A Chance For Models To Show Their Quality: Stochastic Process Model-Log Dimensions

Adam T. Burke[a,*], Sander J.J. Leemans[b], Moe T. Wynn[a],
Wil M.P. van der Aalst[b,a], Arthur H.M. ter Hofstede[a]

[a]*School of Information Systems, Queensland University of Technology, Brisbane, Australia*
[b]*RWTH University, Aachen, Germany*

Abstract

Process models describe the desired or observed behaviour of organisations. In stochastic process mining, computational analysis of trace data yields process models which describe process paths and their probability of execution. To understand the quality of these models, and to compare them, quantitative quality measures are used.

This research investigates model comparison empirically, using stochastic process models built from real-life logs. The experimental design collects a large number of models generated randomly and using process discovery techniques. Twenty-five different metrics are taken on these models, using both existing process model metrics and new, exploratory ones. The results are analysed quantitatively, making particular use of principal component analysis.

Based on this analysis, we suggest three stochastic process model dimensions: adhesion, relevance and simplicity. We also suggest possible metrics for these dimensions, and demonstrate their use on example models.

Keywords: stochastic process mining, process conformance, stochastic Petri nets, adhesion, relevance, simplicity

1. Introduction

It has been said that “the purpose of a system is what it does” [6]; in the same spirit, much can be learnt about a system’s purpose by *how often* it does it. An organisation is a system, and we can understand it through the processes it follows. A process performed hundreds of times daily can be a good target for optimisation, and a rare sequence of events may be important to monitor for legal compliance. In these cases where event frequency is important, to analyse

*Corresponding author

Email addresses: at.burke@qut.edu.au (Adam T. Burke), s.leemans@bpm.rwth-aachen.de (Sander J.J. Leemans), m.wynn@qut.edu.au (Moe T. Wynn), wvdaalst@pads.rwth-aachen.de (Wil M.P. van der Aalst), a.terhofstede@qut.edu.au (Arthur H.M. ter Hofstede)

such organisational behaviours quantitatively, we need a stochastic model of a process, that explicitly represents probability. This research is about measuring and comparing stochastic process models. It draws on, and contributes to, the discipline of process mining [1], which concerns the automatic discovery of process models and their further computational analysis.

Process mining uses sequential data recorded as observations of a process in action. A collection of such sequences, recording many instances of the process executing, is termed an *event log*. Logs are the input to discovery algorithms which output *process models*: computational models that describe the underlying process. Process discovery is then a form of unsupervised learning. In the case of stochastic process mining, the models learnt describe not only which sequences are possible, but, perhaps indirectly, how probable those sequences are. Process mining is used across many industries, and there are many commercial tools [16, 18]. These tools all provide frequency information on activities, but no explicit support for stochastic models. Explicit stochastic process models have been used in insurance [37, 29] and healthcare [33] to optimise workflows and to identify risks.

Quality metrics exist to measure the success of models in representing logs, and support other forms of quantitative comparison. This may be to check compliance against a target model, or to understand ways an official model differs from facts on the ground, or change over time. For instance, analysts at a German hospital used process mining conformance tools to compare the changes in medical treatment between different waves of COVID-19 [7]. Many quantitative measures for such models exist.

When they capture only a control-flow perspective, without a stochastic element, process mining has well-established ideas on how to organise metrics. They are organised under four quality dimensions: fitness, precision, simplicity and generalisation [1, p118]. Having four dimensions supports thinking through design trade-offs in the construction of models, rather than seeking to optimise a singular metric.

Some equivalent of process quality dimensions for stochastic models would help understand quality, support process compliance, and make meaningful model design trade-offs. It is far from clear whether the four control-flow quality dimensions translate to use on stochastic process models. Existing metrics may or may not connect to a control-flow dimension, and are often constrained to subsets of common process model types. More fundamentally, we lack an understanding of how these quality metrics relate to one another. As our investigation shows, two metrics that purport to measure the same concept, such as precision, may give very different measurements for the same log and model. On the other hand, metrics that supposedly measure different concepts may be highly correlated.

Accordingly, in this paper, we investigate *what dimensions may describe the quality of stochastic process models*. There are no established quality dimensions for such models, so we use metrics designed for stochastic models as one starting point, as well as established control-flow process mining dimensions. The *mathematical* space under consideration is not purely analytical, as the underly-

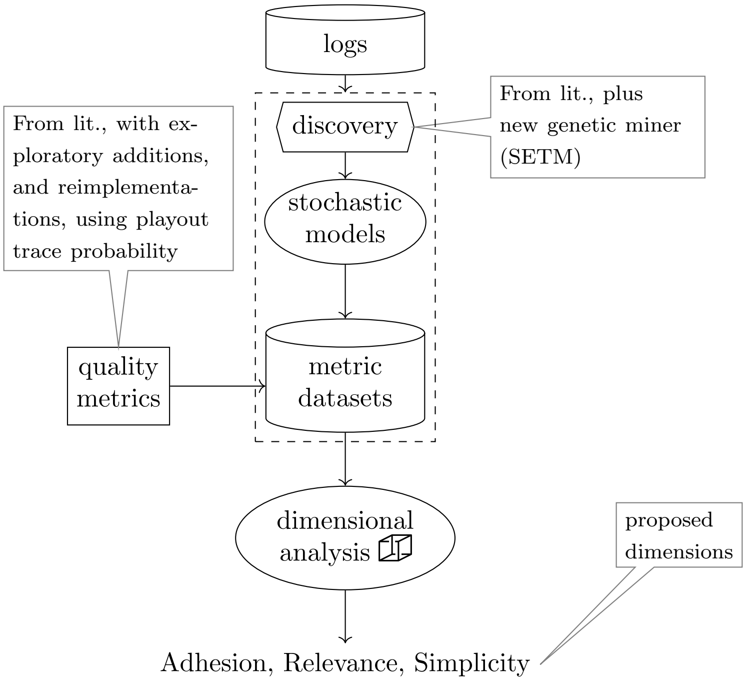

logs
discovery
From lit., plus new genetic miner (SETM)
From lit., with exploratory additions, and reimplementations, using playout trace probability
stochastic models
quality metrics
metric datasets
dimensional analysis
proposed dimensions
Adhesion, Relevance, Simplicity

Figure 1: Research overview.

ing event logs to which models are compared are real-life empirical data on the social behaviour of organisations. This suggests using an exploratory quantitative analysis. Our approach was *empirical*, based on collecting and evaluating stochastic process models for real-life processes. Experiments generated a collection of thousands of models, using a variety of techniques, and based on event data from six real-life logs. Metrics collected included those from the literature and some adapted or designed specifically for this experiment. This empirical data was analysed for correlation and a principal components analysis (PCA) performed. Based on this analysis, we propose three stochastic process model quality dimensions: *adhesion*, *relevance* and *simplicity*. Both the proposed dimensions for stochastic process model quality, and the empirical investigation, are novel.

Figure 1 gives an overview of the research. Stochastic process models are generated from logs using discovery techniques. These models, together with the logs, are inputs to quality metric calculations. The dataset of collected metrics is then analysed quantitatively.

This article extends our earlier conference paper [14], which reported on the experiment design and an initial round of experiment and quantitative analysis. This article adds:

- Additional metrics for entropic relevance [3], and measures based on alpha precision [17], from recently published research;
- Improvements to the experimental design, particularly metric capture and choice;
- A second cycle of experimental evaluation and analysis;
- Candidate metrics based on the three dimensions; and
- Detailed demonstrations of the dimensions and metrics in use on concrete example models.

We also share the following secondary results:

- A practical, approximate solution for trace probability calculation (TRACEPROB) [27], allowing an expanded set of supported models for some metrics, and mentioned in our previous work as a play-out log generation technique;
- New detail on the formal properties of this trace probability calculation;
- A genetic miner for the discovery of stochastic process models, Stochastic Evolutionary Tree Miner (SETM), suitable for laboratory use; and
- A new implementation of the entropic relevance measure [3] applicable to a broader range of models than those in the original paper or public implementation.

The remainder of this article proceeds as follows. Formal foundations are defined in Section 2 and background scholarship is discussed in Section 3. The experimental design is described in Section 4, including metric choice, model generation, and two cycles of experiments. The results of the experiments are presented and analysed in Section 5. Quality dimensions, and metrics suggested by these results, are discussed in Section 6, including applying them to example models with a range of different qualities. Section 8 concludes.

2. Preliminaries

This section defines process mining concepts and mathematical structures used throughout this article.

2.1. Logs and Languages

Sequences are shown as $\langle a_1, \dots, a_n \rangle$ and their concatenation operator as $+$, for example, $\langle a, b\rangle = \langle a \rangle + \langle b \rangle$. The set of multisets (bags) over type C is $\mathcal{B}(C)$ and real-valued multisets are $\mathcal{B}^+(C)$. Real-valued multisets are always positive-valued in this paper, with values $\in \mathbb{R}^+$. The count of item $x \in C$ in bag $B \in \mathcal{B}(C)$ is $B[x]$. Multiset union and intersection are $\sqcup$ and $\sqcap$ respectively. In real-valued multisets, the count of a member is in $\mathbb{R}^+$. As an example, consider real-valued multiset $X = [\langle a \rangle^{3.4}, \langle b, c\rangle^{2.0}]$. Then $X[\langle a \rangle] = 3.4$. The $\cdot$ operator scales all occurrence values by a numeric factor, as in $2 \cdot [\langle a \rangle^1, \langle c, b\rangle^3] = [\langle a \rangle^2, \langle c, b \rangle^6]$.

Definition 1 (Activities and Event Logs). *Let A be a set of activities in a process, and A^* the possible sequences of those activities. Each occurrence of an activity is an* event*. A* trace *$\sigma \in A^*$ is a sequence of activities.* Event logs *are multisets of traces $\mathcal{B}(A^*)$.*

$\mathcal{L}$ is the set of all event logs. $|L|$ is the number of traces in a log $L \in \mathcal{L}$, and $||L||$ the number of events. The number of cases matching trace σ in log $L \in \mathcal{L}$ is $L[\sigma]$.

Definition 2 (Play-out Log). *A play-out log [1, p41] $L_p \in \mathcal{B}^+(A^*)$ is a finite real-valued multiset of traces.*

Real-life event logs have whole-numbered traces, but in our laboratory setup, fractional trace counts are useful in play-out logs to accommodate some side-effects of scaling. These are always positive. The set of all play-out logs is $\mathcal{L}^+ \supset \mathcal{L}$.

Definition 3 (Stochastic Language). *A* stochastic language *$\Theta \subseteq \mathcal{B}^+(A^*)$ for traces over activities A is a real-valued multiset holding probability values for each trace, and summing to 1.*

$$\forall \sigma \in \Theta, \Theta[\sigma] \in [0, 1]$$
$$\sum_{\sigma \in \Theta} \Theta[\sigma] = 1$$

Stochastic languages may be infinite, but this experiment design uses finite approximations derived from *play-out logs*. The corresponding finite stochastic language for a play-out log can be found when scaling by the inverse of the cardinality of the log, $\frac{1}{|L|}$.

2.2. Petri Nets and Other Models

The term Petri net can refer either to a specific formalism or to a family of related transition structures. We refer to the foundational control-flow structure as a place-transition net, following [5].

Definition 4 (Place-transition Nets). *A* place-transition net *is a tuple* (P, T, F, M_0) *of places* P*, transitions* T*, flow relation* $F \subseteq (P \times T) \cup (T \times P)$ *and initial marking* M_0*. Markings are multisets of places* $M \in \mathcal{B}(P)$ *indicating a state of the net.*

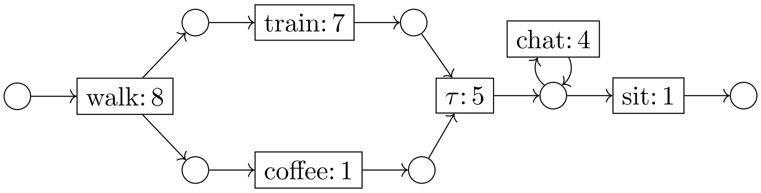
The flow relation F represents a directed connection from the first to the second node. A transition t is *enabled* under marking M when every incoming place is marked, or $\forall (p,t) \in F, p \in M$. When enabled, transitions may *fire*, changing the state of the net by consuming one token from each incoming place and producing one token for each outgoing place.

This structure can be extended to model process probabilities, and to support activity labels.

