Efficient Computation of Causal Behavioural Profiles using Structural Decomposition

Matthias Weidlich[1], Artem Polyvyanyy[1], Jan Mendling[2], and Mathias Weske[1]

[1] Hasso Plattner Institute at the University of Potsdam, Germany
(Matthias.Weidlich,Artem.Polyvyanyy,Mathias.Weske)@hpi.uni-potsdam.de
[2] Humboldt-Universität zu Berlin, Germany
Jan.Mendling@wiwi.hu-berlin.de

Abstract. Identification of behavioural contradictions is an important aspect of software engineering, in particular for checking the consistency between a business process model used as system specification and a corresponding workflow model used as implementation. In this paper, we propose causal behavioural profiles as the basis for a consistency notion, which capture essential behavioural information, such as order, exclusiveness, and causality between pairs of activities. Existing notions of behavioural equivalence, such as bisimulation and trace equivalence, might also be applied as consistency notions. Still, they are exponential in computation. Our novel concept of causal behavioural profiles provides a weaker behavioural consistency notion that can be computed efficiently using structural decomposition techniques for sound free-choice workflow systems if unstructured net fragments are acyclic or can be traced back to S- or T-nets.

1 Introduction

Process modelling has recently become one of the most extensively used approaches for capturing business requirements [1]. These requirements are typically refined and modified in an engineering process, resulting in a workflow model and software artefacts. A workflow model often defines activities of the business process model in more detail, neglects steps that are or do not need to be supported by the system, or adjusts behaviour to the specifics of the workflow system. This raises the question to which degree a process model used as specification and a workflow model used as implementation are behaviourally consistent.

Fig. 1 illustrates this problem. Model (a) assumes a business perspective, whereas (b) shows the workflow implementation of the process. Activities (or sets thereof) that correspond to each other are connected by dash-dotted lines. For this paper, we assume that such correspondences are given. They may stem from a system analyst inspecting the models or from automatic matching. Recently, techniques including structural analysis and natural language processing to automatically identify such correspondences have been introduced for the domain of business process models [2,3]. Moreover, techniques known from the area of
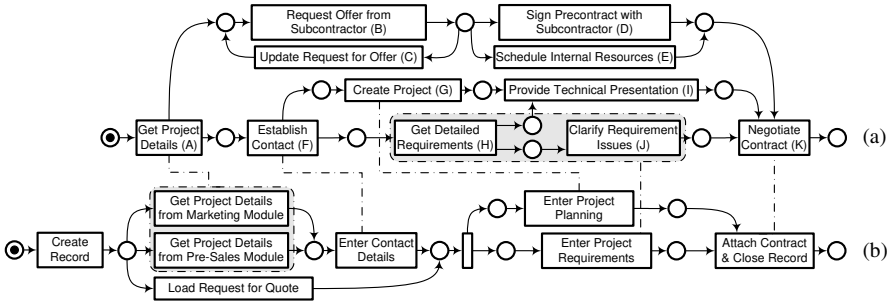
Fig. 1. Example of two Petri net process models, (a) focussing on the business perspective, (b) depicting the workflow implementation

schema matching [4] can be exploited as activities might be regarded as elements of a process model schema.

In order to reason about the relation between two process models, existing notions of behavioural equivalence might be used as a consistency measure. For instance, bisimulation and trace equivalence assume the set of all traces or the branching structure as essential behavioural characteristics that have to be preserved. However, these notions are computationally hard [5], which is particularly a problem for process models including many activities. Furthermore, these notions only provide information whether behaviour is equivalent or not, but do not describe how strong a deviation is in case of a mismatch.

In this paper, we argue that for the use case of comparing business process models and workflow models, a criterion of behavioural equivalence might be weakened in order to compensate for computational efficiency. We define the notion of a *causal behavioural profile*, which includes dependencies in terms of order, exclusiveness, or causality between pairs of activities. It is computed efficiently using structural decomposition techniques for sound free-choice workflow systems if unstructured net fragments are acyclic or can be traced back to S- or T-nets. We also illustrate how these profiles form the basis of a consistency notion that is weaker than existing notions of behavioural equivalence. Note that proofs not given in this paper are available in a separate technical report [6].

This paper is structured accordingly. Section 2 introduces our formal framework. Causal behavioural profiles are defined in Section 3. Section 4 elaborates on graph decomposition and introduces their application to workflow nets. Their application for computing causal behavioural profiles along with experimental results is presented in Section 5. Finally, Section 6 reviews related work, before Section 7 concludes the paper.

2   Preliminaries

We use workflow (WF-) systems [7] as our formal grounding, a class of Petri nets used for process modelling and analysis. Note that Petri net based formalisations have been presented for (parts of) common process modelling languages, such as BPEL, EPCs, and UML, e.g., [8,9,10]. Based on [7,11], we recall basic definitions.

Definition 1 (WF-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 closure of F is denoted by F^+.
- For a node $x \in X$, $\bullet x := \{y \in X \mid (y, x) \in F\}$, $x\bullet := \{y \in X \mid (x, y) \in F\}$, $\bullet(x\bullet) := \{z \in X \mid y \in X \wedge (x, y) \in F \wedge (z, y) \in F\}$.
- A tuple $N' = (P', T', F')$ is a *subnet* of a net $N = (P, T, F)$, if $P' \subseteq P$, $T' \subseteq T$, and $F' = F \cap ((P' \times T') \cup (T' \times P'))$; N' is a *partial subnet* of N, if $F' \subseteq F \cap ((P' \times T') \cup (T' \times P'))$.
- A net N is a *T-net*, if $\forall \, p \in P \, [\, | \bullet p| = 1 = |p \bullet | \,]$, and an *S-net*, if $\forall \, t \in T \, [\, | \bullet t| = 1 = |t \bullet | \,]$.
- A net N is *free-choice*, iff $\forall \, p \in P$ with $|p \bullet | > 1$ holds $\bullet(p\bullet) = \{p\}$.
- A *path* is a non-empty sequence $x_1, \ldots, x_k$ of nodes, $k > 1$, denoted by $\pi_N(x_1, x_k)$, which satisfies $(x_1, x_2), \ldots, (x_{k-1}, x_k) \in F$. We write $x_i \in \pi_N$, if x_i is part of the path π_N. A *subpath* π'_N of a path π_N is a subsequence that is itself a path. A path $\pi_N(x_1, x_k)$ is a *circuit*, if $(x_k, x_1) \in F$ and no node occurs more than once in the path.
- For a net $N = (P, T, F)$ and a partial subnet N' a path $\pi_N(x_1, x_k)$, $k > 1$ and all x_i are distinct, of N is a *handle* of N', iff $\pi_N \cap (P' \cup T') = \{x_1, x_k\}$.
- For a net $N = (P, T, F)$ and two partial subnets N', N'' a path $\pi_N(x_1, x_k)$ ($k > 1$ and all x_i are distinct) of N is a *bridge* from N' to N'', iff $\pi_N \cap (P' \cup T') = \{x_1\}$ and $\pi_N \cap (P'' \cup T'') = \{x_k\}$.
- A Petri net $N = (P, T, F)$ is a *workflow (WF-) net*, iff N has an *initial* place $i \in P$ with $\bullet i = \emptyset$, N has a *final* place $o \in P$ with $o\bullet = \emptyset$, and the *short-circuit* net $N' = (P, T \cup \{t_c\}, F \cup \{(o, t_c), (t_c, i)\})$ of N is strongly connected.

Note that we speak of PP-, TT-, PT-, TP- handles and bridges, depending on the type (place or transition, respectively) of the initial and the final node of the respective path. Further on, we define semantics for WF-nets according to [7].

Definition 2 (WF-net Semantics). Let $N = (P, T, F)$ be a WF-net with initial place i and final place o.
- $M : P \mapsto \mathbb{N}$ is a *marking* of N, $\mathbb{M}$ denotes all markings of N. $M(p)$ returns the number of *tokens* in place p. $[p]$ denotes the marking when place p contains just one token and all other places contain no tokens.
- 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 $M' \in [N, M_i\rangle$, if there exists a firing sequence σ leading from M to M'.
- A *net system*, or a *system*, is a pair (N, M_i), where N is a net and M_i is the *initial marking* of N. A *WF-system* is a pair (N, M_i), where N is a WF-net with initial place i and $M_i = [i]$.

