The Biconnected Verification of Workflow Nets

Artem Polyvyanyy, Matthias Weidlich, and Mathias Weske

Business Process Technology Group
Hasso Plattner Institute at the University of Potsdam
Prof.-Dr.-Helmert-Str. 2–3, D-14482 Potsdam, Germany
(Artem.Polyvyanyy,Matthias.Weidlich,Mathias.Weske)@hpi.uni-potsdam.de

Abstract. Formal representations of business processes are used for analysis of the process behavior. Workflow nets are a widely used formalism for describing the behavior of business processes. Structure theory of processes investigates the relation between the structure of a model and its behavior. In this paper, we propose to employ the connectivity property of workflow nets as an angle to their structural analysis. In particular, we show how soundness verification can be organized using biconnected components of a workflow net. This allows for efficient identification and localization of flaws in the behavior of workflow nets and for supporting process analysts with diagnostic information.

1 Introduction

Business process modeling is the basis for understanding, improving, and implementing working procedures in organizations. Execution semantics of common business process definition languages, such as BPEL or UML activity diagrams, are described rather informally. However, formal investigations rely on a mapping of these languages into a formalism. To this end, various mappings from process definition languages to Petri nets have been proposed, e.g., [1,2].

Workflow nets are a structural subclass of Petri nets with a dedicated source (start) place and a dedicated sink (end) place. *Soundness* is a desirable correctness property of workflow nets [3], since a sound workflow net always terminates properly and each task can contribute to the completion of a process. Consequently, a business process is considered to be correct if the corresponding workflow net is sound.

Soundness verification is a well-studied problem. Most approaches use state space analysis to solve it, so that the resulting state space explosion problem has to be faced. In this paper, we advocate to use structural analysis of workflow nets to investigate soundness, providing insight into an alternative – and in many cases, preferable – way to check soundness. In particular, we employ the biconnected decomposition of workflow nets. That is, we identify components based on cut-vertices, i.e., nodes of the net that when removed yield the net disconnected. Based thereon, we point out how the soundness verification can be organized from the derived components of a workflow net. Where applicable, we draw conclusions on soundness for the general class of workflow nets; otherwise, the results are

restricted to safe nets. Besides formal results on the biconnected decomposition of workflow nets, we provide an outlook on how the triconnected decomposition of biconnected components of workflow nets might also be used to verify soundness. Despite the variety of existing soundness verification techniques, the efficiency and the structural diagnostic information for the general class of workflow nets are unique characteristics of our approach, coming at the expense of verification completeness. Withal, we see a great potential in combining our approach with existing techniques to achieve more mature process model verification.

The paper is structured as follows. The next section provides formal preliminaries for our work. In Sect. 3, we show how soundness verification can be organized following on the biconnected decomposition of workflow nets. A discussion on how this approach can be extended using triconnected decomposition techniques is presented in Sect. 4. Finally, the paper closes with a discussion of related work and conclusions.

2 Preliminaries

We use Petri nets as our formal grounding. A Petri net has a structure given by a net, a marking that represents a state of the net, and the execution semantics that describes the principles of state transitions.

Definition 1 (Petri net). A *Petri net*, or a *net*, is a tuple $N = (P, T, F)$, with P and T as finite disjoint sets of places and transitions, $P \cap T = \emptyset$, and $F \subseteq (P \times T) \cup (T \times P)$ as the flow relation.

We write $X = (P \cup T)$ for all nodes of a net. The transitive closure of F is denoted by F^+. For a node $x \in X$, $\bullet x = \{y \in X \mid (y, x) \in F\}$, $x\bullet = \{y \in X \mid (x, y) \in F\}$. A node $x \in X$ is an *input* (*output*) node of a node $y \in X$, iff $x \in \bullet y$ ($x \in y\bullet$). $in_N(x) = \{(n, x) \mid n \in \bullet x\}$ are the *incoming* flows of x and $out_N(x) = \{(x, n) \mid n \in x\bullet\}$ are the *outgoing* flows of x.

