Generalised Computation of Behavioural Profiles based on Petri-Net Unfoldings

Matthias Weidlich, Felix Elliger, and Mathias Weske

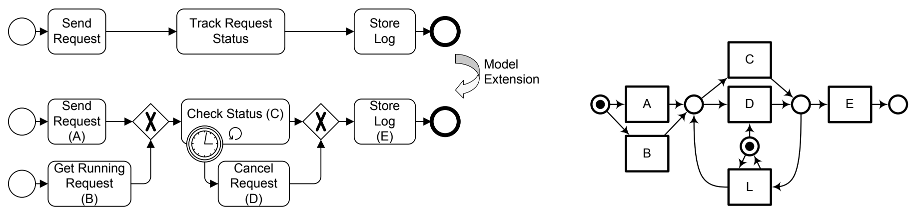
Hasso Plattner Institute at the University of Potsdam, Germany
{Matthias.Weidlich,Mathias.Weske}@hpi.uni-potsdam.de
Felix.Elliger@student.hpi.uni-potsdam.de

Abstract. Behavioural profiles have been proposed as a concept to judge on the behavioural consistency of process models that depict different perspectives of a process. These profiles describe the observable relations between the activities of a process model. Consistency criteria based on behavioural profiles are less sensitive to model projections than common equivalence criteria, such as trace equivalence. Existing algorithms derive those profiles for unlabelled sound free-choice workflow nets efficiently. In this paper, we generalise the computation of behavioural profiles by relaxing the aforementioned assumptions. First, we introduce an algorithm that derives behavioural profiles from the complete prefix unfolding of a bounded Petri net. Hence, it is applicable in a more general case. Second, we lift the concept to the level of labelled Petri nets. We also elaborate on findings of applying our approach to a collection of industry models.

1 Introduction

Notions of behavioural similarity and consistency of business process models have a broad field of application, e.g., for model retrieval [1] or management of process variants [2]. Targeting at the analysis of behavioural consistency between business process models used as a specification and workflow models representing the implementation of the process, behavioural profiles have been proposed as a basis for such consistency analysis [3]. These profiles capture the relations between pairs of activities in terms of their order of potential execution.

Behavioural profiles have been introduced for unlabelled Petri nets along with efficient algorithms for their computation. For sound free-choice Petri nets that have dedicated start and end places (aka workflow nets), the profile is computed in $O(n^3)$ time with n being the number of nodes of the net [3]. Under certain structural assumptions, computation is even more efficient [4]. In this paper, we generalise the computation of behavioural profiles. That is, we introduce an approach for their computation that imposes solely one restriction on a Petri net system — the system has to be bounded. We derive the behavioural profile from the complete prefix unfolding of a Petri net system. We also report on findings of applying our approach to a collection of industry models. In addition, we show how the notion of a behavioural profile is lifted to the level of labelled nets.



(a) (b)

Fig. 1. Process models in BPMN (a); one of the corresponding net systems (b)

Due to their focus on order dependencies, consistency measurement based on behavioural profiles is not affected by model extensions (or model projections, respectively) that impact on the causal dependencies between activities. For instance, Fig. 1(a) depicts two process models in BPMN, the lower one being an extended version of the upper one. A new action to start the process, i.e., *get running request (B)*, has been introduced in the course of model refinement. Apparently, this activity breaks the causal dependency that can be observed between activities *send request* and *store log* in the upper model. Nevertheless, the behavioural profile states that both activities show the same order of potential execution. There are further areas of application for behavioural profiles. They can be used to quantify behavioural deviation between two process models [3], to measure the compliance of process logs [5], and enable change propagation [6].

As mentioned before, there are efficient algorithms for the computation of behavioural profiles that impose the following restrictions.

- The Petri net has to be *sound*. This property is traced back to liveness and boundedness and implies the freedom of deadlocks and livelocks along with the absence of dead transitions. Still, a recent study observed that solely half of the process models in an industry model collection are sound [7]. Moreover, it has been argued that soundness is a rather strict correctness criterion for some use cases. Nevertheless, several weaker correctness criteria, such as weak soundness [8], do not allow for unbounded systems either. Therefore, we argue that boundedness is a reasonable assumption for process models.
- The Petri net has to be *free-choice*. This restriction requires that conflicts and synchronisations of transitions do not interfere. Various constructs of common process modelling languages, such as BPMN, BPEL, and EPCs, can be formalised in free-choice Petri nets [9,10,11]. However, formalisation of exception handling imposes various challenges [12] and typically results in non-free-choice nets (see [9,10]).

The importance of these restrictions is illustrated in Fig. 1(b), which depicts the Petri net formalisation for the lower model in Fig. 1(a). The net is not free-choice due to the modelling of the time-out.

The remainder of this paper is structured as follows. The next section introduces formal preliminaries. Section 3 introduces our approach of deriving the behavioural profile from the complete prefix unfolding along with experimental results. Section 4 defines behavioural profiles for labelled Petri net systems. Finally, Section 5 reviews related work before Section 6 concludes the paper.

2 Background

This section introduces the background of our work. Section 2.1 recalls basic definitions for net systems. Section 2.2 discusses Petri net unfoldings, while Section 2.3 introduces behavioural profiles.

2.1 Net Syntax & Semantics

We recall basic definitions on net syntax and semantics.

Definition 1 (Net Syntax).

- A *net* is a tuple $N = (P, T, F)$ with P and T as finite disjoint sets of places and transitions, and $F \subseteq (P \times T) \cup (T \times P)$ as the flow relation. We write $X = (P \cup T)$ for all nodes. The transitive (reflexive) closure of F is denoted by $<$ ($\leq$). A net is *acyclic*, iff $\leq$ is a partial order.
- For a node $x \in X$, the preset is $\bullet x := \{y \in X \mid (y, x) \in F\}$ and the postset is $x\bullet := \{y \in X \mid (x, y) \in F\}$. For a set of nodes X', $\bullet X' = \bigcup_{x \in X'} \bullet x$ and $X'\bullet = \bigcup_{x \in X'} x\bullet$.
- A tuple $N' = (P', T', F')$ is a *subnet* of a net $N = (P, T, F)$, if $P' \subseteq P$, $T' \subseteq T$, and $F' = F \cap ((P' \times T') \cup (T' \times P'))$.