Note that the *final marking* is denoted by M_o. Without stating it explicitly, we assume a net of a system to be defined as $N = (P, T, F)$. Moreover, when the context is clear, we refer to WF-systems and short-circuit nets as WF-nets. Finally, we recall the *soundness* property, which requires WF-systems (1) to always terminate, and (2) to have no dead transitions (proper termination is implied for WF-systems) [12].

Definition 3 (Liveness, Boundedness, Soundness).
- A system (N, M_i) is *live*, iff for every reachable marking $M \in [N, M_i\rangle$ and $t \in T$, there exists a marking $M' \in [N, M\rangle$ such that $(N, M')[t\rangle$.
- A system (N, M_i) is *bounded*, iff the set $[N, M_i\rangle$ is finite.
- A WF-system (N, M_i) is *sound*, iff the short-circuit system (N', M_i) is live and bounded.

3   The Notion of a Causal Behavioural Profile

This section introduces causal behavioural profiles. They are based on the notion of behavioural profiles, which we recall in Section 3.1. We introduced these profiles in an earlier work [13] to reason on execution ordering constraints only. Thus, optionality of transition execution or causality between transitions is not captured. These aspects are addressed by the novel concept of a causal behavioural profile introduced in Section 3.2. Section 3.3 discusses our concepts in the light of existing behavioural models defined for Petri nets. Finally, we discuss the application of causal behavioural profiles for consistency checking in Section 3.4.

3.1   Execution Order Constraints: The Behavioural Profile

Behavioural profiles aim at capturing behavioural aspects in terms of order constraints of a process in a fine-grained manner [13]. They are grounded on the set of possible firing sequences of a WF-system and the notion of *weak order*.

Definition 4 (Weak Order). Let (N, M_i) be a WF-system. A pair (x, y) is in the *weak order relation* $\succ \subseteq T \times T$, iff there exists a firing sequence $\sigma = t_1, \ldots, t_n$ with $(N, M_i)[\sigma\rangle$, $j \in \{1, \ldots, n-1\}$, $j < k \le n$, for which holds $t_j = x$ and $t_k = y$.

Thus, 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. Depending on how two activities of a process model are related by weak order, we define three relations forming the behavioural profile.

Definition 5 (Behavioural Profile). Let (N, M_i) be a WF-system. A pair $(x, y) \in (T \times T)$ is in at most one of the following 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 $y \succ x$.

Given a set $T' \subseteq T$, the set of all relations $BP_{T'} = \{\rightsquigarrow, +, ||\}$ defined over $T' \times T'$ is the *behavioural profile* of (N, M_i) for T'.

Computing the behavioural profile for all transitions of the system (a) in Fig. 1, for instance, it holds $C \rightsquigarrow E$ as there exists no firing sequence, such that E occurs before C. However, strict order does not imply the actual occurrence. That is, there are firing sequences containing only one of the two transitions, or even none of them. It holds $D + E$ as both transitions will never occur in a single firing sequence and $B||G$ as both transitions can occur in any order. Note that the three relations are mutually exclusive and (together with *reversed* strict order) partition the Cartesian product of transitions over which they are defined [13]. With respect to itself, a transition is either in the exclusive relation (if it can occur at most once, e.g., $D + D$) or in the interleaving order relation (if it can occur more than once, e.g., $B||B$).

3.2   Occurrence Constraints: The Causal Behavioural Profile

Behavioural profiles as introduced above relate pairs of transitions according to their *order* of potential occurrence. It is important to see that for the case of validating a workflow implementation against a process model specification, the relation associating corresponding transitions of both models to each other is typically partial. That is, certain transitions of one model are without counterpart in the other model, cf., Fig. 1. As

(a)

(b)

Fig. 2. Optionality

a consequence, information on ordering constraints is not sufficient to draw conclusions on optionality and causality of transition occurrences.

Optionality of a transition is given, if there is a firing sequence leading from the initial to the final marking of the system that does not contain the transition. Optionality can be lifted from single transitions to sets of transitions. A set of transitions is considered to be jointly optional, if any firing sequence from the initial to the final marking contains all or none of the transitions. As illustrated by Fig. 2(a) and Fig. 2(b) this property cannot be derived from the knowledge about optionality of single transitions. In both systems, B and C are optional, but only in Fig. 2(b) the set $\{B, C\}$ is jointly optional.

Closely related to optionality is *causality*, which requires that one transition can only occur after the occurrence of another transition. Thus, causality comprises two aspects, a certain *order* of occurrences and a *causal coupling* of occurrences. While the former is addressed by the behavioural profile in terms of the strict order relation, the latter is not captured. For instance, B is a cause of C in Fig. 2(b), but not in Fig. 2(a). Note that two transitions in interleaving order cannot show causality according to our definition. For both systems in Fig. 3, it holds $B||C$, as there

(a)

(b)

Fig. 3. No causality for transitions (B,C) in a cycle

is no distinct order relation between *all* occurrences of both transitions. Thus,

interleaving order is interpreted as the absence of any dependency regarding the order of occurrence. Thus, it is reasonable to define causality as a dependency between *all* occurrences of two transitions, instead of considering causal dependencies between their *single* occurrences (cf., *response/leads-to* dependencies [14]). There is no causality between B and C in either system in Fig. 3.

In order to cope with the aforementioned aspects, we introduce the co-occurrence relation and the causal behavioural profile. Two transitions are co-occurring, if any firing sequence from the initial to the final marking that contains the first transition contains also the second transition.

Definition 6 (Causal Behavioural Profile). Let (N, M_i) be a WF-system.
- A pair $(x, y) \in (T \times T)$ is in the *co-occurrence relation* $\gg$, if for all firing sequences σ with $(N, M_i)[\sigma\rangle(N, M_o)$, it holds $x \in \sigma \Rightarrow y \in \sigma$.
- Given a set $T' \subseteq T$, the set of all relations $CBP_{T'} = \{\rightsquigarrow, +, ||, \gg\}$ defined over $T' \times T'$ is the *causal behavioural profile* of (N, M_i) for T'.

Trivially, it holds $t \gg t$ for all $t \in T$. We derive optionality and causality as follows. A single transition $t \in T$ is optional, if $t_i \not\gg t$ for some $t_i \in i\bullet$ with i as the initial place. A set $T_1 \subseteq T$ of transitions is optional, if all transitions themselves are optional and they are pairwise co-occurring to each other $((T_1 \times T_1) \subseteq \gg)$.

Further on, there is a causal dependency between two transitions $t_1, t_2 \in T$, if they are in strict order ($t_1 \rightsquigarrow t_2$) and occurrence of the first implies occurrence of the second ($t_1 \gg t_2$). Note that, in contrast to the behavioural profile, the *causal* behavioural profile differs for both systems in Fig. 2.

3.3   Relation to Existing Behavioural Models

There is a large body of research on behavioural relations for formal models specifying dynamic systems in general, and for Petri nets in particular. Focussing on the order of occurrence, the relations proposed in [15] for workflow mining are close to our relations, yet different. We base our definitions on the notion of an *indirect* weak order dependency, whereas the relations in [15] are grounded on a *direct* sequential order. As a result, the notion of exclusiveness is restricted to *'pairs of transitions that never follow each other directly'* [15], whereas we capture exclusiveness for transitions that might occur at different stages of a firing sequence. While the notion of direct sequential order is appropriate for workflow mining, it leads to undesired effects in our setting. Consider, for instance, transitions G and K of model (a) in Fig. 1. They are exclusive according to the relations proposed in [15], whereas their counterparts in model (b) are in a sequential order. The behavioural profile, in turn, yields equal relations in both models. The respective transitions are in strict order in both models, (a) and (b).

Obviously, the well-known notions of *conflicting* and *concurrent* transitions are related to our *observed* relations as well. In a sound free-choice WF-system, two transitions in conflict, which are not part of a common control flow cycle will be exclusive in the behavioural profile. This follows from Lemma 3 in [13] and the fact that sound free-choice WF-systems are safe (a place carries at most one token

in all markings, cf., Lemma 1 in [24]). Similarly, all transitions that are enabled concurrently in some reachable marking (cf., the concurrency relation [16]) are in interleaving order in the behavioural profile.