Definition 5 (Stochastic Labelled Petri Net). *An* SLPN *[29] is a tuple* $(P, T, F, M_0, W, \lambda)$ *such that* (P, T, F, M_0) *is a place-transition net. A weight function* $W \colon T \to \mathbb{R}^+$ *assigns each transition a weight. Labelling function* $\lambda \colon T \to A \cup \{\tau\}$ *then provides a mapping from transitions to a symbol library of activities* A*.* τ *is a silent label where* $\tau \notin A$*.*

When transitions $T_e \subseteq T$ are enabled in a particular marking, a transition $t \in T_e$ fires according to the probability given by $\frac{W(t)}{\sum_{t' \in T_e} W(t')}$. The sequences of activity labels, without silent label τ, generated by a series of transitions through the model forms a trace, and the collection of such traces and their probabilities is the SLPN's stochastic language. We assume traces to end in deadlock, where no transitions are enabled.

Figure 2 is an example SLPN describing a commuter travelling to work. Every trip starts with them walking to the train station. They always take the train and buy a coffee, but sometimes they buy the coffee from the cafe at their departure station, before embarking, and sometimes after taking the train, from a cafe close to their work. When they arrive at work, they may go straight to their desk to start the day, or they may chat with one or more colleagues. In the model, at each point where more than one transition is enabled, the probability is determined by weighted choice. Transitions are annotated with labels and weights in the diagram, with τ a silent label. For example, after walking to the station (walk: 8), the commuter usually takes the train (train: 7) before buying their coffee (coffee: 1), with a probability of $\frac{7}{8}$. The probability of the entire



$$\begin{aligned}\Theta_{commute} = [&\langle \text{walk}, \text{train}, \text{coffee}, \text{sit}\rangle^{0.18}, \langle \text{walk}, \text{train}, \text{coffee}, \text{chat}, \text{sit}\rangle^{0.14},\\ &\langle \text{walk}, \text{train}, \text{coffee}, \text{chat}, \text{chat}, \text{sit}\rangle^{0.12}, ...,\\ &\langle \text{walk}, \text{coffee}, \text{train}, \text{sit}\rangle^{0.03}, \langle \text{walk}, \text{coffee}, \text{train}, \text{chat}, \text{sit}\rangle^{0.02},\\ &\langle \text{walk}, \text{coffee}, \text{train}, \text{chat}, \text{chat}, \text{sit}\rangle^{0.02}, ...]\end{aligned}$$

Figure 2: Example SLPN for a commuter travelling to work, with its stochastic language. The language has an infinite number of traces due to the presence of the loop. An SLPN without weights and a label function is a place-transition net.

trace $\langle \text{walk}, \text{train}, \text{coffee}, \text{sit}\rangle$ is 0.18, as shown in the excerpt from the infinite stochastic language generated by the model.

We name the set of all SLPNs as $\mathcal{N}$. SLPNs are a labeled variant of *Stochastic Petri Nets* (SPNs) [5] and Generalized SPNs (GSPNs) [5]. SPNs use timed transitions, where firing happens according to a reverse exponential function. GSPNs have both immediate transitions, as in SLPNs, and timed transitions, as in SPNs.

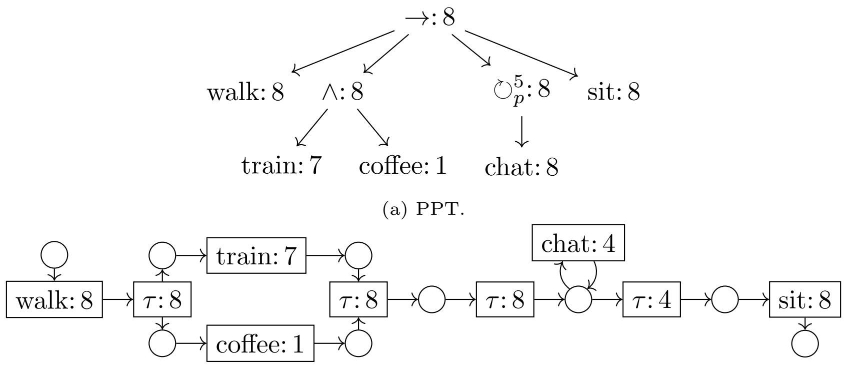
All process models used in this research are either SLPNs, or structures translatable to SLPNs. A Probabilistic Process Tree (PPT) [12] is a tree of weighted nodes.

Definition 6 (Probabilistic Process Trees). *Let $x\colon w$ be a node, where x is the unweighted portion and $w \in \mathbb{R}^+$ a weight. The universe of PPTs over activity set A is recursively defined as:*

1. *A single activity. For $a \in A$, $a\colon w \in \mathcal{PPT}_{\mathcal{A}}$.*
2. *A silent activity, represented by the constant τ, such that $\tau \notin A \wedge \tau : w \in \mathcal{PPT}_{\mathcal{A}}$.*
3. *An n-ary operator $\oplus_n$ over one or more child trees. Given $m \geq 1$, $u_1, ..., u_m \in \mathcal{PPT}_{\mathcal{A}}$, then $\oplus_n(u_1, ..., u_m)\colon w \in \mathcal{PPT}_{\mathcal{A}}$. $\oplus_n \in \{\rightarrow, \times, \wedge\}$.*
   - *$\rightarrow(x_1\colon w, ..., x_n\colon w)\colon w$ is a sequence of trees, executed serially.*
   - *$\times(x_1\colon w_1, ..., x_n\colon w_n)\colon w$ is a weighted exclusive choice between processes.*
   - *$\wedge(x_1\colon w_1, ..., x_n\colon w_n)$: describes child processes which execute concurrently. The next process to progress is determined by a weighted race.*
4. *A unary operator $\oplus_1$. Given $u \in \mathcal{PPT}_{\mathcal{A}}$, then $\oplus_1(u)\colon w \in \mathcal{PPT}_{\mathcal{A}}$. $\oplus_1 \in \{\circlearrowleft_n, \circlearrowleft_p\}$.*
   - *$\circlearrowleft_n^m(x\colon w)\colon w$ is a fixed loop which repeats process $x\colon w$ m times.*

- $\circlearrowright^{\rho}_{p}(x\colon w)\colon w$ *is a probabilistic loop which repeats process* $x\colon w$ *zero or more times, with exit probability* $\frac{1}{\rho}$.

PPTs can be translated to SLPNs and are used in multiple ways as part of the model generation in these experiments. Figure 3 is an example PPT describing the same commuter travel process as Figure 2, and with the same stochastic language.Figure 3b shows the automatic translation of the PPT to an SLPN, including the silent transitions which maintain the block structure. For example, the concurrent subtree describing taking the train and having a coffee becomes a block with a race between "train" and "coffee" Petri net transitions.



(a) PPT.

(b) SLPN.

Figure 3: Example PPT for a commuter travelling to work (3a), and its SLPN translation (3b).

2.3. Metrics and Measures

In this work, *metrics* and *measures* are particular classes of functions used to evaluate log and model quality.

Definition 7 (Metrics and Measures). *A* metric m *is a function comparing models and logs,* $m\colon \mathcal{N} \times \mathcal{L}^{+} \to \mathbb{R}$*, which returns a real number. A* measure μ *is a metric with range* $[0, 1]$*, or* $\mu\colon \mathcal{N} \times \mathcal{L}^{+} \to [0, 1]$.

Measures are a subset of metrics with a guaranteed finite range that make some forms of comparison and analysis simpler. Where a metric or measure does not return values for all SLPN models ($\mathcal{N}$), we term it *delicate*. This may be a formally identified limitation, or a practical observation about the behaviour of a particular metric implementation. To give some examples from the literature, Entropic Relevance [3] is a metric reported in bits, which has no upper bound. Entropy Recall [30], by contrast, is a measure, formally defined in a way that guarantees a $[0, 1]$ range; it is also a delicate measure, which is defined only for determinstic models.

Some metrics are designed to work with play-out logs instead of models.

Definition 8 (Play-out Metrics and Measures). *A* play-out metric *is a function comparing play-out logs and event logs,* $\pi\colon \mathcal{L}^+ \times \mathcal{L}^+ \to \mathbb{R}$*, and a* play-out measure π_μ *is a play-out metric with range* $[0,1]$.

3. Related Work

This research builds on other scholarly work on stochastic process mining and models. This includes the discovery of such models, their quantitative measurement and comparison (conformance), and the dimensions along which they may be compared.

3.1. Discovery of Stochastic Process Models

Many control-flow discovery algorithms exist [19]. Stochastic process discovery algorithms are more limited in number, and may directly annotate models discovered by control-flow techniques [38, 13, 28, 32] or construct stochastic models directly [36, 12]. Table 1 summarises existing discovery techniques, including whether a technique depends on a non-stochastic model from another miner, the type of model output, whether the output is convertible to an SLPN, and the existence of a public implementation. For completeness, we include two techniques published after our experiments were run [28, 32]. Both use SLPN variants, one with dynamic weights [28], the second with data-sensitive guards [32]. Most discovery techniques leverage a control-flow technique as an initial step, such as the use of Inductive Miner [26] by GDT_SPN discovery [38]. Many of techniques have public implementations. There is a diversity of output model types, but most of them support conversion to SLPNs. Other formats, such as the annotated BPMN used by Simod [15], may potentially support SLPN conversion with further research beyond the scope of the current paper.

Table 1: Related work on stochastic process discovery.

Technique	Miner dependency	Model output	SLPN conversion	Public impl.	Year
GDT_SPN discovery [38]	Inductive miner [26]	GDT_SPN	Yes	Yes	2014
Non-classical Bayesian net discovery [36]	No	Bayesian net variant	No	No	2018
PLTL discovery [31]	No	ProbDeclare	No	No	2019
MCMC prefix tree discovery [21]	No	Prefix automaton	No	No	2020
Simod [15]	Split miner [4]	Annotated BPMN	No	Yes	2020
Weight estimators (6 variants) [13]	Yes	SLPN	Yes	Yes	2021
Toothpaste Miner [12]	No	PPT	Yes	Yes	2021
SLPN-SD discovery [28]	Yes	SLPN variant	Yes	Yes	2023
Data-Based Stochastic Discovery [32]	Yes	SLPN variant	Yes	Yes	2023

For our experiment, we use existing GSPN and SLPN discovery techniques with public implementations [38, 13, 12]. Other recent discovery research has shown techniques for discovering probability-annotated BPMN models [15], probabilistic declarative models [31], non-classical probability Bayesian networks [36] and Bayesian models for place-transition Petri nets [21]. The Bayesian technique [21] also has potential applications for model comparison and new conformance measures.

The current study builds directly on the analysis of genetically-mined control-flow models [11], both in study design, and direct extension of the Evolutionary Tree Miner code [10]. That work conducted a qualitative study on classes of models generated with different genetic miner constraints. In this work, the dimensions derived through quantitative analysis in Section 6 are applied qualitatively in Section 6.4. The Stochastic Evolutionary Tree Miner (SETM), a laboratory discovery technique suitable for exploring alternative models, is introduced in Section 4.3.

3.2. Quality Dimensions

For control-flow process models and process mining, quality measures are typically considered to be measuring one of four quality dimensions: fitness, precision, simplicity and generalisation [11], [1, p118]. Fitness measures indicate how well the model can reproduce the behaviour of the log. Precision measures how much of the model is used to reproduce log behaviour. A model may describe not just all traces in the log, but many other traces besides: such a model has high fitness but low precision. The simplicity dimension considers simpler models as higher quality, in both an application of Ockham's Razor [39] and a recognition that simpler models are easier to understand [35]. Generalisation measures whether the model is applicable to more than the current sample (in process mining, a specific event log). In contrast to approaches in statistical learning where metrics aspire to represent overall model quality [41], in process mining, model quality is usually presented as a way to make design trade-offs against four quality criteria which are inherently in tension.

Though many studies investigate particular techniques quantitatively, quantitative experiments on the basis for quality dimensions are rarer. There is at least one quantitative study of the relationship between control-flow quality dimensions [22]. This used a collection of quality measures on models from a variety of control-flow discovery techniques. Factor analysis on the results found fitness and precision components with a clear correspondence to existing measures. An established consensus on what control-flow dimensions were meaningful preceded the experiment, and the empirical components supported those concepts. For the stochastic process context, there are no pre-known dimensions, so this study has a more exploratory character.

