On the Suitability of Generalized Behavioral Profiles for Process Model Comparison

Abel Armas-Cervantes[1], Marlon Dumas[1],
Luciano García-Bañuelos[1], and Artem Polyvyanyy[2]

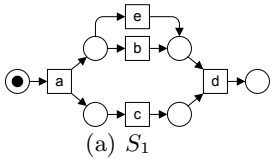
[1] Institute of Computer Science, University of Tartu, Estonia
{abel.armas,marlon.dumas,luciano.garcia}@ut.ee
[2] Queensland University of Technology, Brisbane, Australia
artem.polyvyanyy@qut.edu.au

Abstract. Given two process models, the problem of behavioral comparison is that of determining if these models are behaviorally equivalent (e.g., by trace equivalence) and, if not, identifying how can the differences be presented in a compact manner? Behavioral profiles have been proposed as a convenient abstraction for this problem. A behavioral profile is a matrix, where each cell encodes a behavioral relation between a pair of tasks (e.g., causality or conflict). Thus, the problem of behavioral comparison can be reduced to matrix comparison. It has been observed that while behavioral profiles can be efficiently computed, they are not accurate insofar as behaviorally different process models may map to the same behavioral profile. This paper investigates the question of how accurate existing behavioral profiles are. The paper shows that behavioral profiles are fully behavior preserving for the class of acyclic unlabeled nets with respect to configuration equivalence. However, for the general class of acyclic nets, existing behavioral profiles are exponentially inaccurate, meaning that two acyclic nets with the same behavioral profile may differ in an exponential number of configurations.

1 Introduction

Pairwise process model comparison is a basic primitive in the context of management of process model collections. Such comparison can be made from a lexical, syntactical and/or behavioral perspective. This paper deals with the latter. In this context, behavioral profiles [1] have been proposed as an abstract representation of process models for the purpose of comparison. A behavioral profile of a process model can be seen as a complete graph over the set of tasks of the model, where edges are annotated by types of behavioral relations. Alternatively, a behavioral profile is a matrix where rows and columns represent tasks and each cell is labeled by a behavioral relation between a pair of tasks.

Thus, the problem of behavioral comparison of process models can be mapped to that of comparing two matrices. This provides a convenient basis for computing behavioral similarity between pairs of process models [2].



(a) S_1

	a	b	c	d	e
a	+	↦	↦	↦	↦
b	↤	+	‖	↦	+
c	↤	‖	+	↦	‖
d	↤	↤	↤	+	↤
e	↤	+	‖	↦	+

(b) $BP|_w$

Fig. 1. Net system and its behavioral profile

Fig. 1 depicts a process model represented as a Petri net system and alongside its behavioral profile.[3] In the matrix representation, the strict order relation (↦) denotes causal precedence between a pair of tasks in all the computations of the model. Exclusive order relation (+) denotes that a pair of tasks never occurs in the same computation. Finally, interleaving (‖) represents the absence of order in the execution of a pair of tasks.

In the context of behavioral comparison of process models, the adoption of a notion of behavioral equivalence is crucial since it establishes the ground rules for the comparison. Unfortunately, since the introduction of the concept of behavioral profiles [1], the authors pointed out that this representation does not correspond to any of the well-known notions of behavioral equivalence, i.e., two behaviorally different models (e.g., by trace equivalence), may have the same matrix representation. Interestingly, different families of binary relations have been proposed as extensions of [1]; causal behavioral profile [3] and the relations of the 4C spectrum [4] are cases in point. However, none of them has been shown to preserve any well-known notion of equivalence.

In this paper, we analyze three different behavioral profiles and study their suitability for the representation of the behavior of a process model. All the considered behavioral profiles use $O(|\Lambda|^2)$ space to capture behavior.[4] In the light of the above, the contributions we make in this paper are as follows:

(i) we give an execution semantics to the behavioral profiles proposed in [1],
(ii) we show that, for a restricted family of (unlabeled) Petri net systems, behavioral profiles can ensure configuration equivalence [5], and
(iii) we show that, even for the family of acyclic labeled Petri net systems, the three considered behavioral profiles cannot provide an accurate representation of behavior.

The rest of the paper is structured as follows. Section 2 introduces basic concepts of Petri nets, workflow nets, flow nets and flow event structures. Section 3 develops the contributions (i) and (ii) listed above, whereas the contribution (iii) is presented in Section 4. Finally, Section 5 concludes the paper.

2 Background

This section introduces the necessary concepts used in further sections. Firstly, we recall basic definitions of Petri nets and present two families of nets: workflow and flow nets. Then, we review basic definitions of flow event structures and introduce the adopted notion of behavioral equivalence, configuration equivalence.

[3] The behavioral profile in Fig. 1 is computed using the relations in [1].
[4] In the rest of the paper, we will say that the size of the behavioral profile of a process model is $O(|\Lambda|^2)$, where Λ is the set of tasks of the model.

2.1 Petri nets

Definition 1 ((labeled) Petri net, net system).
A Petri net*, or a* net*, is a tuple* (P,T,F)*, where* P *is a set of* places*,* T *is a set of* transitions*, such that* $P \cap T = \emptyset$ *and* $P \cup T \neq \emptyset$*, and* $F \subseteq (P \times T) \cup (T \times P)$ *is the* flow relation*. A labeled net is a tuple* (P,T,F,Λ,λ)*, where* (P,T,F) *is a net,* Λ *is a set of* labels*, and* $\lambda : T \to \Lambda \cup \{\tau\}$ *is a function that maps transitions to labels, where* τ *is a special label,* $\tau \notin \Lambda$*. If* $\lambda(t) = \tau$*, where* $t \in T$*, then* t *is said to be* silent*; otherwise* t *is* observable*. A* marking *of a net* (P,T,F) *is a function* $M : P \to \mathbb{N}_0$ *that maps places to natural numbers (viz., tokens). A* Petri net system*, or a* net system*, is a pair* (N,M)*, where* $N = (P,T,F)$ *is a Petri net and* M *is a* marking *of* N*.* ⌟

Places and transitions are conjointly referred to as *nodes*. We write $\bullet y = \{x \in P \cup T \mid (x,y) \in F\}$ and $y\bullet = \{z \in P \cup T \mid (y,z) \in F\}$ to denote the *preset* and *postset* of the node $y \in P \cup T$, respectively. Similarly, for a set of nodes $X \subseteq P \cup T$, $\bullet X = \bigcup\{\bullet x \mid x \in X\}$ and $X\bullet = \bigcup\{x\bullet \mid x \in X\}$.

