A critical evaluation study of model-log metrics in Process Discovery

Jochen De Weerdt[1], Manu De Backer[1,2,3], Jan Vanthienen[1], and Bart Baesens[1]

[1] Department of Decision Sciences and Information Management, Katholieke Universiteit Leuven, Naamsestraat 69, B-3000 Leuven, Belgium
Jochen.DeWeerdt@econ.kuleuven.be

[2] Department of HABE, Hogeschool Gent, Universiteit Gent, Voskenslaan 2, B-9000 Ghent, Belgium

[3] Department of Management Information Systems, University of Antwerp, Prinsstraat 13, B-2000 Antwerp, Belgium

Abstract. Recent years have witnessed the ability to gather an enormous amount of data in a large number of domains. These data, for the most part, remain in repositories where they are never used. There is an urgent need to beneficially use these data to retrieve useable knowledge. Process discovery from event logs is one such attempt to generate actionable knowledge. This field of study is part of process mining research, which can be defined as the gathering of useful knowledge from information system audit trails or event logs. However, process discovery's main objective is the extraction of control-flow models from event logs. As many authors have indicated, process discovery faces many challenges. Consequently, a well-defined evaluation framework for process discovery techniques is vital. With this paper, we aim to provide a tangible analysis of the currently available model-log evaluation metrics for mined control-flow models. Also, we will indicate strengths and weaknesses of the existing metrics and propose a number of opportunities for future research.

Key words: process discovery, evaluation metrics, machine learning

1 Introduction

The topic of process mining is relatively new and can be situated at the intersection of the fields of Business Process Management (BPM) and data mining [1]. It is inherently related to data mining and to the more general domain of knowledge discovery in databases (KDD) since the nature of its objectives is extracting useful information from large data repositories. Likewise, process discovery is strongly associated with BPM because of its purpose of gaining insight into business processes. As a result, process mining fits flawlessly into the BPM life cycle framework [2]. It should be noted that business process mining comprises process discovery because process mining describes a family of a-posteriori analysis techniques for extracting knowledge from event logs while process discovery only deals with extracting control-flow models. However, most of the attention in the process mining literature has been given to process discovery techniques.

Processes occur in a more or less structured fashion, containing structures such as or-joins, or-splits, and-joins, and-splits, and loops. The learning task for any

process discovery technique can be formulated as follows: given an event log that contains the events about a finite number of process instances, find a model that correctly summarizes the behavior in the event log, striking the right balance between generality (allowing enough behavior) and specificity (not allowing too much behavior). One important complexity for process mining techniques is that there are no natural negative cases available in an event log. In most KDD applications, negative instances are generally available. Due to the situation of learning from positive instances only, the straightforward application of traditional data mining techniques is non-trivial.

What is more, not only the development of process mining techniques is challenging, also the definition of appropriate metrics is a complex encounter. The quantification of accuracy as well as the trade-off between generality and precision is difficult. Desirably, a process discovery metric should measure only one dimension of a mined process model in reference with its event log. This is because in case a metric captures multiple dimensions, the metric quickly becomes incomprehensible.

We acknowledge the existence of a discussion in current literature concerning the absence of a "perfect model". However, we think that the availability of proficient metrics remains vital. As such, we will evaluate existing model-log evaluation metrics for mined control-flow models. Furthermore, we will discuss strengths and weaknesses of the existing metrics and propose a number of opportunities for future research.

In order to realize our goal of elucidating and assessing currently available process discovery metrics, this paper is structured as follows. In section 2, we will provide an overview of existing process discovery metrics. In section 3, a number of key metrics will be clarified in a simplified example. Furthermore, the metrics will also be illustrated and assessed within a more comprehensive Driver's License example. Finally, section 4 outlines the conclusions and some important opportunities for future research.

2 Overview of process discovery metrics

When evaluating the quality of mined process models, appropriate metrics need to be at hand. However, the quality of process models can be evaluated along different perspectives and by using different methods. One method is to compare the traces in the event log and the model mined from this event log (model-log metrics). Another approach is to compare an apriori model with the discovered model, but then an apriori model needs to be available (model-model metrics). In this paper, we will only consider model-log metrics. These metrics can be applied in any setting, whether a predefined process model exists or not.

2.1 Available model-log metrics and their dimensions

Existing model-log metrics evaluate mined process models along four important dimensions, as illustrated in table 1. It can be seen that most evaluation metrics

reflect the recall dimension. However, good recall is certainly not the only dimension a model should score well on. Also very important for comprehensible and useful models is a good balance between precision and generality.