In order to cope with concurrency and the interleaving problem, the *unfolding* of a Petri net (or its prefix, respectively) might be exploited for behaviour analysis [17,18]. That is, a *true concurrent* model is created in which a transition (i.e., an *event*) corresponds to a certain *occurrence* of a transition in the original net. Events can be related as being in a *weak causal predecessor*, *conflict*, or *concurrency* relation. While these relations resemble the relations of our casual behavioural profile, they are defined for transition occurrences instead of transitions. Thus, we might derive our relations by lifting these relations to the level of transitions again. For instance, if all events representing two transitions are in conflict in the (potentially infinite) unfolding, both transitions are exclusive according to the behavioural profile. However, an algorithm for the derivation of causal behavioural profiles from the prefix of an unfolding is beyond the scope of this paper. Usage of unfoldings is also inappropriate w.r.t. the class of systems we address in this paper, as the construction of unfoldings is computationally much harder than the approach introduced in the remainder of this paper.

With respect to common notions of behavioural equivalence, we see that two WF-systems with equal causal profiles are not necessarily trace equivalent. For instance, both systems in Fig. 3 have the same causal profile, whereas they are not trace equivalent. Evidently, the same holds true for bisimulation equivalences, as the profile neglects the branching structure of a system. However, it is easy to see that trace equivalence of two WF-systems implies equivalence of their causal behavioural profiles for all transitions, as all behavioural relations formulate statements about the existence of firing sequences.

3.4   Application of Causal Behavioural Profiles

We motivated the definition of causal behavioural profiles with the need for a notion of behavioural consistency that enables analysis of related process models in an efficient manner. Under the assumption of an alignment relation between transitions of two WF-systems, we define a degree of consistency as follows.

Definition 7 (Degree of Consistency). Let (N_1, M_{i_1}) and (N_2, M_{i_2}) be two WF-systems and $\sim \subseteq T_1 \times T_2$ a correspondence relation with $\sim \neq \emptyset$.

- The set $T_1^\sim = \{t_1 \in T_1 \mid \exists\, t_2 \in T_2 \, [\, t_1 \sim t_2 \,]\}$ contains all *aligned transitions* of (N_1, M_{i_1}). $T_2^\sim$ is defined analogously.
- With $\mathcal{R}_1$ and $\mathcal{R}_2$ as the relations of the causal behavioural profile for the WF-systems, the set $CT_1^\sim \subseteq (T_1^\sim \times T_1^\sim)$ contains all *consistent transition pairs* (t_x, t_y), such that
    - if $t_x = t_y$, then $\forall\, t_s \in T_2^\sim$ with $t_x \sim t_s$ it holds $t_x \mathcal{R}_1 t_x \Rightarrow t_s \mathcal{R}_2 t_s$,
    - if $t_x \neq t_y$, then $\forall\, t_s, t_t \in T_2^\sim$ with $t_s \neq t_t$, $t_x \sim t_s$, and $t_y \sim t_t$ it holds either $t_x \mathcal{R}_1 t_y \Rightarrow t_s \mathcal{R}_2 t_t$ or $t_x \sim t_t$ and $t_y \sim t_s$.
  The set $CT_2^\sim$ is defined analogously.
- The *degree of consistency* of $\sim$ is defined as $\mathcal{D}^\sim = \frac{|CT_1^\sim| + |CT_2^\sim|}{|(T_1^\sim \times T_1^\sim)| + |(T_2^\sim \times T_2^\sim)|}$.

The general idea behind this degree can be summarised as follows. For each pair of transitions, for which there are corresponding transitions in the other model, we check whether they share the same constraints. Since there can be complex 1:n correspondences as in Fig. 1, we have to count these correspondences from the perspective of each model. Applying this degree to the scenario in Fig. 1, we see that the order of potential occurrence is preserved for all aligned transitions. However, transition (A) is mandatory in model (a), whereas its counterparts are optional in model (b). Consequently, causality between transition (A) and, for instance, transition (K) is not preserved in model (b) either, which is taken into account in the causal behavioural profile. For our example, the degree of consistency is $\mathcal{D}^{\sim} = \frac{28+27}{36+36} \approx 0.76$, as both models (a) and (b) contain six transitions with correspondences yielding 36 transition pairs in the profile, while the profile relations are preserved for 28 (or 27, respectively) pairs.

The degree of consistency as defined above shows the characteristics of a semimetric for the comparison of two causal behavioural profiles. That is, the degree of consistency is non-negative and symmetric measure that equals one (or zero if it is subtracted from one, respectively), if and only if both profiles are equal. For the assessment of two profiles, however, the degree of consistency is not a metric as it does not satisfy the triangle inequality. That is due to the fact that the degree is a criterion for the quality of an alignment, i.e., a set of a correspondences. Hence, it is normalised by the size (the number of transitions) of the alignment but independent of the size of the respective WF-systems and, therefore, causal behavioural profiles. Still, we see that the relations of the causal behavioural profile are transitive in the sense that equal relations between a first and a second model, and the second and a third model imply the equivalence for the relations between the first and the third model. Thus, triangle inequality holds for the comparison of the degree of consistency of different alignments when considering solely those pairs of transitions that are part of all alignments.

For our proposal of assessing the consistency between business process models and their implementation as a workflow model, we got positive feedback from process analysts. Currently, we are evaluating the results of an empirical study that relates our degree of consistency to the consistency perception of process experts in a broader setting. Here, preliminary findings confirm a good approximation of perceived consistency by our degree. Clearly, there is a need for a multitude of consistency criteria in order to be able to graduate consistency requirements for a concrete setting. Nevertheless, an interval scale and efficient computation methods have to be seen as core requirements on such notions.

It is worth to mention that we already showed how behavioural profiles can be applied to support change propagation between related process models [19].

4   Graph Decomposition Techniques for WF-Systems

First, Section 4.1 introduces the Refined Process Structure Tree (RPST), a structural decomposition technique for workflow graphs. Second, Section 4.2 enriches the RPST for WF-systems with behavioural annotations.
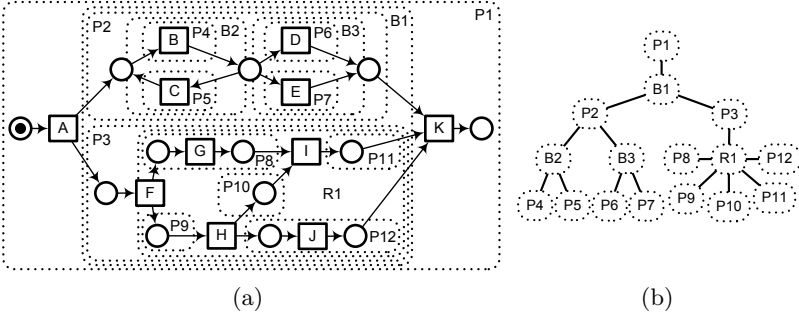
(a)   (b)

Fig. 4. (a) A WF-system and its canonical fragments, (b) the RPST of (a)

4.1 The Refined Process Structure Tree

The RPST [20,21] is a technique for detecting the structure of a workflow graph. A workflow graph can be parsed into a hierarchy of *fragments* with a single entry and a single exit, such that the RPST is a containment hierarchy of *canonical* fragments of the graph. The RPST is unique for a given workflow graph and can be computed in linear time [20,21]. Although the RPST has been introduced for workflow graphs, the technique can be applied to other graph based behavioural models such as WF-systems in a straight-forward manner. Basic terms of the RPST are defined for WF-nets as follows.

