrightarrow e \in \sigma s$$

The support predicate is used to filter the guidance dataset by removing any processes that do not support an allowed activity.

Définition 8 (Support Filtering).

We define the filtered dataset based on the support of the allowed activity $e \in E$ *and the partial trace* ρ *as following :*

$$Ds(\tau,\rho,e) = [\sigma \in D \mid s(\tau,\rho,\sigma,e)]$$

The mined traces of the filtered guidance dataset $Ds(\tau,\rho,e)$ support the allowed activity e and they are used for the recommendation of e according to a guidance goal τ.

Then we define a weighting function (ω) to express the importance of each mined trace for the recommendation of an allowed activity e.

The Support of an allowed activity determines which part of the guidance dataset serves as a recommendation.

5. Weighting of relevant processes

However, the processes that support an activity do not have the same importance, that is why, some of the mined traces can match the partial trace better than the others.

Therefore, we define weighting functions: a weight for each trace. The weight of a trace can be between 1 and 0. The value 1 indicates that two traces correspond entirely and 0 that they do not correspond at all. The calculation of the degree of matching depends on the abstraction of the trace. The weight of the mined trace is defined as the fraction of partial trace activities and the trace abstraction divided by the partial activities.

Définition 9 (Weighting Functions).

We define $\omega(\tau,\rho,\sigma)$, that is, the relative importance of a process σ When we consider the partial trace ρ and the target function τ.

- $\omega s(\tau,\rho,\sigma) = |\rho s \cap \sigma s| \; DIV \; |\rho s|$

---

6. Expected Result

Definition 5 indicates that a recommendation of an allowed activity `e` contain predictive information about the target value.

We define the expected result of a target value `(value do)`, When `e` is a executed in the next step.

The target value of each trace of `Ds(τ,ρ,e)` is weighted `(ω)` on the basis of degree of matching with the partial trace.

Définition 10 (do).

The target value expected after the execution of e is defined as following:

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

Similarly, we define the expected target value of not performing an allowed activity via the function dont.

Définition 11 (dont).

La valeur cible attendue après la non exécution de l'activité e est définie comme suit:

$$Dont\ (e,\tau,\rho,D) = \frac{\sum_{e' \in E \setminus \{e\}} \sum_{\sigma \in Ds(\tau,\rho,e) \setminus Ds(\tau,\rho,e')} \omega\ (\tau,\rho,\sigma) * \tau\ (\sigma)}{\sum_{e' \in E \setminus \{e\}} \sum_{\sigma \in Ds(\tau,\rho,e') \setminus Ds(\tau,\rho,e)} \omega\ (\tau,\rho,\sigma)} \quad \textbf{DIV}$$

The implementation of the ConfiLog algorithm must order the allowed activities according to the difference between the results of do and dont.

- The most important difference first if we try to minimize the time.
- The smallest difference first if we try to maximize flexibility.
- The most important difference first if we try to maximize customization..

Last note :)

It should be noted that the ConfiLog algorithm respects the domain constraints of the Product Line Model: the constraints "requires", the constraints "excludes" and the cardinalities of choice ". Indeed, the guidance dataset never contains activities that violate these constraints. Moreover, this is considered to be an implicit guidance operated by a constraint propagation.

Demonstration of the ConfiLog algorithm with a basic example

Let's suppose that:

- ρ = D,F is a partial trace.
- Allowed activities are E = {A, B, C}
- The user introduced a guidance goal = τ : Rapidity

Recommendation request = (τ,ρ,E)

According to our process mining treatments, processes below are the relevant according to τ : rapidity.

The guidance dataset is given by process mining

D={(A,B,C) , (D,B,C)...} with

[Processes that were executed rapidly <mean duration]

τ(A, B, C) = 900 secondes,

τ(D,B,C) = 500 secondes,

τ(F,B,C) = 500 secondes,

τ(D,F,A) = 1000 secondes,

[Processes that were executed without backtracks?]

τ(D,F,B) = 1500 secondes,

τ(D,F,C) = 2000 secondes,

τ(D,F,H) = 1260 secondes,

τ(C,C,A) = 1680 secondes,

Support predicate and weighting

Trace	Predicate			Weighting
	ss(τ,ρ,σ,e) ⇐⇒ e ∈ σs			ωs (τ,ρ,σ)
	A	B	C	
(A,B,C)	T	T	T	0
(D,B,C)	F	T	T	0.5
(F,B,C)	F	T	T	0.5
(D,F,A)	T	F	F	1
(D,F,B)	F	T	F	1
(D,F,C)	F	F	T	1
(D,F,H)	F	F	F	1
(C,C,A)	T	F	F	0

The trace (D,F,H) is to be removed from our guidance dataset because it contains no activity allowed in this case.

The user tries to minimize the configuration time. The function do and dont are calculated as follows :

- **do ((A,B,C), Rapidité, (D,F), D)**

do(A, Rapidity, D,F , L) = [0 * 900 + 1 * 1000 + 0 * 1680] DIV [0+1+0] = 1000

do(B, Rapidity, D,F , L) = [0 * 900 + 0.5 * 500 + 0.5 * 500 + 1 * 1500] DIV [0+0.5+0.5+1] = 1000

do(C, Rapidity, D,F , L) = [0 * 900 + 0.5 * 500 + 0.5*500 + 1*2000 + 0 * 1680] DIV [0+0.5+0.5+1+0] = 1250

- **dont ((A,B,C), Rapidity, (D,F), D)**

dont(A, Rapidity, D,F , L) = (0.5 * 500 + 0.5 * 500 + 1 * 1500) + (0.5 * 500 + 0.5 * 500 + 1 * 2000) 0.5+0.5+1+0.5+0.5+1 = 1125

dont(B, Rapidity, D,F , L) = (1 * 1000 + 0 · 1680) + (1 * 2000 + 0 * 1680) 1+0+1+0 = 1500

dont(C,Rapidity,D,F , L) = (1 * 1000) + (1 * 1500) 1+1 = 1250

The implementation of the ConfiLog algorithm must order the allowed activities according to the difference between do and dont.

In our case :

(A, 1000, 1125) = -125

(B, 1000, 1500) = - 500

(C, 1250, 1250) = 0

- **The most important difference first if we try to minimize time.**

So the recommended order is (B,A,C)