One approach to define the execution semantics of a net system is in terms of markings. A marking M of a net $N = (P,T,F)$ *enables* a transition $t \in T$, denoted as $M[t\rangle$, iff $\forall p \in \bullet t : M(p) > 0$. Moreover, an occurrence of t, such that $M[t\rangle$, leads to a fresh marking M' of N, denoted as $M \xrightarrow{t} M'$, where $M'(p) = M(p) - 1$ if $p \in \bullet t \setminus t\bullet$, $M'(p) = M(p) + 1$ if $p \in t\bullet \setminus \bullet t$, and $M'(p) = M(p)$ otherwise.

Definition 2 (firing sequence, execution).
Let $S = (N, M_0)$*,* $N = (P,T,F)$*, be a Petri net system.*

- *A sequence of transitions* $\sigma = t_1 \dots t_n$ *in* N*, where* $n \in \mathbb{N}_0$*, is a* firing sequence *in* S *iff* σ *is empty or it holds that* $M_0 \xrightarrow{t_1} M_1 \xrightarrow{t_2} M_2 \dots \xrightarrow{t_n} M_n$*. In the latter case, we say that* σ *leads from* M_0 *to* M_n*.*
- *A marking* M *is* reachable *in* S *iff* $M = M_0$ *or there exist a firing sequence* σ *that leads from* M_0 *to* M*. The notation* $M' \in [N,M\rangle$ *represents that* M' *is reachable in* (N,M)*.*
- *A marking* M *of* N *is* terminal *iff there exist no transition enabled at* M*.*
- *A firing sequence* σ *that leads from* M_0 *to* M*, where* M *is terminal, is called an* execution*. By* $\Theta(S)$*, we denote the set of all executions of* S*.* ⌟

A marking M of a net $N = (P,T,F)$ is *n-safe* iff for every place $p \in P$ it holds that $M(p) \leq n$, $n \in \mathbb{N}_0$. A net system S is said to be *n-safe* if all its reachable markings are *n-safe*. We restrict the subsequent discussions to 1-*safe* net systems. Note that one can identify a 1-*safe* marking M of a net (P,T,F) as the set of places $\{p \in P \mid M(p) = 1\}$.

Workflow nets [6] are a class of nets with a dedicated source and sink place, such that every transition is on a path from the source to the sink place.

Definition 3 (WF-net, WF-system). *A Petri net* $N = (P,T,F)$ *is a* workflow net*, or a* WF-net*, iff* N *has a dedicated* source *place* $i \in P$*, with* $\bullet i = \emptyset$*,* N *has a dedicated* sink *place* $o \in P$*, with* $o\bullet = \emptyset$*, and the* short-circuit *net*

$N^* = (P, T \cup \{t^*\}, F \cup \{(o, t^*), (t^*, i)\})$ *of* N *is strongly connected,* $t^* \notin T$*. A* WF-net system *is a net system* (N, M)*, where* N *is a WF-net with the source place* i *and* $M = \{i\}$*.* ⌟

Soundness [7] is the commonly adopted criterion of correctness for WF-net systems. For example, a sound WF-net system guarantees that every execution ends with one token in the sink place and no tokens elsewhere.

Definition 4 (liveness, boundedness, soundness).
Let $S = (N, M)$*,* $N = (P, T, F)$*, be a Petri net system.*

- S *is* live *iff for every reachable marking* $M' \in [N, M\rangle$ *and for every transition* $t \in T$ *there exist a marking* $M'' \in [N, M'\rangle$ *such that* $M''[t\rangle$*.*
- S *is* bounded *iff there exist a number* $n \in \mathbb{N}_0$ *such that* S *is* n*-safe.*

A WF-net system (N, M) *is* sound *iff the net system* (N^*, M)*, where* N^* *is the short-circuit net of* N*, is live and bounded.* ⌟

Flow nets form another family of Petri nets [8]. This type of nets is semantically acyclic, meaning that in any firing sequence, a place cannot be marked more than once. Thus, all the transitions in a firing sequence are distinct. In the context of flow nets, the notion of causal dependency between transitions is defined w.r.t. places. A transition t_j causally depends on a transition t_i iff $\exists\, p \in t_i \bullet \cap \bullet t_j$, such that whenever both transitions occur in a firing sequence, then t_i is the only transition that puts a token in p; p is said to be a *strong postcondition* of t_i. Finally, a flow net is defined as follows.

Definition 5 (flow net, flow net system).
A net system $S = (N, M)$*,* $N = (P, T, F)$*, is a* flow net system *and* N *is a* flow net *iff for every firing sequence* $\sigma = t_1\, t_2\, \ldots\, t_n$ *in* S *and for every* $i, j \in \mathbb{N}$*, s.t.* $1 \le i < j \le n$*, it holds that:*

- *a place* $p \in P$ *is in a preset of at most one transition of* σ*,i.e.,* $\bullet t_i \cap \bullet t_j = \varnothing$*,and*
- *if* $t_i \bullet \cap \bullet t_j \neq \varnothing$ *then* $\exists\, p \in t_i \bullet \cap \bullet t_j$*, s.t.,* p *is a strong postcondition of* t_i*.* ⌟

An alternative way to define the execution semantics of a net system is using the notion of a configuration. The main difference between firing sequences and configurations is that the former capture the interleaving semantics, whereas the latter describe the partial order semantics (aka true concurrency). In the case of flow nets, firing sequences and configurations are in the close relation, which is due to the next definition.

Definition 6 (flow net configuration).
A configuration of a flow net system $S = (N, M)$*,* $N = (P, T, F)$*, is a subset* $C \subseteq T$ *of transitions of* N *such that there exist a firing sequence* σ *in* S *that consists of the transitions in* C*, i.e.,*

$$\sigma = t_1\, t_2\, \ldots\, t_n \quad \textit{and} \quad C = \{t_1, t_2, \ldots, t_n\}$$

The set of all configurations of S *is denoted by* $Conf(S)$*.* ⌟

Set inclusion ($\subseteq$) defines an order over configurations. We say that a configuration C *evolves into* a configuration C' if $C \subseteq C'$.

