On the deadlock control in parallel resource-constrained workflows

Vladimir A. Bashkin
and Nadezhda Yu. Panfilova
Yaroslavl State University
Yaroslavl, Russia 150000
Email: v_bashkin@mail.ru lillian007@mail.ru

***Abstract*—We study the verification of the soundness property for workflow nets extended with resources. A workflow is sound if it terminates properly (no deadlocks and livelocks are possible). A class of resource-constrained workflow nets (RCWF-nets) is considered, where resources can be used by a process instance, but cannot be created or spent.**

Two sound RCWF-net, using the same set of resources, can be put in parallel. This parallel composition in some cases may produce additional deadlocks. A problem of deadlock avoidance in parallel workflows is studied, some methods of deadlock search and control are presented.

***Keywords*—Petri net, workflow, soundness, deadlock, RCWF-net, parallel composition**

I. Introduction

Workflow management systems provide the automated support and coordination of business and technological processes to reduce costs and flow times and to increase quality of service and productivity. Workflows orchestrate people, resources, technology and information flow. Workflow nets [1], [2], a particular class of Petri nets, have become one of the standard ways to model and analyze workflow processes.

Workflow net is an abstraction of the workflow that can be used to check the so-called soundness property. This property guarantees the absence of livelocks, deadlocks, and other anomalies that can be detected without domain knowledge. Nowadays there exists a number of soundness notions (see [3] for a survey). Informally, the classical soundness ensures that from any reachable state the system may terminate properly.

A workflow consists of a set of coordinated tasks describing the flow of work within the organization. In real world the occurrence of those tasks may depend on resources, such as machines, manpower, and raw material. To take resources into account different extensions of a base formalism of WF-nets have been introduced, coursing different versions of soundness.

In [4], [5] a specific class of WFR-nets with decidable soundness was studied. In [10], [12] a more general class of Resource-Constrained Workflow Nets (RCWF-nets) was defined. Informally, the authors impose two constraints on resources. First, they require that all resources that are initially available are available again after terminating of all cases. Second, they also require that for any reachable marking, the number of available resources does not override the number of initially available resources.

In [6], [7] a more general case of arbitrary resource transformations was studied.

In [10] it was proven that for RCWF-nets with a single resource type generalized soundness can be effectively checked in polynomial time. In [12] it was proven that generalized soundness is decidable in RCWF-nets with an arbitrary number of resource places (by reducing to the home-space problem).

Although soundness is decidable, there is so far no efficient decision algorithm because the proposed algorithm decides a home-space property, which requires a finite but (in general) too high number of reachability checks [12]. In addition, the problem of the calculation of the smallest number of resources for which soundness can be proved, remains open.

In this paper we consider a compositional approach to this problem. We investigate possible ways of minimal resource partitioning in control-independent and resource-dependent parallel branches of a workflow. We define a natural notion of parallel composition of two RCWF-nets, sharing common resource places. Parallelism may introduce additional deadlocks here, but we prove that these deadlocks (and other soundness violations) are avoidable by an enlargement of the initial resource. We present an approach, that allows to compute a nontrivial subset of minimal sound resources of a decomposable RCWF-net.

The main result of the paper is a method of deadlock avoidance for parallel workflows. We show that under certain circumstances a composite workflow can be restructured in such a way that the resulting net would require not a sum but a union of minimal sound resources of its parallel subnets. This allows to save a significant part of resources without any violation of the soundness property.

The paper is organized as follows. In Section 2 basic definitions of multisets and Petri nets are given. In Section 3 resource-constrained workflow nets and their soundness properties are formally defined. In Section 4 we study reachability properties of sound RCWF-nets. In Section 5 a notion of parallel composition of RCWF-nets is introduced. Several result are formulated, describing how minimal resources of a composite workflow can be obtained from minimal resources

This work is licensed under the Creative Commons Attribution License

of its parallel subnets. In Section 6 deadlock/livelock avoidance methods are presented. The first one can be applied for any pair of sound workflows, but requires specific run-time control, not incorporated into the net itself. The second one uses the original Petri net structure, but is applicable to the safe nets only. Section 7 contains some conclusions.