3.3. Conformance of Stochastic Process Models

Stochastic conformance metrics are those which specifically take stochastic process models as input. We make use of most of the metrics surveyed below in Section 4.1, either directly, or by introducing alternatives inspired by them (defined formally in Appendix A). Existing metrics in the literature often consider probability mass or the probability of particular traces as parameters in metric calculation. Table 2 summarises existing metrics, including whether their design is based on a control-flow dimension, restrictions on model inputs from the full set of SLPN models, and whether there is a public implementation. There are a small number of metrics, all a product of research from the last four years. There is no metric for the control-flow Generalisation dimension, and a third of the models have no control-flow dimension analogue. Most metrics have restrictions on which models they can be applied to. Three apply only to SDFA-equivalent models. The set of models for which the Earth-Movers' Distance measure (EM) [25] is not practical has been noted, but is not a recognised formal class of models. Public implementations are available for most, but not all, of the published metrics.

Table 2: Related work on stochastic process conformance.

Technique	Related Control-flow dimension	SLPN restriction (delicate)	Public impl.	Year
Earth-movers' distance [25]	No	Some	Yes	2019
Entropy projection precision [30]	Precision	SDFA only	Yes	2020
Entropy projection recall [30]	Fitness	SDFA only	Yes	2020
Entropic behavioural simplicity [24]	Simplicity	No	No	2020
Alpha precision [17]	Precision	No	No	2022
Entropic relevance [3]	No	SDFA only	Yes	2020

Calculating the probability of a particular trace through a process model is a non-trivial algorithmic problem, Trace-Prob [27]. Though solutions now exist, efficient approaches for broad classes of models are still an active research challenge. Two recent approaches use firstly, linear programming [27] and secondly, an expectation-minimisation (EM) algorithm on Probabilistic Context Free Grammar trees [42]. In Section 4.2 we show an alternative solution which uses SLPN model play-out - a variant of the Petri net token game [1, p41] - to give the trace probability for all traces in a model beyond a given probability threshold. The result is represented as a play-out log.

The Earth-Movers' Distance measure (EM) [25] combines the well-known concepts of Levenshtein string edit cost - in this case in comparing traces - with the Earth-Movers' distance over the possible traces of model and log. As the set of possible model traces can be infinite, this includes a truncated measure using a specific fraction of the probability mass, for tractability. This theoretical constraint was seen in practice on some models built by existing discovery techniques from real-life logs.

Entropy has also been used for model quality measurement. In projection-based precision and recall [30], Stochastic Deterministic Finite Automata (SDFAs) are constructed for both log and model. New SDFAs can be computed from a projection of the log over the model, and vice versa. Entropy ratios then provide measures for precision (H_P) and recall/fitness (H_F), both used in this study. We also employ measures inspired by entropy projection, but

Table 3: Event logs

Log	Traces	Variants	$\|A\|$	Domain
BPIC 2013 closed	1487	183	4	Issue tracking
BPIC 2013 incidents	7554	1511	3	Incident tracking
BPIC 2018 control	43808	59	7	EU Agriculture policy
BPIC 2018 reference	43802	515	6	EU Agriculture policy
Road Traffic Fines	150370	231	11	Italian policing
Sepsis	1054	846	16	Hospital diagnosis

not limited to SDFAs (Play-out Entropy Fitness and Precision measures HIFT, HIPT, HJFT, HJPT). In entropic relevance [3], the process model is considered as a way of encoding the log. The entropy of the resulting encoding is calculated using trace probability, accounting for the encoding cost according to a background cost model. Three defined metrics correspond to three background cost models: Universal (HRU), Zero Order (HRZ) and Restricted Zero Order (HRR). In the original work trace probability is calculated for SDFAs only. In our experiments, we use a trace probability calculation from play-out logs when calculating these metrics, as seen in Section 4.1. The result is in bits and so not constrained to a $[0, 1]$ range. Entropy has also been used to formulate behavioural simplicity measures [24] for control-flow models.

The Alpha Precision measure [17] uses the stochastic language of the model and the event log, and inferences about the underlying system that generated the log. Probability in the (otherwise unknown) underlying system is estimated using attributes of the log, including the traces, and reasoning from a Dirichlet distribution. Traces are included in the calculation when their probability in the underlying system exceeds a parameter called alpha significance. This alpha significance parameter varies across domains, making the comparison of models across logs difficult. The Existential Precision metric (XPU) is the alternative we introduce to allow such comparisons.

In summary, current discovery techniques use a variety of techniques and output model types, but are somewhat comparable using a common denominator of SLPNs. Control-flow quality dimensions suggest starting points for the quality of stochastic models, but translation of the concepts to a stochastic setting is non-obvious, and new concepts may apply. Existing metrics for stochastic models are inconsistently related to control-flow dimensions, and have restrictions on supported model types due to the challenges of calculating stochastic languages. This landscape provides the challenges and constraints for our experimental design.

4. Experiment Design

In this section, we introduce the experiment design in more detail, including components and phases. The experiment aimed to build a dataset of metrics for quantitative analysis. Figure 4 shows the detailed experiment design.

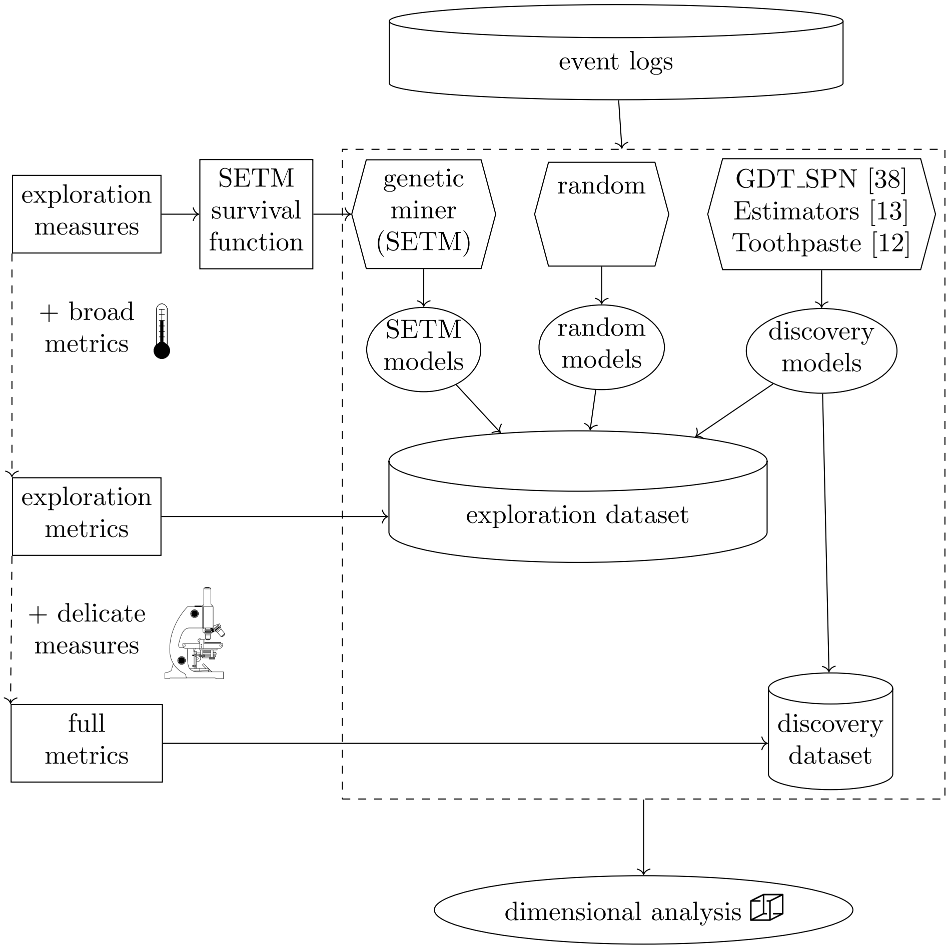


Figure 4: Experiment design, generating a broad range of models from event logs from different domains, then applying metrics to them for analysis.

We started with real-life event log data, covering a number of domains, as summarised in Table 3. A large number of process models were generated with these input logs, using random generation and discovery techniques (including genetic miner SETM). This was designed to obtain a large number of models of varying quality, and an abundance of metrics for quantitative and qualitative analysis from different perspectives. Measures applicable to all models, including low-quality ones, are termed *exploration measures*. Some *broad metrics* applicable to all models (such as the number of edges) were added, and this set of *exploration metrics* were calculated for all models to form the *exploration dataset*. From laboratory experience, some measures in the literature only reliably return values on higher-quality models. We designate these *delicate measures*, and use them only on models generated by established discovery techniques, the *discovery models*. Though such models are only a smaller part of the larger universe of possible models, they are among the most relevant to process mining in practice. This full set of metrics was collected on the discovery models, yielding the *discovery dataset*.

Two full iterations of the experiments were run, with small variations in the choice of metrics for each cycle. Finally, dimensional analysis was performed on both datasets.

In detailing the components of the experiment design, we first introduce the metrics collected. Secondly, we examine the model representations which make the metrics calculation practicable, stochastic play-out logs, and the technique used to construct them. Thirdly, we introduce the SETM genetic miner for model discovery, used to generate a large volume of models across a quality gradient. Fourthly, we detail the random and discovery model generation techniques.

4.1. Choice of Metrics

As reviewed in Section 3.3, metrics in stochastic process mining are an ongoing research challenge with an expanding literature. In choosing and designing the metrics in these experiments, we deliberately covered a number of different design concepts, based on a detailed study of metrics in the literature. We drew on the four control-flow quality dimensions of fitness, precision, simplicity and generalisation, using existing stochastic metrics where possible (e.g., Entropy Precision (H_P) [25]). To supplement these metrics and to explore a larger quality space, we constructed stochastic versions of control-flow measures, such as Play-out Entropy Precision (HIPT) or the small changes to Generalisation measures by trace floor and trace uniqueness [2]. We also explored the stochastic quality concepts Earth Movers' Distance, Probability Mass, and Entropy. The metrics used in the two experiment cycles are listed in Table 4. This includes the design concept behind a metric's inclusion, the abbreviation for it throughout the article, and which experiments it was used in. Categories correspond to those in Figure 4. Formal definitions for metrics are found in Appendix A.

Table 4: Metrics and their design rationale.

			Experiment	
Abbrv.	Metric Name	Design Concept	1	2
Exploration measures				
EMT	Earth Movers' With Play-out Trace	Earth Movers'	✓	✓
TOR	Trace Overlap Ratio	Probability Mass	✓	✓
TMO	Trace Probability Mass Overlap	Probability Mass	✓	
ARG	Activity Ratio Gower	Fitness	✓	✓
TRG2	Trace Ratio Gower length 2	Fitness	✓	✓
TRG3	Trace Ratio Gower length 3	Fitness	✓	✓
TRG4	Trace Ratio Gower length 4	Fitness	✓	✓
HIFT	Play-out Entropy Intersection Fitness	Fitness	✓	✓
HIPT	Play-out Entropy Intersection Precision	Precision	✓	✓
HJFT	Play-out Entropy Projection Fitness	Fitness	✓	✓
HJPT	Play-out Entropy Projection Precision	Precision	✓	✓
XPU	Existential Precision	Precision		✓
SSENC	Structural Simplicity by entity count [34]	Simplicity	✓	✓
SSEDC	Structural Simplicity by edge count [34]	Simplicity	✓	✓
SSS	Structural Simplicity incl. stochastic ratio	Simplicity	✓	✓
TGF1	Generalisation by Trace Floor (1) [2]	Generalisation	✓	
TGF5	Generalisation by Trace Floor (5) [2]	Generalisation	✓	✓
TGF10	Generalisation by Trace Floor (10) [2]	Generalisation	✓	
TGDU	Generalisation by trace uniqueness [2]	Generalisation	✓	✓
Exploration metrics				
CSS	Structural Complexity incl. stochastic	Simplicity		✓
MEC	Model Entity Count	Simplicity		✓
MGC	Model Edge Count	Simplicity		✓
HRU	Entropic Relevance w. Uniform [3]	Entropy		✓
HRZ	Entropic Relevance w. Zero Order [3]	Entropy		✓
HRR	Entropic Relevance w. Restricted Zero Order [3]	Entropy		✓
Delicate measures - discovery only				
EM	Earth Movers truncated 0.8 [25]	Earth Movers	✓	✓
H_P	Entropy Precision [30]	Precision	✓	✓

Table 4 – continued from previous page

Abbrv.	Metric Name	Design Concept	Experiment 1	 2
H_F	Entropy Recall [30]	Fitness	✓	✓
Log metrics				
LTC	Log Trace Count	Log	✓	✓
LTE	Log Event Count	Log	✓	✓

In the first experiment cycle, we found that some of these exploratory measures were very highly correlated. As this is uninformative, and the measures were therefore excluded from much of the statistical analysis, Trace Overlap Ratio (TOR), and two Generalisation by Trace Floor measures (TGF1, and TGF10) were excluded in the second cycle of experiments. In the second experiment cycle, we were able to add new metrics based on new scholarly work, particularly Alpha Precision [17] and Entropic Relevance [3]. The Existential Precision (XPU) measure is based on Alpha Precision.