Definition 8 (Edges, Entry, Exit, Canonical Fragment).
Let $N = (P, T, F)$ be a WF-net.
- For a node $x \in X$ of a net $N = (P, T, F)$, $in_N(x) = \{(n, x) \mid n \in \bullet x\}$ are its *incoming edges* and $out_N(x) = \{(x, n) \mid n \in x \bullet\}$ are its *outgoing edges*.
- A node $x \in X'$ of a connected subnet $N' = (P', T', F')$ of a net N is a *boundary node*, if $\exists\, e \in in_N(x) \cup out_N(x)\ [\ e \notin F'\ ]$. If x is a boundary node, it is an *entry* of N', if $in_N(x) \cap F' = \emptyset$ or $out_N(x) \subseteq F'$, or an *exit* of N', if $out_N(x) \cap F' = \emptyset$ or $in_N(x) \subseteq F'$.
- Any connected subnet ω of N, is a *fragment*, if it has exactly two boundary nodes, one entry and one exit denoted by $\omega_\triangleleft$ and $\omega_\triangleright$, respectively.
- A fragment is *place bordered* if its boundary nodes are places.
- A fragment is *transition bordered* if its boundary nodes are transitions.
- A fragment $\omega = (P_\omega, T_\omega, F_\omega)$ is *canonical* in a set of *all* fragments Σ of N, iff $\forall\, \gamma = (P_\gamma, T_\gamma, F_\gamma) \in \Sigma\ [\ \omega \neq \gamma \Rightarrow (F_\omega \cap F_\gamma = \emptyset) \vee (F_\omega \subset F_\gamma) \vee (F_\gamma \subset F_\omega)\ ]$.

Fig. 4 exemplifies the RPST for the WF-system from Fig. 1(a). Fig. 4(a) illustrates its canonical fragments, each of them formed by a set of edges enclosed in or intersecting the region with a dotted border. Fig. 4(b) provides an alternative view, where each node represents a canonical fragment and edges hint at containment relation of fragments. Observe that one obtains a tree structure—the RPST. For instance, fragment $B1$ has two boundary transitions: entry A and exit K, is contained in fragment $P1$, and contains fragments $P2$ and $P3$.

If the RPST is computed for a *normalized* workflow graph, i.e., a workflow graph that does not contain nodes with multiple incoming and multiple outgoing edges, each canonical fragment can be classified to one out of four structural

classes [21,22]: A *trivial* (*T*) fragment consists of a single edge. A *polygon* (*P*) represents a sequence of nodes (fragments). A *bond* (*B*) stands for a collection of fragments that share common boundary nodes. Any other fragment is a *rigid* (*R*). Note that we use frag-

Fig. 5. Node-splitting

ment names that hint at their structural class, e.g., $R1$ is a rigid fragment. Every workflow graph can be normalized by performing a node-splitting pre-processing step, illustrated for WF-nets in Fig. 5. The WF-system in Fig. 4(a) is normalized.
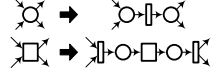
4.2   An Annotated RPST: The WF-Tree

The structural patterns derived by the RPST can be related to behavioural properties of the underlying WF-system. In this section, we concretise RPST fragments by annotating them with behavioural characteristics. We refer to the containment hierarchy of annotated canonical fragments of a WF-system as the RPST with behavioural annotations, or *WF-tree* for short. The WF-tree is defined for sound free-choice WF-systems. It is well-known that the free-choice and soundness properties are required to derive behavioural statements from the structure of a system, as both together imply a tight coupling of syntax and semantics (cf., [23,24]).

Definition 9 (WF-Tree). Let (N, M_i) be a sound free-choice WF-system. The RPST with behavioural annotations, the *WF-Tree* of N, is a tuple $\mathcal{T}_N = (\Omega, \chi, t, b)$, where:

○ Ω is a set of *all* canonical fragments of N,
○ $\chi : \Omega \to \mathcal{P}(\Omega)$ is a function that assigns to fragment its child fragments,
○ $t : \Omega \to \{T, P, B, R\}$ is a function that assigns a type to a fragment,
○ $b : \Omega_B \to \{B_\circ, B_\diamond, L\}$, $\Omega_B = \{\omega \in \Omega \mid t(\omega) = B\}$, is a function that assigns a refined type to a bond fragment, where $B_\circ$, $B_\diamond$, and L types stand for place bordered, transition bordered, and loop bonds, respectively.

Further on, we define auxiliary concepts for the WF-tree.

Definition 10 (Parent, Child, Root, Ancestor, Descendant, LCA, Path). Let $\mathcal{T}_N = (\Omega, \chi, t, b)$ be the WF-tree.

○ For any fragment $\omega \in \Omega$, ω is a *parent* of γ and γ is a *child* of ω, if $\gamma \in \chi(\omega)$. By χ^+ we denote the transitive closure of χ.
○ The fragment $\omega \in \Omega$ is a *root* of $\mathcal{T}$, denoted by ω_r, if it has no parent.
○ The partial function $\rho : \Omega \setminus \{\omega_r\} \to \Omega$ assigns parents to fragments.
○ For any fragment $\omega \in \Omega$, ω is an *ancestor* of ϑ and ϑ is a *descendant* of ω, if $\vartheta \in \chi^+(\omega)$.
○ For any two fragments $\{\omega, \gamma\} \in \Omega$ their *lowest common ancestor* (LCA), denoted by $lca(\omega, \gamma)$, is the shared ancestor of ω and γ that is located farthest from the root of the WF-tree. By definition, $lca(\omega, \omega) = \omega$.
○ For any fragment $\omega_0 \in \Omega$ and its descendant $\omega_n \in \Omega$, a *downward path* from ω_0 to ω_n, denoted by $\pi_\mathcal{T}(\omega_0, \omega_n)$, is a sequence $(\omega_0, \omega_1, \ldots, \omega_n)$, such that ω_i is a parent of ω_{i+1} for all $i \in \mathbb{N}_0$. In addition, $\pi_\mathcal{T}(\omega_0, \omega_n, i) = \omega_i$ and $\pi_\mathcal{T}\{\omega_0, \omega_n\}$ is a set which contains all fragments of $\pi_\mathcal{T}(\omega_0, \omega_n)$.

Fig. 6 shows the WF-tree of the WF-system from
Fig. 4(a). Note that trivial fragments are not vi-
sualised. The WF-tree is isomorphic to the RPST
of the WF-system, cf., Fig. 4(b). Given the RPST,
adding the behavioural annotation is a trivial task
for most fragments, except of the following cases:
A bond fragment $\gamma = (P_\gamma, T_\gamma, F_\gamma) \in dom(b)$
of $\mathcal{T}_N = (\Omega, \chi, t, b)$ is assigned the L type, if
$\gamma_\lhd = \omega_\rhd$ with ω being a child of γ. Otherwise,
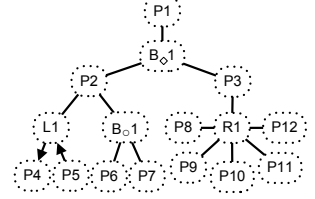$b(\gamma) = B_\circ$ if $\gamma_\lhd \in P_\gamma$, or $b(\gamma) = B_\diamond$ if $\gamma_\lhd \in T_\gamma$.

Fig. 6. The WF-tree

Children of a polygon fragment are arranged with respect to their execution
order. A partial function $order : \Omega' \to \mathbb{N}_0$, $\Omega' = \{\omega \in \Omega \setminus \{\omega_r\} \mid t(\rho(\omega)) =$
$P\}$ assigns to children of polygon fragments their respective order positions;
$order(\omega) = 0$, if $\omega_\lhd = \gamma_\lhd$ with $\gamma = \rho(\omega)$ being the parent, and $order(\omega) = i$,
$i \in \mathbb{N}$, if $\omega_\lhd = \vartheta_\rhd$ for some $\vartheta \in \Omega$, such that $order(\vartheta) = i - 1$. Observe that the
orders of two nodes are only comparable if they share a common parent. For
instance, in Fig. 6, $order(L1) = 1$ and $order(B_\circ 1) = 2$. This means that fragment
$L1$ is always executed before fragment $B_\circ 1$ inside of polygon $P2$. The layout of
child fragments of polygon fragments hints at their order relations.