II. Preliminaries

Let S be a finite set. A *multiset* m over a set S is a mapping $m : S \to \mathit{Nat}$, where *Nat* is the set of natural numbers (including zero).

For two multisets m, m' we write $m \subseteq m'$ iff $\forall s \in S : m(s) \leq m'(s)$ (the inclusion relation). The sum and the union of two multisets m and m' are defined as usual: $\forall s \in S : m+m'(s) = m(s)+m'(s),\ m \cup m'(s) = max(m(s), m'(s))$. By $\mathcal{M}(S)$ we denote the set of all finite multisets over S.

Let P and T be disjoint sets of *places* and *transitions* and let $F : (P \times T) \cup (T \times P) \to \mathit{Nat}$. Then $N = (P, T, F)$ is a *Petri net*. A *marking* in a Petri net is a function $M : P \to \mathit{Nat}$, mapping each place to some natural number (possibly zero). Thus a marking may be considered as a multiset over the set of places. Pictorially, P-elements are represented by circles, T-elements by boxes, and the flow relation F by arcs. Places may carry tokens represented by filled circles.

For a transition $t \in T$ the *preset* ${}^\bullet t$ and the *postset* $t^\bullet$ are defined as the multisets over P such that ${}^\bullet t(p) = F(p, t)$ and $t^\bullet(p) = F(t, p)$ for each $p \in P$. Similarly, for a place $p \in P$ we define ${}^\bullet p$ and $p^\bullet$ as the multisets over T such that ${}^\bullet p(t) = F(t, p)$ and $p^\bullet(t) = F(p, t)$ for each $t \in T$.

A transition $t \in T$ is *enabled* in a marking M iff $\forall p \in P\ M(p) \geq F(p, t)$. An enabled transition t may *fire* yielding a new marking $M' =_{\text{def}} M - {}^\bullet t + t^\bullet$, i. e. $M'(p) = M(p) - F(p, t) + F(t, p)$ for each $p \in P$ (denoted $M \xrightarrow{t} M'$, or just $M \to M'$). We say that M' is reachable from M iff there is a sequence $M = M_1 \to M_2 \to \cdots \to M_n = M'$. For a Petri net N by $\mathcal{R}(N, M_0)$ we denote the set of all markings reachable from its initial marking M_0.

A net (N, M_0) is *bounded* iff $\mathcal{R}(N, M_0)$ is finite.

A net (N, M_0) is *safe* iff $\forall M \in \mathcal{R}(N, M_0), p \in P$ we have $M(p) \leq 1$. Places in safe nets can be considered as boolean variables (no tokens – false, 1 token – true).

III. RCWF-nets

A resource-constrained workflow net (RCWF-net for short) is a tuple $N = (P_c, P_r, T, F_c, F_r, i, o)$ s.t.

- P_c is a finite set of control places;
- P_r is a finite set of resource places, $P_c \cap P_r = \emptyset$;
- T is a finite set of transitions, $P_c \cap T = P_r \cap T = \emptyset$;
- $F_c : (P_c \times T) \cup (T \times P_c) \to \mathit{Nat}$ is a multiset of control arcs;
- $F_r : (P_r \times T) \cup (T \times P_r) \to \mathit{Nat}$ is a multiset of resource arcs;
- $\forall t \in T\ \exists p \in P_c : F_c(p, t) + F_c(t, p) > 0$ (each transition is incident to some control place);
- $i \in P_c$ is a source place and $o \in P_c$ is a sink place (input and output), such that ${}^\bullet i = o^\bullet = \emptyset$;
- every node from $P_c \cup T$ is on a path from i to o, and this path consists of nodes from $P_c \cup T$.