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

- $M : P \to \mathbb{N}_0$ 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 when place p contains just one token and all other places contain no tokens.
- For any transition $t \in T$ and any marking $M \in \mathbb{M}$, t is *enabled* in M, denoted by $(N, M)[t\rangle$, iff $\forall p \in \bullet t : M(p) \geq 1$.
- Marking M' is reached from M by *firing* of t, denoted by $(N, M)[t\rangle(N, M')$, such that $M' = M - \bullet t + t\bullet$, i.e., one token is taken from each input place of t and one token is added to each output place of t.
- A *firing sequence* of length $n \in \mathbb{N}$ is a function $\sigma : \{0, \ldots, n-1\} \to T$. For $\sigma = \{(0, t_x), \ldots, (n-1, t_y)\}$, we also write $\sigma = t_0, \ldots, t_{n-1}$.
- For any two markings $M, M' \in \mathbb{M}$, M' is *reachable* from M in N, denoted by $M' \in [N, M\rangle$, iff there exists a firing sequence σ leading from M to M'.
- A *system* is a pair (N, M_0), where N is a net and M_0 its *initial marking*.

Workflow (WF-)nets form a subclass of Petri nets. WF-nets were proposed in [4] for modeling workflow definitions. A WF-net is a net with two special places: one to mark the initialization and the other to mark the completion of a workflow.

Definition 3 (WF-net, Short-circuit net, WF-system).
A Petri net $N = (P, T, F)$ is a *workflow net* (or a *WF-net*), iff N has a dedicated *source* place $i \in P$, with $\bullet i = \emptyset$, N has a dedicated *sink* place $o \in P$, with $o\bullet = \emptyset$, and the *short-ciruit* net $N' = (P, T \cup \{t^*\}, F \cup \{(o, t^*), (t^*, i)\})$, $t^* \notin T$, of N is strongly connected. A *WF-system* is a pair (N, M_i), where $M_i = [i]$.

Soundness is the basic correctness property of workow nets [3]. A sound workow net always terminates and each transition can contribute to the completion of the workflow by following certain route through the net.

Definition 4 (Liveness, Boundedness, Safeness, Soundness).
- A system (N, M_0) is *live*, iff for every reachable marking $M \in [N, M_0\rangle$ and $t \in T$, there exists a marking $M' \in [N, M\rangle$ such that $(N, M')[t\rangle$.
- A system (N, M_0) is *bounded*, iff the set $[N, M_0\rangle$ is finite.
- A system (N, M_0) is *safe*, iff $\forall\ M \in [N, M_0\rangle\ \forall\ p \in P : M(p) \leq 1$.
- A WF-system (N, M_i) with $N = (P, T, F)$ is *sound*, iff the short-circuit system (N', M_i) is live and bounded.

For the purpose of structural analysis of nets, we give the following definitions.

Definition 5 (Subnet, Path).
Let $N' = (P', T', F')$ and $N = (P, T, F)$ be two nets, and let $P' \subseteq P$, $T' \subseteq T$. N' is a *subnet* of N, denoted $N' \subseteq N$, iff $F' = F \cap ((P' \times T') \cup (T' \times P'))$. A *path* is a non-empty sequence $\langle x_1, \ldots, x_k\rangle$ of nodes, $k > 1$, denoted by $\pi(x_1, x_k)$, which satisfies $(x_1, x_2), \ldots, (x_{k-1}, x_k) \in F$. We write $x_i \in \pi$, if x_i is on the path.

The structure of a Petri net (P, T, F) is defined by the graph (X, F). Connectivity is a basic property of a graph. A graph is *connected* if every pair of distinct vertices in the graph is connected; otherwise the graph is *disconnected*. A graph is *biconnected* if there exists no vertex whose removal renders the graph disconnected. If such a vertex exists, it is called a *cutvertex*. Note that removal of a vertex implies removal of all incident edges. After removing a cutvertex from a graph, the graph gets decomposed into disconnected subgraphs (or *components*).