Table 1. Overview of process mining evaluation metrics: model-log metrics

Name	Symbol	Author	Available in ProM	Range	Model input type	Measured aspect			
						Recall	Specificity	Precision	Generality
Parsing Measure	PM	Weijters et al. [3]	✓	[0,1]	Heuristic net	✓			
Soundness		Greco et al. [4]		[0,1]	Workflow schema			✓	
Completeness		"		[0,1]	Workflow schema	✓			
Fitness	f	Rozinat and Van der Aalst [5]	✓	[0,1]	Petri net	✓			
Behavioral Appropriateness	a_B	"	✓	[0,1]	Petri net			✓	
Advanced Behavioral Appropriateness	a'_B	"	✓	[0,1]	Petri net			✓	
Structural Appropriateness	a_S	"	✓	[0,1]	Petri net				✓
Advanced Structural Appropriateness	a'_S	"	✓	[0,1]	Petri net				✓
Completeness	$PF_{complete}$	Alves de Medeiros et al. [6]	✓	[-∞,1]	Heuristic net	✓			
Behavioral Recall	r^p_B	Goedertier et al. [7]		[0,1]	Petri net	✓			
Behavioral Specificity	s^n_B	"		[0,1]	Petri net		✓		

The recall dimension. Recall or sensitivity is a very important aspect. This dimension reflects how much behavior present in the event log is captured by the model. For every process discovery algorithm, it is of utmost importance to render models with good recall because representing the control-flow behavior in an event log is the major objective of any technique. Hence, it is definitely satisfying that a number of researchers have proposed different measures to capture recall.

- The *parsing measure (PM)* was proposed by Weijters et al. [3]. It quantifies the percentage of traces in the log that can be replayed by the discovered process model. It should be noted that PM is a coarse-grained metric. A single missing arc in a Petri net can result in parsing failure for all traces.
- A very similar metric is *completeness* as defined by Greco et al. [4]. This is the percentage of traces in the event log that are compliant with the workflow schema or process model. Completeness always ranges between 0 and 1.
- *Fitness (f)* is a metric that is obtained by trying whether each trace in the event log can be reproduced by the generative model. This procedure is called

sequence replay [5]. During replay, the transitions in the Petri net will produce and consume tokens to reflect the state transitions. Consequently, the fitness measure punishes for tokens that must additionally be created in the marked Petri net and also for tokens that remain after replay.
- *Completeness ($PF_{complete}$)* proposed by Alves de Medeiros et al. [8] is very similar to the fitness metric, but it additionally exploits trace frequencies in order to take into account the severity of missing and remaining tokens.
- *Behavioral recall (r_B^p)* as defined by Goedertier et al. [7] is the percentage of correctly classified positive events in the event log. During sequence replay, it is verified whether every positive event can be parsed by the model. Note that this measure originates from a process discovery technique that makes use of inducing artificial negative events in order to mine control-flow models. By verifying the parsing of positive events, recall can be quantified.

The specificity dimension. Specificity is the counterpart of recall. It captures the percentage of correctly classified negative cases. Of course, in process discovery, negative events or negative traces are typically not available. However, Goedertier et al. [7] propose a technique that generates artificial negative events. These artificial negative events can be used to define a state-of-the-art specificity metric. The availability of both recall and specificity metrics allows for the definition of an approved accuracy measure for process discovery models.

- *Behavioral specificity (s_B^n)* is the percentage of correctly classified negative instances during sequence replay. Negative events can be generated with the technique developed by Goedertier et al. However, the definition of the metric does not exclude the use of negative events stemming from other techniques.

The precision and generality dimensions. The trade-off between precision and generality is a major challenge in process discovery. Although models should be precise, generalizing beyond observed behavior is also a necessity. This is because assuming that all behavior is included in an event log is a much too strong completeness assumption. So, process discovery algorithms should be able to balance between underfitting (overly general models) and overfitting (overly precise models). Therefore, superior precision and generality metrics should be at hand. The following list enumerates the currently available measures.

- *Soundness* (Greco et al. [4]) is the first of three precision metrics. Soundness is the percentage of traces compliant with the process model that have been registered in the log. Calculating soundness is not straightforward because enumerating all possible paths in a process model is hard. Even for smaller process models, it might be impossible to determine all the traces that are compliant with a process model.
- The first of four appropriateness measures defined by Rozinat and Van der Aalst [5] is the simple *behavioral appropriateness (a_B)*. This simple approach measures the amount of possible behavior to determine a mean number of enabled transitions during log replay. Because this metric is not independent of structural properties, it is advised to use the advanced behavioral appropriateness.