The discussions throughout the paper consider Petri nets in the intersection of two families: sound WF-nets and flow nets (shorthanded as WF-flow nets, see Fig. 2). More specifically, the focus of this work is on WF-nets that are acyclic, sound and have the properties of flow nets (Def. 5).

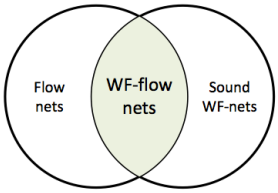


Fig. 2. WF-flow nets

2.2 Flow event structures

Flow Event Structures [8] (FES) is a well-known model of concurrency. It describes the behavior of a net system by means of events (occurrences of actions) and two binary behavioral relations, namely *flow* and *conflict*.

Definition 7 ((labeled) flow event structure).
A flow event structure *(FES) is a tuple* $\mathbb{F} = (E, \#, \prec)$*, where*
- E *is a set of events,*
- $\# \subseteq E \times E$ *is the* conflict *relation, which is symmetric, and*
- $\prec \subseteq E \times E$ *is the* flow *relation, which is irreflexive.*

A labeled *FES is a tuple* $\mathbb{F} = (E, \#, \prec, \Lambda, \lambda)$*, where* $(E, \#, \prec)$ *is a FES,* Λ *is a set of labels, and* $\lambda : E \to \Lambda \cup \{\tau\}$ *is a function that maps events to labels. Again, if* $\lambda(e) = \tau$*, where* $e \in E$*, then* e *is said to be* silent*; otherwise* e *is* observable. ⌟

Intuitively, the flow relation represents possible immediate precedence, meaning that if two events e and e' are in the flow relation, i.e., $e \prec e'$, then event e can potentially occur before e'. The conflict relation represents mutual exclusion. Two events e and e' in the conflict relation, i.e., $e \# e'$, cannot occur together in the same execution. The conflict relation is reflexive and then self-conflicting events are allowed. Even though self-conflicting events never occur in any configuration, in general, they cannot be removed from a FES without affecting the set of configurations [8].

Similar to flow nets, the behavior of FESs can be given in terms of configurations, and it is defined as follows.

Definition 8 (FES configuration).
A configuration of a FES $\mathbb{F} = (E, \#, \prec)$ *is a subset* $C \subseteq E$ *of events of* $\mathbb{F}$ *s.t.:*
- C *is conflict free, i.e.,* $\forall\, e, e' \in C : \neg(e \# e')$,
- C *has no flow cycles, i.e., the transitive closure of* $\prec$ *between the events in* C $(\prec^*_C)$ *is a partial order, and*
- *for all events* $e' \in C$ *and* $e \notin C$ *such that* $e \prec e'$ *it holds that there exist an event* $e'' \in C$ *such that* $e \# e''$ *and* $e'' \prec e'$.

The set of all configurations of $\mathbb{F}$ *is denoted by* $Conf(\mathbb{F})$. ⌟

An alternative formulation of configurations is done using proving sequences.

Definition 9 (proving sequence). *A* proving sequence *in a FES* $\mathbb{F} = (E, \#, \prec)$ *is a (finite or infinite) sequence* $\sigma = e_1 \dots e_n \dots$ *of distinct non-conflicting events, s.t.* $\forall i \forall e \in E : e \prec e_i \Rightarrow (\exists\, j < i : e \# e_j \wedge e_j \prec e_i)$. ⌟

A subset of events $C \subseteq E$ is a configuration of a FES $\mathbb{F} = (E, \#, \prec)$, s.t. $C = \{e_1, \dots, e_n\}$, if and only C is conflict free and for every event $e_k \in C$, $k \leq n$ it holds that $e_1 \dots e_k$ is a proving sequence in $\mathbb{F}$, cf. [8].

In [8], Boudol shows that FES corresponds to the family of flow nets, i.e., it is always possible to compute a FES for a given flow net system, where configurations of FES are derived from firing sequences in the system. Interestingly, for a sound WF-flow net, it is possible to establish a bijection between its transitions and events in the corresponding FES representation.[5]

The next definition suggests how to construct a FES from a flow net system. Different from [8], we do not consider self-conflicting events, thus there is a bijection between the transitions in the net and the events in the FES. Hence, we use T to represent both, events and transitions, indistinctively.

Definition 10 (FES of a flow net).
Let $S = (N, M)$, $N = (P, T, F)$, *be a flow net system. The* FES of S *is the tuple* $\mathbb{F} = (T, \#, \prec)$, *where for every two transitions* $t \in T$ *and* $t' \in T$ *it holds that:*

- $t \# t' \Leftrightarrow_{def} \forall\, C \in Conf(S) : \{t, t'\} \nsubseteq C$, *and*
- $t \prec t' \Leftrightarrow_{def} \neg(t \# t') \wedge t\bullet \cap \bullet t' \neq \emptyset$. ⌟

2.3 Configuration equivalence

The equivalence notion adopted in this work is configuration equivalence, a well-known notion of equivalence in the spectrum of true concurrency [5].[6] Note that this notion is stronger than trace equivalence. As showed before, the behavior of a flow net system or a FES can be described, in partial order semantics, by means of configurations. Intuitively, a pair of structures (in our context, either flow event structures or flow net systems) are configuration equivalent if 1. there is a bijection between elements (events or transitions), and 2. the structures represent, essentially, the same set of configurations. Below, we provide a definition of the configuration equivalence for flow net systems, but it can be straightforwardly adapted to FESs.

Definition 11 (configuration equivalence $\approx_{conf}$). *Let* $S = (N, M_0)$ *and* $S' = (N', M_0')$ *be two flow net systems, where* $N = (P, T, F)$ *and* $N' = (P', T', F')$, *and let* $\gamma : T \to T'$ *be a bijective function between the transitions of the nets. Let* $S \eqsim_{conf} S'$ *denote that for any configuration* C *in* S *there is a corresponding*

[5] Additional self-conflicting events can be required in a FES when, in the context of WF-nets, a net system does not meet the property of liveness. We refer the reader to [8] for more details about the introduction of self-conflicting events.