Definition 2 (Net Semantics). Let $N = (P, T, F)$ be a net.

- $M : P \mapsto \mathbb{N}$ is a *marking* of N, $\mathbb{M}$ denotes all markings of N. $M(p)$ returns the number of *tokens* in place p. We also identify a marking M with the multiset containing $M(p)$ copies of p for every $p \in P$.
- For any two markings $M, M' \in \mathbb{M}$, $M \geq M'$ if $\forall\, p \in P$ [$M(p) \geq M'(p)$].
- For any transition $t \in T$ and any marking $M \in \mathbb{M}$, t is *enabled* in M, denoted by $(N, M)[t\rangle$, iff $\forall\, p \in \bullet t$ [$M(p) \geq 1$].
- Marking M' is reached from M by *firing* of t, denoted by $(N, M)[t\rangle(N, M')$, such that $M' = M - \bullet t + t\bullet$, i.e., one token is taken from each input place of t and one token is added to each output place of t.
- A *firing sequence* of length $n \in \mathbb{N}$ is a function $\sigma : \{0, \ldots, n-1\} \mapsto T$. For $\sigma = \{(0, t_x), \ldots, (n-1, t_y)\}$, we also write $\sigma = t_0, \ldots, t_{n-1}$.
- For any two markings $M, M' \in \mathbb{M}$, M' is reachable from M in N, denoted by $M' \in [N, M_0\rangle$, if there exists a firing sequence σ leading from M to M'.
- A *net system*, or a *system*, is a pair (N, M_0), where N is a net and M_0 is the *initial marking* of N.
- A system (N, M_0) is *bounded*, iff the set $[N, M_0\rangle$ is finite.

2.2 Unfoldings of Net Systems

Any analysis of the state space of a net system has to cope with the state explosion problem [13]. Unfoldings and their complete prefixes have been proposed as a technique to address this problem [14,15]. The unfolding of a net system is another, potentially infinite net system, which has a simpler, tree-like structure. We recall definitions for unfoldings based on [16].

Definition 3 (Occurrence Net, Ordering Relations).

- A pair of nodes $(x, y) \in (X \times X)$ of a net $N = (P, T, F)$ is in the *conflict relation* $\#$, iff $\exists\ t_1, t_2 \in T\ [\ t_1 \neq t_2 \wedge \bullet t_1 \cap \bullet t_2 \neq \emptyset \wedge t_1 \leq x \wedge t_2 \leq y\ ]$.
- A net $N = (P, T, F)$ is an *occurrence net*, iff (1) N is acyclic, (2) $\forall\ p \in P\ [\ |\bullet p| \leq 1\ ]$, and (3) for all $x \in X$ it holds $\neg(x\#x)$ and the set $\{y \in X \mid y < x\}$ is finite. In an occurrence net, transitions are called *events*, while places are called *conditions*.
- For an occurrence net $N = (P, T, F)$, the relation $<$ is the *causality relation*. A pair of nodes $(x, y) \in (X \times X)$ of N is in the *concurrency relation* co, if neither $x \leq y$ nor $y \leq x$ nor $x\#y$.
- For an occurrence net $N = (P, T, F)$, $Min(N)$ denotes the set of minimal elements of X w.r.t. $\leq$.

The relation between a net system $S = (N, M_0)$ with $N = (P, T, F)$ and an occurrence net $O = (C, E, G)$ is defined as a homomorphism $h : C \cup E \mapsto P \cup T$ such that $h(C) \subseteq P$ and $h(E) \subseteq T$; for all $e \in E$, the restriction of h to $\bullet e$ is a bijection between $\bullet e$ and $\bullet h(e)$; the restriction of h to $e\bullet$ is a bijection between $e\bullet$ and $h(e)\bullet$, the restriction of h to $Min(O)$ is a bijection between $Min(O)$ and M_0; and for all $e, f \in E$, if $\bullet e = \bullet f$ and $h(e) = h(f)$ then $e = f$.

A *branching process* of $S = (N, M_0)$ is a tuple $\pi = (O, h)$ with $O = (C, E, G)$ being an occurrence net and h being a homomorphism from O to S as defined above. A branching process $\pi' = (O', h')$ is a *prefix*, if $O' = (C', E', G')$ is a subnet of O, such that if $e \in E'$ and $(c, e) \in G$ or $(e, c) \in G$ then $c \in C'$; if $c \in C'$ and $(e, c) \in G$ then $e \in E'$; h' is the restriction of h to $C' \cup E'$.

The maximal branching process of S is called *unfolding*. The unfolding of a net system can be truncated once all markings of the original net system and all enabled transitions are represented. This yields the complete prefix unfolding.

Definition 4 (Complete Prefix Unfolding). Let $S = (N, M_0)$ be a system and $\pi = (O, h)$ a branching process with $N = (P, T, F)$ and $O = (C, E, G)$.

- A set of events $E' \subseteq E$ is a *configuration*, iff $\forall\ e, f \in E'\ [\ \neg(e\#f)\ ]$ and $\forall\ e \in E'\ [\ f < e \Rightarrow f \in E'\ ]$. The *local configuration* $\lceil e \rceil$ for an event $e \in E$ is defined as $\{x \in X \mid x < e\}$.
- A set of conditions $C' \subseteq C$ is called *co-set*, iff for all distinct $c_1, c_2 \in C'$ it holds $c_1\ co\ c_2$. If C' is maximal w.r.t. set inclusion, it is called a *cut*.
- For a finite configuration C', $Cut(C') = (Min(O) \cup C'\bullet) \setminus \bullet C'$ is a cut, while $h(Cut(C'))$ is a reachable marking of S, denoted $Mark(C')$.
- The branching process is *complete*, iff for every marking $M \in [N, M_0\rangle$ there is a configuration C' of π such that $M = Mark(C')$ and for every transition t enabled in M there is a finite configuration C' and an event $e \notin C'$ such that $M = Mark(C')$, $h(e) = t$, and $C' \cup \{e\}$ is a configuration.
- An *adequate order* $\lhd$ is a strict well-founded partial order on local configurations such that for two events $e, f \in E$ $\lceil e \rceil \subset \lceil f \rceil$ implies $\lceil e \rceil \lhd \lceil f \rceil$.
- An event $e \in E$ is a *cut-off event* induced by $\lhd$, iff there is a *corresponding event* $f \in E$ with $Mark(\lceil e \rceil) = Mark(\lceil f \rceil)$ and $\lceil f \rceil \lhd \lceil e \rceil$.