The Entropic Relevance [3] metrics are originally restricted to SDFAs according to their formal definition and in the public implementation. A new implementation for this work removes this restriction by using stochastic play-out logs.

Two log metrics were included as controls: Log Trace Count (LTC) and Log Event Count (LTE).

4.2. Stochastic Language Estimation with Play-out Logs

Play-out logs [1, p41] are an established process mining technique for generating event log traces based on process models. For place-transition Petri nets, a standard way of generating play-out logs is by "playing the token game": noting the traces generated when the model advances from the initial marking through subsequent states. Play-out logs in a stochastic setting have an advantage over those with control-flow models: the stochastic model eliminates the need for arbitrary choices and assumptions when choosing between enabled transitions. Often, some assumed probability distribution is used for play-out logs on control-flow models. Stochastic models, such as SLPNs, already include explicit probability functions which define behaviour when multiple transitions are enabled. The play-out log can then substitute for the model when comparing other logs or models, allowing measurement of models which otherwise could not be practically included in the experiment.

By using a finite representation to approximate the possibly infinite stochastic language of the model, a stochastic play-out log eliminates or greatly reduces the need for multiple samples to represent possible traces Alternatives, such as random walks, will converge to representative values over many runs, and are necessary when a distribution is not well known, or when there are effects that emerge only after iterative calculation. The information in an SLPN allows alternative paths to be calculated in proportion directly when the goal is to obtain representative proportions of valid traces.

The stochastic play-out log generator implemented for these experiments is represented as function *spg*, which takes an SLPN and returns a play-out log. This can be thought of as a breadth-first search on possible traces, pruning improbable traces. To describe it, we use function $eb\colon \mathcal{N} \times \mathcal{B}(P) \to \mathbb{P}(T)$, which returns all enabled transitions for a net and a marking. Trace marking function $tg\colon \mathcal{N} \times \mathcal{B}(P) \times T \to \mathcal{B}(P)$ returns the new marking after a transition fires. Lastly, function *lab* gives a transition label as an activity sequence.

$$\begin{aligned} lab&\colon \mathcal{N} \times T \to A^* \\ lab((P,T,F,M_0,W,\lambda),t) &= \langle\rangle \text{ if } \lambda(t) = \tau \text{ else } \langle\lambda(t)\rangle \\ &\quad \text{where } t \in T \end{aligned}$$

Definition 9 (Stochastic Play-out Generation). *Let g be an SLPN model such that $g = (P,T,F,M_0,W,\lambda)$. Then sdlg is a play-out log generation functions taking an SLPN (g), a marking (m), a number of traces to be generated (b), and a maximum path length (ω). Function spg is a specialisation which starts from the initial marking. Function surplus allocates rounded amounts to specific traces.*

$$\begin{aligned} sdlg&\colon \mathcal{N} \times \mathcal{B}(P) \times \mathbb{N} \times \mathbb{N} \to \mathcal{L} \\ sdlg(g,m,b,\omega) &= \biguplus_{t \in eb(n,m)} [\sigma^f \mid \sigma = lab(g,t) + \sigma_{tl} \\ &\qquad\qquad \wedge d = \text{floor}\left(\frac{bW(t)}{W_s}\right) + surplus(g,t,m,b) \\ &\qquad\qquad \wedge r = sdlg(g, tg(n,m,t), d, \omega - 1) \\ &\qquad\qquad \wedge \sigma_{tl} \in r \\ &\qquad\qquad \wedge f = r[\sigma_{tl}]] \\ &\textit{where } W_s = \sum_{t' \in eb(g,m)} W(t') \textit{ if } eb(g,m) \neq \emptyset \textit{ and } \omega > 0 \\ sdlg(g,m,b,\omega) =& [\langle\rangle^b] \textit{ if } eb(g,m) = \emptyset \vee \omega = 0 \\ spg(g,b,\omega_0) =& sdlg(g, M_0, b, \omega_0) \end{aligned}$$

The *spg* function takes a target size as a trace "budget", then recursively splits the budget according to each possible state in a token game, and the relative weights of enabled transitions. The maximum path length ensures termination even for models that include potential livelock, or infinite loops. Traces affected by maximum path lengths are truncated.

Rounding is controlled by the *surplus* function. The value rounded across all enabled transitions is the difference between the budget b and the sum of the weighted natural number allocations to those transitions. The rounding order is first to silent transitions, then by lexical order of the transition labels, then to the transition with the least allocation, then arbitrarily. Prioritising silent transitions favours the representation of loop exit states in the SLPN translation of PPT models. Least allocation refers to the enabled transition which will receive the least trace budget from the weighted allocation. This favours representation of rarer traces on the margin. In general, the design intent is for easily reproducible outputs, where variation is limited, and which discourages the computationally expensive process of sampling over multiple runs. Instead, if a given granularity is insufficient for a particular use, larger values for the log size and maximum path length parameters can be used to achieve more granularity.

As play-out logs can be straightforwardly converted to stochastic languages, this provides a practical approximation to the Trace-Prob problem for SLPNs [27]. A play-out log M, generated without maximum path restrictions, will include all traces from paths which have a probability exceeding $\frac{1}{|M|}$. Some models have stochastic languages which fall outside this guarantee, when they have highly probable traces which exceed the maximum trace length. These models are rare in practice, and often amenable to inclusion by using a different maximum trace length. For example, long traces are mostly due to loop constructs in an underlying Petri net. In an SLPN which terminates, each iteration around the loop will construct new traces of monotonically decreasing probability. Therefore very long traces produced by loops are often also very improbable.

The algorithm has a worst-case computational complexity of $O(b \cdot \omega_0)$, where b is the target log size, and ω_0 is the maximum path length. In many cases, a large maximum path length ($\omega_0 >> 0$) parameter is desirable to ensure representative and non-truncated traces. The use of the reachability graph (implicitly, via the token game) makes the algorithm combinatorial below this ceiling. The combination of the play-out log size limit and the maximum trace limit make it highly practical across a variety of models, as observed in this experimental work. Most of the exploration measures make use of this form of trace probability estimation, by using the frequency of traces in play-out logs. This approach both increased the range of possible models and radically decreased calculation times.

In the implementation, the play-out log size was set to 1000 traces. The maximum path length was set to 5000 for the first round of experiments and 500 in the second. Almost all play-out logs experiencing maximum path truncation were from random models, and on a minority of traces.

4.3. Genetic Miner (SETM)

Genetic algorithms try a broad range of solutions according to a process loosely inspired by the "survival of the fittest" genetic adaptation of biological species to their environment. These algorithms usually start with some randomly generated potential solutions. The solutions are evaluated according to a survival function, and the best kept. These are then mutated randomly according to set rules, and the process is repeated for many iterations, or *generations*. When employed for process discovery, these algorithms are termed genetic miners [10].

A novel genetic miner for discovering stochastic process models, the Stochastic Evolutionary Tree Miner (SETM), was implemented for these experiments.

It is based on the Evolutionary Tree Miner [10]. The SETM generates random PPTs for the initial generation of models. Four possible mutations are then applied: to add a node (including control-flow nodes and silent transitions), mutate a single node, remove a subtree, or remove useless nodes (specifically to apply Preserving Compression rules [12]). These mutations also select elements randomly, while preserving valid and consistent tree weights. Models were exported as SLPNs. SETM is suitable for exploring model alternatives in a laboratory setting, with the quality of final generation models being far higher than random models, but not at the same level as those from other discovery techniques. An example mutation is shown in Figure 5.

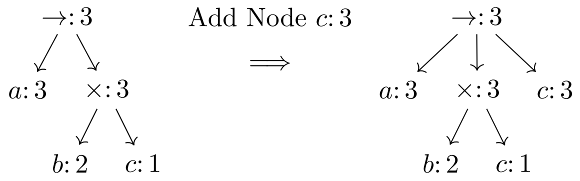


Figure 5: An example of applying an Add Node mutation on a PPT. The activity to add and the location in the tree are chosen randomly during mutation.

In our experiments, the SETM was run across 1000 generations with a survival function incorporating all the exploration measures for that cycle, with equal weight. The model with the highest survival score in each generation was added to the exploration dataset, generating a spectrum of models of moderate quality. Any additional exploration metrics were also collected for each model. The genetic miner yielded results for four of the logs in this experiment; due to timeouts after forty hours, the two logs with the most activities gave partial results, and were excluded from the dataset.

4.4. Model Generation

As well as genetic mining, the two other classes of model generation techniques employed were firstly, random generation, and secondly, existing stochastic discovery techniques.

Random models were created by randomly choosing nodes of PPTs. The random generation included silent, activity and control-flow nodes. Models larger than the arbitrary cutoffs of a tree depth of 30 or 1000 transitions were discarded, and substituted for another generated model. Models generated randomly were anticipated to have lower quality.

Models generated by existing stochastic discovery techniques were also included, and were anticipated to be of higher quality. Public implementations of stochastic process discovery techniques for GSPNs created a further 103 models relating to the selected event logs. This included GDT_SPN discovery [38], Tootpaste miner [12], and multiple estimation techniques [13] which add weight information to control-flow models. For estimation, the input control-flow models were constructed using established miners Inductive Miner [26], Fodina [9] and Split Miner [4], and all combinations of weight estimation techniques [13] and the three control-flow discovery algorithms were used. State of the art stochastic discovery techniques yield higher quality models than random models or SETM, but still yield low-quality models in a number of cases. This meant the discovery dataset still contained a wide range of metric values.

A total of 9301 models were generated. Metric implementations, metric reuse, and other experimental scaffolding, were all implemented in Java using the ProM framework[1]. Experiments were run on a Linux clustered data centre using 50 Gb of RAM.

5. Results

An exploratory quantitative analysis was performed on model metrics from Experiments 1 and 2. As Experiment 2 is a refinement of Experiment 1, we foreground Experiment 2 results in this section.

5.1. Quantitative Analysis For Component Identification

We performed analyses of correlations and principal components [23] to determine commonality and orthogonality between metrics that indicated potential quality dimensions. To weigh the sources of models equally, sources with less than 1000 models had data points repeated as if resampled. Sample sizes are quoted without resampling. We used scaled PCA, centring all input parameters to a zero mean and scaling to unit variance. This allowed metrics such as HRU to be included in the analysis, even though they could not be included in the genetic miner survival function, as their range was not known in advance.

In Experiment 1, some measures were very highly correlated (> 0.99), and these were excluded in Experiment 2, as indicated in Table 4. We examined Experiment 2 metric correlation for the exploration and discovery datasets; exploration metrics are shown in Figure 6. Correlation is indicated in blue and anti-correlation in red, with colour intensity and circle size indicating the strength of correlation. A number of correlated groups of measures can be observed in these results, and the metrics are ordered so they are clearer visually. A number of groups are already related by concept and implementation: metrics for logs (LTC,LTE), model complexity (MEC,MGC,CSS), Trace Ratios Gower (TRG2-4), simplicity (SSENC,SSEDC,SSS), and Entropic Relevance (HRZ,HRR,HRU). Some metrics showed high correlations even though they were included under different concepts. While Trace Overlap Ratio (TOR) and Earth Movers' With Play-out Trace (EMT) are included under Probability Mass and Earth Movers' respectively, the Earth Movers' Distance measure definition is closely related to probability mass. Play-out Entropy Intersection Fitness and Precision (HIFT,HIPT) are correlated, though they are intended to measure quite distinct control-flow concepts. These measures do share implementation similarities, in that they both use an entropy calculation over a

[1]. All source code is accessible at `https://github.com/adamburkegh/spm_dim`

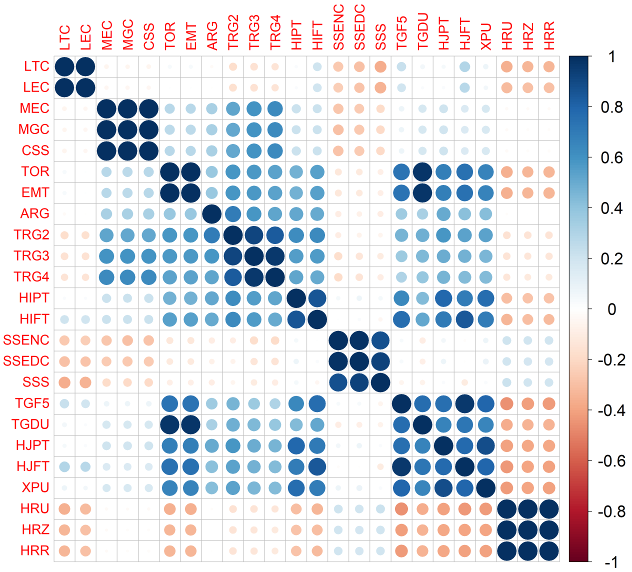


Figure 6: Correlation between exploration metrics, Experiment 2.

trace projection. More surprising, perhaps, is the group of five partially correlated metrics TRG5-XPU, which includes metrics intended to measure fitness, precision and generalisation, all together.