Children of a loop fragment are classified as *forward* ($\Rightarrow$) or *backward* ($\Leftarrow$).
A partial function $\ell : \Omega'' \to \{\Leftarrow, \Rightarrow\}$ with $\Omega'' = \{\omega \in \Omega \setminus \{\omega_r\} \mid b(\rho(\omega)) = L\}$
assigns an orientation to children of loop fragments. $\ell(\omega) = \Rightarrow$ if $\omega_\lhd = \gamma_\lhd$ with
$\gamma = \rho(\omega)$, otherwise $\ell(\omega) = \Leftarrow$. In Fig. 6, $P4$ and $P5$ are forward and backward
fragments, respectively, which is visualised by the direction of edges.

We introduce two lemmas that prove the completeness of the codomain of
function b by showing that a bond fragment is either place or transition bordered,
and that each loop fragment is place bordered. Note that a rigid fragment bordered
with a place and a transition can still be free-choice and sound (see [25]).

Lemma 1. *Let $\mathcal{T}_N = (\Omega, \chi, t, b)$ be the WF-tree of a sound free-choice WF-
system (N, M_i), $N = (P, T, F)$. No bond fragment $\omega \in \Omega$, $t(\omega) = B$, has $\{p, t\}$
boundary nodes, where $p \in P$ and $t \in T$.*

Proof. Assume ω is a bond fragment with $\{p, t\}$ boundary nodes. There exists a
circuit Γ in a short-circuit net of N that contains $\{p, t\}$. Let Γ_ω be a subpath
of Γ inside ω. There exists a child fragment γ of ω that contains Γ_ω. A bond
fragment has $k \geq 2$ child fragments, cf., [21,22]. Let ϑ be a child of ω, $\vartheta \neq \gamma$. We
distinguish two cases:

- Let H be a path from p to t contained in ϑ. H is a PT-handle of Γ. In a live
  and bounded free-choice system, H is bridged to Γ_ω through a TP-bridge K,
  cf., Proposition 4.2 in [26]. This implies that $\vartheta = \gamma$; otherwise bond fragment
  ω contains path K that is not inside of a single child fragment, cf., [22,21].
  Thus, ω has a single child fragment, a contradiction with the assumption of
  ω being a bond fragment.
- Let H be a path from t to p contained in ϑ. H is a TP-handle of Γ. In a live
  and bounded free-choice system, no circuit has TP-handles, cf., Proposition 4.1
  in [26], which yields a contradiction with our assumptions. □

Lemma 2. *Let $\mathcal{T}_N = (\Omega, \chi, t, b)$ be the WF-tree of a sound free-choice WF-system, (N, M_i), $N = (P, T, F)$. A loop fragment $\omega = (P_\omega, T_\omega, F_\omega) \in \Omega$, $b(\omega) = L$, is place bordered, i.e., $\{\omega_\triangleleft, \omega_\triangleright\} \in P$.*

Proof. Because of Lemma 1, ω is either place or transition bordered. Assume ω is transition bordered. There exists place p such that $p \in \bullet\omega_\triangleleft \cap P_\omega$, $M_i(p) = 0$. Transition $\omega_\triangleleft$ is enabled if there exists a marking $M \in [(N, M_i)\rangle$ with $M(p) > 0$. Since ω is a connected subnet, for all $t \in T_\omega \setminus \{\omega_\triangleleft, \omega_\triangleright\}$ all edges are in ω, i.e., $(in_N(t) \cup out_N(t)) \subseteq F_\omega$. Thus, every path from i to p visits $\omega_\triangleleft$. Thus, $M(p) > 0$ is only possible, if $\omega_\triangleleft$ has fired before. We reached a contradiction. Transition $\omega_\triangleleft$ is never enabled and N is not live, and hence, not sound. Since any loop fragment is not transition bordered, it is place bordered (Lemma 1). □

For sound free-choice WF-systems, the WF-tree can be derived efficiently.

Corollary 1. *The following problem can be solved in linear time. Given a sound free-choice WF-system, to compute its WF-tree.*

Proof. Given a workflow graph, its RPST can be computed in time linear to the number of edges of the graph [20,21]. The number of canonical fragments in the RPST is linear to the number of edges in the workflow graph [21,27,28]. Given the RPST of a WF-system, we iterate over all bond fragments and assign the behavioural annotations. Here, it suffices to check the type of the entry node, either a place or transition, and to determine whether the entry is also the exit of a child fragment. That can be decided in constant time for each fragment. Finally, child fragments of a polygon can be ordered in linear time. We introduce a hash function that returns a child fragment with the given node as an entry and iterate over the children of the polygon. □

5   Efficient Computation of Causal Behavioural Profiles

This section shows how a WF-tree is applied to compute the causal behavioural profile. Section 5.1 introduces the approach for transition pairs that do not require analysis of rigid fragments. Afterwards, we discuss analysis of rigid fragments in Section 5.2 and present experimental performance results in Section 5.3.

5.1   Computation without Analysis of Rigid Fragments

For the computation of the causal behavioural profile for a pair of transitions, we assume that each transition has one incoming and one outgoing flow arc. If this is not the case, we apply the pre-processing illustrated in Fig. 5, which preserves the behaviour of the system (cf., [29]) and, therefore, does not change the causal behavioural profile. Given a pre-processed WF-system (N, M_i) with $N = (P, T, F)$ and its WF-tree $\mathcal{T}_N = (\Omega, \chi, t, b)$, each transition $t \in T$ is a boundary node of at most two trivial fragments of $\mathcal{T}_N$. Thus, it suffices to show how the behavioural relations are determined for the entries of two trivial fragments.

Our computation is based on two elementary properties of free-choice sound WF-systems. If (N, M_i) is free-choice and sound, it is safe (cf., Lemma 1 in [24]), i.e., $\forall\, p \in P$, $M(p) < 2$ in all reachable markings. Thus, a single transition cannot be enabled concurrently with itself.
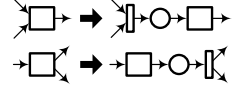

Fig. 7. Pre-processing

In addition, if (N, M_i) is free-choice and sound, the existence of a path $\pi_N(x, y)$ between places x and y implies the existence of a firing sequence containing all transitions on $\pi_N(x, y)$ (cf., Lemma 4.2 in [23]). While the implication actually requires the marking $M_y = [y]$ to be a home marking (a marking reachable from every marking that is reachable from the initial state), it can be lifted to all home markings with $M_y(y) > 0$. Due to soundness of the system (N, M_i), the short-circuit system (N', M_i) is live and bounded, such that all markings $M \in [N, M_i\rangle$ are home markings in (N', M_i). Thus, all markings $M_y(y) > 0$ are reachable from markings $M_x(x) > 0$, if $M_y, M_x \in [N', M_i\rangle$.

In the absence of rigid fragments on certain paths, the execution ordering relations and the co-occurrence relation of the causal behavioural profile are computed as follows. Proofs of both propositions can be found in [6].

Proposition 1. Let $\mathcal{T}_N = (\Omega, \chi, t, b)$ be the WF-tree and $\alpha, \beta \in \Omega$ two trivial fragments. Let $\gamma = lca(\alpha, \beta)$ and $\forall\, \omega \in \pi_\mathcal{T}\{\omega_r, \gamma\}\ [\ t(\omega) \neq R\ ]$.
 1. If $\alpha = \beta$, then $\alpha_\lhd || \beta_\lhd$, iff $\exists\, \omega \in \pi_\mathcal{T}\{\omega_r, \gamma\}\ [\ b(\omega) = L\ ]$. Otherwise, $\alpha_\lhd + \beta_\lhd$.
 2. If $\alpha \neq \beta$,
    - $\alpha_\lhd \rightsquigarrow \beta_\lhd$, iff (1) $t(\gamma) = P \wedge order(\pi_\mathcal{T}(\gamma, \alpha, 1)) < order(\pi_\mathcal{T}(\gamma, \beta, 1))$, and (2) $\forall\, \omega \in \pi_\mathcal{T}\{\omega_r, \gamma\}\ [\ b(\omega) \neq L\ ]$.
    - $\alpha_\lhd + \beta_\lhd$, iff (1) $b(\gamma) = B_\circ$, and (2) $\forall\, \omega \in \pi_\mathcal{T}\{\omega_r, \gamma\}\ [\ b(\omega) \neq L\ ]$.
    - $\alpha_\lhd || \beta_\lhd$, iff (1) $b(\gamma) \in \{B_\diamond, L\}$, or (2) $\exists\, \omega \in \pi_\mathcal{T}\{\omega_r, \gamma\}\ [\ b(\omega) = L\ ]$.