[6] The authors of [5] use *pomset-trace equivalence*. A pomset is basically a set of configurations augmented with the order induced by set inclusion. Since we are not interested in such order, we keep the equivalence at the level of configurations.

configuration C' in S' consisting of the images of C. I.e., $\forall C \in Conf(S) \exists C' \in Conf(S') : C' = \{\gamma(t) \mid t \in C\}$.

The pair of net systems S, S' are configuration equivalent*, denoted $S \approx_{conf} S'$, iff $S \sim_{conf} S'$ and $S' \sim_{conf} S$.* ⌟

3 Behavioral profiles of acyclic unlabeled WF-flow nets

This section proposes an execution semantics for the behavioral profiles defined in [1], hereinafter referred to as *classic behavioral profiles* and denoted as $BP|_w$. We show that, for the case of acyclic unlabeled WF-flow nets, classic behavioral profiles are behavior preserving under configuration equivalence. The section is organized as follows, Subsection 3.1 presents a generalized notion of behavioral profiles. Subsection 3.2 presents a behavioral profile $BP|_{fes}$ derived from flow event structures. Then, in Subsection 3.3, a transformation from $BP|_w$ into $BP|_{fes}$ is proposed and it is shown that $BP|_w$ is behaviorally preserving for acyclic unlabeled WF-flow nets. Finally, Subsection 3.4 discusses about the suitability of the $BP|_w$ for the behavioral comparison of processes.

3.1 Generalized behavioral profiles

Behavioral profiles can be seen as a framework that is concretely defined according to a set of behavioral relations. Roughly speaking, a behavioral profile $BP|_R$ of a process model is a complete graph over the set of tasks' labels, which uses a set of relations R as edge labels. This general notion of behavioral profiles results useful for uniformly analyze the different formalisms considered in this paper.

A behavioral profiles is said *behavior preserving*, if a pair of net systems that are behavior-equivalent (under certain notion of equivalence) have isomorphic behavioral profiles (denoted as $\equiv_{iso}$); and vice-versa. The intuition above is captured in the following definition.

Definition 12 (behavior-preserving $BP|_R$). *Let $\mathcal{N}$ be a class of nets and $\approx$ be an equivalence relation on $\mathcal{N}$. A behavioral profile $BP|_R$ is behavior-preserving on $\mathcal{N}$, if for any $N, N' \in \mathcal{N}$ with net systems $S = (N, M_0), S' = (N', M_0')$ and behavioral profiles $BP|_R(S)$ and $BP|_R(S')$, respectively, the following holds:*

$$S \approx S' \Leftrightarrow BP|_R(S) \equiv_{iso} BP|_R(S').$$

⌟

3.2 Behavioral profiles and FES

The correspondence between FESs and flow nets investigated by Boudol et al. [8] shows that given a flow net, it is possible to construct a FES, such that the firing sequences in the net are configurations in the FES; and vice-versa.

The following proposition restates the results proved in [8] for flow nets. Although, we refer concretely to the set of all WF-flow nets and represent by $\mathcal{N}$.

Proposition 1 (Proposition 3.4 in [8]). *Let $S = (N, M_0)$ be a WF-flow net system, with a net $N = (P, T, F) \in \mathcal{N}$, and let $\mathbb{F}$ be its corresponding FES (see Definition 10), we have that $Conf(S) = Conf(\mathbb{F})$. More precisely, a sequence $t_1 \dots t_n$ is firable in S if and only if it is a proving sequence in $\mathbb{F}$.*

Thus, a result from Proposition 1 is that a pair of configuration equivalent WF-flow nets have, similarly, configuration equivalent FESs.

Corollary 1. *Let N, N' be nets in $\mathcal{N}$. Moreover, let $\mathbb{F}$ and $\mathbb{F}'$ be the FESs of S and S', respectively. Then, it holds that:*

$$S \approx_{conf} S' \Leftrightarrow \mathbb{F} \approx_{conf} \mathbb{F}'$$

A type of behavioral profiles for WF-flow nets can be defined using FESs when there is a bijection between the transitions in the net and the events in event structure. Thus, let us refer to the behavioral profile of a net system S given by the FES as $BP|_{fes}(S)$. In this case, the tasks in $BP|_{fes}(S)$ are the events in the FES, and the set of binary relations are flow and conflict. Finally, one can notice that it is possible to define a behavioral profile $BP|_{fes}$ that contains all the behavior of a net system S, such that any conclusion (w.r.t. behavior) derived from $BP|_{fes}$ holds in S.

The following proposition shows that every place in a WF-flow net is, in fact, a strong postcondition of a transition w.r.t. a firing sequence.

Proposition 2. *Let $S = (N, M_0)$ be a WF-flow net system, with a net $N = (P, T, F)$, and let $\sigma = t_1\ t_2 \dots t_n \in \Theta(S)$ be an execution of S. Then, a place $p \in t_j\bullet$ is a strong postcondition of a transition t_j, where $1 \leq j \leq n$.*

Proof. In this case, it is shown that the property holds for an execution σ, but then it also holds for any firing sequence, which elements are part of σ.

Note that if p is the source place then it cannot be a postcondition of any transition, since $\bullet p = \varnothing$. Conversely, consider the case when p is not the source place, thus $\bullet p \neq \varnothing$. Then, let us show that there is a unique t_j, where $1 \leq j \leq n$, such that $p \in t_j\bullet$. Suppose that there is another transition t_k, where $1 \leq k \leq n$, such that $k \neq j \ \wedge \ p \in t_k\bullet$. The place p cannot be the sink place, because if t_j and t_k fire, then p would have two tokens, but it contradicts the fact that the net is 1-safe. Thus, it means that p was marked in two different occasions and, since the net is 1-safe, it was necessary to consume one of the tokens before the other was set. Hence, p was the preset of two different transitions in σ. Nevertheless, it violates the condition 1 in Definition 5. Therefore, if $p \in t_j\bullet$ then p is a strong postcondition of t_j. □

The proposition above, however, implies that every place between a pair of transitions in a WF-flow net defines a flow relation. As a result, the behavioral profiles $BP|_{fes}$, w.r.t. Def. 10, are not behavior preserving (see Def. 12). For instance, a net system can have implicit places that would define unnecessary flow relations between the events in the FES. Nevertheless, we believe that by providing a more elaborated definition for constructing a FES of a net system, one can find a behavior preserving $BP|_{fes}$. Although, the last is left for future work, since Definition 10 is enough for the scope of this paper.