3 Biconnected Verification of WF-nets

The soundness of a WF-net can be verified by checking liveness and boundedness of the corresponding short-circuit net. Short-circuit nets are connected, but not necessarily biconnected. This section explains how soundness verification of a WF-net can be broken down into checks of its biconnected components.

The classic sequential algorithm for computing biconnected components in a connected graph, proposed in [5], runs in linear time. Let (X, F) be a connected graph, then the algorithm requires time and space proportional to $max(|X|, |F|)$. Biconnected components can be arranged in a tree structure—the *tree of the biconnected components*. The tree has two types of nodes that refer either to cutvertices or to biconnected components. Edges of the tree connect cutvertices with associated biconnected components, i.e., there is an edge between a cutvertex and a biconnected component if the biconnected component contains the cutvertex.

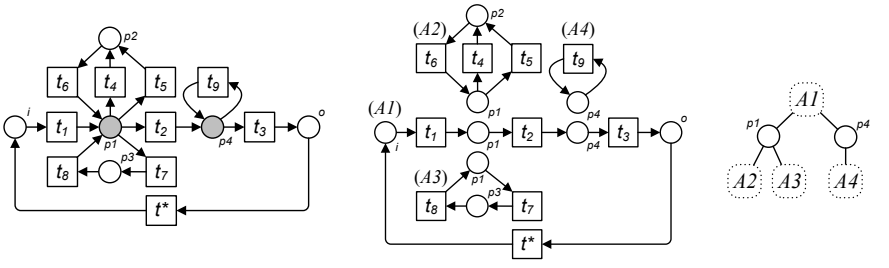


Fig. 1. (a) A short-circuit net, (b) biconnected subnets, and (c) the 2-WF-tree

The number of nodes in the tree is $O(|X|)$ and, hence, the space required to store the tree and all the biconnected components is $O(max(|X|, |F|))$.

A *biconnected subnet* is a biconnected component of a short-circuit net. The *tree of the biconnected subnets*, or the *2-WF-tree*, is the tree of the biconnected components of a short-circuit net.

Definition 6 (The tree of the biconnected subnets).
Let $N = (P, T, F)$ be a WF-net and let N' be its short-circuit net with the extra transition t^*. The *tree of the biconnected subnets*, or the *2-WF-tree*, of N is a tuple $\mathcal{T}_N^2 = (\mathcal{B}, \mathcal{C}, \rho, \eta, \triangle)$, where:

- $\mathcal{B}$ is a set of all biconnected subnets and $\mathcal{C}$ is a set of all cutvertices of N',
- $\rho = (P_\rho, T_\rho, F_\rho) \in \mathcal{B}$ is the root of $\mathcal{T}_N^2$, iff $t^* \in T_\rho$,
- $\eta : \mathcal{B} \to \mathcal{P}(\mathcal{B})$ assigns to biconnected subnet its child biconnected subnets,
- $\triangle \subseteq \mathcal{B} \times \mathcal{C} \times \mathcal{B}$, $(b_1, c, b_2) \in \triangle$, iff c is shared by b_1 and b_2, and $b_2 \in \eta(b_1)$.

Fig. 1 exemplifies the biconnected decomposition of a WF-net: Fig. 1(a) shows a short-circuit net. The net has two place cutvertices $p1$ and $p4$, which are highlighted with grey background. The cutvertices induce four biconnected subnets $A1$–$A4$, cf., Fig. 1(b). Finally, Fig. 1(c) organizes the subnets in the 2-WF-tree with the root node that corresponds to the biconnected subnet $A1$.

One observation is that a WF-net can be sound only if all the cutvertices of the corresponding short-circuit net are places.