Proposition 2. Let $\mathcal{T}_N = (\Omega, \chi, t, b)$ be the WF-tree and $\alpha, \beta \in \Omega$ two trivial fragments, $\alpha \neq \beta$. Let $\gamma = lca(\alpha, \beta)$, $\Pi = \pi_\mathcal{T}\{\gamma, \beta\}$, and $\forall\, \omega \in \Pi\ [\ t(\omega) \neq R\ ]$. Then, $\alpha_\lhd \gg \beta_\lhd$, iff for all $\omega \in (\Pi \setminus \{\beta\})$ one of the following conditions holds:
 1. $t(\omega) = P$,
 2. $t(\omega) = B$ and $b(\omega) = B_\diamond$, or
 3. $t(\omega) = B$, $b(\omega) = L$, and with $\Theta = \{\vartheta \in \chi(\omega) \mid \ell(\vartheta) ==\Rightarrow\}$ it holds $\forall\, \vartheta \in \Theta\ [\ \beta \in \chi^+(\vartheta)\ ]$.

We illustrate both propositions using our example from Fig. 4(a). For instance, transitions B and E are in strict order, $B \rightsquigarrow E$, as the LCA of the trivial fragments that have B and E as entries is the polygon fragment $P2$, cf., Fig. 4(b) and Fig. 6. Here, the *order* value for the child fragment of $P2$ containing B is lower than the one for the child fragment that contains E, while the path from the root of the tree $P1$ to $P2$, i.e., $\pi_\mathcal{T}(P1, P2)$, does not contain any loop fragment. It holds $D + E$ for transitions D and E due to the LCA being fragment $B3$ in Fig. 4(b) or $B_\circ 1$ in Fig. 6, respectively. The fragment $B_\circ 1$ is a place bordered bond and, again, the path $\pi_\mathcal{T}(P1, B_\circ 1)$ does not contain any loop fragments. Transitions B and C, in turn, are an example for interleaving order, $B || C$, as their LCA is fragment $B2$ in Fig. 4(b). This fragment corresponds to the loop type fragment $L1$ in Fig. 6. Derivation of the co-occurrence is illustrated using

transitions B and C. We see that the path from the respective LCA (i.e., $B2$ in Fig. 4(b), $L1$ in Fig. 6) to the trivial fragments having B and C as entries contains solely polygon fragments ($P4$ and $P5$, respectively). However, the LCA itself is a loop fragment, such that the orientation of its child fragments $P4$ and $P5$ needs to be considered. There is only one child with forward orientation, namely $P4$. It contains transition B. Therefore, we derive $C \gg B$, but $B \not\gg C$ according to Proposition 2.

Using these propositions, computation of the causal behavioural profile for a pair of transitions in a sound free-choice WF-system is very efficient.

Corollary 2. *The following problem can be solved in linear time.*
Given a sound free-choice WF-system (N, M_i) and its WF-tree $\mathcal{T}_N$, to compute the causal behavioural profile for a pair of transitions (a, b) if b is not contained in any rigid fragment.

Proof. Let a and b be two transitions and β be a trivial fragment of $\mathcal{T}_N$ with $b = \beta_\triangleleft$. Each of the behavioural relations, cf., propositions 1 and 2, requires analysis of fragments on a subpath from the root of $\mathcal{T}_N$ to β. The analysis of a single fragment is performed in constant time. In the worst case, the length of the subpath is linear in size to the number of fragments in $\mathcal{T}_N$. The number of fragments in $\mathcal{T}_N$ is linear to the number of flows in the WF-system [21,27,28].   □

5.2   Computation for Rigid Fragments

Given the WF-tree, the computation of the causal behavioural profile for two transitions a and b of a WF-system as introduced above assumes that there is no rigid fragment on the path from the root of the tree to b. If b is part of a rigid fragment, derivation of the behavioural relations is more costly.

In [13], we introduced a computation of the (non-causal) behavioural profile for all transitions in $O(n^3)$ time for sound free-choice WF-systems with n as the number of nodes. This approach, however, has the drawback that the behavioural profile cannot be calculated for a single pair of transitions, but solely for the Cartesian product of transitions leading to increased computational complexity. For the problem of this paper, this implies computational overhead as various transitions are irrelevant for consistency analysis. Not in all cases, such irrelevant transitions might be removed in a pre-processing step without changing semantics.

While for the behavioural profile computation in polynomial time complexity is possible for sound free-choice WF-systems, the co-occurrence relation of the causal behavioural profile imposes serious challenges. In the following, we show how this relation can be derived efficiently for three subclasses, namely sound workflow T- and S-systems, and sound free-choice WF-systems that are acyclic.

First, we need an auxiliary lemma for the relation between (forwards and backwards) conflict-free paths and the co-occurrence relation. As usual, given a WF-net $N = (P, T, F)$ a path $\pi_N(x_1, x_k)$ is *forwards conflict-free*, iff $x_i \in P$ implies $|x_i \bullet| = 1$ for $1 \le i < k$. The path $\pi_N(x_1, x_k)$ is *backwards conflict-free*, iff $x_i \in P$ implies $|\bullet x_i| = 1$ for $1 < i \le k$. The proof can be found in [6].

Lemma 3. *For two transitions x and y in a sound WF-system holds,*
- *if there is a forwards conflict-free path from x to y, then $x \gg y$.*
- *if there is a backwards conflict-free path from x to y, then $y \gg x$.*

The co-occurrence relation for sound workflow T-systems is derived as follows.

Lemma 4. *All pairs of transitions of a sound workflow T-system are in the co-occurrence relation.*

Proof. Let (N, M_i) be a sound workflow T-system. Let $i\bullet = \{t_i\}$ be the initial transition (there is only one due to the structure of T-systems). For any transition $t \in T$ any path $\pi_N(t_i, t)$ is forwards conflict-free. Thus, $t_i \gg t$ (Lemma 3). Consequently, all firing sequences starting with t_i imply the occurrence of every $t \in T$. Due to soundness, such firing sequences lead to the final marking M_o. Thus, all firing sequences σ with $(N, M_i)[\sigma\rangle(N, M_o)$ contain all transitions $t \in T$.   □

Regarding our example in Fig. 4(a), we see that Lemma 4 suffices to derive the co-occurrence relation for all pairs of transitions that can not be treated according to Proposition 2 introduced before as they are part of a rigid fragment. The subnet represented by fragment $R1$ in Fig. 4(b) and Fig. 6 is a T-Net, such that all transitions inside are pairwise co-occurring (e.g., $F \gg J$ and $J \gg F$). This knowledge, in turn, is used to derive co-occurrence for pairs of transitions, in which one transition is outside the rigid. For instance, we already know $D \gg K$, as the trivial fragment having transition K as entry is directly contained in fragment $P1$ (Proposition 2 can be applied to decide co-occurrence for D and K). As K is also the exit of the rigid fragment $R1$, it is co-occurring to all transitions inside $R1$. Thus, D is co-occurring to all these transitions, e.g., $D \gg H$.

For sound workflow S-systems, the co-occurrence relation can be traced back to the notion of dominators and post-dominators known from graph theory. For a WF-net $N = (P, T, F)$, i and o as its initial and final place, and two nodes $x, y \in X$, x is a dominator of y, iff for all paths $\pi_N(i, y)$ it holds $x \in \pi_N(i, y)$. x is a post-dominator of y, iff for all paths $\pi_N(y, o)$ it holds $x \in \pi_N(y, o)$.

Lemma 5. *For two transitions x and y of a sound workflow S-system holds, $x \gg y$, iff y is dominator or post-dominator of x.*

Proof. Let (N, M_i) be a sound workflow S-system and $x, y \in T$ two transitions. In a workflow S-system, every reachable marking $M \in [N, M_i\rangle$ marks exactly one place, as only i is marked initially and for all transitions $t \in T$ we know $|\bullet t| = 1 = |t \bullet|$. Therefore, for every firing sequence $\sigma = t_1, \ldots, t_n$ we know that there is a path $\pi_N(t_1, t_n)$ containing all transitions of σ in the respective order.
- $\Rightarrow$ Let y be a dominator or a post-dominator of x, assume $x \not\gg y$. If y is a dominator of x, then $y \in \pi_N(i, x)$ for every path $\pi_N(i, x)$. Thus, any firing sequence σ with $(N, M_i)[\sigma\rangle(N, M_1)$ with $(N, M_1)[x\rangle$ is required to contain y, i.e., $x \gg y$. If y is a post-dominator of x, the argument can be turned around. for all paths $\pi_N(x, o)$.