3.3 An execution semantics for the classic behavioral profiles

In this subsection, we focus on the classic behavioral profiles ($BP|_w$). This type of behavioral profiles uses three behavior relations, namely *strict order* ($\mapsto$), *exclusive order* ($+$) and *interleaving* ($|||$). The following definition formalizes the behavior relations comprising any $BP|_w$, along with its computation.

Definition 13 ($BP|_w$, see [1]). *Let $S = (N, M)$ be a net system, with $N = (P, T, F)$. A pair of transitions $t, t' \in T$ is in one of the following relations:*

- Strict order relation*, denoted by $t \mapsto t'$, if for every firing sequence $\sigma \in \Theta(S)$, with $\sigma = t_1\ t_2 \dots t_n$ such that $t_i = t$ and $t_j = t'$, it holds $1 \leq i < j \leq n$.*
- Exclusive order relation*, denoted by $t + t'$, if for every firing sequence $\sigma \in \Theta(S) : \sigma = t_1\ t_2 \dots t_n$ there are no t_i, t_j, where $1 \leq i, j \leq n$, s.t. $i \neq j$, $t_i = t$ and $t_j = t'$.*
- Interleaving relation*, denoted by $t \,|||\, t'$, if $\neg(t \mapsto t')$, $\neg(t' \mapsto t)$ and $\neg(t + t')$.*

For technical reasons, we also define the direct strict order*. Transitions t and t' are in direct strict order, denoted by $t \twoheadrightarrow t'$, iff*

$$t_i \mapsto t_j \ \wedge\ t_i \bullet \cap \bullet t_j \neq \varnothing$$

The set $BP|_w(S) = \{\mapsto, +, |||\}$ is the classic behavioral profile *of S.* ⌟

The following definition formalizes the transformation from $BP|_w$ to $BP|_{fes}$.

Definition 14 ($BP|_w^{fes}$). *Let $BP|_w(S) = \{\mapsto, +, |||\}$ be a classic behavioral profile of a WF-flow net system $S = (N, M)$, with $N = (P, T, F)$. Let $+'$ be the exclusive order relation without the self-exclusive order relations, i.e., $+' = + \backslash \{(t, t) \mid t \in T\}$. Then $BP|_{fes}$ of $BP|_w(S)$ is defined as $BP|_w^{fes}(S) = \{\twoheadrightarrow, +'\}$.* ⌟

The definition presented above gives $BP|_w$ an execution semantics on FES through $BP|_w^{fes}$. Thus, given a $BP|_w$ and the $\twoheadrightarrow$ relation, one can derive the configurations of the corresponding net by reusing the notion of configuration of FES over $BP|_w^{fes}$. The following proposition shows that the $BP|_w^{fes}(S)$ computed from $BP|_w(S)$ is isomorphic to $BP|_{fes}(S)$, i.e., $BP|_w^{fes}(S)$ captures the same relations as $BP|_{fes}(S)$ and so $BP|_w^{fes}(S)$ captures all the behavior of S. Note that for the reminder of this section, we focus on unlabeled WF-flow nets, which will be denoted as $\mathcal{N}_{\overline{\lambda}}$.

Proposition 3. *Let $S = (N, M)$ be a net system, where $N = (P, T, F) \in \mathcal{N}_{\overline{\lambda}}$ is an unlabeled WF-flow net, and $BP|_w(S) = \{\mapsto, +, |||\}$ be the classic behavioral profile of S. Thus, $BP|_w^{fes}(S) = \{\twoheadrightarrow, +\}$ is isomorphic to $BP|_{fes}(S) = \{\prec, \#\}$, in specific, for any two transitions $x, y \in T$ 1. $x + y \ \Leftrightarrow\ x \# y$, and 2. $x \twoheadrightarrow y \ \Leftrightarrow\ x \prec y$.*

Proof. Given that the nets are unlabeled, then for any transition in T there is a task in $BP|_{fes}$ and in $BP|_w$ (and so in $BP|_w^{fes}$). Therefore, let us consider the same set of elements T throughout the different structures.

1. $x + y \ \Leftrightarrow\ x \# y$. It is easy to check that the definition of exclusive ordering relation ($+$) in $BP|_w$ and conflict ($\#$) in $BP|_{fes}$ is the same. So, the conflict in the FES coincides with the exclusive order relation in $BP|_w$.

2. $x \twoheadrightarrow y \;\Leftrightarrow\; x \prec y$. ($\Rightarrow$) Consider a pair of transitions $x, y \in T : x \twoheadrightarrow y$. By Definition 13, $\exists \sigma = t_1\ t_2 \ldots, t_n : x = t_i, y = t_j \;\wedge\; i < j$. Additionally, since the causal relation is direct, then there is at least a place $p \in x\bullet \cap \bullet y$. Thus, by Definition 10, $x \prec y$ in $BP|_{fes}(S)$.
($\Leftarrow$) Suppose $x \prec y$ in $BP|_{fes}(S)$, and $\neg(x \twoheadrightarrow y)$ in $BP|_w^{fes}(S)$. By Definition 10, since $x \prec y$ then $x\bullet \cap \bullet y \neq \varnothing$ and $\neg(x \# y)$. Let $p \in x\bullet \cap \bullet y$ be a strong postcondition of x. The first thing to notice is that $\neg(x + y)$, since $\neg(x \# y)$ – see previous case. Thus, there exist a firing sequence $\sigma = t_1\ t_2 \ldots, t_n : x = t_i, y = t_j$, where $1 \leq i < j \leq n$. Hence, the only case where $\neg(x \mapsto y)$ can hold is if there exist an execution $\sigma' \in \Theta(S)$, s.t., $\sigma' = t'_1, t'_2, \ldots, t'_n$ and $\exists t'_i = x, t'_j = y : 1 \leq j < i \leq n$, in which case it would hold $x \,|||\, y$. Given that y occurred in σ' and $p \in \bullet y$, then p had a token prior the firing of y, and as p is a strong postcondition of x, then x occurred before y (by Proposition 2), but then σ' has two occurrences of x because $t'_i = x, t'_j = y : 1 \leq j < i \leq n$. The last contradicts the fact that N is a flow net because, by Definition 5(1), the places in the present of x cannot be marked more than once by the transitions in σ'. Thus, if $x \prec y$ then $x \twoheadrightarrow y$. □

In what follows, $BP|_w$ is shown to be behavior preserving for the $\mathcal{N}_{\overline{\lambda}}$.