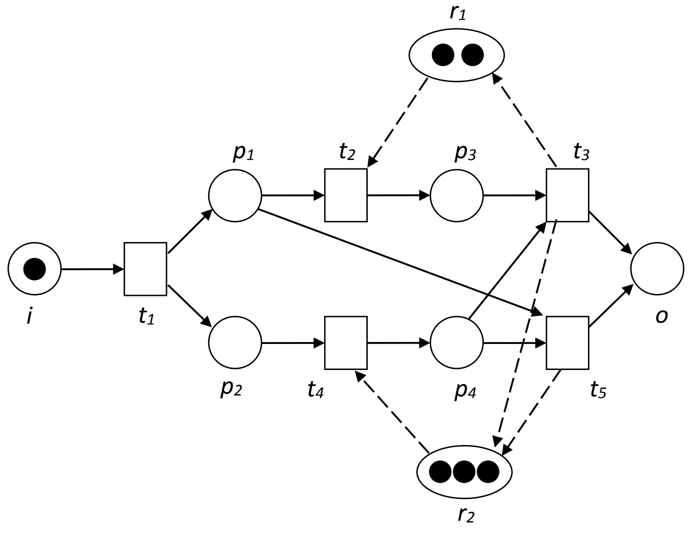


Fig. 1. RCWF-net

In RCWF-nets Petri net places are divided into control and resource ones. Note that all transitions are necessarily linked to control places — this guarantees the absence of "uncontrolled" resource modifications.

A marking is also divided into control and resource parts. For a multiset $c + r$, where $c \in \mathcal{M}(P_c)$ and $r \in \mathcal{M}(P_r)$, we write $c|r$.

For a net N a *resource* is a multiset over P_r. A *controlled resource* (a state) is a multiset over $P_c \cup P_r$.

Fig. 1 represents an example of a RCWF-net, where resource places r_1 and r_2 are depicted by ovals, resource arcs — by dotted arrows.

Every RCWF-net $N = (P_c, P_r, T, F_c, F_r, i, o)$ contains its *control subnet* $N_c = (P_c, T, F_c, i, o)$, which forms a RCWF-net with the empty set of resources.

A *marked net* $(N, i|r)$ is a net N together with some initial marking $i|r$ (here i denote a multiset, containing a single token in the input place i).

Let N be an RCWF-net. N is (r)-*sound* for some resource $r \in \mathcal{M}(P_r)$ iff $\forall c|r' \in \mathcal{R}(N, i|r)$ we have:

1) $r' \leq r$;
2) $o|r \in \mathcal{R}(N, c|r')$.

N is *sound* iff there exists some resource $r \in \mathcal{M}(P_r)$ such that N is (r')-sound for any $r' \geq r$.

For example, the net on Fig. 1 is sound, $(r_1 + r_2)$-sound and not (r_1)-sound.

Thus soundness for an RCWF-net means, that, first, this workflow net can terminate properly from any reachable state, and, moreover, adding any extra resource does not violate the proper termination property.

In [12] it was proven that the soundness problem is decidable even in a more general case of multiple input tokens.

Definition 1: For a sound RCWF-net N by $\mathbf{res}(N)$ and $\mathbf{mres}(N)$ we denote the sets of sound and minimal sound resources:

- $\mathbf{res}(N) =_{\text{def}} \{r \in \mathcal{M}(P_r) \mid N \text{ is } (r+r') - \text{sound for any } r' \in \mathcal{M}(P_r)\}$;
- $\mathbf{mres}(N) =_{\text{def}} \{r \in \mathbf{res}(N) \mid \nexists r' \in \mathbf{res}(N) : r` < r\}$.

Obviously, $\mathbf{mres}(N)$ is finite.

For example, for the net on Fig. 1 we have $\mathbf{mres}(N) = \{r_1 + r_2\}$.

As it was stated in [12], the problem of finding $\mathbf{mres}(N)$ is still open. In this paper we introduce and evaluate a promising approach to this problem, based on the parallel composition/decomposition of RCWF-nets.