- The branching process π is the *complete prefix unfolding* induced by $\lhd$, iff it is the greatest prefix of the unfolding of S that does not contain any event after a cut-off event.

We see that the definition of a cut-off event and, therefore, of the complete prefix unfolding is parametrised by the definition of an adequate order $\lhd$. Multiple definitions have been proposed in the literature, cf., [15]. The differences between these definitions can be neglected for our approach and are relevant solely for the experimental evaluation in which we rely on the definition presented in [16]. As we leverage the information on cut-off events in our approach, we include them in the complete prefix for convenience.

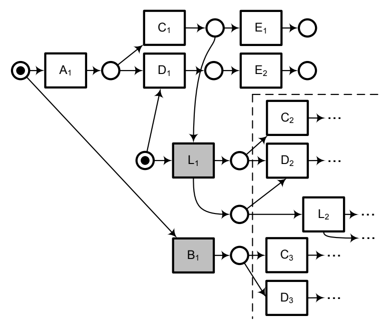


Fig. 2. Complete prefix unfolding of the net system in Fig. 1(b)

Fig. 2 illustrates the concept of an unfolding and its complete prefix for the net system in Fig. 1(b). Here, the labelling of transitions and the initial marking hints at the homomorphism between the two systems. Apparently, the unfolding of the net system is infinite due to the control flow cycle. In Fig. 2, cut-off events are highlighted in grey and the complete prefix unfolding is marked by dashed lines.

2.3 Behavioural Profiles

Behavioural profiles capture behavioural aspects of a system in terms of order constraints [3]. They are based on the set of possible firing sequences of a net system and the notion of *weak order*. Informally, two transitions t_1, t_2 are in weak order, if there exists a firing sequence reachable from the initial marking in which t_1 occurs before t_2.

Definition 5 (Weak Order). Let (N, M_0) be a net system with $N = (P, T, F)$. Two transitions x, y are in the *weak order relation* $\succ \subseteq T \times T$, iff there exists a firing sequence $\sigma = t_1, \ldots, t_n$ with $(N, M_0)[\sigma\rangle$, $j \in \{1, \ldots, n-1\}$, $j < k \leq n$, for which holds $t_j = x$ and $t_k = y$.

Depending on how two transitions of a system are related by weak order, we define three relations forming the behavioural profile.

Definition 6 (Behavioural Profile). Let (N, M_0) be a net system with $N = (P, T, F)$. A pair of transitions $(x, y) \in (T \times T)$ is in at most one of the following *profile relations*:

- The *strict order relation* $\rightsquigarrow$, if $x \succ y$ and $y \not\succ x$.
- The *exclusiveness relation* $+$, if $x \not\succ y$ and $y \not\succ x$.
- The *interleaving order relation* $||$, if $x \succ y$ and $y \succ x$.

$\mathcal{B} = \{\rightsquigarrow, +, ||\}$ is the *behavioural profile* of (N, M_0).

For our example net system in Fig. 1(b), for instance, it holds $A \rightsquigarrow C$, as there exists no firing sequence, in which C occurs before A. As no firing sequence contains both transitions, A and B, it holds $A + B$. Due to the control flow cycle, it holds $C || L$. That is, both transitions can occur in any order in a firing sequence of the system. With $\rightsquigarrow^{-1}$ as the inverse relation for $\rightsquigarrow$, the relations $\rightsquigarrow, \rightsquigarrow^{-1}, +$, and $||$ partition the Cartesian product of transitions.

3 Generalised Computation of Behavioural Profiles

This section introduces our approach of computing the behavioural profile of a bounded net system from its complete prefix unfolding. First, Section 3.1 relates the ordering relations of an unfolding to the relations of the profile. Second, Section 3.2 presents an algorithm for the computation of behavioural profiles. Finally, Section 3.3 presents experimental results.

3.1 Behavioural Relations

The computation of the behavioural profile from the complete prefix unfolding is based on the ordering relations introduced for occurrence nets in Definition 3. The causality, conflict, and concurrency relation partition the Cartesian product of events of an occurrence net and, therefore, of the complete prefix unfolding, similar to the relations of the behavioural profile. However, the ordering relations of an occurrence net relate to *events*, i.e., *occurrences* of transitions of the original net system. Formally, this is manifested in the homomorphism between the net system and its complete prefix unfolding, which may relate multiple events to a single transition. For the net system in Fig. 1(b), for instance, transition C relates to two events in the prefix in Fig. 2, C_1 and C_2 (the latter being a cut-off event). Both events represent an occurrence of transition C in the original system and, hence, have to be considered when deriving the behavioural profile.

In general, the order of potential execution, i.e., the weak order relation of the behavioural profile, can be deduced from the concurrency and the causality relation of the complete prefix unfolding. The existence of a firing sequence containing two transitions of the original system is manifested in two events in the prefix that relate to these transitions and are concurrent or in causality. The former represents two transitions that can be enabled concurrently in the original system. Thus, there is a firing sequence containing both transitions in either order. Two events in causality in the prefix, in turn, represent two transitions in the original net that can occur in a firing sequence in the respective order.