Proposition 4. *Consider a pair of nets $N = (P, T, F)$ and $N' = (P', T', F')$ in $\mathcal{N}_{\overline{\lambda}}$, such that there is a bijection between the transitions $\gamma : T \to T'$, and let $S = (N, M_0)$ and $S' = (N', M'_0)$ be net systems with initial markings M_0 and M'_0. Thus, the following holds:*

$$BP|_w(S) \equiv_{iso} BP|_w(S') \Leftrightarrow S \approx_{conf} S'.$$

Proof. ($\Rightarrow$) Firstly, let us show that if $BP|_w(S) \equiv_{iso} BP|_w(S')$ then $S \approx_{conf} S'$.

Suppose that $BP|_w(S) \equiv_{iso} BP|_w(S')$, but $\neg(S \approx_{conf} S')$. By Corollary 1, we have $\neg(BP|_w^{fes}(S) \approx_{conf} BP|_w^{fes}(S'))$ since $\neg(S \approx_{conf} S')$.

Assume a configuration $C \subseteq T$ in $BP|_w^{fes}(S)$ and its mapping $C' = \{\gamma(t') \mid t' \in C\}$ in S', such that C' is not a configuration in $BP|_w^{fes}(S')$. By Definition 8, the configuration C (i) is conflict free, (ii) for all $e' \in C$ and $e \notin C$, s.t., $e \prec e'$ there exist an $e'' \in C$ s.t. $e \# e'' \prec e'$, and (iii) has no flow cycles. Therefore, we must consider the following cases:

(i) Conflict freeness. Since C is a configuration in $BP|_w^{fes}(S)$, then for any $e, e' \in C$ it holds $\neg(e \# e')$ and, in consequence, $\neg(e + e')$ in $BP|_w(S)$ by Proposition 3(1). Then, by the assumption on the isomorphism of the $BP|_w$'s, $\exists e_1, e'_1 \in C' : \gamma(e) = e_1 \;\wedge\; \gamma(e') = e'_1$, such that $\neg(e_1 + e'_1)$ and thus $\neg(e_1 \# e'_1)$. So, C' is also conflict free iff C is conflict free, and every pair of $e_1, e'_1 \in C'$ is either in $|||$ or $\mapsto$ ordering relations.

(ii) For any $e''_1 \in C'$ and $e_1 \notin C'$, s.t., $e_1 \prec e''_1$, there exist an $e'_1 \in C' : e_1 \# e'_1 \prec e''_1$. Suppose that there is an event $e_1 \notin C'$, such that $\exists e''_1 \in C' : e_1 \prec e''_1$ and $\forall e'_1 \in C' : \neg(e_1 \# e'_1)$. Given that $e_1 \prec e''_1$, then $e_1 \mapsto e''_1$ (more specifically, $e_1 \twoheadrightarrow e''_1$), and since $\neg(e_1 \# e'_1)$ then $\neg(e_1 + e'_1)$ for any $e'_1 \in C'$, by Proposition 3. Hence, by the isomorphism of BP_w's, then $\exists e \notin C, e'' \in C : \gamma(e) = e_1 \;\wedge\; \gamma(e'') = e''_1 \;\wedge\; e \mapsto e''$ and for any $e' \in C$ it holds $\neg(e + e')$. However, the last

contradicts the fact that C is a configuration in $BP|_w^{fes}(S)$, because e would necessarily be in C. Henceforth, condition 2 also holds for C'.

(iii) Free of flow cycles. The only case remaining, so that C' is not a configuration in $BP|_w^{fes}(S')$, is when C' contains cycles, i.e., $\prec_{C'}^*$ is not a partial order. This case simply cannot happen because WF-flow nets are acyclic and any firing sequence contains at most one occurrence of each activity.

Therefore, if C is a configuration in $BP|_w^{fes}(S)$, then C' must also be a configuration in $BP|_w^{fes}(S')$.

($\Leftarrow$) The opposite case, $S \approx_{conf} S' \Rightarrow BP|_w(S) \equiv_{iso} BP|_w(S')$, follows directly from the construction of the $BP|_w$, see Definition 13. □

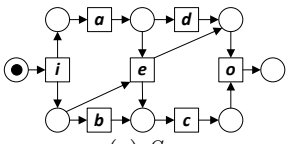
Armed with the above, one can easily see that $BP|_w$ is behavior-preserving for the class of $\mathcal{N}_{\overline{\lambda}}$. This fact is captured in the following Corollary.

Corollary 2. *The behavioral profiles $BP|_w$ is behavior-preserving for the class $\mathcal{N}_{\overline{\lambda}}$, w.r.t. configurations equivalence $\approx_{conf}$.*

The above results also holds for the different extensions of the classic behavioral profiles that have strict and exclusive order relations, e.g., causal behavioral profile [3] and behavioral profiles based on the relations of the 4C spectrum [4].

3.4 On the interpretation of behavioral differences using $BP|_w$

The process model comparison aims not just at determining if a pair of process models are (behaviorally) equivalent, but also at explaining the existing differences between the process models. This section analyzes the suitability of the classic behavioral profiles when used to interpret encountered differences between a pair of behavioral profiles.