IV. Properties of sound resources

The next statement formally defines a well-known "proper completion" property of sound workflows:

Fact 1: For any (r)-sound net N

$$c|r \in \mathcal{R}(N, i|r) \;\Rightarrow\; c = o \;\vee\; c \cap o = \emptyset.$$

Proof: Assume the converse: $o + m|r \in \mathcal{R}(N, i|r)$ for some non-empty m.

From the second requirement of soundness we have $o|r \in \mathcal{R}(N, o + m|r)$. However, since the place o doesn't have outgoing arcs, we have $\emptyset|r \in \mathcal{R}(N, m|r)$. But every transition in N has at least one *output* control place, thus $m = \emptyset$ — a contradiction. ■

Another established fact is the soundness and boundedness of the control subnet:

Fact 2: For any sound RCWF-net $N = (P_c, P_r, T, F_c, F_r, i, o)$ and its control subnet $N_c = (P_c, T, F_c, i, o)$ (that may be considered as an RCWF-net with an empty set of resources) we have:

1) N_c is $(\emptyset)$-sound;
2) $(N_c, i|\emptyset)$ is bounded;
3) if $c|\emptyset, c + c'|\emptyset \in \mathcal{R}(N, i|r)$ then $c' = \emptyset$.

Proof: (1) Assume the converse: N_c is not $(\emptyset)$-sound. Since N_c contains no resource places, only the second part of the soundness definition is violated:

$$\exists c|\emptyset \in \mathcal{R}(N_c, i|\emptyset) \;:\; o|\emptyset \notin \mathcal{R}(N_c, c|\emptyset).$$

Denote the corresponding transition sequence by σ (so we have $i|\emptyset \xrightarrow{\sigma} c|\emptyset$).

Now consider N. Obviously, there exists some large initial resource r s.t. $c|r' \in \mathcal{R}(N, i|c)$ for some r' — it is sufficient to sum all resources, required by transitions of σ.

On the other hand, no resource x would be enough to reach the final state $o|y$ from $c|x$ (for any y), since it is unreachable even in the "resource-free" control subnet N_c. Hence N is not sound.

(2) Otherwise an infinite run is possible in $(N_c, i|\emptyset)$, containing an infinite number of different markings, and hence a pair of markings $c_1 < c_2$ with $i \to c_1 \to c_2 \to \ldots$. From the soundness of N_c we have $c_1 \xrightarrow{\sigma} o$ for some sequence of transitions $\sigma \in T^*$. But from $c_1 < c_2$ the same sequence is possible in c_2 : $c_2 \xrightarrow{\sigma} o + (c_2 - c_1)$ — a contradiction to the proper completion property.

(3) Assume the converse. From the soundness property we have two transition sequences: $i|\emptyset \to c|\emptyset \to o|\emptyset$ and $i|\emptyset \to c + c'|\emptyset \to o|\emptyset$.

From the first sequence and the monotonicity of Petri nets we have $i + c'|\emptyset \to c + c'|\emptyset \to o + c'|\emptyset$. Combining with the second sequence, we obtain $i|\emptyset \to c + c'|\emptyset \to o + c'|\emptyset$ — a contradiction to the soundness property. ■

Every reachable $c \in \mathcal{M}(P_r)$ (a *control state* of a control subnet N_c) corresponds to a single reachable resource value:

Lemma 1: If N is sound, $r \in \mathbf{res}(N)$ and $c|r_1, c|r_2 \in \mathcal{R}(N, i|r)$, then $r_1 = r_2$.

Proof: Assume the converse: let $r_1 \neq r_2$.

Consider some $r' = r_1 + \delta_1 = r_2 + \delta_2$. From $r_1 \neq r_2$ we have $\delta_1 \neq \emptyset$ or $\delta_2 \neq \emptyset$ or both. Additionally, $\delta_1 \neq \delta_2$.