– *Advanced behavioral appropriateness* (a'_B) allows to compare the behavior that is specified by the model with the behavior that is actually needed to describe the behavior in the event log. Therefore, this metric makes use of an analysis of "follows" and "precedes" relations, both in the model and the event log. Comparing the variability of these relations allows the definition of a precision metric that penalizes extra behavior.

Balancing between precision and generality also involves metrics that quantify generality. The structural and advanced structural appropriateness measures are the only currently available model-log metrics that quantify generality.

– *Structural appropriateness* (a_S) is based on the number of different task labels in relation to the graph size of the model. As identified by Rozinat and Van der Aalst [5], this metric's applicability is limited because it is only based on the graph size of the model.
– *Advanced structural appropriateness* (a'_S) is a generality metric that evaluates two specific design guidelines for expressing behavioral patterns. This measure will punish for both alternative duplicate tasks and redundant invisible tasks. Note that these guidelines are definitely not the only behavioral preferences of control-flow models. However, a'_S is the only metric that quantifies generality in some way. Ideally, a process model does not contain redundant invisible tasks nor alternative duplicate tasks. Accordingly, this measure will punish for models that simply enumerate all traces in the event log and for models that entail to much irrelevant invisible tasks.

2.2 Discussion

So far, we have discussed a number of process discovery evaluation metrics. Four important dimensions were identified along which process models should be judged: recall, specificity, precision and generality. Many evaluation metrics have already been proposed in literature. However, existing model-log metrics might insufficiently capture all of the underlying dimensions. For example, the currently available precision and generality measures are insufficiently capable of capturing all the complexities related to the trade-off between underfitting and overfitting.

3 Illustration of key metrics in process discovery

In this section, we will illustrate the most important currently available process mining metrics and show how they capture the different dimensions discussed in the previous section. We have selected five metrics: fitness, advanced behavioral appropriateness, advanced structural appropriateness, behavioral recall and behavioral specificity, so that the four identified dimensions are covered by at least one metric. Although the other metrics definitely have value, they do not add to this analysis because most of them are very comparable to one of the metrics selected. Furthermore, metrics like soundness and both simple appropriateness measures suffer from different shortcomings and are therefore left out.

3.1 A simplified example to elucidate the most important metrics

The event log of this simplified example contains 50 traces "ABCDA" and 50 traces "ACBDA". Accordingly, the best model representing the traces in the log is model 1(a). Table 2 shows that this model scores 1 on every metric and thus can be considered excellent along all evaluation dimensions.

Table 2. Model-log metrics for a simple artificial event log and four different models

Metric	**Fitness**	**Adv. Behavioral Appropriateness**	**Adv. Structural Appropriateness**	**Behavioral Recall**	**Behavioral Specificity**
Symbol *(dimension)*	f *(recall)*	$a_B^{'}$ *(precision)*	$a_S^{'}$ *(generality)*	r_B^p *(recall)*	s_B^n *(specificity)*
Best Model	1	1	1	1	1
Incomplete Model	0.92	1	1	0.90	0.79
Flower Model	1	0.17	1	1	0
Explicit Model	1	1	0.40	1	1

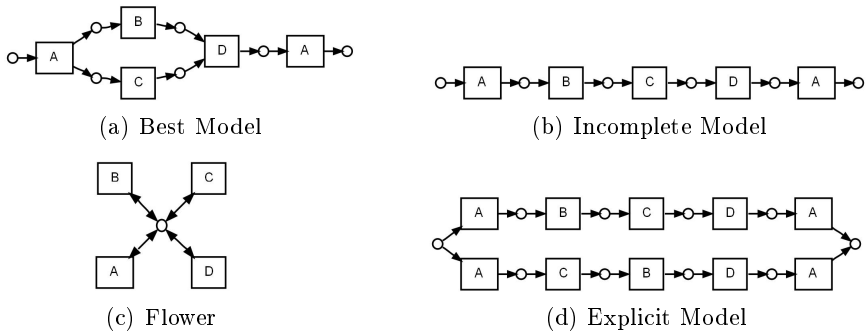


Fig. 1. Different control-flow models for the simple event log

The other models in figure 1 are in one way or the other erroneous. The incomplete model does not recall all traces in the log, the flower model allows too much behavior and the explicit model is only a mere enumeration of the traces in the log. We will now elucidate how different metrics are able to identify the dimension(s) along which a mined process model misses the mark.