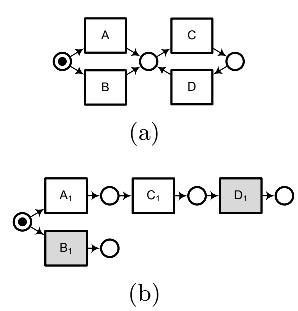


Fig. 3. Net system (a); its complete prefix unfolding (b)

Another observation relates to the fact that not all firing sequences are visible in the complete prefix unfolding directly. Events that relate to two transitions might not show causality or concurrency although the respective transitions might occur in a firing sequence. Fig. 3 illustrates this issue. Apparently, transition D might be observed in firing sequences that commence with transition B in the net system in Fig. 3(a). The events that represent occurrences of both transitions, B_1 and D_1, are in conflict in the complete prefix unfolding in Fig. 3(b). Hence, the information that is hidden due to the cut of the unfolding has to be taken into account. In Fig. 3(b), firing of B_1 leads to a marking that is already contained in the prefix. That is, firing of A_1 reaches at the same marking and enables the event C_1. Therefore, it can be deduced that an event relating to transition D of the original net can follow any firing of event B_1.

While this example illustrates solely a simple dependency, Fig. 4 illustrates the general case. Consider the events e and f and assume that both events represent the occurrences of different transitions t_e and t_f of the original net system and that both events are in conflict. Still, the transitions t_e and t_f might occur in a firing sequence of the net system due to the cut-off events c_1 and c_2 and their corresponding events c'_1 and c'_2. Event e is in a causal relation with the cut-off event c_1. The respective cut of its local configuration is highlighted by a dashed line. This cut corresponds to the cut of the local configuration of event c'_1, which, in turn, is concurrent to another cut-off event c_2. Similarly, a relation can be established between c_2 and f via c'_2. This example illustrates that such dependencies between two events may span multiple cut-off events.

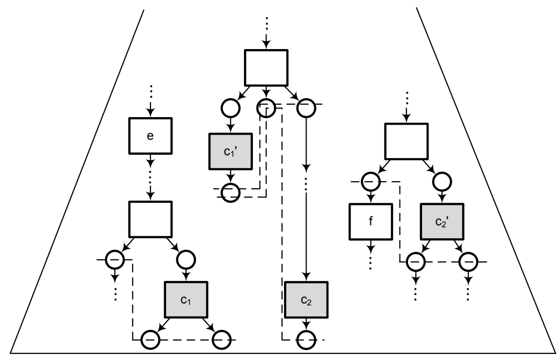


Fig. 4. Sketch of a complete prefix unfolding; e and f can occur together in a firing sequence

Based on this observation, we establish the relation between the ordering relations of the complete prefix unfolding and the weak order relation as follows.

Proposition 1. Let $S = (N, M_0)$ be a bounded system and $\pi = (O, h)$ its complete prefix unfolding including the cut-off events with $N = (P, T, F)$ and $O = (C, E, G)$. Then, two transitions $x, y \in T$ are in weak order, $x \succ y$, iff there are two events $e, f \in E$ with $h(e) = x$ and $h(f) = y$ and either

- they are causally related or concurrent, i.e., $(e < f) \vee (e \; co \; f)$, or
- there is a sequence of cut-off events $(g_1, g_2, \ldots, g_n)$ with $g_i \in E$ for $1 \leq i \leq n$ and a sequence of corresponding events $(g'_1, g'_2, \ldots, g'_n)$ with $g'_i \in E$, such that $(e = g_1) \vee (e < g_1)$, $\exists \; c \in Cut(\lceil g'_j \rceil) \; [\; c < g_{j+1} \;]$ for $1 \leq j < n$, and either $g'_n = f$ or $\exists \; c \in Cut(\lceil g'_n \rceil) \; [\; c < f \;]$.

Proof. Let x, y, $(g_1, g_2, \ldots, g_n)$, and $(g'_1, g'_2, \ldots, g'_n)$ be defined as above.
($\Rightarrow$) Let $x \succ y$. Then, there is a reachable firing sequence in S containing

transition x before y. As the prefix is complete, both occurrences of transitions are represented by corresponding events $e, f \in E$ with $h(e) = x$ and $h(f) = y$. If there is a firing sequence in O starting in its initial marking induced by $Min(O)$ that contains e before f, they are causally related or concurrent, i.e., $(e < f) \vee (e \; co \; f)$ (first statement of our proposition). If there is no such firing sequence, then either $e \# f$ or $f < e$. If there are no cut-off events after firing of e in O, all markings that are reached after firing of x in S have their counterparts in π directly. Those do not enable f. Therefore, there has to be a cut-off event k that can occur after firing of e (or which is equal to e). Hence, either $e \leq k$, or $e \; co \; k$. Assume that the sequence of cut-off events is one. Then, the corresponding event k' for the cut-off event k has to occur together with the event f, i.e., $k' < f$, $f < k'$ or $f \; co \; k'$. If $e \; co \; k$, then there has to be an event e' with $h(e) = h(e')$ and $e' \; co \; k'$. In this case, we, again, arrive at $(e' < f) \vee (e' \; co \; f)$. Consider $e \leq k$. To observe a firing of x before y, we have to exclude $f < k'$ from the possible relations between f and k'. That is because $f < k'$ implies that y is observed before the marking represented by the cut of $\lceil k \rceil$ is reached as $Mark(\lceil k \rceil) = Mark(\lceil k' \rceil)$. Hence, we have $k' < f$ or $f \; co \; k'$ (the second statement of our proposition). The same argument can be applied to all intermediate cut-off events in case the sequence of cut-off events is longer than one.

