Fundamenta Informaticae 113 (2011) 1–37
IOS Press

Causal Behavioural Profiles – Efficient Computation, Applications, and Evaluation

Matthias Weidlich [C]
Hasso Plattner Institute, University of Potsdam
Prof.-Dr.-Helmert-Str. 2-3, 14482 Potsdam, Germany
matthias.weidlich@hpi.uni-potsdam.de

Artem Polyvyanyy
Hasso Plattner Institute, University of Potsdam
Prof.-Dr.-Helmert-Str. 2-3, 14482 Potsdam, Germany
artem.polyvyanyy@hpi.uni-potsdam.de

Jan Mendling
Humboldt University
Unter den Linden 6, D-10099 Berlin, Germany
jan.mendling@wiwi.hu-berlin.de

Mathias Weske
Hasso Plattner Institute, University of Potsdam
Prof.-Dr.-Helmert-Str. 2-3, 14482 Potsdam, Germany
mathias.weske@hpi.uni-potsdam.de

Abstract. Analysis of behavioural consistency is an important aspect of software engineering. In process and service management, consistency verification of behavioural models has manifold applications. For instance, a business process model used as system specification and a corresponding workflow model used as implementation have to be consistent. Another example would be the analysis to what degree a process log of executed business operations is consistent with the corresponding normative process model. Typically, existing notions of behaviour equivalence, such as bisimulation and trace equivalence, are applied as consistency notions. Still, these notions are exponential in computation and yield a Boolean result. In many cases, however, a quantification of behavioural deviation is needed along with concepts to isolate the source of deviation.

In this article, we propose causal behavioural profiles as the basis for a consistency notion. These profiles capture essential behavioural information, such as order, exclusiveness, and causality between pairs of activities of a process model. Consistency based on these profiles is weaker than trace equivalence, but can be computed efficiently for a broad class of models. In this article, we introduce techniques for the computation of causal behavioural profiles 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 elaborate on the findings of applying our technique to three industry model collections.

Keywords: Causal Behavioural Profiles, Formal Methods, Behavioural Abstraction, Structural Decomposition, Exclusiveness, Concurrency, Order Relations, Causality, Optionality

[C] Corresponding author

1. Introduction

Consistency verification is a central aspect of software engineering. Focussing on the domain of process and service management, consistency between behavioural models needs to be assessed at different stages of the implementation of business processes. First and foremost, process modelling has become one of the most extensively used approaches for capturing business requirements [14]. 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 not implemented 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 what degree a process model used as specification and a workflow model used as implementation are behaviourally consistent.

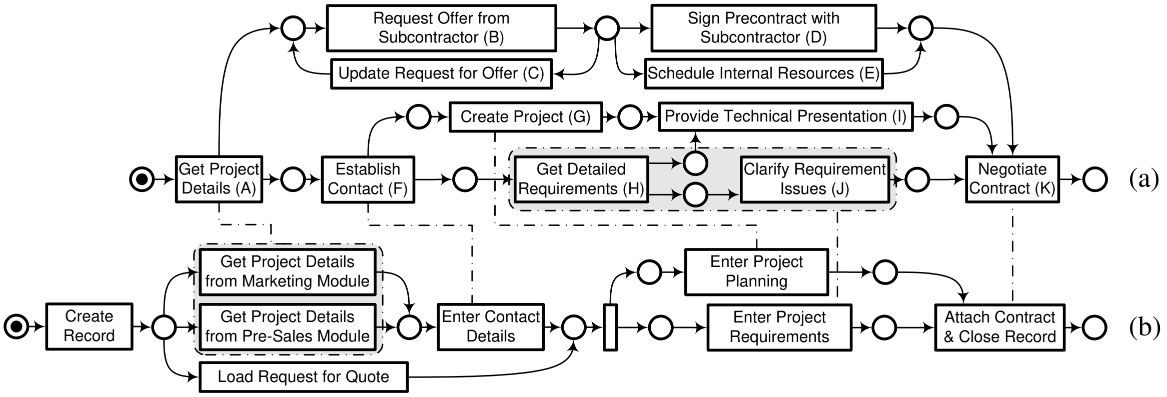

Request Offer from Subcontractor (B)
Sign Precontract with Subcontractor (D)
Update Request for Offer (C)
Schedule Internal Resources (E)
Create Project (G)
Provide Technical Presentation (I)
Get Project Details (A)
Establish Contact (F)
Get Detailed Requirements (H)
Clarify Requirement Issues (J)
Negotiate Contract (K)
(a)
Get Project Details from Marketing Module
Get Project Details from Pre-Sales Module
Enter Project Planning
Create Record
Enter Contact Details
Enter Project Requirements
Attach Contract & Close Record
Load Request for Quote
(b)

Figure 1. Example of two Petri net process models, (a) focussing on the business perspective, (b) depicting the workflow implementation

Figure 1 illustrates the problem addressed in this article with two process models that relate to a project handling process. 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. If process models assume different perspectives on a common business process, such correspondences rarely express semantic equivalence between the matched activities. Consider the activities *'Establish Contact'* and *'Enter Contact Details'* of the example in Figure 1. Apparently, establishing a contact involves more than just entering the contact details. Still, we say that both activities correspond to each other against the background of aligning process models that assume different perspectives. As for the model in general, both activities assume either a business perspective or a workflow perspective on a particular unit of work.

For this article, we assume that correspondences between activities 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 modelling [18, 50, 64]. Moreover, techniques from the area of schema and ontology matching can be exploited [27, 56]. Then, activities are regarded as elements of a process model schema.

Consistency verification between behavioural models is not only relevant during design-time. The analysis to what degree a process log of executed business operations is consistent with the corresponding normative process model is an example for consistency verification at run-time. Here, it may be the case that the actual processing observed in information systems is not in line with the specified processing. Deviations may stem from information systems that do not explicitly enforce the predefined execution order of activities or from people working around the system [5].

For both use cases, existing notions of behaviour equivalence may 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 complex [29], 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 many cases, however, a quantification of behavioural deviation is needed and the source of deviation needs to be identified.

In this article, we argue that for certain scenarios of consistency verification, a criterion of behaviour equivalence might be weakened in order to compensate for computational efficiency. To this end, we define the notion of a *causal behavioural profile*. Such a profile represents a behavioural abstraction that 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.

This article is an extended and revised version of our previous work [69]. As an extension, we included a broader discussion of the application of causal behavioural profiles. In particular, we do not only illustrate how they form the basis of a consistency notion for related process models, but also summarise recent work on their application for consistency analysis of process logs (aka conformance checking). Further, we revised the section on the computation of causal behavioural profiles and integrated the results obtained for structured fragments with those obtained for unstructured fragments. Hence, we are able to present a complete computation algorithm for the proposed approach. Finally, we present an extended validation of the techniques for the computation of causal behavioural profiles. In addition to the models used in [69], we applied our approach to two more model collections from industry. Hence, we provide further insights on the applicability of our approach in an industry setting.

The remainder of this article is structured as follows: Section 2 introduces our formal framework. Causal behavioural profiles are defined in Section 3. We discuss applications of causal behavioural profiles for consistency verification in Section 4. Section 5 elaborates on graph decomposition techniques that are applied to workflow nets. How these techniques are used to compute causal behavioural profiles is presented in Section 6. In Section 7, we report on the findings of applying our approach to three industry model collections. Finally, Section 8 reviews related work, before Section 9 concludes the article.

2. Preliminaries

We use workflow (WF-) systems [2] 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 the Business Process Model and Notation (BPMN), the Web Service Business Process Execution Language (BPEL), Event Driven Process Chains (EPCs), and UML Activity Diagrams, e.g., [19, 42, 38, 24, 47]. A survey of these formalisations can be found in [43]. Based on [2, 17], we recall basic definitions.

Definition 2.1. (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 irreflexive transitive closure of F is denoted by F^+.
- For a node $x \in X$, $\bullet x := \{y \in X \mid (y, x) \in F\}$ is the pre-set, $x\bullet := \{y \in X \mid (x, y) \in F\}$ is the post-set, and $\bullet(x\bullet) := \{z \in X \mid y \in X \wedge \ (x, y) \in F \wedge (z, y) \in F\}$.
- A tuple $N' = (P', T', F')$ is a *subnet* of a net $N = (P, T, F)$, if $P' \subseteq P$, $T' \subseteq T$, and $F' = F \cap ((P' \times T') \cup (T' \times P'))$; $N'' = (P'', T'', F'')$ is a *partial* subnet of N, if $P'' \subseteq P$, $T'' \subseteq T$, and $F'' \subseteq F \cap ((P'' \times T'') \cup (T'' \times P''))$.
- A net N is a *T-net*, if $\forall\, p \in P\ [\ |\bullet p| \leq 1 \geq |p \bullet|\ ]$, and an *S-net*, if $\forall\, t \in T\ [\ |\bullet t| \leq 1 \geq |t \bullet|\ ]$.
- 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, $k \in \mathbb{N}$, $k > 1$, denoted by $\pi_N(x_1, x_k)$, which satisfies $(x_1, x_2), \ldots, (x_{k-1}, x_k) \in F$. By $\pi_N\{x_1, x_k\} = \{x_1, \ldots, x_k\}$, we denote the set of all nodes on the path. We write $x_i \in \pi_N$, if $x_i \in \pi_N\{x_1, x_k\}$. A *subpath* π'_N of a path π_N is a subsequence of π_N 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 is more than once part of the path.
- For a net $N = (P, T, F)$ and a partial subnet $N' = (P', T', F')$ a path $\pi_N(x_1, x_k)$, $k > 1$, where all x_i are distinct, of N is a *handle* of N', iff $\pi_N\{x_1, x_k\} \cap (P' \cup T') = \{x_1, x_k\}$.
- For a net $N = (P, T, F)$ and two partial subnets $N' = (P', T', F')$, $N'' = (P'', T'', F'')$, 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 $\pi_N\{x_1, x_k\} \cap (P' \cup T') = \{x_1\}$ and $\pi_N\{x_1, x_k\} \cap (P'' \cup T'') = \{x_k\}$.
- 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-ciruit* net $N' = (P, T \cup \{t_c\}, F \cup \{(o, t_c), (t_c, i)\})$, $t_c \notin T$, of N is strongly connected.

Free-choiceness of a net implies that various behavioural analysis questions can be answered efficiently. Note that the same effect can be achieved with a more relaxed notion, referred to as *extended free-choiceness* [10]. Any extended free-choice net can be transformed into a behaviour equivalent free-choice net [10] – here, behaviour equivalence assumes that transitions inserted by the transformation are ignored. For this article, we stick to the definition presented earlier.

Further, we speak of *PP-,TT-,PT-,TP-* handles and bridges, depending on the type (place or transition) of the initial and the final node of the path. We define semantics for WF-nets according to [2].

Definition 2.2. (WF-net Semantics)

Let $N = (P, T, F)$ be a WF-net with initial place i and final place o.

- $M : P \mapsto \mathbb{N}_0$ 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 where 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 reachable 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 sequence of transitions $\sigma = t_1, \ldots, t_n$, $n \in \mathbb{N}_0$, is a *firing sequence*, iff there exist markings $M_0, \ldots, M_n \in \mathbb{M}$, such that for all $i \in \mathbb{N}$, $1 \leq i \leq n$ it holds $(N, M_{i-1})[t_i\rangle(N, M_i)$. We say that

σ is enabled in M_0, denoted by $(N, M_0)[\sigma\rangle$. For $n = 0$, we refer to $\sigma = \langle\rangle$ as the empty firing sequence.

- For any two markings $M, M' \in \mathbb{M}$, M' is reachable from M in N, denoted by $M' \in [N, M\rangle$, if there exists a firing sequence σ leading from M to M'. Firing of σ in M is denoted by $(N, M)[\sigma\rangle(N, M')$.
- 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 $M_i = [i]$.

Given a (free-choice, S-, T-) WF-net N with M_i as its initial marking, the tuple $S = (N, M_i)$ is a (free-choice, S-, T-) *WF-system*. The *final marking* is denoted by $M_o = [o]$ with o being the final place of a WF-net. Without stating it explicitly, we assume a net of a system to be defined as $N = (P, T, F)$. If 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 [1]. Both requirements imply proper termination of the WF-system, i.e., for all reachable markings holds that a token in the final place implies the absence of tokens for all other places. Soundness of a WF-system is traced back to liveness and boundedness of its short-circuit system, see [1].

Definition 2.3. (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

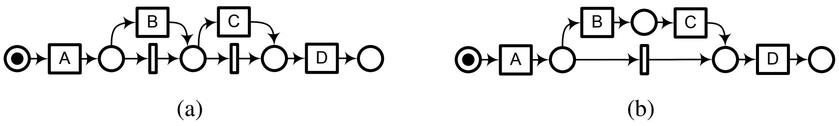
This section introduces causal behavioural profiles. They are based on the notion of behavioural profiles, which we recall in Section 3.1. We introduced these profiles in an earlier work [66] to reason on order constraints only. Optionality of transition execution or causality between transitions is 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 relations defined for Petri nets.

3.1. Execution Order Constraints: The Behavioural Profile

Behavioural profiles capture behavioural aspects in terms of order constraints of a process in a fine-grained manner [66]. They are grounded on the set of possible firing sequences of a WF-system and the notion of *weak order*.

Definition 3.1. (Weak Order)
Let (N, M_i), $N = (P, T, F)$, be a WF-system. A pair $(x, y) \in (T \times T)$ is in the *weak order relation* $\succ$, iff there exists a firing sequence $\sigma = t_1, \ldots, t_n$ with $(N, M_i)[\sigma\rangle$ and indices $j, k \in \mathbb{N}$, $1 \leq j < k \leq n$, for which holds $t_j = x$ and $t_k = y$.

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. Using weak order, we define three relations forming the behavioural profile.



(a) (b)

Figure 2. Optionality

Definition 3.2. (Behavioural Profile)
Let (N, M_i), $N = (P, T, F)$, be a WF-system and $T' \subseteq T$ a set of transitions. A pair of transitions $(x, y) \in (T' \times T')$ can be in 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 $y \succ x$.

$\mathcal{B}_{T'} = \{\rightsquigarrow, +, ||\}$ is the *behavioural profile* of (N, M_i) over T'.