$\Leftarrow$ Let $x \gg y$ and assume that y is neither a dominator nor a post-dominator of x. $x \gg y$ implies that any firing sequence σ with $x \in \sigma$ and $(N, M_i)[\sigma\rangle(N, M_o)$ contains y as well. Thus, all paths $\pi_N(i, o)$ that contain x also contain y, i.e, y is a dominator (if y F^+ x) or post-dominator (if x F^+ y) of x. $\qquad \square$

For the more generic case of sound free-choice WF-systems that are acyclic, the co-occurrence relation can be traced back to the exclusiveness relation. Note that it is easy to see that two transitions that are exclusive to each other are not co-occurring. Therefore, this case is not considered in the following lemma.

Lemma 6. *In a sound free-choice WF-system holds, two transitions x and y that are not exclusive ($x \not{+} y$), while y is not part of a control flow cycle (y $\not{F^+}$ y) are co-occurring, if and only if, all transitions exclusive to y are exclusive to x.*

Again, the proof of Lemma 6 is detailed in [6]. Based thereon, computation of the causal behavioural profile is efficient for the respective system classes.

Corollary 3. *The following problem can be solved in $O(n^3)$ time with n as the number of nodes of the system. For a sound WF-system that is a T- or S-system, or free-choice and acyclic, to compute the causal behavioural profile for a pair of transitions.*

Proof. Given any sound free-choice WF-system, the relations of the behavioural profile can be computed in $O(n^3)$ time [13] (T- and S-systems are free-choice). The co-occurrence relation for the causal profile is set directly in case of a T-system (cf., Lemma 4). In case of an S-system, dominators and post-dominators are determined in linear time [30]. Based thereon, co-occurrence is decided based on Lemma 5. For the case of acyclic free-choice WF-systems, co-occurrence is traced back to exclusiveness according to Lemma 6. That requires an iteration over the Cartesian product of transitions, while for each pair all other transitions are analysed, which yields a time complexity of $O(n^3)$. Thus, overall time complexity is $O(n^3)$ with n as the number of nodes of the system. $\qquad \square$

5.3 Implementation & Experimental Results

In order to validate our approach of deriving behavioural characteristics, we implemented the computation of the causal behavioural profiles based on WF-trees and conducted an experiment using the SAP reference model [31]. This reference model describes the functionality of the SAP R/3 system and comprises 737 EPC models. From these models, we selected those that are non-trivial (more than one element), syntactically correct, free of deadlocks or livelocks (cf., [32]), and have unambiguous instantiation semantics (cf., [33]). We also normalised multiple start and end events, and replaced OR-split and OR-join connectors with AND connectors (which does not impact on the behavioural profile, but on the causal behavioural profile). For 493 EPC models, these pre-processing steps led to a model that could be transformed into a sound free-choice WF-system following on common EPC formalisations (eg., [9]).

In our experiment, we computed the (non-causal and causal) behavioural profiles for all transitions of all 493 WF-systems separately. We grouped the models

according to their size, i.e., the number of EPC nodes (the WF-systems are larger in size). Fig. 8 shows the average computation time for each model group in three experiment runs. First, we computed the behavioural profile using the approach introduced in [13] (BP-Net). Second, we derived the same pro-

Fig. 8. Computation time relative to the size of the EPC model (Intel Core 2 Duo, 1.2 GHz, 4 GB RAM, Java 1.6).

file using the WF-trees as introduced in this paper (BP-Tree). Third, we computed the causal behavioural profile (including co-occurrence) using WF-trees (CBP). Note that two WF-systems contained a rigid fragment. Both could be mapped to an S-system and, therefore, be handled as introduced in Section 5.2. To illustrate the extent to which the models of our collection suffer from the state explosion problem [34], Fig. 8 shows the average computation time for a n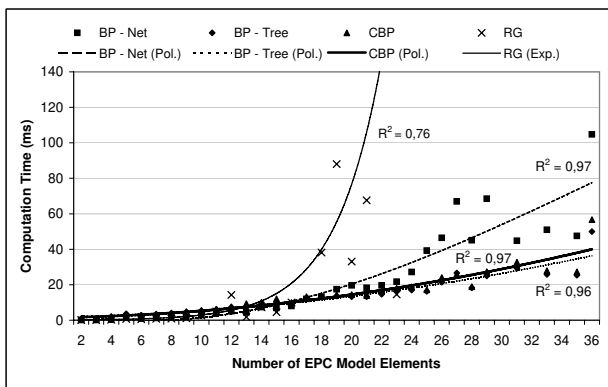aive creation of the reachability graph (RG). While all reachability graphs are finite (due to soundness of the WF-systems), computation takes up to tens of seconds. For all four computations, Fig. 8 depicts the polynomial (or exponential for RG) least squares regression.

We see that the usage of WF-trees as introduced in this paper, speeds up the computation of the behavioural profile significantly compared to the existing approach. In addition, the overhead implied by our extension of the behavioural profile yielding the causal behavioural profile is negligible. Moreover, any trace-based consistency assessment would have to explore the state space and, therefore, deal with the same computational complexity as the creation of the reachability graphs. Despite the availability of state space reduction techniques, the applicability of such an approach for real-world scenarios seems questionable.

6   Related Work

Clearly, our work relates to other behavioural models that have been defined for Petri nets. While we discussed causal behavioural profiles in the light of relations proposed for workflow mining [15], the well-known concurrency relation [16], and Petri net unfoldings [17,18] already in Section 3.3, their relation to common notions of behavioural equivalence deserves further explanation.

When applied in the context of model refinement and adaptation, the multitude of equivalence criteria from the linear time – branching time spectrum [35,5] has three major drawbacks. First and foremost, these notions yield a true or false answer, which has been criticised in [36]. Such notions cannot be applied to assess the amount of potential behavioural deviation. Second, it is well-known

that interleaving equivalences are not invariant under *forgetful refinements* of activities [37], i.e., projection of activities. However, our initial example shows that projections are a substantial part of refining and adapting a process model towards a workflow model. These phenomena, in turn, can be quantified using the causal behavioural profile. Moreover, work on equivalence-preserving refinements for Petri nets, refer to [38] for a thorough survey, illustrates that common notions of equivalence are preserved solely under certain refinement operators. Similarly, work on net morphisms [39] and behaviour inheritance [40,41] shows that any extension of a net has to be done in a structured manner in order to preserve common equivalences. Third, notions of behavioural equivalence are computationally hard, which precludes an application for large scale industrial process models. As discussed in Section 3.3, equivalence of causal behavioural profiles is weaker than trace equivalence in order to compensate for computational efficiency.

Relations similar to those of the behavioural profile have been proposed to reason on the consistency of hardware specifications and requirements imposed by operational modules [42]. To this end, transitions of a Petri net can be classified as being sequential or parallel depending on whether there is an order between all their occurrences in all traces. In addition, these relations along with an exclusiveness relation are also defined for operations of a programming language. The authors of [42] derive these relations from the parse of an acyclic program. This, in turn, is very similar to our approach of leveraging the RPST decomposition technique. Still, the causal behavioural profile comprises further details and our approach is also applicable for cyclic nets.

The degree to which causal behavioural profiles of two related Petri nets are preserved can be used as a behavioural similarity measure. Therefore, work on causal footprints as a behavioural abstraction for determining the similarity between processes [43] or on a trace-based similarity metric for process mining [36] is related. Further references on behavioural similarity can be found in [44].

Related work includes further applications of the tree-based decomposition for behavioural models, e.g., model transformation [20] or model abstraction [22].