($\Leftarrow$) For all events $e, f \in E$ with $h(e) = x$ and $h(f) = y$ let (1) $(e < f) \vee (e \; co \; f)$, or (2) there is a sequence of cut-off events $(g_1, g_2, \ldots, g_n)$ and corresponding events $(g'_1, g'_2, \ldots, g'_n)$, such that $(e = g_1) \vee (e < g_1)$, $\exists \, c \in Cut(\lceil g'_j \rceil) \; [\; c < g_{j+1} \;]$ for all $j \in \{1, \ldots, n-1\}$, and $g'_n = f$ or $\exists \, c \in Cut(\lceil g'_n \rceil) \; [\; c < f \;]$. Assume that $x \not\succ y$. From the initial marking of the occurrence net that is induced by $Min(O)$, we can fire all events of the local configuration of f, $\lceil f \rceil$, in their order induced by $<$ to reach a marking M_1. Then, event f can be fired in this marking to reach marking M_2. If $e < f$ then event is e is part of $\lceil f \rceil$ and, therefore, has been fired already. If $e \; co \; f$, event e has not been fired as part of $\lceil f \rceil$ to reach M_1. Still, there must be an event g in $\lceil f \rceil$ such that for every condition $c_f \in \bullet f$ there is a condition $c_g \in g\bullet$ with $c_g < c_f$ or $c_g = c_f$. Due to $e \; co \; f$, $g < f$, and $g < e$, we also know $c_g \not< e$ for all those conditions c_g. All conditions c_g are marked in M_1. They are also marked in M_2 reached via firing of e as $c_g \not< e$. Hence, there has to be a firing sequence starting in M_2 and containing all events that are part of $\lceil f \rceil \setminus \lceil e \rceil$. Then, event f can be fired in the reached marking. Thus, in both cases there is a firing sequence containing both events e and f. When all fired events are resolved according to h, there is a firing sequence in S containing x before y, which is a contradiction with $x \not\succ y$. Consider case (2). Following on the argument given in the previous case, we know that there is a firing sequence in the occurrence net that contains event e before event g_1. For the corresponding event g'_1, we know $Mark(\lceil g_1 \rceil) = Mark(\lceil g'_1 \rceil)$. Hence, the marking in S reached via firing the transitions that correspond to all events $\lceil g_1 \rceil$ is equal to the marking reached by firing the corresponding transitions in $\lceil g'_1 \rceil$. Assume that the sequence of cut-off events is one, i.e., $\exists \, c \in Cut(\lceil g'_1 \rceil) \; [\; c < f \;]$. Again, there is a firing sequence in the occurrence net that contains g'_1 before event f. Due to $Mark(\lceil g_1 \rceil) = Mark(\lceil g'_1 \rceil)$, there is a firing sequence in S that contains the transition represented by the event g_1 before the one represented by f, i.e., transition y. From $e < g_1$, we get that such a firing sequence can contain the transition represented by event e, i.e., x, before the one represented by g_1 and, therefore, also before y. Thus, there is a firing sequence in S in which x is followed by y and we arrive at a contradiction with $x \not\succ y$. Again, the same argument can be applied to all intermediate cut-off events in case the sequence of cut-off events is longer than one. □

3.2 Computation Algorithm

As we have seen in the previous section, the weak order relation of the behavioural profile can be traced back to the ordering relations of the complete prefix unfolding. Based thereon, Algorithm 1 shows how the behavioural profile is computed for a bounded system given its complete prefix unfolding.

First and foremost, the algorithm comprises the computation of the ordering relations, i.e., the causality, conflict, and concurrency relation, for the complete prefix unfolding (line 1). The respective algorithm can be found in [17].

Second, we capture relations between cut-off events (lines 2 to 12). We compute the relation between events and the conditions that belong to the cut induced by their local configuration (set $\mathcal{L}$). Then, causality between these conditions of one event and another event is captured in set $\mathcal{LC}$. The set $\mathcal{C}_{cut}$ is filled with all cut-off events, while their corresponding events are added to the set $\mathcal{C}_{cor}$. The relation between them is stored as an entry in the relation $\mathcal{C}$. We check for each event that corresponds to a cut-off event, whether it is in a causal relation with a condition of the cut relating to the local configuration of another cut-off event. If so, this information is stored in the relation $\mathcal{C}$. The intuition behind is that the transitive closure of $\mathcal{C}$ hints at the existence of a sequence of cut-off and corresponding events as defined in Proposition 1.

Third, all pairs of events of the complete prefix unfolding are assessed according to Proposition 1 (lines 13 to 21). If the respective requirements are met, the weak order relation is captured for the transitions that are represented by these events.

Finally, the relations of the behavioural profile are derived from the weak order relation according to Definition 6 (lines 22 to 27).

Proposition 2. Algorithm 1 terminates and after termination $\mathcal{B} = \{\rightsquigarrow, +, ||\}$ is the behavioural profile of S.

Proof. Termination: The algorithm iterates over sets that are derived from E, C, $\mathcal{C}_{cut} \subseteq E$, $\mathcal{C}_{cor} \subseteq E$, and T. T is finite by definition. Due to boundedness of the net system the complete prefix unfolding and, therefore, the sets of events E and conditions C are finite as well. Hence, the algorithm terminates.
Result: The set $\mathcal{L}$ contains all pairs of events and conditions, such that the condition belongs to the cut of the local configuration of the respective event. That is achieved by considering all conditions of the postset of the event and all concurrent conditions that are either initially marked or are part of a postset of another event that is in a causal relation with the former event. Then, an

Algorithm 1: Algorithm for the computation of the behavioural profile

Input: $S = (N, M_0)$, a bounded system with $N = (P, T, F)$.
$\pi = (O, h)$, its complete prefix unfolding including cut-off events with $O = (C, E, G)$.
Output: $\mathcal{B} = \{\rightsquigarrow, +, ||\}$, the behavioural profile of S.