(a) S_2

	a	b	c	d	e	i	o
a		‖	‖	↠	↠	↞	↦
b	‖		↠	‖	+	↞	↦
c	‖	↞		‖	↞	↤	↠
d	↞	‖	‖		+	↤	↠
e	↞	+	↠	+		↞	↠
i	↠	↠	↦	↦	↠		↦
o	↤	↤	↞	↞	↞	↤	

(b) $BP|_w$

Fig. 3. Net system S_2 and its $BP|_w$

Figure 3 presents an example showing that $BP|_w$ does not necessarily provide a detailed representation of the behavior of a WF-flow net. In this example, there is a WF-flow net system[7] and its behavioral profile $BP|_w$ aside. Let us draw you attention to transitions a and c, for which $BP|_w$ asserts an interleaving relation. However, in all the configurations where e occurs it is always the case that a precedes c. It is only in the set of configurations where e does not occur where a and c occur in any order. The fact is that these subtle kind of differences requires a diagnostic with

[7] This net corresponds to the FES N presented in [8].

contextual information in addition to the local information provided by a binary relation. It should be clear that it is possible to derive such sets of configurations from the $BP|_w^{fes}$.

A solution to disambiguate the situation presented in Fig. 3 is to reason not in terms of actions, but in terms of instances of actions (events), where it is possible to set a single relation between a pair of transitions (causality, conflict or concurrency). In this regard, alternative representations of the behavior of a WF-flow net, e.g., by means of a branching process [9], can result useful. For instance, the branching process of the net system in S_2 (Fig. 3) is displayed in Fig. 4. Although, the price to pay is that a branching process can contain several instances of a single activity, and the $O(|\Lambda|^2)$ size of the representation is no longer guaranteed.

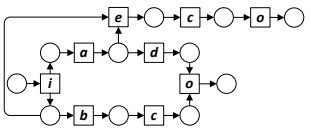


Fig. 4. Branching process of net system S_2 (Fig. 3(a))

Another approach to tackle the ambiguity of the $BP|_w$ is to use a larger set of behavioral relations. For instance, the 4C spectrum [4] defines a repertoire of eighteen basic behavioral relations that capture such behavioral phenomena as co-occurrence, conflict, causality, and concurrency. One can employ the relations of the 4C spectrum to construct an abstract representation of behavior of a process model, i.e., its behavioral profile. Note that due to the nature of the 4C spectrum, a pair of tasks can be associated with several behavioral relations. Nevertheless, behavioral profiles that are based on the relations of the 4C spectrum are guaranteed to be captured using $O(|\Lambda|^2)$ space. Even though this approach solves the problem of the ambiguity for the family of unlabeled net systems, it falls short when trying to generalize the solution to the case of net systems with silent transitions (as discussed in the next section).

4 Behavioral profiles and acyclic labeled WF-flow nets

This section extends the analysis of the behavioral profiles to labeled WF-flow nets. It is shown that for this family of nets neither the notion of classic behavioral profile nor its extensions, including that based on the relations of the 4C spectrum, provide behavior-preserving representations of process models.

Proponents of classic behavioral profiles search for providing a representation that only considers the observable behavior. When it comes to representing labeled net systems, the common approach is to omit the columns and rows in the matrix that would be associated with silent transitions. This decision, however, comes with a loss of accuracy of the representation. For example, consider the net system S_3 in Fig. 5. Its classic behavioral profile is isomorphic with the one of S_1, cf. Fig. 1. However, S_3 differs from S_1 in that it has two additional configurations: $\{a, b, d\}$ and $\{a, e, d\}$.

In order to preserve behavior, as for the case of unlabeled WF-flow nets, one possibility is to explicitly represent silent transitions in the matrix, as illustrated in Fig. 6. It is easy to see that, using this approach, the behavior of S_1 and S_3

would be represented with two non-isomorphic matrices. However, this approach does not provide a complete solution since multiple net systems may exist with different numbers of silent transitions exhibiting the same observable behavior.

The use of a larger number of behavior relations can be seen as a way to tackle the above problem. For instance, both causal behavioral profiles and behavioral profiles that are based on the relations of the 4C spectrum provide non-isomorphic representations for S_1 and S_3. However, none of them provides representations that distinguish the WF-flow net systems S_4 and S_5 in Fig. 7 w.r.t. configuration equivalence. Interestingly, there is only one configuration that distinguishes S_4 from S_5, namely $\{i, o\}$. It turns out that the set of configurations that is common to both systems, namely $\{\{i, a, o\}, \{i, b, o\}, \{i, a, b, o\}\}$, gives rise to the same representation based on the relations of the 4C spectrum.

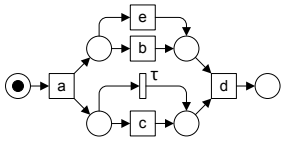


Fig. 5. Net system S_3

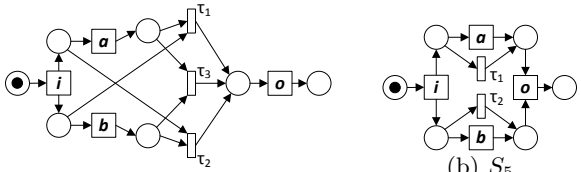
Figure 8 shows two constructions that generalize the net systems in Fig. 7 with a variable amount of transitions n. It turns out that, for any fixed value of $n \in \mathbb{N}$, the system S_6 would comprise the set of configurations $\{\{i, a_1, a_2, \ldots, a_n, o\}\} \cup \{\{i, a_m, o\} \mid m \in [1\,..\,n]\}$, however, it would have the same representation as the system S_7 over the relations of the 4C spectrum. Note that system S_6 encodes $n{+}1$ configurations, whereas system S_7 describes 2^n configurations. Therefore, there exist two net systems for which there is an exponential number of configurations that are indistinguishable when using the representation based on the relations of the 4C spectrum; in specific, $2^n{-}n{-}1$ indistinguishable configurations for systems in Fig. 8. This fact is captured in the next proposition.

	a	b	c	d	τ	e
a		$\prec$	$\prec$		$\prec$	$\prec$
b	$\prec^{-1}$			$\prec$		+
c	$\prec^{-1}$			$\prec$	+	
d		$\prec^{-1}$	$\prec^{-1}$		$\prec^{-1}$	$\prec^{-1}$
τ	$\prec^{-1}$		+	$\prec$		
e	$\prec^{-1}$	+		$\prec$		