Lemma 1. *Let (N, M_i), $N = (P, T, F)$, be a WF-system and N' be the short-circuit net of N. If $t \in T$ is a cutvertex of N', then (N, M_i) is not sound.*

Proof. Because t is a cutvertex of N', there exists $p' \in {\bullet t}$, $p' \neq i$, such that t is on every path $\pi(i, p')$. We show now by induction that t is never enabled, i.e., for every marking $M \in [N, M_i\rangle$ holds $\neg(N, M)[t\rangle$.

base: $\neg(N, M_i)[t\rangle$ as $M_i(p') = 0$, i.e., t is not enabled by the initial marking.

step: Let M' be a marking reachable from M_i by a firing sequence σ that does not contain t, i.e., t was never enabled. Let $t' \in T$ be such that $(N, M')[t'\rangle$. Assume that $t' = t$, then $M'(p') \geq 1$. If $M'(p') \geq 1$, then σ contains all the transitions of some path $\pi(i, p')$ and, hence, contains t. We have reached the contradiction and, therefore, $t' \neq t$.

As t is never enabled, (N', M_i) is not live. Thus, (N, M_i) is not sound. □

A transition cutvertex hints at unsoundness of the net and, therefore, constitutes valuable diagnostic information. In case all cutvertices of a short-circuit net are places, verification procedure should proceed. We show how the verification procedure can be broken down into checks of biconnected subnets of the short-circuit net. First, we explain how to derive WF-nets from biconnected subnets.

Definition 7 (Biconnected sub-WF-net). Let $N = (P, T, F)$ be a WF-net, $\mathcal{T}_N^2 = (\mathcal{B}, \mathcal{C}, \rho, \eta, \triangle)$ its 2-WF-tree, and $b = (P_b, T_b, F_b) \in \mathcal{B}$. A *biconnected sub-WF-net* of N, denoted $b^\star$, $b^\star = (P_{b^\star}, T_{b^\star}, F_{b^\star})$, is obtained from b as follows:

- If $b = \rho$, then $P_{b^\star} = P_b$, $T_{b^\star} = T_b \cap T$, and $F_{b^\star} = F_b \cap F$.
- If $b \neq \rho$ and $a \in \mathcal{B}$, $c \in \mathcal{C}$ are such that there exists $(a, c, b) \in \triangle$, then $P_{b^\star} = (P_b \setminus \{c\}) \cup \{i, o\}$, $T_{b^\star} = T_b$, and $F_{b^\star} = \{(x_1, x_2) \in F_b \mid x_1 \neq c \wedge x_2 \neq c\} \cup \{(i, x) \in \{i\} \times T_b \mid (c, x) \in F_b\} \cup \{(x, o) \in T_b \times \{o\} \mid (x, c) \in F_b\}$.

A WF-net that corresponds to a subtree of b, denoted $b^\triangle$, is obtained by merging sub-WF-net $b^\star$ and all subnets that are descendants of b at shared cutvertices. Biconnected sub-WF-nets are also referred to as biconnected WF-nets.

Fig. 2 presents sub-WF-nets of the short-circuit net provided in Fig. 1(a). The sub-WF-nets correspond to the biconnected subnets in Fig. 1(b). Sub-WF-net $A1$ is obtained from the corresponding biconnected subnet by ignoring transition t^* and adjacent flow relations. In the case when a biconnected subnet is not the root of the 2-WF-tree, cf., Fig. 1(c), the corresponding sub-WF-net is obtained as follows: The cutvertex that corresponds to the parent node in the 2-WF-tree is removed from the subnet and two fresh places are added, a source place i and a sink place o. The flow relations of the cutvertex are rerouted to originate from i or to terminate at o, respectively.

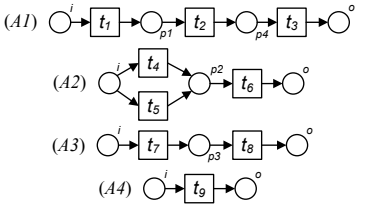


Fig. 2. Biconnected sub-WF-nets

Construction of a WF-net that corresponds to a subtree in the 2-WF-tree is supported by two types of transformations, viz., refinements, of nets.