Compute ordering relations $<$, $\#$, and co of O;

/* Establish relation between cut-off events of O */
$\mathcal{L}, \mathcal{LC}, \mathcal{C}_{cor}, \mathcal{C}_{cut}, \mathcal{C} \longleftarrow \emptyset$;
foreach $(e, c) \in (E \times C)$ **do**
    if $(c \in e\bullet) \vee ((e\ co\ c) \wedge ((\bullet c \times \{e\} \subseteq <) \vee (\bullet c = \emptyset)))$ **then** $\mathcal{L} \longleftarrow (e, c)$;
end
foreach $(e_1, c, e_2) \in (\mathcal{L} \times E)$ **do** **if** $c < e_2$ **then** $\mathcal{LC} \longleftarrow (e_1, e_2)$;
foreach $(e_1, e_2) \in (E \times E)$ **do**
    if $(Mark(\lceil e_1 \rceil) = Mark(\lceil e_2 \rceil)) \wedge (\lceil e_1 \rceil \lhd \lceil e_2 \rceil)$ **then**
        $\mathcal{C}_{cor} \longleftarrow (e_1)$; $\mathcal{C}_{cut} \longleftarrow (e_2)$; $\mathcal{C} \longleftarrow (e_2, e_1)$;
    end
end
foreach $(e_{cor}, e_{cut}) \in (\mathcal{C}_{cor} \times \mathcal{C}_{cut})$ **do** **if** $e_{cor}\ \mathcal{LC}\ e_{cut}$ **then** $\mathcal{C} \longleftarrow (e_{cor}, e_{cut})$;

/* Derive weak order for transitions of N */
$\succ \longleftarrow \emptyset$;
foreach $(e_1, e_2) \in (E \times E)$ **do**
    if $(e_1 < e_2) \vee (e_1\ co\ e_2)$ **then** $\succ \longleftarrow (h(e_1), h(e_2))$;
    else **foreach** $(e_{cut}, e_{cor}) \in (\mathcal{C}_{cut} \times \mathcal{C}_{cor})$ **do**
        if $(e_1 \leq e_{cut}) \wedge (e_{cut}\ \mathcal{C}^+\ e_{cor}) \wedge (e_{cor}\ \mathcal{LC}\ e_2)$ **then**
            $\succ \longleftarrow (h(e_1), h(e_2))$;
        end
    end
end

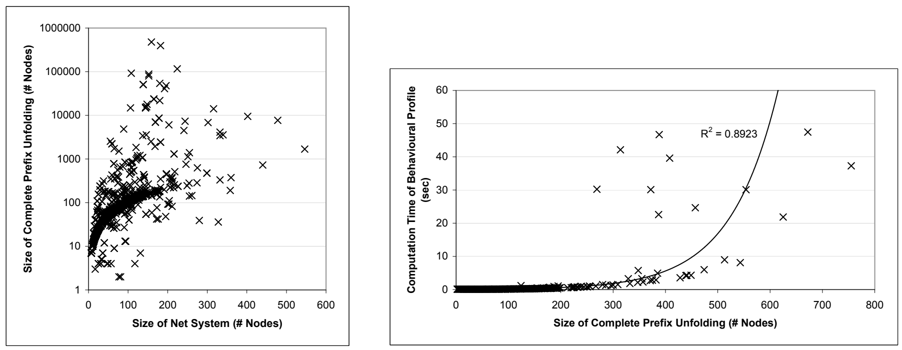
/* Derive relations of behavioural profile of N */
$\rightsquigarrow, +, || \longleftarrow \emptyset$;
foreach $(t_1, t_2) \in (T \times T)$ **do**
    if $(t_1 \succ t_2) \wedge (t_2 \succ t_1)$ **then** $|| \longleftarrow (t_1, t_2)$;
    else if $t_1 \succ t_2$ **then** $\rightsquigarrow \longleftarrow (t_1, t_2)$;
    else $+ \longleftarrow (t_1, t_2)$;
end

entry of set $\mathcal{LC}$ associates an event to all events that are in causality to one of the conditions of the former event's cut. Moreover, set $\mathcal{C}$ is build such that it contains all pairs of cut-off events and their corresponding events. Then, pairs of corresponding events and further cut-off events are added, if there is a causal relation between them in $\mathcal{LC}$. Hence, the transitive closure of $\mathcal{C}$ hints at the existence of a sequence of cut-off events and corresponding events that are causally related. Based thereon, Proposition 1 is implemented directly. The derivation of the profile from the weak order relation follows directly on Definition 6. □

The algorithm runs in polynomial time with respect to size of the complete prefix unfolding. The final step of the algorithm that sets the profile relations based on the weak order relation for all pairs of transitions is neglected at this point.

Proposition 3. The following problem can be solved in $O(n^4)$ time with n as the number of events and conditions of the complete prefix unfolding:
For a bounded net system and its complete prefix unfolding, to compute the weak order relation for the net system.

Proof. Computation of the ordering relations of the complete prefix unfolding can be done in $O(|E| * |C|)$ time [17]. In the second step of the algorithm, we



(a) (b)

Fig. 5. (a) Size of complete prefix unfolding relative to the size of the net system; (b) overall computation time relative to the size of the complete prefix unfolding

iterate over $E \times C$, $E \times C \times E$, and $E \times E$, which takes $O(|E|^2 * |C|)$ time. Due to $\mathcal{C}_{cut} \subseteq E$ and $\mathcal{C}_{cor} \subseteq E$, the third step takes $O(|E|^4)$ time. As a prerequisite for this step the transitive closure of $\mathcal{C}$ has to be computed, which takes $O(|E|^3)$ time. Therefore, overall time complexity is $O(n^4)$ with n as the number of events and conditions of the complete prefix unfolding. □

3.3 Experimental Evaluation

The approach introduced in the previous section is applicable in a general case as it requires only boundedness of the net system. However, the generality is traded for computational complexity. On the one hand, computation of the prefix unfolding is computationally hard, as it is an NP-complete problem. On the other hand, our algorithm runs in polynomial time with respect to the size of the complete prefix unfolding. We are using the partial order defined in [16] for identifying cut-off events, which has been shown to create compact prefixes. Nevertheless, complete prefix unfoldings might have a very large size, such that even a polynomial time algorithm results in long computation times. According to [16], the size of the prefix is at most the size of the reachability graph.

In order to investigate the implications of these issues, we conducted an experiment based on 735 industry process models for which net systems have been derived (see [7] for further details). Although these systems were free-choice, only half of them were sound [7]. Hence, the existing efficient algorithms would be applicable solely for half of these models. In the implementation of our approach, we used Mole[1] to generate the complete prefix unfolding.