It is also interesting to note which metrics are not correlated or are anti-correlated. Activity Ratio Gower (ARG) is not strongly correlated with any other metric, including other subtrace ratios (TRG2-4). Metrics for fitness are not strongly correlated with one another, and similarly for precision and generalisation. The Entropic Relevance metrics (HRZ,HRR,HRU) show some anti-correlation with the TRG5-XPU grouping, and with some other tracewise metrics.

The two log metrics LTC and LTE showed correlations with the trace ratio measures and the simplicity measures (-0.45 and -0.63 respectively). In these cases, either the number of traces or events is a parameter to the measure, so this is to be expected. Correlation between LTC and LTE and other metrics is low. As these properties were already known, we then excluded the two log metrics. An Anderson-Darling test for normality showed no variables fit a normal distribution ($p < 0.001$), ruling out techniques such as factor analysis for both experiment cycles.

A scaled Principal Component Analysis (PCA) was then used to examine the basis for orthogonal components. PCA outputs a change of basis for a dataset with n measures in which the resulting n dimensions can be ranked by their maximisation of variance. The result is guaranteed to produce orthogonal

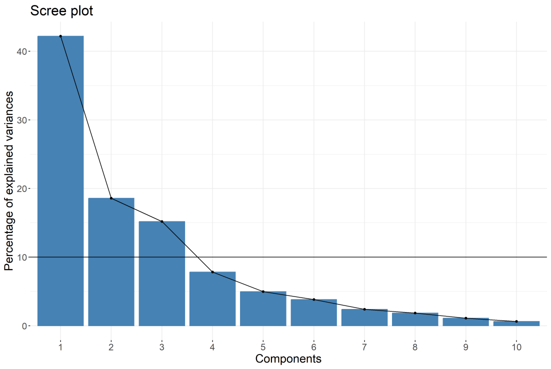


Figure 7: Scree plot of percent of variance explained by each principal component, sorted in descending order, on PCA for exploration metrics in Experiment 2.

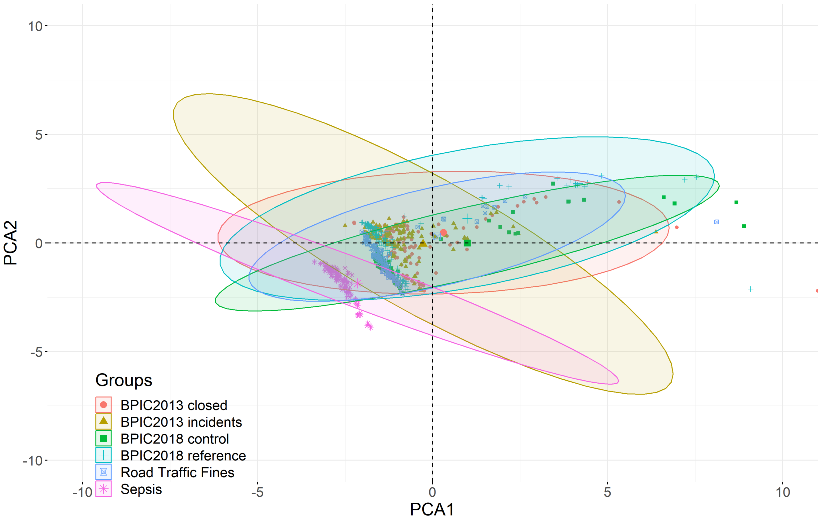


Figure 8: Exploration dataset scatterplot against PCA components 1 and 2. Ellipses and colour distinguish source logs.

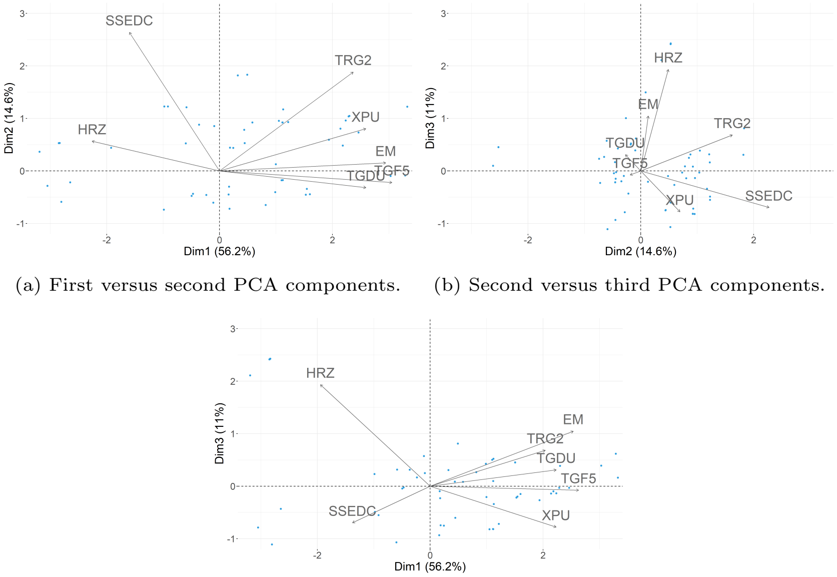
dimensions (in PCA terminology, components), and is often used for dimensional reduction by choosing the highest-ranked components. It is employed here to identify potential orthogonal dimensions with an empirical basis. A scree plot of the variance covered by the components was used to estimate the number of possible dimensions. Figure 7 shows the scree plot for exploration metrics in Experiment 2. These three components explain 39.3%, 27.5% and 11.3% of the variance respectively, and the remaining components explain at most 6% each. Results for the discovery dataset, which includes delicate measures, were similar, explaining 45.9%, 16.9% and 12.3% respectively; these are also similar to Experiment 1's results. The elbow technique and other methods suggest a fourth useful component may exist. Both experiments showed three orthogonal components, with a possible fourth. This fourth component was not clearly identified with an underlying concept, and explained less than 10% of the variance. We chose, conservatively, to exclude it from further analysis.

We performed robustness tests to examine whether components could be identified with any element in the experiment setup itself, and for consistency across data subsets. Specifically, components were compared to log sources and to model generation sources (i.e. random/SETM/discovery) to check whether the PCA was simply identifying these input partitions. Classification by log and by model source varied across PCA components for both experiments. Figure 8 shows one example classification by log across the first two PCA components for the exploration dataset. Though models from different logs, as represented by the ellipses, have different quality profiles, they are not straightforwardly identified with components in either instance. Since this analysis suggests the components reflect deeper underlying regularities in the dataset, a second round of analysis, below, breaks these components down further.

5.2. New Metrics Yield New Components

Different components were identified by the two cycles of experiments. Within experiments, components differed across exploration and discovery datasets, and the order of influence of the second and third components changed, but similar metrics were associated with them in both cases. In both experiments, the first component is associated with the Earth Movers' Distance (EM) and Trace Generalisation by floor (TGF) measures. A second component is associated with Simplicity measures. However, in Experiment 1, the third component has an association with Entropy Precision and Recall (H_P,H_F) and Trace Ratio measures. In Experiment 2, the Structure Stochastic Complexity (CSS) and on the discovery dataset, the Entropy Relevance metrics (HR*), are closely associated with a third component, and not correlated with Entropy Precision and Recall.

To clarify these relationships and seek a more parsimonious description of the data with fewer input metrics, we performed a second round of analysis. Metrics that correlated with another at > 0.9 were pruned. When choosing from a pair of correlated metrics, we prioritised first metrics from published literature, then metrics that correlated to delicate measures on the discovery dataset, then conceptually clearer metrics. By conceptual clarity, we refer to a



(a) First versus second PCA components.

(b) Second versus third PCA components.

(c) First versus third PCA components.

Figure 9: PCA biplots for selected metrics on the discovery dataset, comparing three components. The approximate orthogonality of the Native Metrics Earth Movers' Distance (EM), Simplicity by Edge Count (SSEDC), and Entropic Relevance Zero Order (HRZ) can be observed.

decision about whether to include the complexity metric CSS, or the simplicity metric used as an input to its calculation. Since CSS has a known formal relationship to simplicity measure SSS, one of these metrics could be excluded. Examining PCA biplots, the resulting dimensions were more clearly aligned with named measures, and hence existing concepts, when based on simplicity, so this was the metric included.There were seven metrics remaining after pruning. The six metrics available on all models were Existential Precision (XPU), Generalisation by Trace Floor (5) (TGF5), Generalisation by Trace Uniqueness (TGDU), Entropic Relevance with Zero Order (HRZ), Trace Ratio Gower length 2 (TRG2), and Structural Simplicity by edge count (SSEDC). The seventh is the Earth Movers' Distance (EM), which is a delicate measure, and so not practically available for all models. In choosing these metrics, we evaluated their correlations on both exploration and discovery datasets.]Figure 9 shows PCA biplots for the remaining metrics for the discovery dataset in Experiment 2. A PCA biplot plots input dataset variables as vectors against two selected PCA components. Combinations of the first three components are shown to give a sense of the metrics in the three dimensional space constructed by three components. For example, The Earth Mover's Distance is strongly associated with the first component, labeled Dim1. The six exploration metrics, EM, and the PCA performed on them across both Experiment 2 datasets, are the immediate underlying data for our proposed quality dimensions.

In summary, three PCA components are shown across both experiments and across exploration and discovery datasets. Two components are similar across the two experiments; a third differs in composition under the influence of new metrics. A second analysis, centred on the Experiment 2 metrics, suggests candidates for the quality dimensions proposed in Section 6.

6. Quality Dimensions

From the experimental results above, we propose three quality dimensions, which we name *Adhesion, Simplicity*, and *Relevance*, and which we characterise below. To apply these dimensions for quantitative measurement and comparison, we provide three sets of measures, corresponding to two interpretations of the experiments. In the first view, the experiments and analysis are considered to have revealed hidden underlying regularities, akin to physical laws, and corresponding to PCA components, which the combined weighted measures approximate. We call this view *dimensional realism.* In the second view, the experiments and analysis are used to reveal which metrics effectively partition the quality space, by capturing variance and their orthogonality to other metrics. Those metrics are then used directly, transformed only by scaling, and so this is termed the *native metrics* view. Though the problem of choosing synthetic or direct metrics is not a new one in science, the dimensional realist / native metrics terminology is, to our knowledge, new, at least as applied to the specific problem of dimensional choice.

6.1. Three Model-Log Quality Dimensions

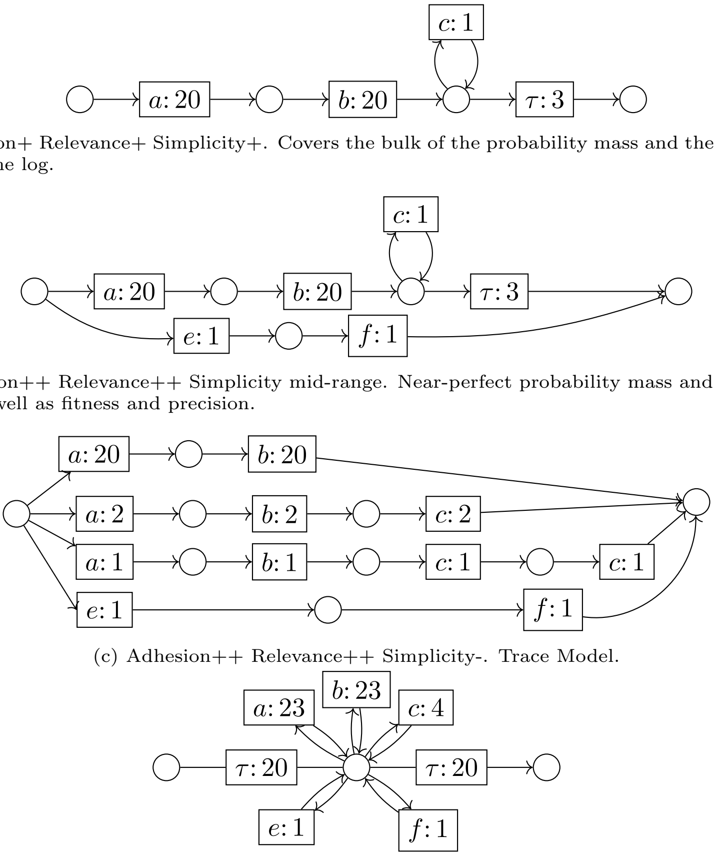
Adhesion. To represent how little effort is required to transform one stochastic language into another, we use the term *adhesion.* Such a transformation can involve both modifying which traces the process accepts, and the probability of those traces. An informal interpretation is how few changes a team needs to make to adhere to a different way of working.