Definition 8 (Self-loop place refinement, Transition refinement).

- Let $N_1 = (P_1, T_1, F_1)$ be a net, $p \in P_1$ a place. A *self-loop place refinement* of p yields a net $N_2 = (P_1, T_1 \cup \{t_p\}, F_1 \cup \{(p, t_p), (t_p, p)\})$, denoted $N_1[p]$.
- Let $N_1 = (P_1, T_1, F_1)$ be a net, $N_2 = (P_2, T_2, F_2)$ a WF-net with source i and sink o, $T_1 \cap T_2 = \emptyset$, $P_1 \cap P_2 = \emptyset$, and $t \in T_1$. A *transition refinement* of t by N_2 yields a net $N_3 = (P_3, T_3, F_3)$, denoted $N_1[t/N_2]$, such that:
  - $P_3 = P_1 \cup (P_2 \setminus \{i, o\})$, $T_3 = (T_1 \setminus \{t\}) \cup T_2$, and
  - $F_3 = \{(x_1, x_2) \in F_1 | x_1 \neq t \wedge x_2 \neq t\} \cup \{(x_1, x_2) \in F_2 | \{x_1, x_2\} \cap \{i, o\}\} \cup \{(x_1, x_2) \in P_1 \times T_2 \mid (x_1, t) \in F_1 \wedge (i, x_2) \in F_2\} \cup \{(x_1, x_2) \in T_2 \times P_1 \mid (t, x_2) \in F_1 \wedge (x_1, o) \in F_2\}$.

A self-loop place refinement preserves liveness and safeness, cf., [6]. The concept of transition refinement is borrowed from [7,8]. Fig. 3(a) shows the self-loop refinement of place p_1 in the WF-net $A1$ from Fig. 2, whereas Fig. 3(b) depicts the transition t_{p_1} refinement in the WF-net from Fig. 3(a) by WF-net $A2$.

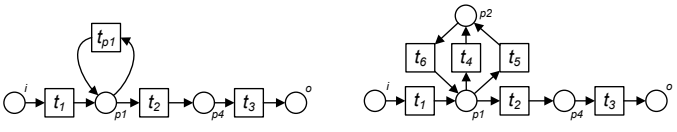


Fig. 3. (a) A self-loop place refinement, and (b) a transition refinement

By using biconnected sub-WF-nets of a WF-net and the net transformations from Definition 8 it is possible to organize soundness verification of the WF-net. In the class of safe systems, the soundness of a system is closely related to the soundness of its biconnected sub-WF-nets.

Theorem 1. *Each biconnected sub-WF-net of a WF-net is safe and sound, iff the WF-net is safe and sound.*

Proof. Let N be a WF-net and let $\mathcal{T}_N^2 = (\mathcal{B}, \mathcal{C}, \rho, \eta, \triangle)$ be the 2-WF-tree of N.
($\Rightarrow$) By structural induction on the tree of the biconnected subnets.

base: If $\mathcal{T}_N^2$ contains only one biconnected subnet, i.e., $|\mathcal{B}| = 1$, then N is a biconnected WF-net. Obviously, the claim holds.

step: Let $b \in \mathcal{B}$ be a biconnected subnet. Suppose that the claim holds for all $a^{\triangle}$ such that $a \in \eta(b)$. We show by induction that the claim is also true for $b^{\triangle}$.

$b^{\star}$ is a biconnected sub-WF-net of N and, hence, is safe and sound. Let $a \in \eta(b)$ and $c \in \mathcal{C}$ be such that $(b, c, a) \in \triangle$. A WF-net $b' = b^{\star}[c]$ with a self-loop transition t_c is safe and sound. A WF-net $b'[t_c/a^{\triangle}]$ is safe and sound, cf., statement 4 of Theorem 3 in [8]. Therefore, after refining $b^{\star}$ with all the biconnected WF-nets that correspond to subnets from $\eta(b)$ one obtains a safe and sound WF-net that is equal to $b^{\triangle}$.