For 4 net systems, creation of the complete prefix unfolding was not possible due to the size of the unfolding. The maximum size of the derived complete prefix unfoldings was 500.000 nodes. Fig. 5(a) illustrates that prefixes can be an order of magnitude larger in size than the original net systems (note the logarithmic scale). Still, the majority of complete prefix unfoldings was rather small. 94%

[1] `http://www.fmi.uni-stuttgart.de/szs/tools/mole/`

of the net systems had a prefix with less than 800 nodes. For these net systems, Fig. 5(b) shows the overall computation time of the behavioural profile including prefix creation relative to the size of the prefix in terms of the number of nodes. For small prefixes with less than 250 nodes, computation of the behavioural profile is done within one or two seconds. However, computation for prefixes with more than 300 nodes may take up to tens of seconds. The inherent exponential complexity is also visible in the exponential least squares regression depicted in Fig. 5(b). Taking into account that 94% of the net systems have a complete prefix unfolding with less than 800 nodes, we see that our approach works for the majority of process models in the collection within a reasonable time.

In order to shade light on the trade-off between efficiency and generality of the different approaches for the profile computation, Fig. 6 compares the generic approach based on unfoldings with the structural approach presented in [3] (along with exponential or polynomial least squares regressions). Due to the assumptions of the latter approach, solely sound systems are considered. The differences in the base effort are due to the usage of an external unfolding tool. We see that effects in terms of efficiency become visible for net systems that have more than 100 nodes. For these systems, the structural characterisation building on the assumption of soundness and free-choiceness is more efficient.

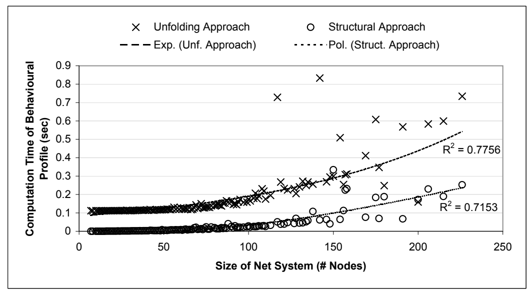


Fig. 6. Overall computation time relative to the size of the net system

4 Behavioural Profile of a Labelled Net System

As the final step of our generalised profile computation, we lift the concept to the level of labelled net systems. While our initial example in Fig. 1 comprises solely unique transitions, this assumption does not hold for real world process models. Multiple transitions with the same label occur as activities might be executed at different stages of a process. For instance, exception handling for an activity might comprise the possibility to redo the activity. Then, there may be two transitions with the same label, one as part of the standard processing and one as part of the exception handling.

Formally, a *labelled net* is a tuple $N = (P, T, F, \Lambda, \lambda)$, where (P, T, F) is a net, Λ is a set of labels, and $\lambda : T \mapsto \Lambda$ is a labelling function. (N, M_0) is called *labelled* system, iff N is a labelled net. The weak order relation can be lifted to labels as follows. Two labels are in weak order, if and only if, two transitions carrying those labels are in weak order. Based thereon, the relations of the behavioural profile can be lifted to labels in a straight-forward manner.

Definition 7 (Weak Order & Behavioural Profile on Labels). Let $S = (N, M_0)$ be a labelled system with $N = (P, T, F, \Lambda, \lambda)$ and $\succ$ the weak order relation of S.

- Two labels l_1, l_2 are in the *weak order on labels relation* $\succ_\Lambda \subseteq \Lambda \times \Lambda$, iff there are two transitions $x, y \in T$ such that $\lambda(x) = l_1$, $\lambda(y) = l_2$, and $x \succ y$.
- A pair of labels $(l_1, l_2) \in (\Lambda \times \Lambda)$ is in at most one of the following relations:
  - The *strict order relation* $\rightsquigarrow_\Lambda$, if $l_1 \succ_\Lambda l_2$ and $l_2 \not\succ_\Lambda l_1$.
  - The *exclusiveness relation* $+_\Lambda$, if $l_1 \not\succ_\Lambda l_2$ and $l_2 \not\succ_\Lambda l_1$.
  - The *interleaving order relation* $||_\Lambda$, if $l_1 \succ_\Lambda l_2$ and $l_2 \succ_\Lambda l_1$.

  $\mathcal{B} = \{\rightsquigarrow_\Lambda, +_\Lambda, ||_\Lambda\}$ is the *behavioural profile on labels* of S.

As behavioural profile on labels is derived directly from the behavioural profile its computation can be done efficiently.

Proposition 4. The following problem can be solved in $O(n^2)$ time with n as the number of transitions:
Given the behavioural profile of a labelled net system, to derive its behavioural profile on labels.

Proof. Both, deriving the weak order relation on labels and setting the relations of the behavioural profile on labels, requires iteration over the Cartesian product of transitions of the net system. Assuming that each label relates to at least one transition, the number of labels is smaller than the number of transitions. Thus, overall time complexity is $O(n^2)$ with n as the number of transitions. □

5 Related Work

Since the unfolding technique has been introduced by McMillan [14], it has been extended and analysed in a huge number of publications, see [15] for a thorough discussion. Moreover, this technique has been applied for various purposes. Unfoldings are used to check common properties of net systems, such as reachability of certain markings, or even for LTL model checking (cf., [18]). Moreover, domain specific problems, e.g., the analysis and synthesis of asynchronous circuits [19], have been addressed using the unfolding technique. Recently, complete prefix unfoldings have been used to restructure process models [20].

Besides applications of unfoldings, our work relates to notions of behavioural consistency or similarity, as behavioural profiles provide a behavioural abstraction. There is a huge body of work on such notions, starting with the equivalence criteria of the linear time – branching time spectrum [21]. Similarity measures may define an edit distance between processes, which, in turn, is based on the n-gram representation of the process language or the underlying automaton [22]. Close to behavioural profiles are causal footprints [23]. These footprints capture causal dependencies between activities and can also be leveraged to determine behavioural similarity. Behavioural relations between activities are also at the core of many process mining algorithms that aim at constructing models from event logs. The α-algorithm for mining process models [24] is based on relations that are similar to those of the behavioural profile, yet different.

6 Conclusion

In this paper, we generalised the computation of behavioural profiles. We introduced an algorithm that derives behavioural profiles from the complete prefix unfolding of a bounded Petri net. Moreover, we showed how behavioural profiles are lifted to the level of labelled net systems. Therefore, our approach is applicable in a more general setting than the existing techniques.