Relevance. Relevance measures the informational cost of reconstructing the complete traces from the event log with the model. The dimension name is directly inspired by the Entropic Relevance metrics [3]. [2] This definition also constrains the concept to a *trace level* view of the model and log, where completed cases are considered relevant, but even slightly differing traces are not.

[2] As Relevance concerns the amount of information shared between log and model, it is related conceptually to Entropy, the name used for the second dimension in our previous work [14]. However, as noted in Section 5, the underlying metrics differ significantly.

Simplicity. Parsimony in models is well-recognized as a virtue in science generally [39], and a desirable dimension in process models specifically [1, p118]. Simplicity represents the number of explicit syntactic features of the model. Encoding costs, or the movement of complexity into a notation, are not considered, but are mitigated in this work by using SLPNs as a common model structure. Syntactic simplicity, as in this dimension, is also distinct from behavioural simplicity [24], for example where a model with many elements might describe a process with very limited behaviour. Behavioural simplicity is more associated with the Relevance dimension.

Example models illustrating these dimensions are examined in Section 6.4, and in Figures 10 and 11.



(a) Adhesion+ Relevance+ Simplicity+. Covers the bulk of the probability mass and the completed traces in the log.

(b) Adhesion++ Relevance++ Simplicity mid-range. Near-perfect probability mass and completed traces, as well as fitness and precision.

(c) Adhesion++ Relevance++ Simplicity-. Trace Model.

(d) Adhesion- Relevance- Simplicity mid-range. Flower Model with event frequencies.

Figure 10: Models exemplifying adhesion and entropy variations relative to log $L_E = [\langle a, b\rangle^{20}, \langle a, b, c\rangle^{2}, \langle a, b, c, c\rangle^{1}, \langle e, f\rangle^{1}]$.

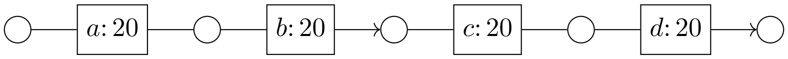


Figure 11: Model exemplifying adhesion and entropy variations relative to logs L_F and L_G . This also shows how the quality of a single model will vary when compared to different logs.
Adhesion+ Relevance- Simplicity+. Partial major trace relative to log $L_F = [\langle a, b, c, d, e\rangle^{20}, \langle a, b\rangle^{1}, \langle b, c\rangle^{1}, \langle c, d\rangle^{1}, \langle d, e\rangle^{1}]$.
Adhesion mid-range(-) Relevance mid-range(+) Simplicity+. Half log coverage relative to log $L_G = [\langle a, b, c, d\rangle^{50}, \langle e, f, g\rangle^{50}]$.

6.2. Dimensional Realist View

In the dimensional realist view, PCA components are taken as the model of the underlying space. After excluding highly correlated metrics, we use those remaining as a synthetic estimator for each dimension, which can be used as a measure for that dimension. To construct this estimator, we start from the definition of a principal component in PCA.

In a Principal Component Analysis, each element is centred by its mean and scaled by its standard deviation. Take the metrics $m_1 ... m_6$ included in the analysis, then m_i to be the i-th metric, $\bar{x_i}, s_i$ to be the corresponding mean and standard deviation, and P_{Xi} the PCA loading for one of the PCA components.

$$
\begin{aligned}
X_D =& P_{X1}\frac{m_1 - \bar{x_1}}{s_1} + P_{X2}\frac{m_2 - \bar{x_1}}{s_2} + ... \\
=& \frac{P_{X1} \cdot m_1}{s_1} - \frac{P_{X1} \cdot \bar{x_1}}{s_1} + \frac{P_{X2} \cdot m_2}{s_2} - \frac{P_{X2} \cdot \bar{x_2}}{s_2} + ... \\
=& \frac{P_{X1} \cdot m_1}{s_1} + \frac{P_{X2} \cdot m_2}{s_2} + ... - \sum_{i=1...6} \frac{P_i \bar{x_i}}{s_i}
\end{aligned}
$$

This linear equation is reorganised for our specific use case by noting that the concluding sum is a constant, and renaming the constant factors $\frac{P_{Xi}}{s_i}$ after their corresponding metrics. For example, $\frac{P_{X1} \cdot m_1}{s_1}$ is replaced by MX_{XPU}. Definition 10 uses the reorganised formula to define three metrics, one for each quality dimension.

Definition 10 (Dimensional Realist Metrics).

$$
\begin{aligned}
X_D =& MX_{XPU} \cdot XPU + MX_{TGF5} \cdot TGF5 + MX_{TGDU} \cdot TGDU \\
& + MX_{HRZ} \cdot HRZ + MX_{TRG2} \cdot TRG2 + MX_{SSEDC} \cdot SSEDC + K_{XD}
\end{aligned}
$$

$$
\textit{where } K_{XD} = \sum_{i=1...6} \frac{P_{Xi} \cdot \bar{x_i}}{s_i}
$$

and $X \in \{A, R, S\}$ for (A)dhesion A_D, (R)elevance R_D and (S)implicity S_D, respectively.

The empirically derived factors and constants are summarised in Table 5.]Min/max values are in Table 6. Min/max scaling is applied in Table 7. The tables show not all factors are of the same importance. For the third dimension, simplicity, the P_X values for the entropic relevance (HRZ) factor is zero at four digits of significance, excluding it. The values for existential precision XPU are very small (-0.00113), such that it could also be excluded. The PCA component for simplicity minimised when input simplicity was highest, so we have reversed the signs of the factors and constant so that high values indicate higher quality. The factors show the conceptual limitations of dimensional realism: the explicit simplicity measure based on edge count SSEDC has a similar contribution to both Relevance and Simplicity DR dimensions, and the relevance metric HRR contributes only slightly more to Relevance than Adhesion.

The calculations in Definition 10 yield metrics rather than measures, as they may range beyond $[0, 1]$, including negative values. Min/max scaling was applied to achieve a measure in an applied setting, as seen in Table 7. They are derived by noting that five of the six input metrics are already normalised within a $[0, 1]$ range. The remaining input metric, Entropic Relevance Zero Order (HRZ), is observed in experimental data to have a maximum of 52.55 and range of 50.68. To calculate scale ranges, we treated HRZ as having range $[0, 60]$, and calculated overall ranges based on the theoretical extremes of each input variable. We expect the scaled (measure) versions of DR metrics in Table 7 to be the most immediately useful to those measuring Adhesion, Relevance and Simplicity, as they are easier to compare and intuitively understand. Unscaled factors, constants and min/max values for this study are in Tables 5 and 6, to allow users of the measures to rescale if different extrema are observed in their applied domain.

Table 5: Dimensional realist factors and constants from exploration metrics.

Metric	Adhesion (MA)	Relevance (MR)	Simplicity (MS)
XPU	1.518	-0.693	-0.087
TGF5	1.540	0.358	-0.709
TGDU	1.425	0.579	0.842
HRZ	-0.0476	-0.0492	0.000
TRG2	1.090	-3.527	-3.582
SSEDC	0.610	-2.157	2.518
Constant	Adhesion (K_{AD})	Relevance (K_{RD})	Simplicity (K_{SD})
	3.09	-2.92	0.97

Table 6: Dimensional realist min/max from exploration metrics.

	Adhesion A_{DM}	Relevance R_{DM}	Simplicity S_{DM}
Minimum	-5.94	-6.40	-5.36
Maximum	3.09	3.87	2.38

Table 7: Dimensional realist factors and constants from exploration metrics, after min/max scaling.

Metric	Adhesion (MA)	Relevance (MR)	Simplicity (MS)
XPU	0.168	-0.0675	-0.00113
TGF5	0.170	0.0349	-0.0916
TGDU	0.158	0.05637	0.109
HRZ	-0.00526	-0.00479	0.000
TRG2	0.121	-0.344	-0.463
SSEDC	0.0674	-0.210	0.325
Constant	Adhesion (K_{AD})	Relevance (K_{RD})	Simplicity (K_{SD})
	0.342	-0.285	0.126

6.3. Native Metrics View

In the native metrics view, a small number of metrics that partition the space are each identified with a particular dimension. The metrics are not fully orthogonal, in the sense that they show partial correlation and do not intersect at perfect right angles. Compared with the dimensional realist view, this loses some information from the excluded metrics, but it is simpler. Since metrics were originally designed with the intent to capture some particular aspect of model quality, it is also conceptually clearer. Native metrics are listed in Table 8. Figure 12 plots a PCA using only the selected metrics on the exploration dataset. As the current implementation of Earth Movers' Distance is a delicate measure, we include a substitute of TGF5 (correlation = 0.75) for when it is unavailable.

Table 8: Native metrics for Adhesion, Relevance and Simplicity, chosen by joint PCA orthogonality and conceptual linkage.

Data Set	Adhesion	Relevance	Simplicity
Exploration	Generalisation by Trace Floor (5) TGF5	Entropic Relevance w. Zero Order HRZ	Structural Simplicity by edge count SSEDC
Discovery	Earth Movers truncated EM	Entropic Relevance w. Zero Order HRZ	Structural Simplicity by edge count SSEDC

6.4. Interpretation of the Adhesion, Relevance, and Simplicity Dimensions Using Example Models

To give a sense of the three dimensions, and metrics that approximate them, Figures 10 and 11 illustrate extreme cases. Cases were informed by using alternative fitness functions for SETM which neglected one or more dimensions, and then the resulting models were optimised for extremity by hand. The corresponding metrics are summarised in Table 9.

The examples in Figure 10 use log $L_E = [\langle a,b\rangle^{20}, \langle a,b,c\rangle^{2}, \langle a,b,c,c\rangle^{1}, \langle e,f\rangle^{1}]$. Note that this log has one frequently occurring trace, $\langle a,b\rangle$, which dominates the probability mass, with some variations, and one completely different trace, $\langle e,f\rangle$. Model 10a achieves high adhesion and relevance by covering the main trace and its variations with plausible weights. In model 10b, almost perfect

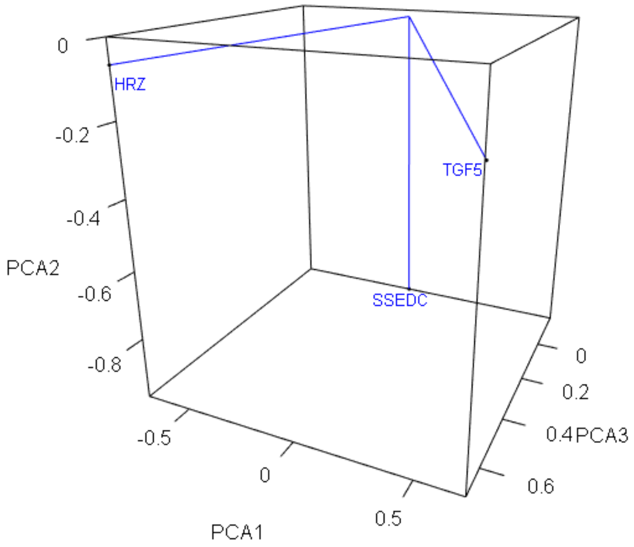


Figure 12: 3D plot of metrics Trace Generalisation by Uniqueness (5) (TGF), Entropic Relevance Zero Order (HRZ), and Simplicity by Edge Count (SSEDC), against PCA dimensions for those three metrics, on the exploration dataset.

Table 9: Quality metrics for paradigm examples in Figures 10 and 11.