If we do not restrict the set of transitions over which the behavioural profile is defined, we assume that it is defined over all transitions. The inverse relation of strict order, $\rightsquigarrow^{-1} = \{(y, x) \in (T' \times T') \mid x \rightsquigarrow y\}$, is referred to as *reverse strict order*. Computing the behavioural profile for all transitions of the system (a) in Figure 1 reveals that, 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. 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 reverse strict order) partition the Cartesian product of transitions over which they are defined [66]. With respect to itself, a transition is either exclusive (if it can occur at most once, e.g., $D + D$) or in interleaving order (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. Even though this information may be sufficient for certain consistency scenarios, these profiles provide a rather coarse-grained behavioural abstraction. 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 Figure 2(a) and Figure 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 Figure 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. The former is addressed by the behavioural profile in terms of the strict order relation, whereas the latter is not captured. For instance, B is a cause of C in Figure 2(b), but not in Figure 2(a). Note that two transitions in interleaving order do not show causality according to our definition. For both systems in Figure 3, it holds $B||C$. We do not observe an ordering between

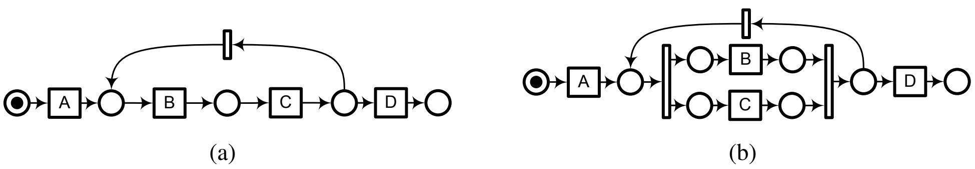


Figure 3. No causality for transitions (B,C) in a cycle

all occurrences of both transitions. 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* or *leads-to* dependencies [23]). There is no causality between B and C in either system in Figure 3.

To cope with the aforementioned aspects, we extend the behavioural profile yielding the causal behavioural profile. The latter is a more fine-grained behavioural abstraction that closer approximates trace semantics of net systems. Hence, we can still achieve efficient computation for a broad class of system, but have a closer approximation of behaviour equivalence for consistency verification. Technically, the causal behavioural profile introduces a co-occurrence relation. 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 3.3. (Causal Behavioural Profile)
Let (N, M_i), $N = (P, T, F)$, be a WF-system and $T' \subseteq T$ a set of transitions.

- ◦ 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 $x \in \sigma \Rightarrow y \in \sigma$.
- ◦ $\mathcal{CB}_{T'} = \{\rightsquigarrow, +, ||, \gg\}$ is the *causal behavioural profile* of (N, M_i) over T'.

Again, if the set of transitions is not restricted explicitly, we assume the causal behavioural profile to be defined over all transitions. 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, 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 Figure 2.

3.3. Relation to Existing Behavioural Relations

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 [6] for workflow mining are close to our relations, yet different. We base our definitions on an *indirect* weak order dependency, whereas the relations in [6] 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'* [6], whereas we capture exclusiveness for transitions that could potentially occur at different stages of a firing sequence. The notion of direct sequential order is appropriate for workflow mining, but leads to undesired effects in our setting. For most scenarios of consistency verification, the relation associating corresponding transitions of both models to each other is partial. Certain transitions of one model are without counterpart in the other model. Figure 1 illustrates this phenomenon for the case of consistency verification of business process models, both captured as WF-nets, used as specification and workflow models used as implementation. Consider, for instance, transitions G and K of model (a) in Figure 1. They are exclusive according to the relations proposed in [6], 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). Therefore, indirect dependencies, as defined by the behavioural profile, are more appropriate for consistency verification in the presence of partial correspondence relations.

The well-known notions of *conflicting* and *concurrent* transitions are related to our relations as well. In a sound free-choice WF-system, two transitions, which are in conflict and are not part of a circuit, will be exclusive in the behavioural profile. This follows from Lemma 3 in [66] 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 [3]). Similarly, all transitions that are enabled concurrently in some reachable marking (cf., the concurrency relation [39]) 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 complete prefix, respectively) may be exploited for behaviour analysis [25, 44]. 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. Even though these relations resemble the relations of our casual behavioural profile, they are defined for transition occurrences instead of transitions. Thus, we can derive our relations by lifting the relations of the complete prefix unfolding to the level of transitions again. Recently, we introduced a computation algorithm for behavioural profiles based on complete prefix unfoldings of bounded systems [65]. However, usage of unfoldings is inappropriate w.r.t. the class of systems addressed in this article, as the construction of unfoldings is computationally much harder than the approach introduced in the remainder of this article.

With respect to common notions of behaviour equivalence, we see that two WF-systems with equal causal profiles are not necessarily trace equivalent. For instance, both systems in Figure 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.

4. Applications of Causal Behavioural Profiles

We motivated the definition of causal behavioural profiles with the need to assess behavioural consistency in an efficient and fine-granular manner. This section reviews applications of causal behavioural profiles. First, Section 4.1 discusses how a degree of consistency is determined for two WF-systems under the assumption of a correspondence relation. Second, Section 4.2 summarises the application of causal behavioural profiles for consistency analysis of process logs. Causal behavioural profiles have already been applied in a much broader context, e.g., for query optimisation in complex event processing [71] and process monitoring [72]. As such, causal behavioural profiles proved to be a valuable behavioural model for many use cases. In this section, however, we focus on their application for consistency measurement.

4.1. Consistency between Related Systems

Once a correspondence relation has been established between the transitions of two systems using structural analysis and natural language processing (see [18, 50, 64]), the degree of consistent behaviour can be quantified based on causal behavioural profiles. The general idea behind the consistency measure is summarised as follows. Given a correspondence relation between the sets of transitions of two WF-systems, we consider all aligned transitions of both system, i.e., transitions in either model that have a corresponding transitions in the other model. For each pair of aligned transitions, we check whether the corresponding transitions show the same constraints as defined by the causal behavioural profile. Since there can be complex n:m correspondences, see Figure 1, we have to count the correspondences from the perspective of each model.

A consistency notion based on the behavioural profile has been introduced in [66]. The novel concept of causal behavioural profiles is applied in the same manner and yields a degree of consistency that is more fine-grained (optionality and causality constraints are also considered) and closer to notions of behaviour equivalence. The degree of consistency based on causal behavioural profiles is defined as follows.

Definition 4.1. (Degree of Consistency)
Let (N_1, M_{i_1}), $N_1 = (P_1, T_1, F_1)$, and (N_2, M_{i_2}), $N_2 = (P_2, T_2, F_2)$, be two WF-systems and $\mathcal{CB}_{T_1} = \{\rightsquigarrow_1, +_1, ||_1, \gg_1\}$ and $\mathcal{CB}_{T_2} = \{\rightsquigarrow_2, +_2, ||_2, \gg_2\}$ their causal behavioural profiles. Let $\sim \subseteq T_1 \times T_2$, $\sim \neq \emptyset$, be a correspondence relation.

- ∘ 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.
- ∘ Two relations $R_1 \in \{\rightsquigarrow_1, +_1, ||_1, \rightsquigarrow_1^{-1}\}$ and $R_2 \in \{\rightsquigarrow_2, +_2, ||_2, \rightsquigarrow_2^{-1}\}$ are *type equivalent*, denoted by $R_1 \simeq R_2$, iff either
  - – $R_1 = \rightsquigarrow_1\ \wedge R_2 = \rightsquigarrow_2$,
  - – $R_1 = \rightsquigarrow_1^{-1}\ \wedge R_2 = \rightsquigarrow_2^{-1}$,
  - – $R_1 = +_1\ \wedge\ R_2 = +_2$, or
  - – $R_1 = ||_1\ \wedge\ R_2 = ||_2$.
- ∘ The set of *behavioural profile consistent transition pairs* $CT_1^{\sim} \subseteq (T_1^{\sim} \times T_1^{\sim})$ for (N_1, M_{i_1}) contains all 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 $(t_x R_1 t_x\ \wedge\ t_s R_2 t_s) \Rightarrow R_1 \simeq R_2$,
  - – if $t_x \neq t_y$, then $\forall\, t_s, t_t \in T_2^{\sim}$ with $t_x \sim t_s$ and $t_y \sim t_t$ it holds either (1) $(t_x R_1 t_y \wedge t_s R_2 t_t) \Rightarrow R_1 \simeq R_2$ or (2) $t_x \sim t_t$ and $t_y \sim t_s$.

  The set $CT_2^{\sim}$ for S_2 is defined analogously.
- ∘ The set of *causal behavioural profile consistent transition pairs* $CCT_1^{\sim} \subseteq CT_1^{\sim}$ for (N_1, M_{i_1}) contains all pairs (t_x, t_y), such that if $t_x \neq t_y$ then for all transitions $t_s, t_t \in T_2^{\sim}$ with $t_x \sim t_s$ and $t_y \sim t_t$ it holds either (1) $t_x \gg_1 t_y \Leftrightarrow t_s \gg_2 t_t$ or (2) $t_x \sim t_t$ and $t_y \sim t_s$. The set $CCT_2^{\sim}$ for S_2 is defined analogously.
- ∘ The *degree of consistency* of $\sim$ is defined as $\mathcal{D}^{\sim} = \frac{|CCT_1^{\sim}|+|CCT_2^{\sim}|}{|(T_1^{\sim} \times T_1^{\sim})|+|(T_2^{\sim} \times T_2^{\sim})|}$.

Applying this degree to the scenario in Figure 1, we see that the order of potential occurrence is preserved for all aligned transitions. Hence, the degree of consistency based on the behavioural profile proposed in [66] yields a value of one. 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 based on causal behavioural profiles is $\mathcal{D}^{\sim} = \frac{27+25}{36+36} \approx 0.722$. This value is derived as follows. Model (a) contains six transitions that are part of correspondences, i.e., transitions $\{(A), (F), (G), (H), (J), (K)\}$. Also, model (b) contains six aligned transitions. Hence, we consider $36 + 36 = 72$ behavioural constraints between pairs of transitions overall. Checking the constraints between transition pairs of model (a), we see that 27 out of 36 constraints are equal for all pairs of corresponding transitions in model (b). For the 36 constraints between aligned transitions of model (b), we observe that only 25 constraints are mirrored by the corresponding transitions in model (a).

The presented degree of consistency shows the characteristics of a semimetric for the comparison of two causal behavioural profiles. It is a 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.

Consistency measurement based on behavioural profiles has been applied successfully in a recent case study on process variants [66]. Clearly, there is a need for a multitude of consistency criteria in order to be able to graduate consistency requirements for a concrete setting. Still, an interval scale and efficient computation methods have to be seen as core requirements on such notions. Being stronger than behavioural profile consistency but weaker than trace equivalence, our proposal based on causal behavioural profiles is one step further towards a spectrum of consistency notions.

It is worth to mention that consistency based on (causal) behavioural profiles enables support for change propagation between related systems. We formalised a change propagation approach using behavioural profiles in [70]. The approach can be lifted to the causal behavioural profiles in a straightforward manner. The general idea behind this approach can be summarised as follows. Once a change in one system is localised by a node or a flow, the location is manifested in the causal behavioural profile. Consequently, the change has to be propagated to the other model so that equal relations of the causal behavioural profile can be observed for the corresponding nodes, respectively. We illustrate the approach to change propagation with the two systems in Figure 4. Assume that both systems were equal before a change operation, i.e., insertion of transition X, has been applied to the upper system. The causal behavioural profile of the upper system locates this change as being in strict order with transition A and in reverse strict order with transition B. Leveraging this information for the corresponding transitions in the lower model isolates the region in which the change has to be applied. Taking the co-occurrence relation of the causal behavioural profile into account, it even becomes clear how a corresponding transition X has to be inserted in the lower model. That is, its execution has to be optional as there is no co-occurrence with neither transition A, nor transition B. Hence, compared to existing work [70], causal behavioural profiles allow for more precise change propagation under the assumption of a stricter notion of consistency.

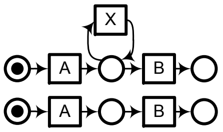
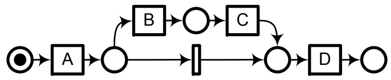


Figure 4. Propagation of inserting transition X

4.2. Conformance of Process Logs

Besides consistency verification between related systems, causal behavioural profiles are applied to assess consistency of recorded execution sequences with a normative model. The former are referred to as *logs* and represent observed executions of transitions as recorded by information systems, e.g., Enterprise Resource Planning (ERP) systems, Customer Relationship Management (CRM) systems, or Workflow Management Systems (WFMS). Consistency of logs is evaluated by *conformance measures* that quantify to what degree the behaviour of a log is captured in the respective model. These measures provide feedback on cases that do not conform to the model and quantify any behaviour deviations. This may be the case when the execution order of transitions is not explicitly enforced by information systems or when people deliberately working around the supporting IT-infrastructure [5]. Conformance measures have to be rather fine-grained to distinguish marginally different logs from completely different logs. Further, they have to be efficiently computable and shall provide diagnostics if non-conformance is observed.

In [67, 68], we showed how causal behavioural profiles are used for conformance analysis of logs that meets the aforementioned requirements. We assess whether the behavioural constraints as imposed by the model for pairs of transitions are satisfied in the log. Causal behavioural profiles are used to capture behavioural constraints. The share of constraints that is satisfied by a log is then used as a conformance measure. We illustrate conformance measurement based on causal behavioural profiles with the example in Figure 5. Here, the upper system is the normative model. Two logs have been recorded as actual executions of the system. Evidently, the log L_1 represents a valid execution sequence of the system. This is manifested by the conformance measure presented in [67, 68], which yields a value of one. All constraints, e.g., the strict order between transitions A and C or the co-occurrence between transitions B and C, are satisfied in the log L_1. In contrast, the log L_2 is not conformant. Constraints as imposed by the system and formalised in the causal behavioural profile are violated. The system allows for at most one execution of transition A (it is exclusive to itself), whereas the log records two executions. Further, the co-occurrence dependency between transitions B and C is violated in the log, as the log contains transition B but no transition C. However, a transition that is in strict order from transition C is already in the log, i.e., transition D. This shows that transition C should have been observed already in the log. Overall, our conformance measure yields a value of $\frac{8+11}{9+12} \approx 0.90$ for log L_2. Eight out of nine constraints of the behavioural profile are satisfied ($A + A$ is violated), and 11 out of 12 co-occurrence constraints are in line with the process model ($B \gg C$ is violated). For co-occurrence, we check 12 constraints since all pairs of transitions that are expected to be in the log are considered, in our example this includes the constraints $C \gg A$, $C \gg B$, and $C \gg D$, whereas self-relations are ignored. Our approach enables root-cause analysis for non-conformant cases, see [68]. For the log L_2, both observed violations would be considered to be independent root causes.