The overall complexity of our approach is dominated by the exponential complexity of computing the prefix unfolding, which is an NP-complete problem. We investigated the implications of this complexity issue by an experimental setup involving 735 industry process models. Our results show that computation of behavioural profiles is feasible within seconds for a certain class of process models, for which the complete prefix unfolding contains several hundreds of nodes. We also proved that our computation based on the complete prefix unfolding runs in low polynomial time. Due to the inherent complexity of the unfolding technique, our approach cannot be applied to all process models. However, we significantly broadened the set of process models for which behavioural profiles can be derived as existing algorithms assume soundness and free-choiceness of net systems. Therefore, our approach gives rise to the usage of the various applications of behavioural profiles (cf., [3,5,6,25]) for a wider class of process models.

There is a lot of potential for the combination of the presented approach with structural decomposition techniques, similar to those used in [4] for the computation of behavioural profiles for sound free-choice systems. Sub-systems that represent single-entry single-exit blocks might be considered in isolation. Then, the profile is computed iteratively for all these sub-systems, which in combination yield the profile for the whole system. The unfoldings of these sub-systems may be significantly smaller in size than the unfolding of the overall system. For sound and free-choice sub-systems, the efficient algorithms from our previous work can be used. We want to investigate the combination of these techniques in future work. Moreover, we want to explore the application of unfoldings for unbounded net systems [26], which would allow for the computation of behavioural profiles even for this class of systems.

Acknowledgements. We would like to thank the anonymous reviewers for the constructive comments. We very much appreciate their suggestions regarding the experimental evaluation and the simplification of the proof of Proposition 1.

References

1. Dumas, M., García-Bañuelos, L., Dijkman, R.M.: Similarity search of business process models. IEEE Data Eng. Bull. **32**(3) (2009) 23–28
2. van der Aalst, W.M.P.: Inheritance of business processes: A journey visiting four notorious problems. In: Petri Net Technology for Communication-Based Systems. Volume 2472 of LNCS., Springer (2003) 383–408
3. Weidlich, M., Mendling, J., Weske, M.: Efficient consistency measurement based on behavioural profiles of process models. IEEE Transactions on Software Engineering (2010) to appear.

4. Weidlich, M., Polyvyanyy, A., Mendling, J., Weske, M.: Efficient computation of causal behavioural profiles using structural decomposition. In: PETRINETS. Volume 6128 of LNCS., Springer (2010) 63–83
5. Weidlich, M., Polyvyanyy, A., Desai, N., Mendling, J.: Process compliance measurement based on behavioural profiles. In: CAiSE. Volume 6051 of LNCS., Springer (2010) 483–498
6. Weidlich, M., Weske, M., Mendling, J.: Change propagation in process models using behavioural profiles. In: IEEE SCC, IEEE CS (2009) 33–40
7. Fahland, D., Favre, C., Jobstmann, B., Koehler, J., Lohmann, N., Völzer, H., Wolf, K.: Instantaneous soundness checking of industrial business process models. In: BPM. Volume 5701 of LNCS., Springer (2009) 278–293
8. Martens, A.: On Compatibility of Web Services. Petri Net Newsletter **65** (2003) 12–20
9. Dijkman, R., Dumas, M., Ouyang, C.: Semantics and Analysis of Business Process Models in BPMN. IST **50**(12) (2009) 1281–1294
10. Lohmann, N.: A feature-complete Petri net semantics for WS-BPEL 2.0. In: WS-FM. Volume 4937 of LNCS., Springer (2008) 77–91
11. Kindler, E.: On the semantics of EPCs: A framework for resolving the vicious circle. In: BPM. Volume 3080 of LNCS. (2004) 82–97
12. Lohmann, N., Verbeek, E., Dijkman, R.M.: Petri net transformations for business processes - a survey. TOPNOC **2** (2009) 46–63
13. Valmari, A.: The state explosion problem. In: Petri Nets. Volume 1491 of LNCS., Springer (1996) 429–528
14. McMillan, K.L.: A technique of state space search based on unfolding. FMSD **6**(1) (1995) 45–65
15. Esparza, J., Heljanko, K.: Unfoldings: a partial-order approach to model checking. Springer (2008)
16. Esparza, J., Römer, S., Vogler, W.: An improvement of mcmillan's unfolding algorithm. FMSD **20**(3) (2002) 285–310
17. Kondratyev, A., Kishinevsky, M., Taubin, A., Ten, S.: Analysis of petri nets by ordering relations in reduced unfoldings. FMSD **12**(1) (1998) 5–38
18. Esparza, J., Heljanko, K.: A new unfolding approach to ltl model checking. In: ICALP. Volume 1853 of LNCS., Springer (2000) 475–486
19. Khomenko, V., Koutny, M., Yakovlev, A.: Logic synthesis for asynchronous circuits based on stg unfoldings and incremental sat. Fundam. Inform. **70**(1-2) (2006) 49–73
20. Polyvyanyy, A., García-Bañuelos, L.L., Dumas, M.: Structuring acyclic process models. In: BPM. (September 2010) (to appear).
21. Glabbeek, R., Goltz, U.: Refinement of actions and equivalence notions for concurrent systems. Acta Inf. **37**(4/5) (2001) 229–327
22. Wombacher, A.: Evaluation of technical measures for workflow similarity based on a pilot study. In: OTM Conf. Volume 4275 of LNCS., Springer (2006) 255–272
23. Dongen, B., Dijkman, R.M., Mendling, J.: Measuring similarity between business process models. In: CAiSE. Volume 5074 of LNCS., Springer (2008) 450–464
24. 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
25. Smirnov, S., Weidlich, M., Mendling, J., Weske, M.: Action patterns in business process models. In: ICSOC. Volume 5900 of LNCS. (2009) 115–129
26. Desel, J., Juhás, G., Neumair, C.: Finite unfoldings of unbounded petri nets. In: ICATPN. Volume 3099 of LNCS., Springer (2004) 157–176