Fitness. Model 1(b) is not able to capture all the behavior that is present in the event log. Thus, metrics quantifying the recall dimension should indicate this problem. The fitness measure (f) is a particularly useful metric to do so. With 50 traces (n_i) for each of the grouped traces, both the number of missing tokens (m_i) and the number of remaining tokens (r_i) amounts to 1 for the second grouped trace ("ACBDA") and 0 for the first grouped trace ("ABCDA"). Furthermore each

of the traces requires 6 tokens to be consumed (c_i) / produced (p_i) in order to be replayed. Accordingly, the fitness of the incomplete model sums up to 0.92. Note that i is an index running over the number of different grouped traces (k), which is two in this simplified case.

– Fitness: $f = \frac{1}{2}\left(1 - \frac{\sum_{i=1}^{k} n_i m_i}{\sum_{i=1}^{k} n_i c_i}\right) + \frac{1}{2}\left(1 - \frac{\sum_{i=1}^{k} n_i r_i}{\sum_{i=1}^{k} n_i p_i}\right)$

$= \frac{1}{2}\left(1 - \frac{50*0+50*1}{50*6+50*6}\right) + \frac{1}{2}\left(1 - \frac{50*0+50*1}{50*6+50*6}\right) = 0.92$

The other models in figure 3 are able to reproduce all the behavior in the event log. Accordingly, this is also demonstrated by the fitness measure. During sequence replay, there are no missing tokens nor remaining tokens for any of the other models and thus the fitness evaluates to 1. Notice that a fitness value of 0.92 for the incomplete model is an unattractive high value for a model that only parses half of the traces correctly.

Advanced behavioral appropriateness. The flower model (1(c)) is a generic model that allows any sequence of activities. Because it overgeneralizes, it is useless for any process intelligence activity. Nevertheless, the flower model captures all the behavior in the event log perfectly, so the model is not penalized by a recall-metric. The advanced behavioral appropriateness does punish the flower model for its overly general representation.

– Adv. behavioral appropriateness: $a_B^{'} = \frac{|S_F^l \cap S_F^m|}{2.|S_F^m|} + \frac{|S_P^l \cap S_P^m|}{2.|S_P^m|} = \frac{2}{2*12} + \frac{2}{2*12} = 0.17$

The advanced behavioral appropriateness for the simplified example flower model is found by calculating the elements in the "sometimes follows" and "sometimes precedes" relations in the model (S_F^m and S_P^m) and in the log (S_F^l and S_P^l) [5]. The calculation for the follows relations is illustrated in figure 2.[1] According to ProM, there are a total of 12 sometimes follows relations and 12 sometimes precedes relations in the model. However, the model and the log have only 2 sometimes follows and 2 sometimes precedes relations in common. Therefore, $a_B^{'}$ adds up to 0.17.

↙	A	B	C	D
A	$\mathbf{A_F}$	S_F	S_F	S_F
B	S_F	$\mathbf{A_F}$	S_F	S_F
C	S_F	S_F	$\mathbf{A_F}$	S_F
D	S_F	S_F	S_F	$\mathbf{A_F}$

↙	A	B	C	D
A	$\mathbf{S_F}$	S_F	S_F	S_F
B	S_F	$\mathbf{S_F}$	S_F	S_F
C	S_F	S_F	$\mathbf{S_F}$	S_F
D	S_F	S_F	S_F	$\mathbf{S_F}$

(a) Model relations: calculated manually (r) and by ProM (l)

↙	A	B	C	D
A	A_F	A_F	A_F	A_F
B	A_F	N_F	S_F	A_F
C	A_F	S_F	N_F	A_F
D	A_F	N_F	N_F	N_F

(b) Log relations

Fig. 2. Follows relations for the flower model (1(c)) and for the simplified event log

[1] For clarity, we make abstraction of the artificial Start and End activities.

For perfectly precise models, $a_B^{'}$ will equal 1. This is illustrated in table 2 as the other three models are not overgeneralizing and their $a_B^{'}$ evaluates to 1 accordingly. For models 1(a) and 1(d), the sometimes follows and sometimes precedes relations in the model and the log are exactly the same. For model 1(b), the relations are not completely identical, but the model is in fact more restrictive with respect to the relations in the log, so the generality measure evaluates to 1. This signifies correctly that the model is not underfitting. Other metrics should signpost the incompleteness of the model.