Fig. 6. $BP|_{fes}(S_3)$

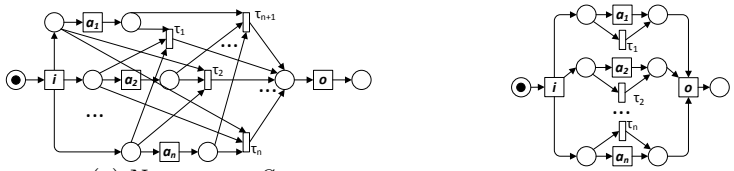
Proposition 5. *There exist two labeled WF-flow net systems that have the same 4C relations over labels and the numbers of distinct configurations that the net systems describe differ in the value which is in the order of $O(2^n)$, where n is the number of distinct labels assigned to transitions of the net systems.*

Observe that there also exist two net systems with no concurrent behavior on observable transitions that have the same 4C relations over labels but induce



(a) S_4 (b) S_5

Fig. 7. Two net systems that have the same 4C relations over labels

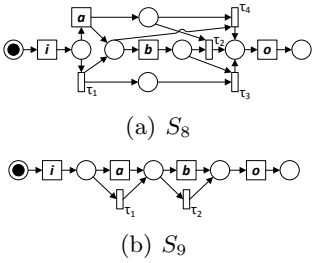


(a) Net system S_6 (b) Net system S_7

Fig. 8. Generalization of the net systems in Fig. 7

different configurations, cf. Fig. 9. Indeed, net systems S_8 and S_9 have the same 4C relations over labels, although net system S_9 describes configuration $\{i, o\}$ which is not captured by net system S_8.

The above observations confirm that existing behavioral profiles are lossy behavioral representations of labeled net systems. So, if one relies on existing behavioral profiles in the context of process model comparison, then one must tolerate inaccurate diagnosis. To address this problem, one must either look for new and more accurate behavioral profiles or, alternatively, explore behavior representations in terms of occurrences of actions; however, the size of such latter representations can be considerably larger than $O(|\Lambda|^2)$.



(a) S_8

(b) S_9

Fig. 9. Two net systems that have the same 4C relations over labels

5 Conclusion

This paper studies the idea of using behavioral profiles for the purpose of behavioral comparison of process models, i.e., deciding if two given models are behaviorally equivalent and, whenever required, providing a convenient representation of their differences. The use of behavioral profiles allows reducing the problem of behavioral comparison to that of matrix comparison, which provides a formal basis for tracing differences between process models that are grounded in behavior. Moreover, the feasibility of the overall idea is validated by showing that behavioral profiles can be used to decide configuration equivalence for a restricted class of acyclic and unlabeled net systems. However, this result ceases to hold (for any currently known notion of behavioral profile) once transitions of net systems are allowed to 'wear' labels. Future works are called to contribute to a better understanding of which behavioral profiles can be employed for the purpose of behavioral comparison of which families of process models under which notions of behavioral equivalence.

The results of this paper have implications in the context of process mining algorithms that rely on matrix-based representations of behavior. A case in point are alpha relations [10], which abstract an event log as an $O(|\Lambda|^2)$ matrix where each cell is annotated by a behavioral relation (direct causality, conflict

or concurrency). Our results hint at the fact that such matrices may miss to encode an exponential number of computations. Thus, designers of process mining algorithms should consider using more faithful intermediate representations.

Alternatively, rather than relying on behavioral relations on tasks, one can consider behavioral comparison that is founded on binary behavioral relations on events, i.e., task occurrences. When viewing the set of binary relations as a matrix, this means that the matrix may be considerably larger than $O(|\Lambda|^2)$, since a task may occur in an exponential number of computations. In a separate work, we have explored the use of event structures as alternative representations for process model comparison [11]. It turns out that the most basic form of event structures requires maintaining a large number of events representing different occurrences of the same task. To tackle this problem, we apply reduction rules to obtain a canonical matrix representation of behavior [12, 13]. However, achieving a quadratic or even a polynomial matrix-based representation of behavior appears to be elusive in the general case.

References

1. Weidlich, M., Mendling, J., Weske, M.: A foundational approach for managing process variability. In: CAiSE. Volume 6741 of LNCS. Springer (2011) 267–282
2. Kunze, M., Weidlich, M., Weske, M.: Behavioral similarity — a proper metric. In: BPM. Volume 6896 of LNCS., Springer (2011) 166–181
3. Weidlich, M., Polyvyanyy, A., Mendling, J., Weske, M.: Causal Behavioural Profiles — Efficient Computation, Applications, and Evaluation. Fundamenta Informaticae **113**(3-4) (2011) 399–435
4. Polyvyanyy, A., Weidlich, M., Conforti, R., Rosa, M.L., ter Hofstede, A.H.M.: The 4C spectrum of fundamental behavioral relations for concurrent systems. In: Petri Nets. Volume 8489 of LNCS. (2014) 210–232
5. van Glabbeek, R., Goltz, U.: Equivalence notions for concurrent systems and refinement of actions. In: Mathematical Foundations of Computer Science 1989. Volume 379 of LNCS. Springer (1989) 237–248
6. van der Aalst, W.M.P.: Verification of workflow nets. In: ICATPN. Springer (1997) 407–426
7. van der Aalst, W.: Workflow verification: Finding control-flow errors using petri-net-based techniques. In: BPM. Volume 1806 of LNCS. Springer (2000) 161–183
8. Boudol, G.: Flow event structures and flow nets. In: Semantics of Systems of Concurrent Processes. Volume 469 of LNCS., Springer (1990) 62–95
9. Engelfriet, J.: Branching processes of Petri nets. ACTA **28**(6) (1991) 575–591
10. van der Aalst, W.M.P., Weijters, T., Maruster, L.: Workflow mining: discovering process models from event logs. IEEE TKDE **16**(9) (2004) 1128–1142
11. Armas-Cervantes, A., García-Bañuelos, L., Dumas, M.: Event structures as a foundation for process model differencing, part 1: Acyclic processes. In: WSFM. Volume 7843 of LNCS. Springer (2013) 69–86
12. Armas-Cervantes, A., Baldan, P., García-Bañuelos, L.: Reduction of event structures under hp-bisimulation. Technical report `http://arxiv.org/abs/1403.7181`.
13. Armas-Cervantes, A., Baldan, P., García-Bañuelos, L., Dumas, M.: Behavioral comparison of process models based on canonically reduced event structures. In: BPM. (2014) Accepted.