7   Conclusions

In this paper, we addressed the problem of finding a behavioural consistency notion that is weaker than existing notions of behavioural equivalence, but can be computed efficiently. Our contribution is the definition of a causal behavioural profile that captures essential behavioural characteristics of a process. Further on, we showed the efficient computation of these profile for sound free-choice workflow systems using structural decomposition techniques under the assumption that unstructured net fragments are acyclic or can be traced back to S- or T-nets. Note that this assumption still allows the system to be cyclic, either in a structured way (bond loop fragment) or in an unstructured way (rigid fragment is a cyclic S-net). We also demonstrated the efficiency by presenting experimental results. The low polynomial complexity of our algorithms opens reasoning on behavioural consistency to industrial applications where trace based approaches do not scale.

In future research, we aim at techniques for computing causal behavioural profiles for a broader class of behavioural models, that is, systems that do not meet our assumptions on free-choiceness, soundness, and the characteristics of rigid fragments. While we addressed the suitability of our degree of consistency in a recent survey, further empirical investigations on the human perception of behavioural consistency are needed and will be tackled in future work.

References

1. Davies, I., Green, P., Rosemann, M., Indulska, M., Gallo, S.: How do practitioners use conceptual modeling in practice? Data Knowl. Eng. **58**(3) (2006) 358–380
2. Nejati, S., Sabetzadeh, M., Chechik, M., Easterbrook, S.M., Zave, P.: Matching and merging of statecharts specifications. In: ICSE, IEEE CS (2007) 54–64
3. Dijkman, R., Dumas, M., García-Bauelos, L., Kääriky, R.: Aligning business process models. In: EDOC, IEEE CS (2009)
4. Rahm, E., Bernstein, P.A.: A survey of approaches to automatic schema matching. VLDB Journal **10**(4) (2001) 334–350
5. Glabbeek, R.: The Linear Time – Brancing Time Spectrum I. The semantics of concrete, sequential processes. In: Handbook of Process Algebra. Elsevier (2001)
6. Weidlich, M., Polyvyanyy, A., Mendling, J., Weske, M.: Efficient Computation of Causal Behavioural Profiles using Structural Decomposition. Technical report 10-2010, Hasso Plattner Institute (January 2010) `http://bpt.hpi.uni-potsdam.de/pub/Public/MatthiasWeidlich/cbp_report.pdf`
7. Aalst, W.: The application of Petri nets to workflow management. Journal of Circuits, Systems, and Computers **8**(1) (1998) 21–66
8. Lohmann, N.: A feature-complete Petri net semantics for WS-BPEL 2.0. In: WS-FM. Volume 4937 of LNCS. (2008) 77–91
9. Kindler, E.: On the semantics of EPCs: A framework for resolving the vicious circle. In: BPM. Volume 3080 of LNCS. (2004) 82–97
10. Eshuis, R., Wieringa, R.: Tool support for verifying UML activity diagrams. IEEE Trans. Software Eng. **30**(7) (2004) 437–447
11. Desel, J., Esparza, J.: Free Choice Petri Nets. Cambridge University Press (1995)
12. Aalst, W.: Verification of workflow nets. In: ICATPN. Volume 1248 of LNCS. (1997) 407–426
13. Weidlich, M., Mendling, J., Weske, M.: Computation of behavioural profiles of process models. Technical report 08-2009, Hasso Plattner Institute (June 2009)
14. Dwyer, M.B., Avrunin, G.S., Corbett, J.C.: Property specification patterns for finite-state verification. In Ardis, M.A., Atlee, J.M., eds.: FMSP, ACM (1998) 7–15
15. 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
16. Kovalyov, A., Esparza, J.: A polynomial algorithm to compute the concurrency relation of free-choice signal transition graphs. In: WODES, The Institution of Electrical Engineers (1996) 1–6
17. McMillan, K.L.: A technique of state space search based on unfolding. Formal Methods in System Design **6**(1) (1995) 45–65
18. Esparza, J., Heljanko, K.: Unfoldings: a partial-order approach to model checking. Springer (2008)
19. Weidlich, M., Weske, M., Mendling, J.: Change propagation in process models using behavioural profiles. In: SCC, IEEE CS (2009)

20. Vanhatalo, J., Völzer, H., Koehler, J.: The refined process structure tree. In: BPM. Volume 5240 of LNCS. (2008) 100–115
21. Polyvyanyy, A., Vanhatalo, J., Völzer, H.: Simplified computation and generalization of the refined process structure tree. Technical Report RZ 3745, IBM (2009)
22. Polyvyanyy, A., Smirnov, S., Weske, M.: The triconnected abstraction of process models. In: BPM. Volume 5701 of LNCS. (2009) 229–244
23. Kiepuszewski, B., Hofstede, A., Aalst, W.: Fundamentals of control flow in workflows. Acta Inf. **39**(3) (2003) 143–209
24. Aalst, W.: Workflow verification: Finding control-flow errors using petri-net-based techniques. In: BPM. Volume 1806 of LNCS. (2000) 161–183
25. Aalst, W., Hirnschall, A., Verbeek, H.: An alternative way to analyze workflow graphs. In: CAiSE. Volume 2348 of LNCS. (2002) 535–552
26. Esparza, J., Silva, M.: Circuits, handles, bridges and nets. In: Applications and Theory of Petri Nets. Volume 483 of LNCS. (1989) 210–242
27. Gutwenger, C., Mutzel, P.: A linear time implementation of SPQR-trees. In: Graph Drawing. Volume 1984 of LNCS. (2001) 77–90
28. Battista, G.D., Tamassia, R.: On-line maintenance of triconnected components with SPQR-trees. Algorithmica **15**(4) (1996) 302–318
29. Murata, T.: Petri nets: Properties, analysis and applications. Proceedings of the IEEE **77**(4) (1989) 541–580
30. Alstrup, S., Harel, D., Lauridsen, P.W., Thorup, M.: Dominators in linear time. SIAM J. Comput. **28**(6) (1999) 2117–2132
31. Curran, T.A., Keller, G., Ladd, A.: SAP R/3 Business Blueprint: Understanding the Business Process Reference Model. Prentice-Hall (1997)
32. Dongen, B., Jansen-Vullers, M., Verbeek, H., Aalst, W.: Verification of the SAP reference models using EPC reduction, state space analysis, and invariants. Computers in Industry **58**(6) (2007) 578–601
33. Decker, G., Mendling, J.: Process instantiation. Data Knowl. Eng. **68** (2009)
34. Valmari, A.: The state explosion problem. In: Petri Nets. Volume 1491 of LNCS. Springer (1996) 429–528
35. Pomello, L., Rozenberg, G., Simone, C.: A survey of equivalence notions for net based systems. In: Advances in Petri Nets: The DEMON Project. Volume 609 of LNCS. Springer (1992) 410–472
36. de Medeiros, A.K.A., Aalst, W., Weijters, A.: Quantifying process equivalence based on observed behavior. Data Knowl. Eng. **64**(1) (2008)
37. Glabbeek, R., Goltz, U.: Refinement of actions and equivalence notions for concurrent systems. Acta Inf. **37**(4/5) (2001) 229–327
38. Brauer, W., Gold, R., Vogler, W.: A survey of behaviour and equivalence preserving refinements of petri nets. In: ATPN. Volume 483 of LNCS. Springer (1989) 1–46
39. Winskel, G.: Petri nets, algebras, morphisms, and compositionality. Inf. Comput. **72**(3) (1987) 197–238
40. Basten, T., Aalst, W.: Inheritance of behavior. JLAP **47**(2) (2001) 47–145
41. Schrefl, M., Stumptner, M.: Behavior-consistent specialization of object life cycles. ACM Trans. Softw. Eng. Methodol. **11**(1) (2002) 92–148
42. Rosenblum, L., Yakovlev, A.: Analyzing Semantics of Concurrent Hardware Specifications. In: ICPP. Volume 3. (1989) 211–218
43. Dongen, B., Dijkman, R.M., Mendling, J.: Measuring similarity between business process models. In: CAiSE. Volume 5074 of LNCS. Springer (2008) 450–464
44. Dumas, M., García-Bañuelos, L., Dijkman, R.M.: Similarity search of business process models. IEEE Data Eng. Bull. **32**(3) (2009) 23–28