An important remark should be made concerning the calculation of $a_B^{'}$. The calculation involves a state space analysis which can be computationally very demanding. Furthermore, the calculation seems approximate because a manual analysis of the metric for the flower model results in a total of 16 sometimes follows and sometimes precedes relations. Apparently, (A,A), (B,B), (C,C) and (D,D) relations are categorized as always follows/precedes (see figure 2(a)), despite the fact that the flower model allows more variability for these relations. As such, $a_B^{'}$, as obtained from ProM, slightly underestimates the overgenerality of the flower model.

Advanced structural appropriateness. Another problem in process discovery are overly precise models or overfitting models. Figure 1(d) shows an explicit model that is a mere enumeration of the traces in the log. Again, such a model is undesired and should be punished by a generality measure. Advanced structural appropriateness is the only currently available metric that allows to quantify some kind of generality. Because model 1(d) contains six alternative duplicate tasks (T_{DA}) and no redundant invisible tasks (T_{IR}), $a_S^{'}$ evaluates to 0.4 (note that $|T|$ denotes the total number of tasks). These alternative duplicate tasks are activities B, C and D, occurring once in each of the branches of the explicit process model. They are alternative duplicate tasks because they never happen together in one execution sequence.

– Structural appropriateness: $a_S^{'} = \frac{|T|-(|T_{DA}|+|T_{IR}|)}{|T|} = \frac{10-(6+0)}{10} = 0.40$

Artificially generated negative event metrics. In [7], Goedertier et al. propose two state-of-the-art metrics originating from a process discovery technique called AGNEs (Artificially Generated Negative Events). This technique involves the induction of negative events in the event log in order to allow the application of advanced machine learning techniques (Inductive Logic Programming) for control-flow discovery (see also [9]). The availability of both positive and artificial negative events allows the definition of behavioral recall r_B^p and behavioral specificity s_B^n, metrics that are grounded in traditional data mining theory. According to their definition, they are able to penalize inaccurate process models.

The induction procedure generates artificial negative events as displayed in table 3. A total of 28 negative events are induced in the event log. In order to calculate the metrics, a confusion matrix can be constructed. The confusion matrix for the incomplete process model (1(b)) is shown in table 4.

Behavioral recall and behavioral specificity for the incomplete model are calculated according to the following formulae.

– Behavioral recall: $r_B^p = \frac{\sum_{i=1}^{k} n_i TP_i}{\sum_{i=1}^{k} n_i TP_i + \sum_{i=1}^{k} n_i FN_i}$
$= \frac{50*5+50*4}{(50*5+50*4)+(50*0+50*1)} = \frac{9}{10} = 0.90$

– Behavioral specificity: $s_B^n = \frac{\sum_{i=1}^{k} n_i TN_i}{\sum_{i=1}^{k} n_i TN_i + \sum_{i=1}^{k} n_i FP_i}$
$= \frac{50*14+50*8}{(50*14+50*8)+(50*0+50*6)} = \frac{22}{28} = 0.79$

Table 3. Artificially generated negative events for the simplified event log

	Trace 1					**Trace 2**				
Positive events	A	B	C	D	A	A	C	B	D	A
Artificially generated negative events	B^n	A^n	A^n	A^n	B^n	B^n	A^n	A^n	A^n	B^n
	C^n	D^n	B^n	B^n	C^n	C^n	D^n	C^n	B^n	C^n
	D^n		D^n	C^n	D^n	D^n		D^n	C^n	D^n

Table 4. Confusion matrix for the negative event metrics

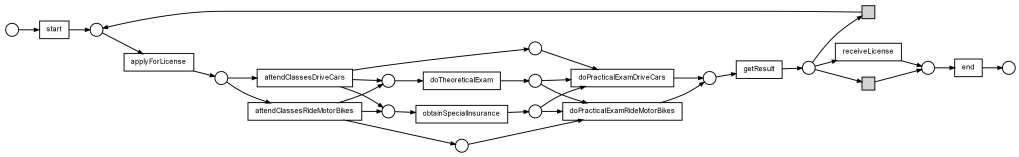
	true pos.	true neg.	total
pred. pos.	9	6	15
pred. neg.	1	22	23
total	10	28	

From this simplified model, it can be concluded that within process discovery research, different evaluation dimensions are covered by existing model-log metrics. However, it cannot be determined whether these metrics are sufficient in order to capture all the complexities of assessing process discovery models. Therefore, we also evaluate the presented metrics within a more extended example.