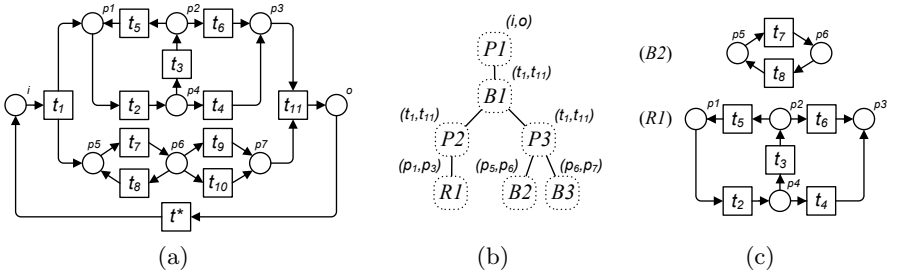
As $\rho^{\triangle}$ is equal to N, the claim holds.

($\Leftarrow$) The claim trivially holds by following ($\Rightarrow$) in the reverse direction. □

Therefore, it suffices to show that at least one biconnected sub-WF-net is not safe and sound in order to conclude that the WF-net is not safe and sound. This biconnected sub-WF-net causes unsoundness and constitutes valuable diagnostic information. Finally, because biconnected sub-WF-nets can be computed in time linear to the size of a net, the biconnected decomposition step does not add to the overall complexity of soundness verification.

4 Towards Triconnected Verification of WF-nets

This section sketches the connectivity-based soundness verification of biconnected WF-nets. Biconnected WF-nets contain no cutvertices; they can, however, contain pairs of vertices that when removed yield the *triconnected* subnets. The sequential algorithm for computing triconnected components in a biconnected graph runs in linear time, cf., [9]. In [10], the authors discuss implementation aspects of the algorithm. Let (X, F) be a biconnected graph, then the algorithm requires time and space proportional to $max(|X|, |F|)$.



(a) (b) (c)

Fig. 4. (a) A short-circuit net, (b) the 3-WF-tree, and (c) triconnected subnets

The triconnected decomposition of WF-nets is illustrated in Fig. 4. Fig. 4(a) shows a WF-net that is biconnected, but not triconnected. That is, it contains several triconnected subnets. For instance, the subnet between places $p1$ and $p3$ is such triconnected subnet, as the removal of both nodes yields the graph disconnected. According to [11], there are four structural classes of triconnected components and, therefore, of triconnected subnets. A subnet is *trivial* if its a flow; a *polygon* if it decomposes into a sequence of subnets where the exit of a subnet is the entry of the next subnet in the sequence; a *bond* if it decomposes into a set of subnets that share boundary nodes; or a *rigid* otherwise. These subnets form a hierarchy that yields the tree of triconnected subnets, similar to the tree of biconnected subnets introduced above. Fig. 4(b) shows this tree for the example net in Fig. 4(a), while Fig. 4(c) also depicts two exemplary triconnected subnets. Note that names of subnets hint at their structural class.

The subnets derived by triconnected decomposition of a WF-net may be leveraged for soundness verification. While a detailed investigation of these nets is beyond the scope of this paper, initial results have been presented already for free-choice nets in [12]. There it was shown that for this class of nets soundness imposes certain requirements on the boundary nodes of triconnected components. For instance, all bond components of a free-choice sound WF-net are either place-bordered or transition-bordered. Moreover, heuristics for soundness verification based on triconnected components have also been proposed in [13].

5 Related Work and Conclusion

In this paper, we have investigated the relation between the connectivity property of a WF-net and its behavioral correctness. We organized soundness verification based on the biconnected decomposition of a WF-net and discussed the potential for leveraging its triconnected decomposition.