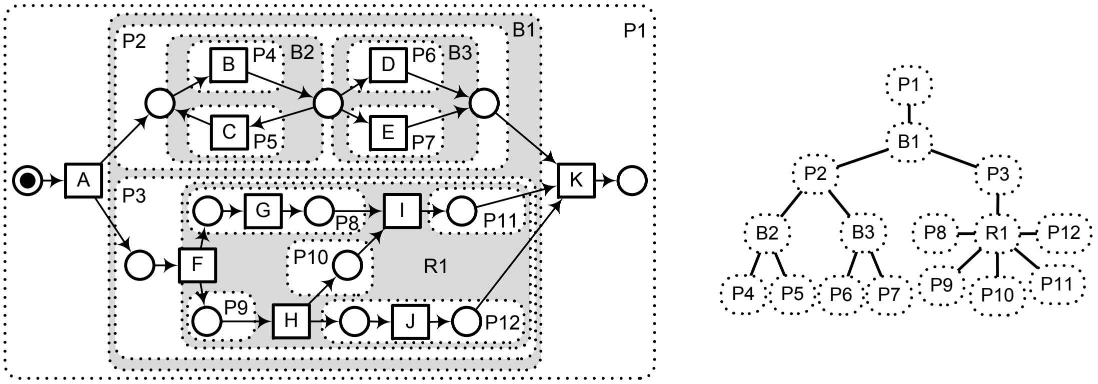


Log L_1: <A, B, C, D>
Log L_2: <A, A, B, D>

Figure 5. Example for conformance analysis

5. Graph Decomposition Techniques for WF-Systems

This section discusses the application of graph decomposition techniques for WF-systems. First, Section 5.1 introduces the Refined Process Structure Tree (RPST), a structural decomposition technique for workflow graphs. Second, Section 5.2 enriches the RPST for WF-systems with behavioural annotations.



(a) (b)

Figure 6. (a) A WF-system and its canonical fragments, (b) the RPST of (a)

5.1. The Refined Process Structure Tree

The RPST [61, 54] 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 [54, 61]. Although the RPST has been introduced for workflow graphs, the technique can be applied to other graph based behavioural models such as WF-nets in a straightforward manner. Basic terms of the RPST are defined for WF-nets as follows.

Definition 5.1. (Edges, Entry, Exit, Canonical Fragment)
Let $N = (P, T, F)$ be a WF-net.

- For a node $x \in X$ of a net $N = (P, T, F)$, $in_N(x) = \{(n, x) \in F \mid n \in \bullet x\}$ are its *incoming edges* and $out_N(x) = \{(x, n) \in F \mid n \in x\bullet\}$ 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 two boundary nodes: one entry, denoted by $\omega_{\triangleleft}$, and one exit, denoted by $\omega_{\triangleright}$.
- 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 $\forall\, \gamma = (P_\gamma, T_\gamma, F_\gamma) \in \Sigma\ [\, \omega \neq \gamma \Rightarrow (F_\omega \cap F_\gamma = \emptyset) \vee (F_\omega \subset F_\gamma) \vee (F_\gamma \subset F_\omega) \,]$.

Figure 6 exemplifies the RPST for the WF-system from Figure 1(a). Figure 6(a) illustrates its canonical fragments, each of them formed by a set of edges enclosed in or intersecting the region with a dotted border. Each edge itself is a fragment, which is neglected in Figure 6 by showing only fragments that comprise more than one edge. As an example, consider fragments $P4$ and $B2$. The former is a subnet that consists of transition B, the place in its pre-set, the place in its post-set, and two edges, which connect these places with transition B. The place in the pre-set of transition B is the entry of this fragment, the place in the post-set of transition B is the exit. Fragment $B2$, in turn, comprises two transitions, B and C, the two places in their pre-sets and post-sets, and four edges that connect these places with both transitions. Fragment $B2$ has the same entry and the same exit as fragment $P4$. All fragments visualised in Figure 6 are canonical. Assume that the subnet $B2$ is extended by transition A and the edge between A and the place in the pre-set of transition B. Then, this subnet would be a fragment, but would not be canonical in the set of fragments illustrated in Figure 6.

For a set of canonical fragments, the containment relation provides us with a hierarchy of fragments. It is illustrated in Figure 6(b), in which each node represents a canonical fragment and edges hint at containment relation of fragments. Observe that one obtains a tree structure—the RPST. Fragment $B2$ mentioned earlier comprises two fragments, $P4$ and $P5$.

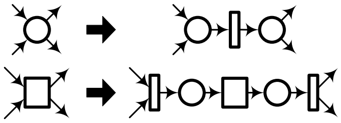


Figure 7. Node-splitting

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 [53, 54]: A *trivial* (T) fragment consists of a single edge. A *polygon* (P) represents a sequence of fragments. This type of fragment represents the most simple control flow structure, a sequence of transitions or fragments, respectively. A *bond* (B) stands for a collection of fragments that share common boundary nodes. This type of fragment represents a well-structured part of the graph, see [37, 52] for a discussion of well-structuredness of workflow graphs. For such a fragment, the behavioural relation between the contained transitions or subnets depends on the entry of the fragment, since there are no edges between the children of a bond fragment. Any fragment that does not match the requirements for the aforementioned classes is a *rigid* (R). In this article, we use fragment 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 Figure 7. The WF-system in Figure 6(a) is normalized.

5.2. 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 the previous section, we already mentioned that the structural characteristics, especially of polygon and bond fragments, are close to control flow routing concepts. In this section, we concretise RPST fragments by annotating them with behavioural characteristics. As such, we explicitly establish the relation between structural and behavioural characteristics, which is the basis for our approach to the computation of causal behavioural profiles.

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., [3, 36]).

Definition 5.2. (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,
- $\chi : \Omega \rightarrow \mathcal{P}(\Omega)$ is a function that assigns to fragment its child fragments,
- $t : \Omega \rightarrow \{T, P, B, R\}$ is a function that assigns a type to a fragment,
- $b : \Omega_B \rightarrow \{B_\circ, B_\diamond, L\}$, $\Omega_B = \{\omega \in \Omega \mid t(\omega) = B\}$, 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.

The intuition behind the annotation of bond fragments is summarised as follows. If the bond is acyclic, a place-bordered bond represents a well-structured part that incorporates an exclusive choice. That is, there is a conflict between the transitions that are part of different children of the fragment for a token in the place that represents the entry of the bond. A transition-bordered bond represents a well-structured part that enables concurrent processing. Firing of the transition that is the entry of such a fragment marks a place for each of the children fragments. A loop bond is a cyclic bond. There is at least one path from the entry to the exit, and vice versa. Since there are no paths between nodes of different children, therefore, a loop bond represents a well-structured control flow cycle.

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

Definition 5.3. (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 irreflexive 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 $\rho : \Omega \setminus \{\omega_r\} \rightarrow \Omega$ 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 $i \in 0 \ldots n-1$. 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 $\pi_\mathcal{T}(\omega_0, \omega_n)$.

Figure 8 shows the WF-tree of the WF-system from Figure 6(a). The WF-tree is isomorphic to the RPST of the WF-system, cf., Figure 6(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 there exists a fragment $\omega \in \Omega$, such that it is a child of γ and $\gamma_\triangleleft = \omega_\triangleright$. Otherwise, $b(\gamma) = B_\circ$ if $\gamma_\triangleleft \in P_\gamma$, or $b(\gamma) = B_\diamond$ if $\gamma_\triangleleft \in T_\gamma$.

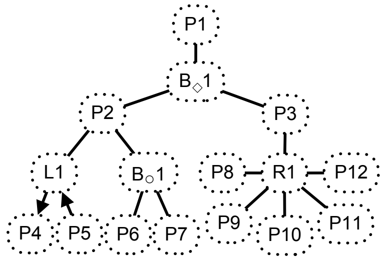


Figure 8. The WF-tree

Children of a polygon fragment are arranged with respect to their execution order. A partial function $order : \Omega' \rightarrow \mathbb{N}_0$, $\Omega' = \{\omega \in \Omega \setminus \{\omega_r\} \mid t(\rho(\omega)) = P\}$ assigns to children of polygon fragments their respective order positions; $order(\omega) = 0$, if $\omega_\triangleleft = \gamma_\triangleleft$ with $\gamma = \rho(\omega)$ being the parent, and $order(\omega) = i$, $i \in \mathbb{N}$, if $\omega_\triangleleft = \vartheta_\triangleright$ 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 Figure 8, $order(L1) = 1$ and $order(B_\circ 1) = 2$. This means that fragment $L1$ is always executed before fragment $B_\circ 1$ inside of polygon $P2$. Note that a trivial fragment resides at position 0 in fragment $P2$. The layout of child fragments of polygon fragments in Figure 8 hints at their order relations.

Children of a loop fragment are classified as *forward* ($\Rightarrow$) or *backward* ($\Leftarrow$). A partial function $\ell : \Omega'' \rightarrow \{\Leftarrow, \Rightarrow\}$ 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 $\omega_\triangleleft = \gamma_\triangleleft$ with $\gamma = \rho(\omega)$, otherwise $\ell(\omega) = \Leftarrow$. In Figure 8, $P4$ 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 [4]).

Lemma 5.1. 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., [54, 53]. 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., [53, 54]. 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 5.2. 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 5.1, 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 $\omega_\triangleleft$. $M(p) > 0$ 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 5.1). □

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

Corollary 5.1. 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 [54, 61]. The number of canonical fragments in the RPST is linear to the number of edges in the workflow graph [9, 34, 54]. 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. □

6. Efficient Computation of Causal Behavioural Profiles

This section shows how a WF-tree is applied to compute the causal behavioural profile. Section 6.1 introduces the approach for transition pairs that do not require analysis of rigid fragments. Afterwards, we discuss analysis of rigid fragments in Section 6.2. Finally, Section 6.3 presents a complete algorithm for the computation of causal behavioural profiles that orchestrates the results of the first two sections.

6.1. 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 Figure 7, which preserves the behaviour of the system (cf., [49]) 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.

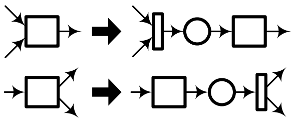


Figure 9. Pre-processing

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 [3]), i.e., $\forall\, p \in P$, $M(p) < 2$ in all reachable markings M. Thus, a single transition cannot be enabled concurrently with itself. 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 [36]). 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 of the WF-tree, the relations of the behavioural profile are computed based on the following proposition.

Proposition 6.1. 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 $\alpha_{\triangleleft} || \beta_{\triangleleft}$, iff $\exists\, \omega \in \pi_{\mathcal{T}}\{\omega_r, \gamma\}$ $[\, b(\omega) = L\,]$. Otherwise, $\alpha_{\triangleleft} + \beta_{\triangleleft}$.

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) $\exists\, \omega \in \pi_\mathcal{T}\{\omega_r, \gamma\}\ [\, b(\omega) = L\,]$.

Proof:
Let $\mathcal{T}_N$, α, β, γ be defined as above, (N, M_i) the respective WF-system, and $\forall\, \omega \in \pi_\mathcal{T}\{\omega_r, \gamma\}\ [\, t(\omega) \neq R\,]$.

1. Let $\alpha = \beta$.

   $\Rightarrow$ Let $\alpha_\lhd || \beta_\lhd$ and assume $\forall\, \omega \in \pi_\mathcal{T}\{\omega_r, \gamma\}\ [\, b(\omega) \neq L\,]$. Due to $\alpha = \beta$, also $\alpha_\lhd = \beta_\lhd$. Thus, we have $\alpha_\lhd || \alpha_\lhd$. Due to safeness of (N, M_i), $\alpha_\lhd || \alpha_\lhd$ cannot be traced back to concurrent enabling of $\alpha_\lhd$. According to Lemma 2 in [66], that implies $\alpha_\lhd\ F^+\ \alpha_\lhd$. Control flow cycles are part of B (if the bond is a loop fragment) or R type fragments. Thus, there has to be a fragment ω, which is an ancestor of α and $t(\omega) = R$ or $b(\omega) = L$. As the LCA of α is $\gamma = \alpha$ by definition, this yields a contradiction with the assumptions.

   $\Leftarrow$ Let $\exists\, \omega \in \pi_\mathcal{T}\{\omega_r, \gamma\}\ [\, b(\omega) = L\,]$ and assume $\alpha_\lhd \not\| \beta_\lhd$. Since $\alpha = \beta$, we have $\alpha_\lhd = \beta_\lhd$. One of the ancestors of α is an B type fragment that is a loop. Thus, $\alpha_\lhd\ F^+\ \alpha_\lhd$. Since (N, M_i) is safe, $\alpha_\lhd$ cannot be enabled concurrently with itself, so that $\alpha_\lhd || \alpha_\lhd$ due to Lemma 2 in [66].

2. Let $\alpha \neq \beta$.

   $\Rightarrow$ Let $\alpha_\lhd \rightsquigarrow \beta_\lhd$ and assume (1) $order(\pi_\mathcal{T}(\gamma, \alpha, 1)) > order(\pi_\mathcal{T}(\gamma, \beta, 1))$ or $t(\gamma) \neq P$, or (2) $\exists\, \omega \in \pi_\mathcal{T}\{\omega_r, \gamma\}\ [\, b(\omega) = L\,]$. According to Theorem 1 in [66], $\alpha_\lhd \rightsquigarrow \beta_\lhd$ implies $\alpha_\lhd\ F^+\ \beta_\lhd$ and $\beta_\lhd \not{F^+}\ \alpha_\lhd$. Thus, assumption (2) cannot hold as an L type fragment that is an ancestor of both, α and β, would imply $\beta_\lhd\ F^+\ \alpha_\lhd$. The first part of assumption (1) cannot hold either: $b(\gamma) = L$ contradicts with the flow dependencies between $\alpha_\lhd$ and $\beta_\lhd$, while $t(\gamma) = R$, $t(\gamma) = B$ and $b(\gamma) \in \{B_\circ, B_\diamond\}$, and $t(\gamma) = T$ (which would imply $\alpha = \beta$) disqualify due to our assumptions. Thus, $t(\gamma) = P$. Obviously, the order in a P type fragment coincidences with the flow dependencies, i.e., $\alpha_\lhd\ F^+\ \beta_\lhd$, which yields a contradiction with $order(\pi_\mathcal{T}(\gamma, \alpha, 1)) > order(\pi_\mathcal{T}(\gamma, \beta, 1))$.