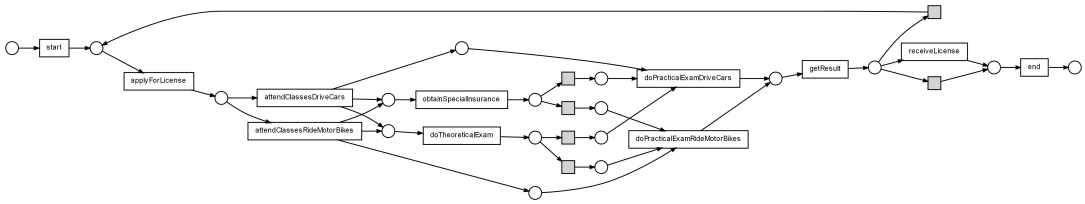
3.2 Further insights using an extended Driver's License example

We apply the evaluation metrics to a more extended example by using a modified version of a Driver's License (from [6]) event log. This event log and the according models contain more complex control-flow constructs such as a loop and non-free choice constructs. As such, this experiment will allow us to verify whether the available metrics can distinguish between worse and better models in a more complex setting. Figure 3 shows the results of four state-of-the-art process discovery techniques. By examining these results, some further elements in the analysis of existing process discovery metrics can be highlighted.

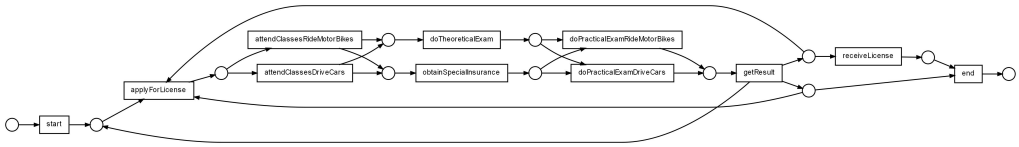
First of all, it can be seen that the fitness metric (f) reveals that every technique, except for the α-algorithm [10], discovers models with perfect recall. This can also be concluded from the behavioral recall metric (r_B^p). Although both metrics seem to measure the same dimension, an important remark should be made. The



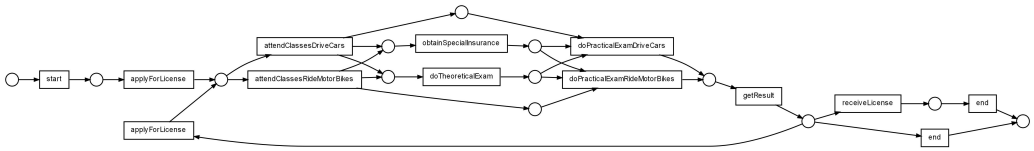
(a) Reference Model



(b) Heuristics Miner result



(c) Alpha result



(d) AGNEs result

Fig. 3. Different control-flow models for the simple event log

interpretation of the fitness measure requires some attention: although it accounts for recall as it punishes for the number of missing tokens that had to be created, it also punishes for the number of tokens that remain in the Petri net after log replay. The latter can be considered extra behavior. Therefore, the fitness metric also has a specificity semantics attached to it. As mentioned previously, metrics desirably measure only one dimension in order to remain comprehensible.

Secondly, some observations concerning the advanced behavioral appropriateness ($a_B^{'}$) are discussed. As for the flower model in the simplified example, the state space analysis for calculating this metric was calculated swiftly. However, for this Driver's License example, the more or less exhaustive simulation of all the behavior in the model was unable to be completed within an acceptable time period.

Table 5. Model-log metrics for the extended Driver's License example

Metric	**Fitness**	**Adv. Behavioral Appropriate-ness**	**Adv. Structural Appropriate-ness**	**Behavioral Recall**	**Behavioral Specificity**
Symbol *(dimension)*	f *(recall)*	$a_B^{'}$ *(precision)*	$a_S^{'}$ *(generality)*	r_B^p *(recall)*	s_B^n *(specificity)*
Reference Model	1	0.927	1	1	0.985
Heuristics Model	1	0.874	1	1	0.985
Alpha Model	0.921	1	1	0.926	0.764
AGNEs Model	1	0.906	0.846	1	0.985
Flower Model	1	0.500	1	1	0

Furthermore, it is also suspicious that $a_B^{'}$ does not evaluate to 1 for the reference model. It is definitely counterintuitive that the reference model is judged not to be completely precise with respect to the event log. Nevertheless, $a_B^{'}$ is capable of identifying the non-detection of the non-free choice construct.