			Adhesion			Relevance		Simplicity	
Model	Fig.	Log	ADM	TGF5	EM	RDM	HRZ	SDM	$\|F\|$
A+ R+ S++	10a	L_E	0.80	0.72	0.91	0.72	2.76	0.67	4
A++ R++ S+	10b	L_E	0.83	0.96	0.94	0.74	1.76	0.35	10
Trace; A++ R++ S-	10c	L_E	0.91	1.00	0.91	0.62	1.08	0.18	22
Flower; A- R mid S mid	10d	L_E	0.28	0.00	0.21	0.88	6.37	0.57	29
Partial Major Trace; A+ R- S+	11	L_F	0.35	0.00	0.73	0.51	14.42	0.52	8
Half log; A mid R mid S+	11	L_G	0.65	0.50	0.50	0.73	6.92	0.42	8

adhesion and relevance has been achieved at the cost of simplicity. This model also has perfect control-flow fitness and precision. Trace model 10c achieves perfect adhesion and relevance at the cost of poor simplicity.

Flower model 10d has poor adhesion and relevance, despite perfect control-flow fitness. The many possible alternative traces generated by a flower model mean little probability mass is devoted to either similar traces (for adhesion) or entire traces (for relevance). This flower model has weights taken from the event frequencies in the log (equivalent to the Frequency Estimator [13]), but this has had little quality impact without further structure constraining the space of possible subtraces.

In Figure 11, two logs are considered versus the same simple sequence model. In log L_F, a single trace, $\langle a, b, c, d, e \rangle$ accounts for the vast majority of cases (partial major trace). Every event except the last is covered, resulting in high (not perfect) adhesion, but poor relevance. In L_G (half log), there are two frequently occurring traces, one of which matches the model perfectly, and one not at all. As half probability is a state that can minimise entropy, we expect somewhat higher relevance, with at best mid adhesion.

In Table 9, we can see how the metrics identified in Sections 6.2 and 6.3 perform against these extreme cases. To calculate these figures, a modification had to be made to the dimensional realist metrics, as the SSEDC metric returns zeroes for very small logs, such as those explored in this section. We substituted a small log variant which divided model edge count by the product of trace variants and average trace length in the dimensional realist measures, and provide edge count ($|F|$) as a substitute for the native metric.

For Adhesion, the earth movers (EM) and ADM metrics reflected the process edits needed across all models. Trace Generalisation by floor (TGF5) generally followed the pattern, but returned zero for the Partial Major Trace scenario, due to the small log.

For Relevance metrics, the Entropic Relevance (HRZ) metric behaves consistently with expectations across the models, though it also shows correlation with Adhesion metrics and makes it difficult to construct a scenario with high Relevance and low Adhesion. The RDM measure does show high relevance for the Flower model. Though being able to map models to all corners of orthogonal dimensions does meet one goal, the conceptual obscurity may undermine use of this as a productive design constraint for model construction.

For Simplicity metrics, both the edge count and the SDM measure show some consistency with expectations. The SDM measure punishes the perfect model more than the trace model, however, which is due to the influence of factors such as Trace Ratio (TRG2), and again works against intuitive understanding.

7. Discussion

So far, as shared in Section 5, we have seen variations and correlations across the twenty-five metrics collected on 9301 models generated from six real-life logs. These have been analysed quantitatively, identifying the three quality dimensions Adhesion, Relevance and Simplicity. We saw that these components were not associated with known regularities in the experimental inputs, in the form of logs or generation sources. We chose metrics based on the dimensions using two alternative dimensional interpretations. We also investigated example models at the extremes of the dimensions, based on these metrics.

In this section, we discuss the two dimensional interpretations, Dimensional Realism versus Native Metrics, in more detail. We also cover potential applications, and limitations of the current design.

7.1. Contrasting Dimensional Interpretations

The two interpretative views we introduce have complementary strengths and weaknesses. Dimensional realism (DR), by treating PCA components as real underlying structures, allows for the revelation of features not directly illustrated by any given metric. Being based directly on the outcome of a principal components analysis, DR is also guaranteed to yield perfectly orthogonal dimensions. Yet that same guarantee, and the "synthetic" nature of PCA, also makes DR dimensions sensitive to the exact metrics chosen as inputs. As new metrics are proposed by the community, or excluded for changing design reasons, the associated dimensions will change.

Native metrics allow for a clearer association between the design concept of a particular metric, the resulting measurement on a specific model, and the dimension it is identified with. However, they lose some information on the underlying quality space, being limited to one metric per dimension. Those metrics also only approximate orthogonality, and may conceal features that DR metrics can indirectly indicate.

7.2. Potential Applications

Stochastic phenomena are widespread in the real world, and stochastic models are used widely in settings from Operations Research [43], to healthcare [33] and performance prediction [40]. For stochastic process models specifically, more automated discovery techniques are emerging, but existing metrics for evaluating their quality are not sufficient. To use these discovered models intelligently, more widely applicable metrics, and a better understanding of their meaning and relations, are needed. We envision the dimensions and metrics proposed above can advance this understanding.

It is often necessary in process modelling and mining to choose among potential representations based on a specific use case. For example, very detailed models with lots of elements may be very accurate, but make a descriptive model difficult to explain. The decision is ultimately one of practitioner judgement. Using these dimensions, and their associated metrics, that judgement can now be better informed in a fine-grained way. A practitioner or modeller can decide how much their model adheres strictly to the process it describes, or how much information on complete traces to sacrifice in working with a simpler model. Better tooling can help share this information with users in the right context. For example, an intelligent slider or two dimensional "colour-picker" widget could allow a user to navigate the right level of quality and complexity for their use case. Commercial process mining tools already make use of frequencies. This implicit use of stochastic models could be made explicit, and our work above can help in designing and implementing such features. For instance, there is little point in calculating two highly correlated metrics.

Another strength of stochastic model comparison can be in exploring change in processes over time, a phenomena known as *concept drift*. The Earth-movers' distance metric (EM) has been used to measure this kind of change [8]. Using these approaches, we can imagine a plant manager seeing their manufacturing process has changed, because of the changing importance of existing paths of workflows in the plant, a condition hard to detect without a stochastic perspective. Tools might visualise the impact of the change, for more rapid, productive troubleshooting.

The metrics based on play-out trace probability (Definition 9) may now be calculated on more types of models, and with relatively low computational cost. Our public reference implementation also shows the feasibility of implementing these metrics in industrial tools.

7.3. Limitations

Although a wide range of models and logs were used, other datasets may reveal other elements. Larger logs of over 200,000 traces or 16 activities were not used, and SETM use was limited by larger numbers of log activities. The stochastic models used were limited to SLPNs, though some of the discovery models were derived from discovery algorithms with BPMN output, and a mix of discovery algorithms was used. The use of PPTs for random generation and for the seed generation in the SETM limits the possible models generated, though it also constrains them to sound models with consistency constraints on stochastic weights.

The example models in Section 6.4 help show the dimensions in use and build an intuition of how these quality dimensions apply in practice. A model with high Relevance and low Adhesion was not identified in this research: the closest was a model with middling to low Adhesion (in Figure 11). Such a model would be informative, make the examples symmetrical, and clarify the relationship between different dimensions in practice.

That new metrics changed the dimensional analysis shows the way this empirical work will have to evolve as new data is available. Models discovered from synthetic data can also be used to deepen our understanding of quality measurement of stochastic processes. As an experimentally derived theory, further experimentation will be the ultimate test of generality for all of the proposed dimensions.

8. Conclusion

Organisations may be understood by what they do, and what they do may be described by stochastic process models. To understand the quality of such models, we conducted two experiments studying stochastic process model quality metrics and relationships. Models were generated from six real-life logs and collected using both random model generation and stochastic process discovery. Analyzing a variety of computationally cheap metrics across thousands of models, and metrics from the literature, three quality dimensions were observed with the help of principal component analysis. We named these dimensions Adhesion, Relevance and Simplicity, evolving our understanding of the three dimensions during the course of the experiments and analysis. Based on the analysis, we suggested possible metrics for these dimensions, and showed their use on example models demonstrating their extremes.

A number of avenues are open for future work. The methods here suggest extensions of existing techniques, and new implementations of those techniques. Large model datasets may be used to expand the empirical foundations of process mining in other ways. By integrating the dimensions and their associated metrics into process mining tools, we can investigate their applied use by practitioners. Lastly, it is a spur for the invention of new metrics based on the proposed dimensions, and for the theory to be challenged with further empirical tests.

Acknowledgements. Thanks to Artem Polyvyanyy for pointing out that the entropic relevance measures could be straightforwardly extended to any model whose stochastic language was known. The eResearch Office at QUT provided computational resources for this work.

References

[1] van der Aalst, W., 2016. Process Mining: Data Science in Action. 2 ed., Springer-Verlag, Berlin Heidelberg. doi:10.1007/978-3-662-49851-4.

[2] van der Aalst, W.M., 2018. Relating process models and event logs - 21 conformance propositions., in: ATAED@ Petri Nets/ACSD, pp. 56–74.

[3] Alkhammash, H., Polyvyanyy, A., Moffat, A., García-Bañuelos, L., 2022. Entropic relevance: A mechanism for measuring stochastic process models discovered from event data. Information Systems 107, 101922. doi:10.1016/j.is.2021.101922.

[4] Augusto, A., Conforti, R., Dumas, M., La Rosa, M., Polyvyanyy, A., 2019. Split miner: automated discovery of accurate and simple business process models from event logs. Knowledge and Information Systems 59, 251–284. doi:10.1007/s10115-018-1214-x.

[5] Bause, F., Kritzinger, P., 2002. Stochastic Petri Nets: An Introduction to the Theory. Vieweg+Teubner Verlag.

[6] Beer, S., 2002. What is cybernetics? Kybernetes 31, 209–219. doi:10.1108/03684920210417283. publisher: MCB UP Ltd.

[7] Benevento, E., Pegoraro, M., Antoniazzi, M., Beyel, H.H., Peeva, V., Balfanz, P., van der Aalst, W.M.P., Martin, L., Marx, G., 2023. Process modeling and conformance checking in healthcare: A COVID-19 case study, in: Montali, M., Senderovich, A., Weidlich, M. (Eds.), Process Mining Workshops, Springer Nature Switzerland, Cham. pp. 315–327. doi:10.1007/978-3-031-27815-0_23.

[8] Brockhoff, T., Uysal, M.S., Aalst, W.M.P.v.d., 2020. Time-aware Concept Drift Detection Using the Earth Mover's Distance, in: 2020 2nd International Conference on Process Mining (ICPM), pp. 33–40. doi:10.1109/ICPM49681.2020.00016.

[9] vanden Broucke, S.K.L.M., De Weerdt, J., 2017. Fodina: A robust and flexible heuristic process discovery technique. Decision Support Systems 100, 109–118. doi:10.1016/j.dss.2017.04.005.

[10] Buijs, J.C.A.M., van Dongen, B.F., van der Aalst, W.M.P., 2012. A genetic algorithm for discovering process trees, in: 2012 IEEE Congress on Evolutionary Computation, pp. 1–8. doi:10.1109/CEC.2012.6256458. ISSN: 1941-0026.

[11] Buijs, J.C.A.M., van Dongen, B.F., van der Aalst, W.M.P., 2014. Quality dimensions in process discovery: The importance of fitness, precision, generalization and simplicity. International Journal of Cooperative Information Systems 23, 1440001. doi:10.1142/S0218843014400012.

[12] Burke, A., Leemans, S.J.J., Wynn, M.T., 2021a. Discovering stochastic process models by reduction and abstraction, in: Application and Theory of Petri Nets and Concurrency, Springer. pp. 312–336. doi:10.1007/978-3-030-76983-3_16.

[13] Burke, A., Leemans, S.J.J., Wynn, M.T., 2021b. Stochastic process discovery by weight estimation, in: Leemans, S., Leopold, H. (Eds.), Process Mining Workshops, Springer International Publishing, Cham. pp. 260–272. doi:10.1007/978-3-030-72693-5_20.

[14] Burke, A.T., Leemans, S.J., Wynn, M.T., van der Aalst, W.M., ter Hofstede, A.H., 2022. Stochastic process model-log quality dimensions: An experimental study, in: 2022 4th International Conference on Process Mining (ICPM), pp. 80–87. doi:10.1109/ICPM57379.2022.9980707.

[15] Camargo, M., Dumas, M., González-Rojas, O., 2020. Automated discovery of business process simulation models from event logs. Decision Support Systems 134, 113284. doi:10.1016/j.dss.2020.113284.

[16] Corallo, A., Lazoi, M., Striani, F., 2020. Process mining and industrial applications: A systematic literature review. Knowledge and Process Management 27, 225–233. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/kpm.1630, doi:https://doi.org/10.1002/kpm.1630. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/kpm.1630.

[17] Depaire, B., Janssenswillen, G., Leemans, S.J.J., 2022. Alpha precision: Estimating the significant system behavior in a model, in: Di Ciccio, C., Dijkman, R., del Río Ortega, A., Rinderle-Ma, S. (Eds.), Business Process Management Forum, Springer International Publishing, Cham. pp. 120–136. doi:10.1007/978-3-031-16171-1_8.