   Let $\alpha_\lhd + \beta_\lhd$ and assume (1) $b(\gamma) \neq B_\circ$ or (2) $\exists\, \omega \in \pi_\mathcal{T}\{\omega_r, \gamma\}\ [\, b(\omega) = L\,]$. According to Lemma 3 in [66], the former implies $\alpha_\lhd \not{F^+}\ \beta_\lhd$ and $\beta_\lhd \not{F^+}\ \alpha_\lhd$. That, in turn, implies that assumption (2) cannot hold and $\gamma \neq P$. Also $\gamma \neq R$ and $\gamma \neq T$ (which would imply $\alpha = \beta$) by our assumptions. Thus, $t(\gamma) = B$. As the flow dependencies preclude $b(\gamma) = L$, we assume $b(\gamma) = B_\diamond$. Then, $\gamma_\lhd$ is a transition. Due to soundness, there are two markings $M_1, M_2 \in [N, M_i\rangle$, such that $(N, M_1)[\gamma_\lhd\rangle(N, M_2)$. As γ is an ancestor of both, α and β, we know $\gamma_\lhd\ F^+\ \alpha_\lhd$ and $\gamma_\lhd\ F^+\ \beta_\lhd$. That implies that both transitions, $\alpha_\lhd$ and $\beta_\lhd$, might get enabled in a firing sequences starting in M_2. That is not in line with $\alpha_\lhd + \beta_\lhd$. Thus, $b(\gamma) = B_\circ$, a contradiction with assumption (1).

   Let $\alpha_\lhd || \beta_\lhd$ and assume (1) $b(\gamma) = B_\circ$ and (2) $\forall\, \omega \in \pi_\mathcal{T}\{\omega_r, \gamma\}\ [\, b(\omega) \neq L\,]$. According to Lemma 2 in [66], $\alpha_\lhd || \beta_\lhd$ implies concurrent enabling of a both transitions in a certain marking, or $\alpha_\lhd\ F^+\ \beta_\lhd$ and $\beta_\lhd\ F^+\ \alpha_\lhd$. The latter is not possible due to assumption (2). Thus, we assume concurrent enabling. Let $x \in \gamma_\lhd\bullet$ be a successor of $\gamma_\lhd$. γ is the LCA of α and β. Consequently, $x\ F^+\ \alpha_\lhd$ implies $x \not{F^+}\ \beta_\lhd$ and vice versa. Thus, concurrent enabling of $\alpha_\lhd$ and $\beta_\lhd$ requires $\gamma_\lhd$ to be a transition. That, in turn, is a contradiction with assumption (1).

$\Leftarrow$ Let (1) $t(\gamma) = P \;\wedge\; order(\pi_{\mathcal{T}}(\gamma, \alpha, 1)) < order(\pi_{\mathcal{T}}(\gamma, \beta, 1))$, and (2) $\forall\, \omega \in \pi_{\mathcal{T}}\{\omega_r, \gamma\}$ $[\; b(\omega) \neq L \;]$ and assume $\alpha_{\triangleleft} \not\rightsquigarrow \beta_{\triangleleft}$. From (1) and (2), we get $\alpha_{\triangleleft}\; F^+\; \beta_{\triangleleft}$ and $\beta_{\triangleleft}\; \not{F}^+\; \alpha_{\triangleleft}$. According to Theorem 1 in [66], this is equivalent to $\alpha_{\triangleleft} \rightsquigarrow \beta_{\triangleleft}$.

Let (1) $b(\gamma) = B_{\circ}$ and (2) $\forall\, \omega \in \pi_{\mathcal{T}}\{\omega_r, \gamma\}$ $[\; b(\omega) \neq L \;]$, and assume $\alpha_{\triangleleft} \not+ \beta_{\triangleleft}$. From (1) and (2), we get $\alpha_{\triangleleft}\; \not{F}^+\; \beta_{\triangleleft}$ and $\beta_{\triangleleft}\; \not{F}^+\; \alpha_{\triangleleft}$. Therefore, we assume that both transitions are enabled concurrently. Due to $b(\gamma) = B_{\circ}$, $\gamma_{\triangleleft}$ is a place. Let $t \in \gamma_{\triangleleft}\bullet$ be a successor of $\gamma_{\triangleleft}$. Due to soundness, there are two markings $M_1, M_2 \in [N, M_i\rangle$, such that $(N, M_1)[t\rangle(N, M_2)$. As γ is the LCA of both, we know that $t\; F^+\; \alpha_{\triangleleft}$ implies $t\; \not{F}^+\; \beta_{\triangleleft}$ and vice versa. Thus, any firing sequence starting in M_2 contains either $\alpha_{\triangleleft}$, $\beta_{\triangleleft}$, or none of the two transitions. As γ is the parent of both, α and β, $\gamma_{\triangleleft}$ is on every path from the initial place i to $\alpha_{\triangleleft}$ and $\beta_{\triangleleft}$. Therefore, there does not exist a firing sequences containing both transitions, which leads to $\alpha_{\triangleleft} + \beta_{\triangleleft}$.

Let (1) $b(\gamma) \in \{B_{\diamond}, L\}$ or (2) $\exists\, \omega \in \pi_{\mathcal{T}}\{\omega_r, \gamma\}$ $[\; b(\omega) = L \;]$, and assume $\alpha_{\triangleleft} \not\parallel \beta_{\triangleleft}$. From requirement (2), we get $\alpha_{\triangleleft}\; F^+\; \beta_{\triangleleft}$ and $\beta_{\triangleleft}\; F^+\; \alpha_{\triangleleft}$. According to Lemma 2 in [66] this is equivalent to $\alpha_{\triangleleft} || \beta_{\triangleleft}$, which is not in line with our assumption. The same holds true for $b(\gamma) = L$. Consider $b(\gamma) = B_{\diamond}$. Then, $\gamma_{\triangleleft}$ is a transition. Let $p_1, p_2 \in \gamma_{\triangleleft}\bullet$ be two successors of $\gamma_{\triangleleft}$ with $p_1\; F^+\; \alpha_{\triangleleft}$ and $p_2\; F^+\; \beta_{\triangleleft}$. The existence of these paths implies the existence of a firing sequence, i.e., $\alpha_{\triangleleft}$ and $\beta_{\triangleleft}$ can get enabled concurrently. That, in turn, is equivalent to $\alpha_{\triangleleft} || \beta_{\triangleleft}$ by Lemma 1 in [66] yielding a contradiction with our assumption.

□

For the derivation of the co-occurrence relation, 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 *forward conflict-free*, iff $x_i \in P$ implies $|x_i \bullet| = 1$ for $1 \leq i < k$. The path $\pi_N(x_1, x_k)$ is *backward conflict-free*, iff $x_i \in P$ implies $|\bullet x_i| = 1$ for $1 < i \leq k$.

Lemma 6.1. For two transitions x and y in a sound WF-system holds,

- ○ if there is a forward conflict-free path from x to y, then $x \gg y$.
- ○ if there is a backward conflict-free path from x to y, then $y \gg x$.

Proof:

Let (N, M_i) be a sound WF-system and $x, y \in T$.

- ○ If $y \in (x\bullet)\bullet$ then every firing sequence σ containing x and ending with o, the final place, contains y as well. For all places $p \in x\bullet$ on the considered path from x to y holds $|p\bullet| = 1$ and, therefore, $p\bullet = \{y\}$, which implies $x \gg y$. If $y \notin (x\bullet)\bullet$, then let $t \in T$ be a transition between x and y, i.e., $x\; F^+\; t$ and $t\; F^+\; y$. For all places $p \in \bullet t$ holds $|p\bullet| = 1$. Thus, $p\bullet = \{t\}$. Consequently, for any two markings M_1, M_2 with $(N, M_1)[\sigma\rangle(N, M_2)$, $(N, M_1)[t\rangle$, and not $(N, M_2)[t\rangle$ we know that $t \in \sigma$. Starting with the transitions in $(x\bullet)\bullet$, therefore, all transitions on $\pi_N(x, y)$ have to be fired once they have been enabled in order to empty the place(s) of their pre-set. Due to soundness of the system, there is a firing sequence to the final marking for all reachable markings that enable x. Consequently, firing of x implies firing of y, which yields $x \gg y$.
- ○ The claim trivially holds by following the above argument in the reverse direction.

□

Using this lemma, we are able to prove the following proposition for the computation of the co-occurrence relation in the absence of rigid fragments.

Proposition 6.2. 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, $\alpha_\triangleleft \gg \beta_\triangleleft$, 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 $\forall\, \vartheta \in \Theta\ [\ \beta \in \chi^+(\vartheta)\ ]$.

Proof:
Let $\mathcal{T}_N$, α, β, γ, Π be defined as above, (N, M_i) the respective WF-system, and $\forall\, \omega \in \Pi\ [\ t(\omega) \neq R\ ]$. For both directions of the proof, let $\delta = \rho(\beta)$ and $\eta = \rho(\delta)$ be the parents of β and δ. Note that we know $t(\delta) = P$ and $t(\eta) \notin \{R, T\}$.

$\Rightarrow$ Let $\alpha_\triangleleft \gg \beta_\triangleleft$ and assume that there is an $\omega \in (\Pi \setminus \{\beta\})$ with $t(\omega) \neq P$ or $b(\omega) \neq B_\diamond$ or if $b(\omega) = L$ then it holds $\exists\, \vartheta \in \Theta\ [\ \beta \notin \chi^+(\vartheta)\ ]$ with $\Theta = \{\vartheta \in \chi(\omega) \mid \ell(\vartheta) = \Rightarrow\}$. For all $\omega \in (\Pi \setminus \{\beta\})$, we know $t(\omega) \neq R$ and $t(\omega) \neq T$ (as $\beta \in \chi^+(\omega)$). We first consider the LCA, i.e., fragment γ. Let $\epsilon \in \chi(\gamma)$ with $\alpha \in \chi^+(\epsilon)$ be the child fragment of γ that contains α (it holds $\epsilon \neq \delta$). We distinguish two cases.

(1) $\gamma_\triangleleft$ is a transition. Then, $t(\gamma) \in \{P, B\}$, while $t(\gamma) = B$ requires $b(\gamma) = B_\diamond$.

(2) $\gamma_\triangleleft$ is a place. Then, $t(\gamma) \in \{P, B\}$, while $t(\gamma) = B$ requires $b(\gamma) \in \{B_\circ, L\}$. We distinguish two cases (I) $b(\gamma) = B_\circ$ and (II) $b(\gamma) = L$.

(I) Let $M_1, M_2 \in [N, M_i\rangle$ be two markings with $M_1(\gamma_\triangleleft) > 1$ and $M_2(\gamma_\triangleright) > 1$. Let σ_1, σ_2 be two firing sequences with $(N, M_1)[\sigma_1\rangle(N, M_2)[\sigma_2\rangle(N, M_o)$, such that σ_2 does not contain any transition that is part of γ. As fragment ϵ represents a path from $\gamma_\triangleleft$ to $\gamma_\triangleright$, σ_1 might contain only transitions that are part of ϵ. Then, it holds $\alpha_\triangleleft \in \sigma_1$. Since $\alpha_\triangleleft \gg \beta_\triangleleft$, also $\beta_\triangleleft \in \sigma_1$ (as $\beta_\triangleleft \notin \sigma_2$). Therefore, $\beta \in \chi^+(\epsilon)$, such that we arrived at a contradiction with the definition of $\gamma = lca(\alpha, \beta)$.

(II) Let $M_1, M_2 \in [N, M_i\rangle$ be two markings as defined for the previous case with $M_1(\gamma_\triangleleft) > 1$ and $M_2(\gamma_\triangleright) > 1$. Consider the case of ϵ having forward orientation, $\ell(\epsilon) = \Rightarrow$. Then, there are two firing sequences σ_1, σ_2 with $(N, M_1)[\sigma_1\rangle(N, M_2)[\sigma_2\rangle(N, M_o)$, such that σ_1 contains only transitions that are part of ϵ, whereas σ_2 does not contain any transition that is part of γ. Then, $\alpha_\triangleleft$ might be part of σ_1. As $\beta_\triangleleft \notin \sigma_2$, but $\alpha_\triangleleft \gg \beta_\triangleleft$, we conclude $\beta_\triangleleft \in \sigma_1$. Thus, $\beta \in \chi^+(\epsilon)$, again, we arrived at a contradiction with the definition of $\gamma = lca(\alpha, \beta)$. Consider the case of ϵ having backward orientation, $\ell(\epsilon) = \Leftarrow$. Then, there is a firing sequence σ_3 with $(N, M_1)[\sigma_1\rangle(N, M_2)[\sigma_3\rangle(N, M_1)[\sigma_1\rangle(N, M_2)[\sigma_2\rangle(N, M_o)$, such that σ_3 contains solely transitions that are part of ϵ. Again, $\alpha_\triangleleft$ can be part of σ_3. From $\beta_\triangleleft \notin \sigma_2$ but $\alpha_\triangleleft \gg \beta_\triangleleft$, it follows $\beta_\triangleleft \in \sigma_1$ or $\beta_\triangleleft \in \sigma_3$. The latter would imply $\beta \in \chi^+(\epsilon)$ (a contradiction as above), which leads to $\beta_\triangleleft \in \sigma_1$. In order to ensure $\alpha_\triangleleft \gg \beta_\triangleleft$, *every* firing sequence σ_1 has to contain $\beta_\triangleleft$. Therefore, all children of fragment γ that represent paths from $\gamma_\triangleleft$ to $\gamma_\triangleright$, i.e., children with forward orientation, have to contain β. As β can only be contained in one child of fragment γ, there is actually only child with forward orientation.

We summarise that $b(\gamma) \neq B_\circ$, while $b(\gamma) = L$ implies that $\forall\, \vartheta \in \Theta\ [\ \beta \in \chi^+(\vartheta)\ ]$ with $\Theta = \{\vartheta \in \chi(\gamma) \mid \ell(\vartheta) = \Rightarrow\}$.

For both cases, $\gamma_\triangleleft$ being a transition or a place, we see that fragment γ does not satisfy the assumptions on a fragment $\omega \in (\Pi \setminus \{\beta\})$ as stated above. We now consider two cases, $\eta = \gamma$ or γ

is an ancestor of η. Due to $t(\delta) = P$, the former yields a contradiction, as $\Pi \setminus \{\beta\} = \{\gamma, \delta\}$ and both fragments do not satisfy our assumption. For γ being an ancestor for η, there is a fragment κ, such that $\kappa \in \chi(\gamma)$ and $\eta \in \chi^+(\kappa)$. Again, we distinguish two cases.

(1) $\kappa_{\lhd}$ is a transition. Then, $t(\kappa) \in \{P, B\}$, while $t(\kappa) = B$ requires $b(\kappa) = B_{\diamond}$.

(2) $\kappa_{\lhd}$ is a place. Now, we distinguish the three possible types of fragments for γ.