Finally, we notice that it is not obvious which model is to be preferred. Models score differently along distinct dimensions, but there are no rules that define how these dimensions should be put together. Are there dimensions that are more important than others? How can this be included in the analysis phase? How should differences along one and multiple dimensions be assessed? We think that these questions bring about the necessity for a more rigorous and comprehensive evaluation framework for discovered process models.

4 Conclusion

With this paper, we discussed currently available process discovery metrics, which can be categorized along four important dimensions: recall, specificity, precision and generality. The analysis was restricted to model-log metrics because these metrics can be applied at all time, even when a predefined model is unavailable. Although the explicit illustration of some key metrics is very insightful, the analysis of strengths and weaknesses of the existent metrics is indispensable. We identified the following shortcomings with respect to the currently available process mining metrics.

- Metrics should be *one-dimensional*. The fitness (f) metric for example does not fulfil this requirement. Metrics that are multi-dimensional in nature will quickly become incomprehensible.
- The currently available precision and generality measures suffer from computational inefficiency. The advanced behavioral and structural appropriateness metrics require an *exhaustive simulation* of the mined process model. This state space analysis procedure is only *approximate* and this causes difficulties with respect to benchmarking new or existing process discovery algorithms. Even though

conceptually proficient, the imprecise calculation of the metrics brings about the necessity of new precision and generality measures. Although not trivial, it is crucial that the trade-off between precision and generality is adequately quantified. This is because this trade-off is a key determinant of model comprehensibility. Accordingly, the definition of new, insightful metrics is definitely a challenge for future research.
- Process discovery metrics developed in the light of *machine learning* theory are definitely of added value. Behavioral recall and behavioral specificity are valuable measures, but their integration with the ProM-framework should allow researchers to exploit these state-of-the-art recall and specificity metrics.

Finally, this analysis demonstrates that, within process discovery, a more rigorous and comprehensive evaluation framework is definitely needed. We think that this paper is a valuable impetus hereto. It can be concluded that in any process discovery analysis, combining different metrics is indispensable, as metrics preferably measure only one aspect of a mined process model and models will always have to be judged along multiple dimensions.

Notes and Comments. We would like to thank the Flemish Research Council for financial support under Odysseus grant B.0915.09. Furthermore, we would like to thank the team of prof. Van der Aalst for their comments and valuable input.

References

1. Tan, P.N., Steinbach, M., Kumar, V.: Introduction to Data Mining. Addison-Wesley (2005)
2. Weske, M.: Business Process Management: Concepts, Languages, Architectures. Springer (2007)
3. Weijters, A.J.M.M., van der Aalst, W.M.P., Alves de Medeiros, A.K.: Process mining with the heuristicsminer algorithm. BETA working paper series 166, Eindhoven University of Technology (2006)
4. Greco, G., Guzzo, A., Pontieri, L., Saccà, D.: Discovering expressive process models by clustering log traces. IEEE Trans. Knowl. Data Eng. **18**(8) (2006) 1010–1027
5. Rozinat, A., van der Aalst, W.M.P.: Conformance checking of processes based on monitoring real behavior. Information Systems **33**(1) (2008) 64–95
6. Alves de Medeiros, A.K.: Genetic Process Mining. PhD thesis, Technische Universiteit Eindhoven (2006)
7. Goedertier, S., Martens, D., Vanthienen, J., Baesens, B.: Robust process discovery with artificial negative events. Journal of Machine Learning Research **10** (2009) 1305–1340
8. Alves de Medeiros, A.K., Weijters, A.J.M.M., van der Aalst, W.M.P.: Genetic process mining: an experimental evaluation. Data Mining and Knowledge Discovery **14**(2) (2007) 245–304
9. Ferreira, H., Ferreira, D.: An integrated life cycle for workflow management based on learning and planning. International Journal of Cooperative Information Systems **15**(4) (2006) 485–505
10. van der Aalst, W.M.P., Weijters, A.J.M.M., Maruster, L.: Workflow mining: Discovering process models from event logs. IEEE Trans. Knowl. Data Eng. **16**(9) (2004) 1128–1142