Our approach relates to other work on the verification of process models. Verification of workflow graphs might be organized based on fragments that have a single-entry edge and a single-exit edge [14]. Albeit related, this work leverages edge-connectivity, whereas our work uses node-connectivity. Soundness checking based on heuristics and state space exploration for a triconnected decomposition of a (free-choice) process graph has been proposed in [13]. Our technique complements this approach and might be integrated to achieve more mature soundness verification. Verification of Petri nets can be based on structural

reductions. Besides the rules by Murata [6], Berthelot proposed a set of rules that reduce live and bounded marked graphs to a single transition [15], while there is a complete kit of rules for free-choice Petri nets [16]. All these rules are incomplete when applied to nets of arbitrary structure.

Despite the large body of related work on the formal verification of process models, we are not aware of any work that employs the connectivity property as an angle to their structural analysis. The biconnected decomposition allows for a *divide and conquer* strategy as suggested by the principles of connectivity-based decomposition outlined in [17]. In future work, we aim at extending our approach towards a holistic verification framework that allows for verification of ordinary Petri nets using step-wise connectivity-based decomposition.

References

1. Lohmann, N.: A feature-complete Petri net semantics for WS-BPEL 2.0. In: WS-FM. Volume 4937 of LNCS. (2008) 77–91
2. Eshuis, R., Wieringa, R.: Tool support for verifying UML activity diagrams. IEEE Trans. Software Eng. **30**(7) (2004) 437–447
3. Aalst, W.: Verification of workflow nets. In: ICATPN. Volume 1248 of LNCS. (1997) 407–426
4. Aalst, W.: The application of Petri nets to workflow management. Journal of Circuits, Systems, and Computers **8**(1) (1998) 21–66
5. Hopcroft, J., Tarjan, R.: Algorithm 447: efficient algorithms for graph manipulation. Commun. ACM **16**(6) (1973) 372–378
6. Murata, T.: Petri nets: Properties, analysis and applications. Proceedings of the IEEE **77**(4) (1989) 541–580
7. Valette, R.: Analysis of petri nets by stepwise refinements. J. Comput. Syst. Sci. **18**(1) (1979) 35–46
8. Aalst, W.: Workflow verification: Finding control-flow errors using petri-net-based techniques. In: BPM. Volume 1806 of LNCS. (2000) 161–183
9. Hopcroft, J., Tarjan, R.: Dividing a graph into triconnected components. SIAM Journal on Computing **2**(3) (1973) 135–158
10. Gutwenger, C., Mutzel, P.: A linear time implementation of SPQR-trees. In: Graph Drawing. Volume 1984 of LNCS. (2001) 77–90
11. Polyvyanyy, A., Vanhatalo, J., Voelzer, H.: Simplified computation and generalization of the refined process structure tree. In: WS-FM, Hoboken, NJ, US (September 2010) to appear.
12. Weidlich, M., Polyvyanyy, A., Mendling, J., Weske, M.: Efficient computation of causal behavioural profiles using structural decomposition. In: Petri Nets. Volume 6128 of LNCS., Springer (2010) 63–83
13. Fahland, D., Favre, C., Jobstmann, B., Koehler, J., Lohmann, N., Völzer, H., Wolf, K.: Instantaneous soundness checking of industrial business process models. In: BPM. Volume 5701 of LNCS. (2009) 278–293
14. Vanhatalo, J., Völzer, H., Leymann, F.: Faster and more focused control-flow analysis for business process models through SESE decomposition. In: ICSOC. Volume 4749 of LNCS. (2007) 43–55
15. Berthelot, G.: Transformations and decompositions of nets. In: Advances in Petri Nets. Volume 254 of LNCS. (1986) 359–376
16. Desel, J., Esparza, J.: Free Choice Petri Nets. Cambridge University Press (1995)
17. Polyvyanyy, A.: Structural abstraction of process specifications. In: ZEUS. (2010)