(I) If $t(\gamma) = P$, without loss of generality, we assume $\gamma_{\lhd}$ and $\gamma_{\lhd}$ to be places (the single places of their post-set or pre-set, respectively, would be taken if $\gamma_{\lhd}$ or $\gamma_{\lhd}$ would be a transition). Let $M_1, M_2 \in [N, M_i\rangle$ be two markings with $M_1(\gamma_{\lhd}) > 1$ and $M_2(\gamma_{\rhd}) > 1$. Let σ_1, σ_2 be two firing sequences with $(N, M_1)[\sigma_1\rangle(N, M_2)[\sigma_2\rangle(N, M_o)$, such that σ_2 does not contain any transition that is part of γ. Due to $t(\gamma) = P$, either $\kappa_{\rhd}\ F^+\ \epsilon_{\lhd}$ and $\epsilon_{\rhd}\ \cancel{F^+}\ \kappa_{\lhd}$, or vice versa. In both cases, $\alpha_{\lhd} \gg \beta_{\lhd}$ requires that a firing sequence σ_3 between two markings $M_3, M_4 \in [N, M_i\rangle$ with $M_3(\kappa_{\lhd}) > 1$ and $M_4(\kappa_{\rhd}) > 1$ contains $\beta_{\lhd}$. That is due to firing sequences leading from M_1 to M_3, or from M_4 to M_2 that contain no transition of fragment κ, but transition $\alpha_{\lhd}$.

(II) If $b(\gamma) = L$, we know that $\forall\ \vartheta \in \Theta\ [\ \beta \in \chi^+(\vartheta)\ ]$ with $\Theta = \{\vartheta \in \chi(\gamma) \mid \ell(\vartheta) =\Rightarrow\}$. As β can only be contained in one child of fragment γ, i.e., fragment κ, we know that $\ell(\kappa) =\Rightarrow$ and, in turn, $\ell(\epsilon) =\Leftarrow$. Let M_1, M_2, σ_1, and σ_2 be defined as for the previous case. Then, we may observe firing sequences σ_4, σ_5 with $(N, M_1)[\sigma_1\rangle(N, M_2)[\sigma_4\rangle$ (N, M_1) and $[\sigma_5\rangle(N, M_2)[\sigma_2\rangle(N, M_o)$, such that σ_4 contains $\alpha_{\lhd}$. Since $\alpha_{\lhd} \gg \beta_{\lhd}$, firing sequence σ_1 or σ_5 must contain $\beta_{\lhd}$. As in the previous case, it follows that any firing sequence σ_3 between two markings $M_3, M_4 \in [N, M_i\rangle$ with $M_3(\kappa_{\lhd}) > 1$ and $M_4(\kappa_{\rhd}) > 1$ must contain $beta_{\lhd}$.

(III) If $b(\gamma) = B_{\diamond}$, then $\epsilon_{\lhd}$ and $\kappa_{\lhd}$ are places in the post-set of transition $\gamma_{\lhd}$ (γ is a transition bordered bond). Let $M_5, M_6, M_7 \in [N, M_i\rangle$ be markings with $M_5(\kappa_{\lhd}) > 1$, $M_5(\epsilon_{\lhd}) > 1$, $M_6(\kappa_{\lhd}) > 1$, $M_6(\epsilon_{\rhd}) > 1$, $M_7(\kappa_{\rhd}) > 1$, $M_7(\epsilon_{\rhd}) > 1$. Then any firing sequence from M_5 to M_6 might contain $\alpha_{\lhd}$. Since $\alpha_{\lhd} \gg \beta_{\lhd}$, again, all firing sequences from M_6 to M_7 must contain $\beta_{\lhd}$.

For all three possible types of fragments for γ, we summarise that we have to ensure that any firing sequence leading from a marking that marks $\kappa_{\lhd}$ to a marking that marks $\kappa_{\rhd}$ must contain transition $\beta_{\lhd}$. Thus, for $\kappa_{\lhd}$ being a place, we know that $b(\kappa) \neq B_{\circ}$, while $b(\kappa) = L$ implies that $\forall\ \vartheta \in \Theta\ [\ \beta \in \chi^+(\vartheta)\ ]$ with $\Theta = \{\vartheta \in \chi(\kappa) \mid \ell(\vartheta) =\Rightarrow\}$ (cf., the argument for the very first case (2), if ϵ would be an arbitrary child of κ).

For both cases, $\kappa_{\lhd}$ being a transition or a place, fragment κ does not satisfy the assumptions on a fragment $\omega \in (\Pi \setminus \{\beta\})$ stated above. As this argument can be applied to all fragments on the path $\pi_{\mathcal{T}}(\kappa, \eta)$, we arrived at a contradiction with our assumption.

$\Leftarrow$ Let $\forall\ \omega \in (\Pi \setminus \{\beta\})$ either $t(\omega) = P$ or $b(\omega) = B_{\diamond}$, or if $(b(\omega) = L$ then $\forall\ \vartheta \in (\chi(\omega) \cap \Pi)\ [\ \ell(\vartheta) =\Rightarrow\ ])]$ with $\Theta = \{\vartheta \in \chi(\omega) \mid \ell(\vartheta) =\Rightarrow\}$. Assume $\alpha_{\lhd} \not\gg \beta_{\lhd}$. With δ as defined above, one path $\pi_N(\delta_{\lhd}, \beta_{\lhd})$ is forward conflict-free, i.e, $\delta_{\lhd} \gg \beta_{\lhd}$ according to Lemma 6.1. Regarding fragment η, we distinguish two cases.

(1) $\eta_{\lhd}$ is a transition. Then, $t(\eta) \in \{P, B\}$, while $t(\eta) = B$ requires $b(\eta) = B_{\diamond}$. Both imply that one path $\pi_N(\eta_{\lhd}, \delta_{\lhd})$ is forward conflict-free, i.e., $\eta_{\lhd} \gg \delta_{\lhd}$. With $\delta_{\lhd} \gg \beta_{\lhd}$ we also get $\eta_{\lhd} \gg \beta_{\lhd}$.

(2) $\eta_{\lhd}$ is a place. Then, $t(\eta) \in \{P, B\}$, while $t(\eta) = B$ requires $b(\eta) = L$. For $t(\eta) = P$, we get $t \gg \delta_{\lhd}$ for all $t \in\ \bullet\eta_{\lhd}$. For $t(\eta) = B$, we have $b(\eta) = L$ and $\forall\ \vartheta \in \Theta\ [\ \beta \in \chi^+(\vartheta)\ ]$ with $\Theta = \{\vartheta \in \chi(\eta) \mid \ell(\vartheta) =\Rightarrow\}$. As only one child of fragment η can contain fragment β, i.e.,

fragment δ, we know $|\Theta| = 1$. That is, there is only one path from $\eta_\triangleleft$ to $\eta_\triangleright$, represented by fragment δ. Therefore, $t \gg \delta_\triangleleft$ for all $t \in \bullet\eta_\triangleleft$. For both cases, $t(\eta) = P$ or $t(\eta) = B$, it also holds $t \gg \beta_\triangleleft$ for all $t \in \bullet\eta_\triangleleft$, since $\delta_\triangleleft \gg \beta_\triangleleft$.

We summarise that for both cases (1) and (2), we derive either $\eta_\triangleleft \gg \beta_\triangleleft$, or $t \gg \beta_\triangleleft$ for all $t \in \bullet\eta_\triangleleft$, respectively. Applying this argument to all fragments on the path $\pi_\mathcal{T}(\gamma, \eta)$ yields $\gamma_\triangleleft \gg \beta_\triangleleft$ or $t \gg \beta_\triangleleft$ for all $t \in \bullet\gamma_\triangleleft$, respectively. Trivially, $\alpha_\triangleleft \gg \gamma_\triangleleft$ if $\gamma_\triangleleft$ is a transition or $\alpha_\triangleleft \gg t$ for all $t \in \bullet\gamma_\triangleleft$ if $\gamma_\triangleleft$ is a place, due to γ being an ancestor of α. Thus, $\alpha_\triangleleft \gg \beta_\triangleleft$, which is a contradiction. □

We illustrate both propositions using our example from Figure 6(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., Figure 6(b) and Figure 8. 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 Figure 6(b) or $B_\circ 1$ in Figure 8, 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 Figure 6(b). This fragment corresponds to the loop type fragment $L1$ in Figure 8. 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 Figure 6(b), $L1$ in Figure 8) 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 6.2.

Using these propositions, computation of the causal behavioural profile for a pair of transitions in a sound free-choice WF-system can be done efficiently.

Corollary 6.1. 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 6.1 and 6.2, 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 [9, 34, 54]. □

6.2. 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 [66], 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. This may imply computational overhead as various transitions may be irrelevant for analysis. For instance, in Figure 1 transitions B, C, D, E, and I in model (a) are without counterpart in model (b). Hence, information of the behavioural profile for these transitions is not needed in order to evaluate the consistency between both models. 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 of rigid fragments, namely sound workflow T- and S-systems, and sound free-choice WF-systems that are acyclic. First, the co-occurrence relation for sound workflow T-systems is derived as follows.

Lemma 6.2. 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 forward conflict-free. Thus, $t_i \gg t$ (Lemma 6.1). 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 Figure 6(a), we see that Lemma 6.2 suffices to derive the co-occurrence relation for all pairs of transitions that are part of the rigid fragment. The subnet represented by fragment $R1$ in Figure 6(b) and Figure 8 is a T-Net, such that all transitions inside are pairwise co-occurring (e.g., $F \gg J$ and $J \gg F$).

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 6.3. 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 claim holds by following the argument for the dominator in reverse direction. 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.

□

For the more general case of sound free-choice WF-systems that are acyclic, the co-occurrence relation can be traced back to the exclusiveness relation. It is easy to see that two transitions that are exclusive to each other are not co-occurring.

Lemma 6.4. In a sound free-choice WF-system holds, two transitions x and y, which are not exclusive ($x \not+ y$) and 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.

Proof:
Let (N, M_i) be a sound WF-system and $x, y \in T$ two transitions with $x \not+ y$, and $y \not F^+ y$. We need the following implications for free-choice sound WF-systems that have been proved in [66].

- ○ Strict order $x \rightsquigarrow y$ implies $x\ F^+\ y$ and $x \not F^+ y$.
- ○ Reverse strict order $x \rightsquigarrow^{-1} y$ implies $y\ F^+\ x$ and $y \not F^+ x$.
- ○ Interleaving order $x || y$ implies either $y\ F^+\ x$ and $y\ F^+\ x$, or there is a marking reachable from the initial marking that enables both transitions.

($\Leftarrow$) Let $t + y \Rightarrow t + x$ for all transitions $t \in T$ and assume $x \not\gg y$. The relations of the behavioural profile partition the set $T \times T$. As we also know $x \not+ y$ we distinguish three cases of how x and y might be related according to the profile.

$x \rightsquigarrow y$ We know $x\ F^+\ y$ and $x \not F^+ y$, such that there is a path $\pi_N(x, y)$. If any path $\pi_N(x, y)$ is forward conflict-free, this yields $x \gg y$ according to Lemma 6.1, a contradiction with our assumption. If there is no path $\pi_N(x, y)$ that is forward conflict-free, there is a $p \in P$ with $p \in \pi_N(x, y)$ for some $\pi_N(x, y)$, such that $|p\bullet| > 1$. If $y \in p\bullet$, we know that there is another transition $t_y \in p\bullet$ with $t_y + y$ due to free-choiceness of the net and $y \not F^+ y$. From $x\ F^+\ t_y$, we get $x \not+ t_y$ (cf., Lemma 3 in [66]), a contradiction. If $y \notin p\bullet$, let $t_1 \in p\bullet$ be a transition. We know $x\ F^+\ t_1$ and, therefore, $x \not+ t_1$. As $y + t_1$ would imply $x + t_1$, we derive $y \not+ t_1$. Thus, it holds either $t_1 \rightsquigarrow y$, $t_1 \rightsquigarrow^{-1} y$, or $t_1 || y$.

$(t_1 \rightsquigarrow^{-1} y)$ We know $y\ F^+\ t_1$ and $t_1 \not F^+ y$. As p is in $\pi_N(x, y)$, we have $p\ F^+\ y$. Thus, there must be a transition $t_2 \in p\bullet$ with $t_2\ F^+\ y$. From $y\ F^+\ t_1$, we get $y\ F^+\ p_1$ for some $p_1 \in \bullet t_1$. Due to the free-choiceness of the net, t_1 and t_2 share all places in their pre-set, such that also $p_1\ F^+\ y$, which yields a contradiction with $y \not F^+ y$.

$(t_1 || y)$ We know either $y\ F^+\ t_1$ and $t_1\ F^+\ y$, or there is a marking that enables both y and t_1. The former is not in line with the assumption of $y \not F^+ y$. The latter is not possible either: let $(N, M)[y\rangle$ and $(N, M)[t_1\rangle$ for some $M \in [N, M_i\rangle$. Due to $p\ F^+\ y$ either also $p \in \bullet y$ or the path implies a firing sequence $(N, M)[\sigma\rangle(N, M_2)$ (property of sound free-choice systems), such that all places of the pre-set of y are marked at least twice. In both cases, the safeness property that holds for sound free-choice systems would be violated.

Therefore, it holds $t_1 \rightsquigarrow y$ for all $t_1 \in p\bullet$ for some $p \in P$ and $\pi_N(x, y)$ with $p \in \pi_N(x, y)$ and $|p\bullet| > 1$. Thus, it also holds $t_1\ F^+\ y$ and $y \not F^+ t_1$ for all these transitions t_1. Now, either one path $\pi_N(t_1, y)$ is forward conflict-free, which yields $t_1 \gg y$ according to Lemma 6.1, or there is a place $p_2 \in P$ with $p_2 \in \pi_N(t_1, y)$ for some $\pi_N(t_1, y)$, such that $|p_2 \bullet| > 1$. In this case, the argument for p can be applied recursively for p_2, as for all transitions $t_2 \in p_2\bullet$ it holds $t_2\ F^+\ y$. Note that the recursive step is only initiated, if the respective place has not been visited before (which might be the case due to control flow cycles). Consequently, we arrive at $t_1 \gg y$ for all transitions $t_1 \in p\bullet$ for some $p \in P$ and $\pi_N(x, y)$ with $p \in \pi_N(x, y)$ and $|p \bullet| > 1$. Therefore, we deduce $x \gg y$, a contradiction.

$(x \rightsquigarrow^{-1} y)$ The argument for the previous case $(x \rightsquigarrow y)$ is followed in reverse direction leading to a contradiction.

$(x || y)$ We know either $x\ F^+\ y$ and $y\ F^+\ x$, or there is a marking that enables both x and y. Again, the former is not in line with the assumption of $y \not{F^+} y$. Now, we consider two cases, whether or nor there is a path $\pi_N(i, y)$ that is forward conflict-free. If so, it holds $i \gg y$, i.e., all firing sequences starting in M_i and leading to M_o contain transition y, such that the assumption of $x \not\gg y$ is violated. If not, there is a $p \in P$ with $p \in \pi_N(i, y)$ for some $\pi_N(i, y)$, such that $|p \bullet| > 1$. For such a place p, we prove two properties.