[18] Galic, G., Wolf, M., 2021. Global Process Mining Survey 2021: Delivering Value with Process Analytics - Adoption and Success Factors of Process Mining. Technical Report. Deloitte. URL: https://www2.deloitte.com/de/de/pages/finance/articles/global-process-mining-survey-2021.html.

[19] Garcia, C.d.S., Meincheim, A., Faria Junior, E.R., Dallagassa, M.R., Sato, D.M.V., Carvalho, D.R., Santos, E.A.P., Scalabrin, E.E., 2019. Process mining techniques and applications – A systematic mapping study. Expert Systems with Applications 133, 260–295. doi:10.1016/j.eswa.2019.05.003.

[20] Gower, J.C., 1971. A general coefficient of similarity and some of its properties. Biometrics , 857–871.

[21] Janssenswillen, G., Depaire, B., Faes, C., 2020. Enhancing Discovered Process Models using Bayesian Inference and MCMC, in: Proceedings of the 2020 BPI Workshop, pp. 295–307.

[22] Janssenswillen, G., Donders, N., Jouck, T., Depaire, B., 2017. A comparative study of existing quality measures for process discovery. Information Systems 71, 1–15. doi:10.1016/j.is.2017.06.002.

[23] Jolliffe, I., 2011. Principal Component Analysis, in: Lovric, M. (Ed.), International Encyclopedia of Statistical Science. Springer, Berlin, Heidelberg, pp. 1094–1096. doi:10.1007/978-3-642-04898-2_455.

[24] Kalenkova, A., Polyvyanyy, A., La Rosa, M., 2020. A framework for estimating simplicity of automatically discovered process models based on structural and behavioral characteristics, in: Fahland, D., Ghidini, C., Becker, J., Dumas, M. (Eds.), Business Process Management, Springer International Publishing, Cham. pp. 129–146. doi:10.1007/978-3-030-58666-9_8.

[25] Leemans, S.J.J., van der Aalst, W.M.P., Brockhoff, T., Polyvyanyy, A., 2021. Stochastic process mining: Earth movers' stochastic conformance. Information Systems 102, 101724. doi:10.1016/j.is.2021.101724.

[26] Leemans, S.J.J., Fahland, D., van der Aalst, W.M.P., 2018. Scalable process discovery and conformance checking. Software & Systems Modeling 17, 599–631. doi:10.1007/s10270-016-0545-x.

[27] Leemans, S.J.J., Maggi, F.M., Montali, M., 2022. Reasoning on labelled Petri nets and their dynamics in a stochastic setting, in: Di Ciccio, C., Dijkman, R., del Río Ortega, A., Rinderle-Ma, S. (Eds.), Business Process Management, Springer International Publishing, Cham. pp. 324–342. doi:10.1007/978-3-031-16103-2_22.

[28] Leemans, S.J.J., Mannel, L.L., Sidorova, N., 2023. Significant stochastic dependencies in process models. Information Systems , 102223URL: https://www.sciencedirect.com/science/article/pii/S0306437923000595, doi:10.1016/j.is.2023.102223.

[29] Leemans, S.J.J., Polyvyanyy, A., 2020. Stochastic-aware conformance checking: An entropy-based approach, in: Dustdar, S., Yu, E., Salinesi, C., Rieu, D., Pant, V. (Eds.), Advanced Information Systems Engineering, Springer International Publishing, Cham. pp. 217–233. doi:10.1007/978-3-030-49435-3_14.

[30] Leemans, S.J.J., Polyvyanyy, A., 2023. Stochastic-aware precision and recall measures for conformance checking in process mining. Information Systems 115, 102197. doi:10.1016/j.is.2023.102197.

[31] Maggi, F.M., Montali, M., Penaloza, R., 2020. Temporal logics over finite traces with uncertainty, in: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 10218–10225. Issue: 06.

[32] Mannhardt, F., Leemans, S.J.J., Schwanen, C.T., de Leoni, M., 2023. Modelling data-aware stochastic processes - Discovery and conformance checking, in: Gomes, L., Lorenz, R. (Eds.), Application and Theory of Petri Nets and Concurrency, Springer Nature Switzerland, Cham. pp. 77–98.

[33] Mazhar, T.I., Tariq, A., Leemans, S.J., Goel, K., Wynn, M.T., Staib, A., 2023. Stochastic-aware comparative process mining in healthcare, in: International Conference on Business Process Management, Springer. pp. 341–358.

[34] Mendling, J., Neumann, G., van der Aalst, W., 2007a. Understanding the occurrence of errors in process models based on metrics, in: Meersman, R., Tari, Z. (Eds.), On the Move to Meaningful Internet Systems 2007: CoopIS, DOA, ODBASE, GADA, and IS, Springer, Berlin, Heidelberg. pp. 113–130. doi:10.1007/978-3-540-76848-7_9.

[35] Mendling, J., Reijers, H.A., Cardoso, J., 2007b. What makes process models understandable?, in: Alonso, G., Dadam, P., Rosemann, M. (Eds.), Business Process Management, Springer, Berlin, Heidelberg. pp. 48–63. doi:10.1007/978-3-540-75183-0_4.

[36] Moreira, C., Haven, E., Sozzo, S., Wichert, A., 2018. Process mining with real world financial loan applications: Improving inference on incomplete event logs. PLOS ONE 13, e0207806. doi:10.1371/journal.pone.0207806. publisher: Public Library of Science.

[37] Polyvyanyy, A., Smirnov, S., Weske, M., 2008. Reducing the complexity of large EPCs, in: Modellierung betrieblicher Informationssyteme (MobIS): Geschäftsprozessmanagement mit EPK, Gesellschaft für Informatik, Saarbrücken, Germany. pp. 195–207.

[38] Rogge-Solti, A., van der Aalst, W.M.P., Weske, M., 2014. Discovering Stochastic Petri Nets with arbitrary delay distributions from event logs, in: Lohmann, N., Song, M., Wohed, P. (Eds.), Business Process Management Workshops, Springer International Publishing, Cham. pp. 15–27. doi:10.1007/978-3-319-06257-0_2.

[39] Sober, E., 2015. Ockham's Razors: A User's Manual. Cambridge University Press.

[40] Tsironis, L.C., Sfiris, D.S., Papadopoulos, B.K., 2010. Fuzzy performance evaluation of workflow Stochastic Petri Nets by means of block reduction. IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans 40, 352–362. doi:10.1109/TSMCA.2009.2035303.

[41] Von Luxburg, U., Schölkopf, B., 2011. Statistical learning theory: Models, concepts, and results, in: Gabbay, D.M., Hartmann, S., Woods, J. (Eds.), Handbook of the History of Logic. North-Holland. volume 10 of *Inductive Logic*, pp. 651–706. doi:10.1016/B978-0-444-52936-7.50016-1.

[42] Watanabe, A., Takahashi, Y., Ikeuchi, H., Matsuda, K., 2022. Grammar-based process model representation for probabilistic conformance checking, in: 2022 4th International Conference on Process Mining (ICPM), IEEE. pp. 88–95.

[43] Zhou, M., Venkatesh, K., 1999. Modeling, Simulation, and Control of Flexible Manufacturing Systems: A Petri Net Approach. World Scientific.

Appendix A. Detailed Exploration Metrics

This section details the measures summarized in Table 4. For the measure definitions below, let event log $L \in \mathcal{L}$, model $g \in \mathcal{N}$, $\omega_0 \in \mathbb{N}$. To obtain the play-out log $M \in \mathcal{L}^+$, the model g is played out to k traces, then occurrences are scaled to match the original log: $M = \frac{|L|}{k} \cdot spg(g, k, \omega_0)$.

The first measure is a simplification of the stochastic Earth Movers' distance [25].

EMT Earth Movers with play-out trace weighting.

$$EMT(M, L) = 1 - \frac{1}{|L|} \sum_{\sigma \in L} \max(L[\sigma] - M[\sigma], 0)$$

Two measures address how much of the probability mass of the log is in shared traces.

TMO Trace Probability mass overlap.

$$TMO(M,L) = \sum_{\sigma \in L \sqcap M} \frac{(L \sqcap M)[\sigma]}{|L|}$$

TOR Trace overlap ratio.

$$TOR(M,L) = \frac{|L \sqcap M|}{|L|}$$

Analysis of which subtraces occur in both log and model (represented by the play-out log) approximate fitness.

ARG The Gower's similarity [20] between activity count ratio vectors. This measure is designed to be deliberately sensitive to variation between poor quality models, when other measures may be zero. Given log L, take $ST_n(L)$ to be the subtraces of length n, $\sigma_s \# L$ the subtrace frequency of σ_s, with each occurrence in a trace counted, and $||L||_n$ to be the total subtraces of length n. ARG is a special case: ARG=TRG1.

TRGn Subtrace ratios, activity ratios generalized to sub-traces of length n. **TRG2**, **TRG3** and **TRG4** are all measured.

$$\begin{aligned} TRGn(M,L) = & \sum_{\sigma \in ST_n(L \sqcup M)} 1 - y_\sigma \\ \text{where } y_\sigma = & \frac{1}{\max(\sigma \# L, \sigma \# M)} \left| \frac{\sigma \# L}{||L||_n} - \frac{\sigma \# M}{||M||_n} \right| \end{aligned}$$

Two simplified variants of evaluation measure entropy [29], based on play-out logs, are used to define fitness and precision measures. The first uses bag intersection.

HIFT Play-out entropy intersection fitness.

$$HIFT(M,L) = \min(1, \frac{H(L \sqcap M)}{H(L)})$$

HIPT Play-out entropy intersection precision.

$$HIPT(M,L) = \min(1, \frac{H(L \sqcap M)}{H(M)})$$

The second entropy variant uses SDFA projection [29] function $\mathcal{P} \colon \mathcal{L}^+ \times \mathcal{L}^+ \to \mathcal{L}^+$, where traces are used as SDFA tokens.

$$\begin{aligned} \mathcal{P}(L_1, L_2) = & L_P \sqcup [\langle\rangle^{|L_1| - |L_P|}] \\ \text{where } & L_P = [\sigma^i \in L_1 \mid \exists_{j>0}\, \sigma^j \in L_2] \end{aligned}$$

HJFT Play-out entropy projection fitness.

$$HJFT(M,L) = \frac{H(\mathcal{P}(L,M))}{H(L)}$$

HJPT Play-out entropy projection precision.

$$HJPT(M,L) = \frac{H(\mathcal{P}(M,L))}{H(M)}$$

XPU Existential precision adapts Alpha precision [17] by calculating the probability mass of model traces represented at least once in the log.

$$XPU(M, L) = \frac{1}{|M|} \sum_{\sigma in L} M[\sigma]$$

Three simplicity measures are scaled by log size to impose a valid upper bound of 1.

SSENC Structural simplicity by entity count [34].

$$SSENC(g, L) = \max(1 - \frac{|P| + |T|}{|L|}, 0)$$

SSEDC Structural simplicity by edge count [34].

$$SSEDC(g, L) = \max(1 - \frac{|F|}{|L|}, 0)$$

SSS Structural simplicity by all structural components in SLPNs. This accounts for stochastic features not found in existing structural simplicity measures.

$$SSS(g, L) = \max(1 - \frac{1}{|L|}(|P| + |T| + |F| + |\bigcup_{t \in T} W(t)|), 0)$$

The following generalisation measures are at a trace level, and are taken from example measures in [2].

TGF1 Generalisation by trace floor, gen_{L2M_q} [2]. We also use **TGF5** and **TGF10** as measures for trace floors of 5 and 10 respectively.

$$\mathit{TGF1}(M, L) = \frac{|[\sigma \in l | \sigma \in M \wedge L[\sigma] \geq q]|}{|L|} \text{ with } q \geq 1$$

TGDU Generalisation by trace uniqueness difference, $gen_{L2M_{HB}}$ [2].

$$TGDU(M, L) = \frac{|[\sigma \in L | \sigma \in M]| - |L \sqcap M|}{|L|}$$

CSS Structural Complexity incl. stochastic includes both control-flow and stochastic features of a SLPN in a common metric. It is a denormalised inverse of SSS.

$$CSS(g, L) = |P| + |T| + |F| + |\bigcup_{t \in T} W(t)|$$

MEC Model entity count is a count of places and transitions.

$$MEC(g, L) = |P| + |T|$$

MEC Model edge count is a count of connecting arcs.

$$MEC(g, L) = |F|$$