1. If $p\ F^+\ x$, then for all $t_1 \in p\bullet$ it holds $t_1 \not{F^+} y \Rightarrow t_1 \not{F^+} x$. Assume that this implication does not hold, i.e., there is a $t_1 \in p\bullet$ with $t_1 \not{F^+} y$ and $t_1\ F^+\ x$. From $p\ F^+\ y$ we know that there must be a $t_2 \in p\bullet$ with $t_2 = y$ or $t_2\ F^+\ y$. The former leads to $t_1 + y$ due to $y \not{F^+} y$. Therefore, it holds $t_1 + x$, yielding a contradiction with $t_1\ F^+\ x$. In case of $t_2\ F^+\ y$, we know $y \not{F^+} p$ from $y \not{F^+} y$. Further, $y \not{F^+} p$ implies $y \not{F^+} t_1$. Thus, either $y + t_1$ or $y || t_1$. The latter implies the existence of a marking $M \in [N, M_i\rangle$ with $(N, M)[y\rangle$ and $(N, M)[t_1\rangle$. Due to $p\ F^+\ y$ this would violate the safeness property of sound free-choice systems. Thus, $y + t_1$ and, therefore, $x + t_1$, a contradiction with $t_1\ F^+\ x$.
2. If $p \not{F^+} x$, then for all $t_1 \in p\bullet$ it holds $t_1\ F^+\ y$. Assume that this is not the case, i.e., there is a $t_1 \in p\bullet$ with $t_1 \not{F^+} y$. From $y \not{F^+} y$ we get $y \not{F^+} p$ and, therefore, $y \not{F^+} t_1$. As for the previous property, $y || t_1$ would violate safeness of the system. Thus, $y + t_1$. From $p \not{F^+} x$, we get $t_1 \not{F^+} x$, while $x \not{F^+} t_1$ holds as well in order to satisfy $x \not{F^+} y$. Thus, either $t_1 + x$ or $t_1 || x$. There is a marking $M \in [N, M_i\rangle$ with $(N, M)[y\rangle$ and $(N, M)[x\rangle$, hence, there is also a marking $M \in [N, M_i\rangle$ with $(N, M)[t_1\rangle$ and $(N, M)[x\rangle$, as $p\ F^+\ y$ and $t_1 \in p\bullet$. Thus, it holds $t_1 || x$, which yields a contradiction as $t_1 + y$ requires $t_1 + x$.

Now, consider all conflicts that might lead to y not being part of a firing sequence starting in M_i and leading to M_o. These conflicts are places p on a path $\pi_N(i, y)$ with $|p \bullet| > 1|$. If $p\ F^+\ x$, the first property ensures that if y will not be part of the firing sequence due to firing of $t_1 \in p\bullet$ with $t_1 \not{F^+} y$, x cannot be part either, that is, $t_1 \not{F^+} x$ holds true. We also know that x and y are enabled concurrently in some marking. Thus, once there is a conflict at place p on a path $\pi_N(i, y)$ and $p \not{F^+} x$, it has to be ensured that y is fired eventually. Here, the second property guarantees $t_1\ F^+\ y$ for all $t_1 \in p\bullet$. That, in turn, implies $t_2 \gg t_1$ for all $t_2 \in \bullet p$ and, as the property holds for all respective places p, also $t_2 \gg y$. Consequently, it holds $x \gg y$, a contradiction with our assumption.

$\Rightarrow$ Let $x \gg y$ and assume that there is a transition $t \in T$ with $t + y$ and $t \not+ x$. Due to $t \not+ x$, there is a firing sequence σ with $(N, M_i)[\sigma\rangle(N, M_o)$ that contains both transitions, t and x. From $x \gg y$, we know that also $y \in \sigma$. Thus, $x, y, t \in \sigma$ is a contradiction with the assumption of $t + y$.

□

Based on these results, computation of causal behavioural profiles is done efficiently.

Corollary 6.2. 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 [66] (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 6.2). In case of an S-system, dominators and post-dominators are determined in linear time [7]. Based thereon, co-occurrence is decided based on Lemma 6.3. For the case of acyclic free-choice WF-systems, co-occurrence is traced back to exclusiveness according to Lemma 6.4. 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. □

6.3. Complete Computation Algorithm

This section organizes the results of the previous two sections into a comprehensive algorithm. The algorithm expects a sound free-choice WF-system and a pair of transitions as input. Given the input, the algorithm determines the profile relation (the relation according to the behavioural profile) and checks the co-occurrence relation for the pair of transitions. Besides the already presented theory, the algorithm exploits the result of Lemma 6.5.

Lemma 6.5. Let $S = (N, M_0)$, $N = (P, T, F)$, be a sound free-choice WF-system and let $\mathcal{T}_N = (\Omega, \chi, t, b)$ be the WF-tree of N. Let $\omega = (P', T', F') \in \Omega$ be a fragment of N. If there exists a path $\pi_N(\omega_\triangleright, \omega_\triangleleft)$, then $t_1 \mid\mid t_2$, for all $t_1, t_2 \in T'$.

Proof:
The existence of a path $\pi_N(x, y)$, where $x, y \in T \cup P$, implies the existence of a firing sequence containing all transitions on $\pi_N(x, y)$ (cf., Lemma 4.2 in [36]). The claim immediately holds from the fact that there exist two paths: $\pi_N(\omega_\triangleright, \omega_\triangleleft)$ and $\pi_N(\omega_\triangleleft, \omega_\triangleright)$. □

Lemma 6.5 defines a link between the interleaving order relation for a pair of transitions and the existence of a cyclic path that contains both these transitions. Finally, Algorithm 1 details the steps to take when computing behavioural relations for a pair of transitions.

Algorithm 1 comprises three stages: First, required data structures are initialized (lines 1–5). Second, computation of the profile relation for the given pair of transitions takes place (lines 6–14). Last, the pair of transitions is checked for being in the co-occurrence relation (lines 15–26). If there exists no rigid fragment on the path from the root of the WF-tree to fragment γ, the profile relation is derived using Proposition 6.1 (line 7); otherwise, the algorithm checks if given transitions are in the interleaving order relation using Lemma 6.5 (line 9), or computes profile relations for the whole fragment γ (line 11) and extracts the relation for the requested pair of transitions (line 12). The check of the co-occurrence relation, in the absence of rigid fragments in the WF-tree on the path from γ to β, relies on Proposition 6.2; otherwise the checks depend on the structural class of fragment γ (lines 18-23). Lastly, if γ is cyclic and

Algorithm 1: Compute Behavioural Relation for a Pair of Transitions

```
   Input: A sound free-choice WF-system $(N, M_i)$, $N = (P, T, F)$ and $a, b \in T$ two transitions of
          $N$
   Output: Relation of the behavioural profile and the co-occurrence relation for $a$ and $b$
1  $\mathcal{T}_N = (\Omega, \chi, t, b)$—the WF-tree of $N$
2  $\alpha \in \Omega$—a trivial fragment with entry $a$
3  $\beta \in \Omega$—a trivial fragment with entry $b$
4  $\omega_r \in \Omega$—the root fragment of $\mathcal{T}_N$
5  $\gamma = lca(\alpha, \beta)$
   // Compute profile relation
6  if $\forall\, \omega \in \pi_{\mathcal{T}}\{\omega_r, \gamma\}\ [\, t(\omega) \neq R\, ]$ then
7      Get profile relation for $a$ and $b$ using Proposition 6.1
8  else
9      Use Lemma 6.5 on subnet $\gamma$ of $N$ to check interleaving order relation for $a$ and $b$
10     if not $a \mid\mid b$, according to Lemma 6.5 then
11         $BP$—behavioural profile of subnet $\gamma$, computed as in [65]
12         Get profile relation for $a$ and $b$ in $BP$
13     end
14 end
   // Check co-occurrence relation
15 if $\forall\, \omega \in \pi_{\mathcal{T}}\{\gamma, \beta\}\ [\, t(\omega) \neq R\, ]$ then
16     Check co-occurrence relation for $a$ and $b$ using Proposition 6.2
17 else
18     if $\gamma$ is a T-net then
19         Check co-occurrence relation for $a$ and $b$ using Lemma 6.2
20     else if $\gamma$ is an S-net then
21         Check co-occurrence relation for $a$ and $b$ using Lemma 6.3
22     else if $\gamma$ is acyclic then
23         Check co-occurrence relation for $a$ and $b$ using Lemma 6.4
24     else
25         Perform state space exploration to check co-occurrence relation for $a$ and $b$
26     end
27 end
```

free-choice, the check of the co-occurrence relation must rely onto costly state space exploration, which can be integrated in the approach proposed in [65] (line 25).

7. Evaluation

To evaluate our approach of deriving behavioural characteristics, we implemented the computation of causal behavioural profiles based on WF-trees and conducted a series of experiments. We took three model collections from industry and applied our approach to them. On the one hand, this provides us with insights on the characteristics found for models in an industry setting. Based thereon, we are able to judge on the impact of our assumptions with respect to free-choiceness and soundness. On the other hand, the absolute computation times illustrate for which sizes of models our approach yields instantaneous creation of causal behavioural profiles. Section 7.1 to 7.3 discuss the three experiments in detail. All experimental results were obtained using the same IT setup.

7.1. The SAP Reference Model

The SAP reference model [13] describes the functionality of the SAP R/3 system. It comprises 604 process diagrams with 20 nodes on average [46]. Those are expanded to 737 models in EPC notation as some diagrams contain multiple disconnected EPCs. These EPC models capture different functional aspects of an enterprise, such as sales or accounting.

Model characteristics. From the models of the SAP reference model, we excluded 23 models that are trivial, i.e., they consist of one element. Further, four models were not considered due to syntax errors that cannot be interpreted unambiguously. Events or functions with more than one incoming or outgoing flow arc are examples for these errors. 169 models show ambiguous instantiation semantics and could, therefore, not be considered either, cf., [16]. It is well-known that the SAP reference model contains behavioural anomalies, such as deadlocks and livelocks [20, 48]. To meet our requirement of soundness, 84 models have been excluded from our experiment due to behavioural errors. We also normalised multiple start and end events, so that each model had a unique entry point and a unique exit point. Some of the models comprise OR-connectors that cannot be mapped to free-choice Petri net constructs. Therefore, we replaced block-structured 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 [38]. When applying our approach, we encountered two WF-systems that contained a rigid fragment. Both fragments could be mapped to an S-system and, therefore, be handled as introduced in Section 6.2.

Computation results. 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 transitions of the WF-systems. Figure 10 shows the average computation time for each model group in three experiment runs. First, we computed the behavioural profile using the approach introduced in [66] (BP-Net). Second, we derived the same profile using the WF-trees as introduced in this article (BP-Tree). Third, we computed the causal behavioural profile (including co-occurrence) using WF-trees (CBP). For all three computations, Figure 10 depicts the polynomial least squares regressions.

For this model collection, we see that the use of WF-trees as introduced in this article speeds up the computation of the behavioural profile significantly compared to the existing approach. In addition, the overhead implied by our extension to the behavioural profile yielding the causal behavioural profile is negligible. Moreover, the close approximation of our values by the respective regression lines indicates that the relation between computation times and model sizes is very homogeneous. Finally, computation is done in tens of milliseconds even for the biggest models of this collection – a prerequisite for instantaneous consistency analysis.

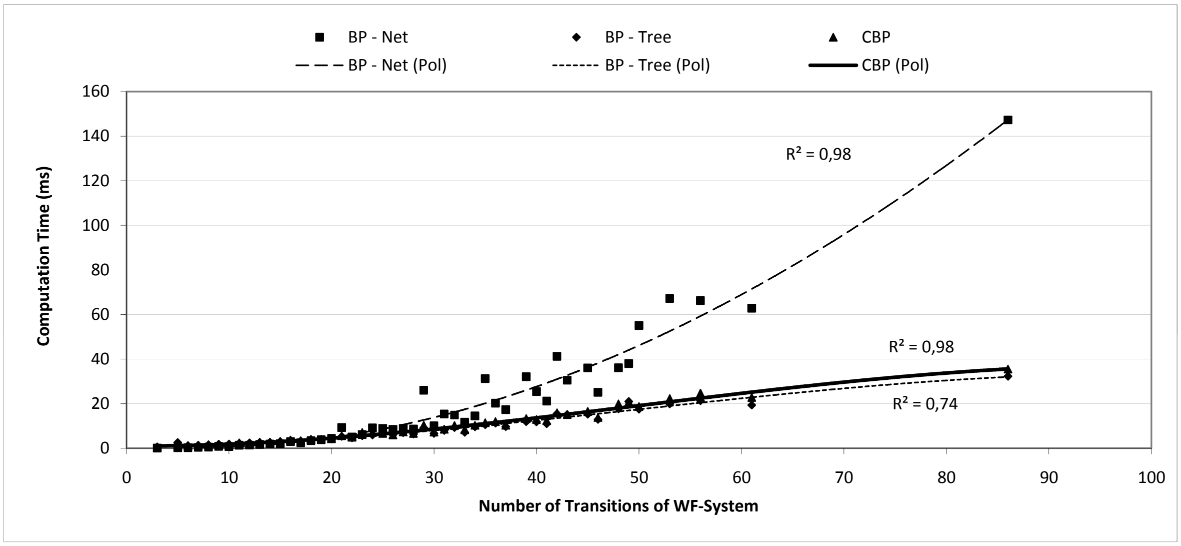


Figure 10. SAP reference model: computation time relative to the size of the EPC model.

7.2. The BIT Process Library

The BIT process library comprises process models that were created in process automation projects in the various industry domains, such as financial services, automotive, telecommunications, construction, supply chain, health care, and customer relationship management. These models were collected and used for a study on soundness verification [28]. Originally, all models were captured in the IBM WebSphere Business Modeler in a notation similar to UML activity diagrams. The authors of [28] already provided Petri net formalisations for all models.

Model characteristics. For the experiment, we used the parts A, $B3$, and C of the model collection, a set of 735 unique models. All models of the collection were already available as free-choice WF-systems, one of the assumptions of our approach. As reported in [28], around half of the systems were not sound. Hence, we had to exclude 361 systems due to behavioural anomalies. For the remaining systems, we assessed whether our approach was applicable. That is, we checked 145 systems that comprised rigid fragments, whether these rigids are acyclic or could be traced back to S- or T-systems. For 13 out of 145 systems this was not the case, so that the respective systems were not included in the experiment. Therefore, 361 WF-system were considered in the experiment.

Computation results. Again, we computed the (non-causal and causal) behavioural profiles for all transitions of all 361 WF-systems and grouped the systems according to their size, i.e., the number of transitions of the WF-systems. Figure 11 shows the average computation time for each group of systems up to a size of 100 transitions. Only one system is larger in size and comprises 285 transitions. The results for this system are not visualised in Figure 11. As in the previous experiment, behavioural profiles were computed with the existing approach (BP-Net) and the approach based on WF-trees (BP-Tree). Moreover, the causal behavioural profile (including co-occurrence) was also computed using WF-trees (CBP). For all three computations, Figure 11 depicts the polynomial least squares regressions.

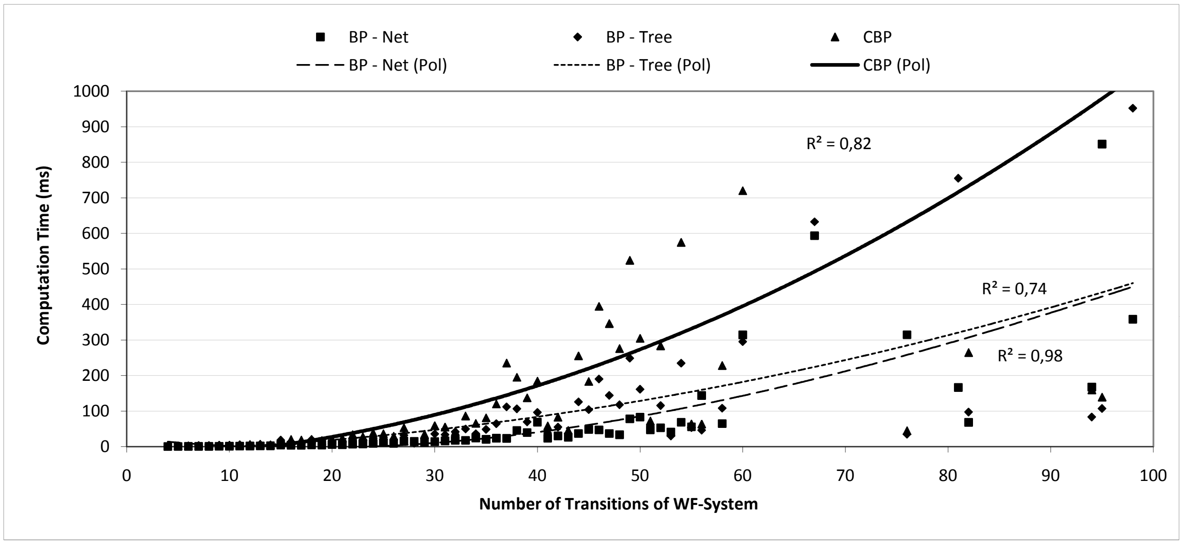


Figure 11. BIT Process Library: computation time relative to the size of WF-system.

The plot illustrates that, for this model collection, the use of WF-trees does not improve the time needed to compute behavioural profiles. The existing approach and the approach based on WF-trees yield comparable results. In contrast to the model collection of the SAP reference model, the overhead needed to compute the co-occurrence relation of the causal behavioural profile is not negligible. Computation of causal behavioural profiles takes significantly more time than the computation of behavioural profiles. Nevertheless, the absolute computation times are above one second only for the very big model that is not considered in Figure 11. For this particular model, all computations took up to 25 seconds. Another observation relates to the dependency of computation times and the model size. For the WF-tree based computations, the regressions show a weaker approximation as in case of the SAP reference model.

7.3. Process Models from a Health Insurance Company

The models used for this experiment have been provided by a health insurance company. These models describe the business functions from an organisational perspective and have mainly been applied for staff planning. The collection comprises 1029 process diagrams in EPC notation. Some diagrams contain more than one process model, so that these diagrams could be expanded to 1350 models.

Model characteristics. From the models of this collection, one model had to be excluded due to a syntax error and four models turned out to be not sound. For the remaining EPC models we tried to normalise multiple start and end events, so that each model had a unique entry point and a unique exit point. For six models, normalisation of end events was not possible. Further, all 1014 models were created using XOR- and AND-connectors, which allows for a direct translation of the EPC models into free-choice WF-systems [38]. It is worth to mention that the amount of concurrency in the behaviour of the systems was rather low. A small number of systems, 48 out of 1014, showed concurrency. Further, we assessed whether our assumptions regarding rigid fragments are met by the WF-systems. We detected 142

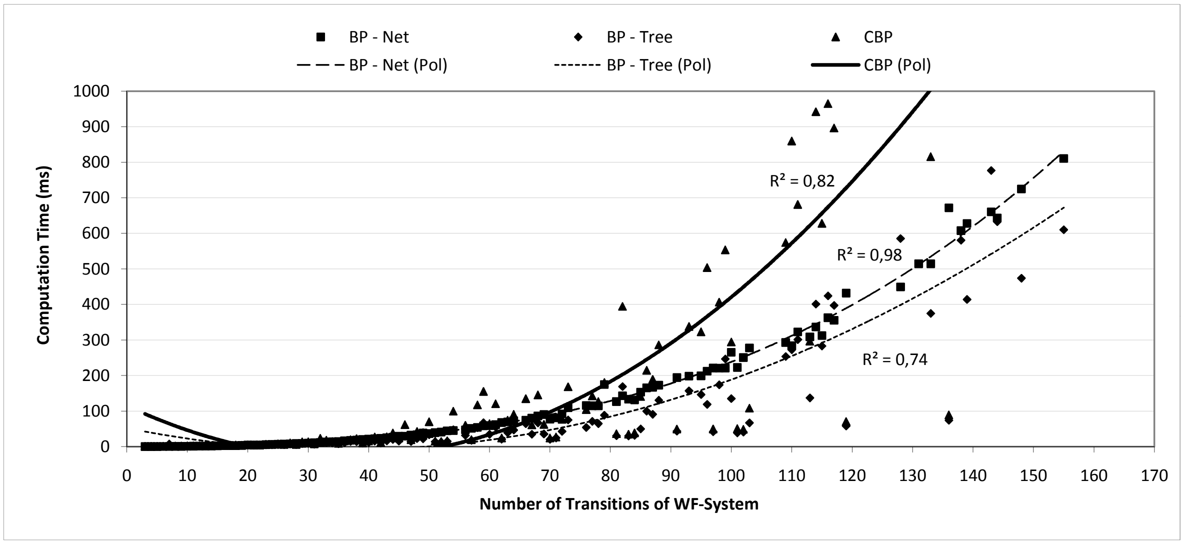


Figure 12. Process models from a health insurance company: computation time relative to the size of WF-system.

systems that comprised a rigid fragment. However, in all cases these fragments were acyclic or could be traced back to S- or T-systems, respectively. Hence, our approach was applicable for all of these systems.

Computation results. As in the previous experiments, we computed the (non-causal and causal) behavioural profiles for all transitions of the 1014 WF-systems. We grouped all systems according to their size. Figure 12 shows the average computation time for each group of systems up to a size of 170 transitions. The biggest system contained 456 transitions, whereas only a few system had more than 170 transitions and are not considered in Figure 12. Again, behavioural profiles were computed with the existing approach (BP-Net) and the approach based on WF-trees (BP-Tree). Computation of causal behavioural profiles based on WF-trees corresponds to the data series CBP. For all three computations, the polynomial least squares regressions are also depicted in Figure 12.

The results obtained for the previous experiments, are close to those derived for this model collection. Computation of the behavioural profile is more efficient using the WF-trees introduced in this article compared to the existing approach. We observe a computational overhead induced by the co-occurrence relation of the causal behavioural profile, even though it appears to be smaller than in case of the BIT process library. Further observations are similar as obtained for the BIT process library. Absolute computation times above one second are observed only for the very big models not considered in the plot. Computations for these models took up to tens of seconds.

8. Related Work

Clearly, our work relates to other behavioural relations that have been defined for Petri nets. We discussed causal behavioural profiles in the light of relations proposed for workflow mining [6], the well-known concurrency relation [39], and Petri net unfoldings [44, 25] already in Section 3.3. Sill, their relation to common notions of behaviour 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 [55, 29] has three major drawbacks. First and foremost, these notions yield a true or false answer, which has been criticised in [45]. 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 [30], 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. Work on equivalence-preserving refinements for Petri nets, refer to [11] for a thorough survey, illustrates that common notions of equivalence are preserved solely under certain refinement operators. Similarly, work on net morphisms [74] and behaviour inheritance [8, 59] shows that any extension of a net has to be done in a structured manner in order to preserve common equivalences. Third, notions of behaviour 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 [57]. 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 [57] 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 overlap of the relations of the behavioural profiles of two related systems can be used as a behavioural similarity measure. The question of *behavioural similarity* has been addressed from various angles, see [22] for an overview. In [50] an approach for merging statechart specifications is introduced. In order to realise the merge operation, the authors consider the preservation of bisimilarity between the statecharts in the similarity score. Similarity measures can also be based on edit distances between behavioural models [75, 76]. There are different groundings for such an edit distance, among them the language of the model, the underlying automaton, or the n-gram representation of the language. Other approaches to behavioural similarity are based on change operations needed to transform one behavioural model into another [41]. Similar to the concept of causal behavioural profiles are causal footprints [21]. These footprints represent a behavioural abstraction that has also been applied for determining the similarity between processes. Note that most of these approaches require a reasoning on the state space of the model, whereas the techniques introduced for causal behavioural profiles are purely structural.

Close to the question of behavioural similarity are conformance measures for process logs. Various conformance measures have been proposed in the literature [31, 32, 45, 58, 73]. All of these measures ground on state-based techniques that involve replaying process logs. Conformance may be computed as the share of process logs that can be replayed in the process model [32, 73]. Other approaches try to replay a process log step-wise in the process model [31, 45, 58]. The number of execution steps that are in line with the semantics induced by the process model is quantified and used as a conformance measure. Due to their grounding in state-based concepts, all of these approaches have to cope with the state explosion problem. Hence, the application of heuristics that have to be tuned for a specific setting is required to achieve efficient computation. In contrast, conformance measures that can be computed efficiently may be grounded on causal behavioural profiles as proposed in this article.

Related work includes further applications of the tree-based decomposition for behavioural models. The RPST has been introduced in [60] with the focus on a mapping between process languages. Tree-based decompositions have also been used to refactor process models [52, 63]. Among other applications, tree-based process model decompositions have also been used for control-flow analysis [35, 62], process comparison [40], pattern application in process modelling [33], and process model abstraction [53].

9. Conclusions

In this article, we addressed the problem of behavioural consistency with a notion that is weaker than existing notions of behaviour equivalence, but can be computed efficiently. Our contribution is the definition of a causal behavioural profile that captures essential behavioural characteristics of a workflow system. We discussed two fields of applications for these profiles. Consistency between related systems can be quantified based on the relations of the causal behavioural profile once correspondences have been identified. We also discussed how causal behavioural profiles form the basis of a conformance measure for process logs. Further, we showed the efficient computation of these profiles 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-systems. 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 traced back to a cyclic S-system). We evaluated our approach for the computation in three experiments using model collections from industry. Here, we saw that our assumptions are mostly satisfied by process models that can be observed in practice. That is, in two collections all models have been free-choice and only a few models per collection could not be transformed into a workflow system structure. Further, for only 13 out of nearly 2,000 WF-systems our assumptions on the structure of rigid fragments were not met, which prevented us from computing the causal behavioural profile for these systems. In contrast to these structural assumptions, we observed that the soundness property is often violated in industry models. Still, it may be argued that these models are erroneous, so that behavioural anomalies should be resolved before these models are the basis for any consistency analysis. Finally, the low polynomial complexity of our algorithms has been visible also in the absolute computation times obtained in our experiments. Apart from exceptionally large models, computation of causal behavioural profiles is done in tens to hundreds of milliseconds. This opens reasoning on behavioural consistency to industrial applications where state-based approaches do not scale.

Even though our experiments indicate that our assumptions hold for most process models observed in practice, there are models that cannot be traced back to sound free-choice WF-systems or that do not meet our assumptions on the characteristics of rigid fragments. Process modelling languages such as BPMN or BPEL provide means to express exception handling and event-based routing. Petri net-based formalisations of these concepts typically result in non-free-choice net systems, see [43]. Therefore, we need techniques for computing causal behavioural profiles for a broader class of behavioural models. In [65], we took a first step in this direction by computing behavioural profiles from the complete prefix unfolding of a net system, which is a compact representation of the system's state space. This approach is computationally hard since the construction of a complete prefix unfolding cannot be done efficiently in the general case. Still, the approach may be leveraged only for subnets, for which the assumptions underlying the techniques presented in this work are violated. Hence, the expensive approach would be used for certain parts of a net system, which are likely to be significantly smaller than the overall system.

In future work, we aim at lifting the approach based on complete prefix unfoldings to the co-occurrence relation of the causal behavioural profile.

Further work is also needed on the application of causal behavioural profiles to measure consistency between related net systems. Up to now, we assume consistency to be a symmetric concept. This assumption may be challenged in the context of a system specification and its implementation. Further investigations are required to judge on the validity of this assumption.

References

[1] van der Aalst, W. M. P.: Verification of Workflow Nets, *ICATPN*, 1248, 1997, 407–426.

[2] van der Aalst, W. M. P.: The Application of Petri Nets to Workflow Management, *Journal of Circuits, Systems, and Computers*, **8**(1), 1998, 21–66.

[3] van der Aalst, W. M. P.: Workflow Verification: Finding Control-Flow Errors Using Petri-Net-Based Techniques, *BPM*, 1806, 2000, 161–183.

[4] van der Aalst, W. M. P., Hirnschall, A., Verbeek, H. M. W.: An Alternative Way to Analyze Workflow Graphs, *CAiSE*, 2348, 2002, 535–552.

[5] van der Aalst, W. M. P., Reijers, H. A., Weijters, A. J. M. M., van Dongen, B. F., de Medeiros, A. K. A., Song, M., Verbeek, H. M. W. E.: Business process mining: An industrial application, *Inf. Syst.*, **32**(5), 2007, 713–732.

[6] van der Aalst, W. M. P., Weijters, T., Maruster, L.: Workflow Mining: Discovering Process Models from Event Logs, *IEEE Trans. Knowl. Data Eng.*, **16**(9), 2004, 1128–1142.

[7] Alstrup, S., Harel, D., Lauridsen, P. W., Thorup, M.: Dominators in Linear Time, *SIAM J. Comput.*, **28**(6), 1999, 2117–2132.

[8] Basten, T., van der Aalst, W. M. P.: Inheritance of Behavior, *JLAP*, **47**(2), 2001, 47–145.

[9] Battista, G. D., Tamassia, R.: On-Line Maintenance of Triconnected Components with SPQR-Trees, *Algorithmica*, **15**(4), 1996, 302–318.

[10] Best, E.: Structure Theory of Petri Nets: the Free Choice Hiatus, *Advances in Petri Nets* (W. Brauer, W. Reisig, G. Rozenberg, Eds.), 254, Springer, 1986, ISBN 3-540-17905-4, 168–205.

[11] Brauer, W., Gold, R., Vogler, W.: A survey of behaviour and equivalence preserving refinements of Petri nets, *Applications and Theory of Petri Nets* (G. Rozenberg, Ed.), 483, Springer, 1989, ISBN 3-540-53863-1, 1–46.

[12] Bravetti, M., Bultan, T., Eds.: *Web Services and Formal Methods - 7th International Workshop, WS-FM 2010, Hoboken, NJ, USA, September 16-17, 2010. Revised Selected Papers*, vol. 6551 of *Lecture Notes in Computer Science*, Springer, 2011, ISBN 978-3-642-19588-4.

[13] Curran, T. A., Keller, G., Ladd, A.: *SAP R/3 Business Blueprint: Understanding the Business Process Reference Model*, Prentice-Hall, 1997.

[14] 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.

[15] Dayal, U., Eder, J., Koehler, J., Reijers, H. A., Eds.: *Business Process Management, 7th International Conference, BPM 2009, Ulm, Germany, September 8-10, 2009. Proceedings*, vol. 5701 of *Lecture Notes in Computer Science*, Springer, 2009, ISBN 978-3-642-03847-1.

[16] Decker, G., Mendling, J.: Process Instantiation, *Data Knowl. Eng.*, **68**, 2009.

[17] Desel, J., Esparza, J.: *Free Choice Petri Nets*, Cambridge University Press, 1995, ISBN 0-521-46519-2.

[18] Dijkman, R., Dumas, M., García-Bañuelos, L., Käärik, R.: Aligning Business Process Models, *13th International IEEE Enterprise Distributed Object Computing Conference (ECOC)*, 2009.

[19] Dijkman, R. M., Dumas, M., Ouyang, C.: Semantics and analysis of business process models in BPMN, *Information & Software Technology*, **50**(12), 2008, 1281–1294.

[20] van Dongen, B., Jansen-Vullers, M., Verbeek, H., van der Aalst, W. M. P.: Verification of the SAP reference models using EPC reduction, state space analysis, and invariants, *Computers in Industry*, **58**(6), 2007, 578–601.

[21] van Dongen, B. F., Dijkman, R. M., Mendling, J.: Measuring Similarity between Business Process Models, *CAiSE* (Z. Bellahsene, M. Léonard, Eds.), 5074, Springer, 2008, ISBN 978-3-540-69533-2, 450–464.

[22] Dumas, M., García-Bañuelos, L., Dijkman, R. M.: Similarity Search of Business Process Models, *IEEE Data Eng. Bull.*, **32**(3), 2009, 23–28.

[23] Dwyer, M. B., Avrunin, G. S., Corbett, J. C.: Property Specification Patterns for Finite-State Verification, *FMSP* (M. A. Ardis, J. M. Atlee, Eds.), ACM, 1998, ISBN 0-89791-954-8, 7–15.

[24] Eshuis, R., Wieringa, R.: Tool Support for Verifying UML Activity Diagrams, *IEEE Trans. Software Eng.*, **30**(7), 2004, 437–447.

[25] Esparza, J., Heljanko, K.: *Unfoldings: a partial-order approach to model checking*, Springer, 2008.

[26] Esparza, J., Silva, M.: Circuits, Handles, Bridges and Nets, *Applications and Theory of Petri Nets*, 483, 1989, ISBN 0-387-53863-1, 210–242.

[27] Euzenat, J., Shvaiko, P.: *Ontology Matching*, Springer-Verlag, 2007.

[28] Fahland, D., Favre, C., Jobstmann, B., Koehler, J., Lohmann, N., Völzer, H., Wolf, K.: Instantaneous Soundness Checking of Industrial Business Process Models, in: Dayal et al. [15], 278–293, 278–293.

[29] van Glabbeek, R. J.: *Handbook of Process Algebra*, chapter The Linear Time – Branching Time Spectrum I. The semantics of concrete, sequential processes, Elsevier, 2001.

[30] van Glabbeek, R. J., Goltz, U.: Refinement of Actions and Equivalence Notions for Concurrent Systems, *Acta Inf.*, **37**(4/5), 2001, 229–327.

[31] Goedertier, S., Martens, D., Vanthienen, J., Baesens, B.: Robust Process Discovery with Artificial Negative Events, *Journal of Machine Learning Research*, **10**, 2009, 1305–1340.

[32] Greco, G., Guzzo, A., Pontieri, L., Saccà, D.: Discovering Expressive Process Models by Clustering Log Traces, *IEEE Trans. Knowl. Data Eng.*, **18**(8), 2006, 1010–1027.

[33] Gschwind, T., Koehler, J., Wong, J.: Applying Patterns during Business Process Modeling, *BPM'08: Proceedings of the 6th International Conference on Business Process Management*, Springer-Verlag, Berlin, Heidelberg, 2008, ISBN 978-3-540-85757-0, 4–19.

[34] Gutwenger, C., Mutzel, P.: A Linear Time Implementation of SPQR-Trees, *Graph Drawing*, 1984, 2001, ISBN 3-540-41554-8, 77–90.

[35] Johnson, R., Pearson, D., Pingali, K.: The Program Structure Tree: Computing Control Regions in Linear Time, *PLDI*, 1994, 171–185.

[36] Kiepuszewski, B., ter Hofstede, A. H. M., van der Aalst, W. M. P.: Fundamentals of Control Flow in Workflows, *Acta Inf.*, **39**(3), 2003, 143–209.

[37] Kiepuszewski, B., ter Hofstede, A. H. M., Bussler, C.: On Structured Workflow Modelling, *CAiSE* (B. Wangler, L. Bergman, Eds.), 1789, Springer, 2000, ISBN 3-540-67630-9, 431–445.

[38] Kindler, E.: On the semantics of EPCs: A Framework for Resolving the Vicious Circle, *BPM*, 3080, 2004, 82–97.

[39] Kovalyov, A., Esparza, J.: A Polynomial Algorithm to Compute the Concurrency Relation of Free-Choice Signal Transition Graphs, *Prof. of the International Workshop on Discrete Event Systems, WODES'96*, The Institution of Electrical Engineers, Edinburgh, 1996, 1–6.

[40] Küster, J., Gerth, C., Förster, A., Engels, G.: Detecting and Resolving Process Model Differences in the Absence of a Change Log, *BPM*, 5240, 2008, 244–260.

[41] Li, C., Reichert, M., Wombacher, A.: On Measuring Process Model Similarity Based on High-Level Change Operations, *ER* (Q. Li, S. Spaccapietra, E. S. K. Yu, A. Olivé, Eds.), 5231, Springer, 2008, ISBN 978-3-540-87876-6, 248–264.

[42] Lohmann, N.: A Feature-Complete Petri Net Semantics for WS-BPEL 2.0, *WS-FM*, 4937, 2008, 77–91.

[43] Lohmann, N., Verbeek, E., Dijkman, R. M.: Petri Net Transformations for Business Processes - A Survey, *T. Petri Nets and Other Models of Concurrency*, **2**, 2009, 46–63.

[44] McMillan, K. L.: A Technique of State Space Search Based on Unfolding, *Formal Methods in System Design*, **6**(1), 1995, 45–65.

[45] de Medeiros, A. K. A., van der Aalst, W. M. P., Weijters, A. J. M. M.: Quantifying Process Equivalence based on Observed Behavior, *Data Knowl. Eng.*, **64**(1), 2008.

[46] Mendling, J.: *Metrics for Process Models: Empirical Foundations of Verification, Error Prediction, and Guidelines for Correctness*, vol. 6 of *Lecture Notes in Business Information Processing*, Springer, 2008, ISBN 978-3-540-89223-6.

[47] Mendling, J., van Dongen, B. F., van der Aalst, W. M. P.: Getting rid of OR-joins and multiple start events in business process models, *Enterprise IS*, **2**(4), 2008, 403–419.

[48] Mendling, J., Verbeek, H. M. W., van Dongen, B. F., van der Aalst, W. M. P., Neumann, G.: Detection and prediction of errors in EPCs of the SAP reference model, *Data Knowl. Eng.*, **64**(1), 2008, 312–329.

[49] Murata, T.: Petri nets: Properties, analysis and applications, *Proceedings of the IEEE*, **77**(4), 1989, 541–580.

[50] Nejati, S., Sabetzadeh, M., Chechik, M., Easterbrook, S. M., Zave, P.: Matching and Merging of Statecharts Specifications, *ICSE*, IEEE Computer Society, 2007, 54–64.

[51] Pernici, B., Ed.: *Advanced Information Systems Engineering, 22nd International Conference, CAiSE 2010, Hammamet, Tunisia, June 7-9, 2010. Proceedings*, vol. 6051 of *Lecture Notes in Computer Science*, Springer, 2010, ISBN 978-3-642-13093-9.

[52] Polyvyanyy, A., García-Bañuelos, L., Dumas, M.: Structuring Acyclic Process Models, *BPM* (R. Hull, J. Mendling, S. Tai, Eds.), 6336, Springer, 2010, ISBN 978-3-642-15617-5, 276–293.

[53] Polyvyanyy, A., Smirnov, S., Weske, M.: The Triconnected Abstraction of Process Models, in: Dayal et al. [15], 229–244, 229–244.

[54] Polyvyanyy, A., Vanhatalo, J., Völzer, H.: Simplified Computation and Generalization of the Refined Process Structure Tree, in: Bravetti and Bultan [12], 25–41, 25–41.

[55] Pomello, L., Rozenberg, G., Simone, C.: A survey of equivalence notions for net based systems, *Advances in Petri Nets: The DEMON Project* (G. Rozenberg, Ed.), 609, Springer, 1992, ISBN 3-540-55610-9, 410–472.

[56] Rahm, E., Bernstein, P. A.: A survey of approaches to automatic schema matching, *VLDB Journal*, **10**(4), 2001, 334–350.

[57] Rosenblum, L. Y., Yakovlev, A.: Analyzing Semantics of Concurrent Hardware Specifications, *ICPP (3)*, 1989, 211–218.

[58] Rozinat, A., van der Aalst, W. M. P.: Conformance checking of processes based on monitoring real behavior, *Inf. Syst.*, **33**(1), 2008, 64–95.

[59] Schrefl, M., Stumptner, M.: Behavior-consistent Specialization of Object Life Cycles, *ACM Trans. Softw. Eng. Methodol.*, **11**(1), 2002, 92–148.

[60] Vanhatalo, J., Völzer, H., Koehler, J.: The Refined Process Structure Tree, *BPM*, 5240, 2008, 100–115.

[61] Vanhatalo, J., Völzer, H., Koehler, J.: The Refined Process Structure Tree, *Data and Knowledge Engineering (DKE)*, **68**(9), 2009, 793–818.

[62] Vanhatalo, J., Völzer, H., Leymann, F.: Faster and More Focused Control-Flow Analysis for Business Process Models Through SESE Decomposition, *ICSOC*, 4749, 2007, 43–55.

[63] Vanhatalo, J., Völzer, H., Leymann, F., Moser, S.: Automatic Workflow Graph Refactoring and Completion, *ICSOC* (A. Bouguettaya, I. Krüger, T. Margaria, Eds.), 5364, 2008, ISBN 978-3-540-89647-0, 100–115.

[64] Weidlich, M., Dijkman, R. M., Mendling, J.: The ICoP Framework: Identification of Correspondences between Process Models, in: Pernici [51], 483–498, 483–498.

[65] Weidlich, M., Elliger, F., Weske, M.: Generalised Computation of Behavioural Profiles Based on Petri-Net Unfoldings, in: Bravetti and Bultan [12], 101–115, 101–115.

[66] Weidlich, M., Mendling, J., Weske, M.: Efficient Consistency Measurement Based on Behavioral Profiles of Process Models, *IEEE Trans. Software Eng.*, **37**(3), 2011, 410–429.

[67] Weidlich, M., Polyvyanyy, A., Desai, N., Mendling, J.: Process Compliance Measurement Based on Behavioural Profiles, in: Pernici [51], 499–514, 499–514.

[68] Weidlich, M., Polyvyanyy, A., Desai, N., Mendling, J., Weske, M.: Process compliance analysis based on behavioural profiles, *Inf. Syst.*, **36**(7), 2011, 1009–1025.

[69] Weidlich, M., Polyvyanyy, A., Mendling, J., Weske, M.: Efficient Computation of Causal Behavioural Profiles Using Structural Decomposition, *Petri Nets* (J. Lilius, W. Penczek, Eds.), 6128, Springer, 2010, ISBN 978-3-642-13674-0, 63–83.

[70] Weidlich, M., Weske, M., Mendling, J.: Change Propagation in Process Models Using Behavioural Profiles, *IEEE SCC*, IEEE Computer Society, 2009, 33–40.

[71] Weidlich, M., Ziekow, H., Mendling, J.: Optimising Complex Event Queries over Business Processes Using Behavioural Profiles, *Business Process Management Workshops* (M. zur Muehlen, J. Su, Eds.), 66, Springer, 2010, ISBN 978-3-642-20510-1, 743–754.

[72] Weidlich, M., Ziekow, H., Mendling, J., Günther, O., Weske, M., Desai, N.: Event-based Monitoring of Process Execution Violations, *Proceedings of the 9th International Conference on Business Process Management (BPM 2011)* (S. Rinderle, F. Toumani, K. Wolf, Eds.), number 6896 in LNCS, Spring, 2011, 182–198, To appear.

[73] Weijters, A. J. M. M., van der Aalst, W. M. P., de Medeiros, A. K. A.: *Process Mining with the HeuristicsMiner Algorithm*, BETA Working Paper Series WP 166, Eindhoven University of Technology, Eindhoven, 2006.

[74] Winskel, G.: Petri Nets, Algebras, Morphisms, and Compositionality, *Inf. Comput.*, **72**(3), 1987, 197–238.

[75] Wombacher, A.: Evaluation of Technical Measures for Workflow Similarity Based on a Pilot Study, *OTM Conferences (1)* (R. Meersman, Z. Tari, Eds.), 4275, Springer, 2006, ISBN 3-540-48287-3, 255–272.

[76] Wombacher, A., Rozie, M.: Evaluation of Workflow Similarity Measures in Service Discovery, *Service Oriented Electronic Commerce* (M. Schoop, C. Huemer, M. Rebstock, M. Bichler, Eds.), 80, GI, 2006, ISBN 3-88579-174-9, 51–71.