We have $i|r \to c|r_1 \to o|r$ and hence (from the monotonicity of Petri nets) $i|r + \delta_1 \to c|r_1 + \delta_1 \to o|r + \delta_1$. Similarly, $i|r + \delta_2 \to c|r_2 + \delta_2 \to o|r + \delta_2$. But $r_1 + \delta_1 = r' = r_2 + \delta_2$ and hence we have $i|r + \delta_1 \to c|r' \to o|r + \delta_2$. From the $(r + \delta_1)$-soundness property it should be $\delta_1 = \delta_2$ — a contradiction. ■

Note that we cannot replace in the statement of Lemma 1 "$r \in \mathbf{res}(N)$" by "N is (r)-sound", because an (r)-sound net is not necessarily $(r + \delta)$-sound.

For (r)-soundness we have a weaker property:

Lemma 2: If N is (r)-sound and $c|r_1, c|r_2 \in \mathcal{R}(N, i|r)$, then

$$r_1 \not< r_2 \;\text{ and }\; r_1 \not> r_2.$$

Proof: Similar to the previous Lemma. Assume the converse: $r_1 < r_2$ and hence $r_2 = r_1 + \delta_1$ with $\delta_1 \neq \emptyset$.

We have $i|r \to c|r_1 \to o|r$ and $i|r \to c|r_2 = c|r_1 + \delta_1 \to o|r$, and so $c|r_1 + \delta_1 \to o|r + \delta_1$ — a contradiction to the (r)-soundness. ■

Since any set of incomparable vectors over $\mathit{Nat}^{|P_r|}$ is finite, we have an obvious

Corollary 1: If N is (r)-sound then $\mathcal{R}(N, i|r)$ is finite.

A particular consequence of Lemma 1 is an inability of a cycle to modify a resource:

Proposition 1: If N is sound, $r \in \mathbf{res}(N)$, $c|r_1 \in \mathcal{R}(N, i|r)$ and $c|r_2 \in \mathcal{R}(N, c|r_1)$, then $r_1 = r_2$.

Proof: Immediately from Lemma 1. ■

Moreover, a sound net can perform only fixed resource transformations:

Proposition 2: If N is sound, $r \in \mathbf{res}(N)$, $c|r' \in \mathcal{R}(N,i|r)$ and $u \in \mathcal{M}(P_r)$, then for any $c|v \in \mathcal{R}(N,i|r+u)$ we have $v = r'+u$.

Proof: Assume the converse. Hence $c|v, c|r'+u \in \mathcal{R}(N,i|r+u)$ with $v \neq r'+u$ — a contradiction to Lemma 1. ■

V. Compositions of RCWF-nets

Nets with the same sets of resource places can be composed in parallel:

Definition 2: Let N_1 and N_2 be RCWF-nets with

- $N_1 = \big((P_c)_1, P_r, T_1, (F_c)_1, (F_r)_1, i_1, o_1\big)$ and
- $N_2 = \big((P_c)_2, P_r, T_2, (F_c)_2, (F_r)_2, i_2, o_2\big)$.

A *parallel composition* of N_1 and N_2 (denoted by $N = N_1 \| N_2$) is an RCWF-net $N = (P_c, P_r, T, F_c, F_r, i, o)$ with

- $P_c =_{\text{def}} (P_c)_1 \cup (P_c)_2 \cup \{i, o\}$,
- $T =_{\text{def}} T_1 \cup T_2 \cup \{t_i, t_o\}$,
- $F_c =_{\text{def}} (F_c)_1 \cup (F_c)_2 \cup \{(i,t_i),(t_i,i_1),(t_i,i_2),(t_o,o),(o_1,t_o),(o_2,t_o)\}$,
- $F_r =_{\text{def}} (F_r)_1 \cup (F_r)_2$.

We put two workflows in parallel, adding common source and sink places.

Examples of simple RCWF-nets compositions are given on Fig. 2. In the case Fig. 2.(a) both subnets has the same minimal sound resource r, and the composition is also sound with this resource. The case Fig. 2(b) is quite different. Note that $r_1 + r_2$ is a minimal sound resource for both subnets, but the composition is not (r_1+r_2)-sound because of a deadlock $p_1 + p_2|\emptyset$, reachable from $i|r_1 + r_2$. Any larger resource is sound.

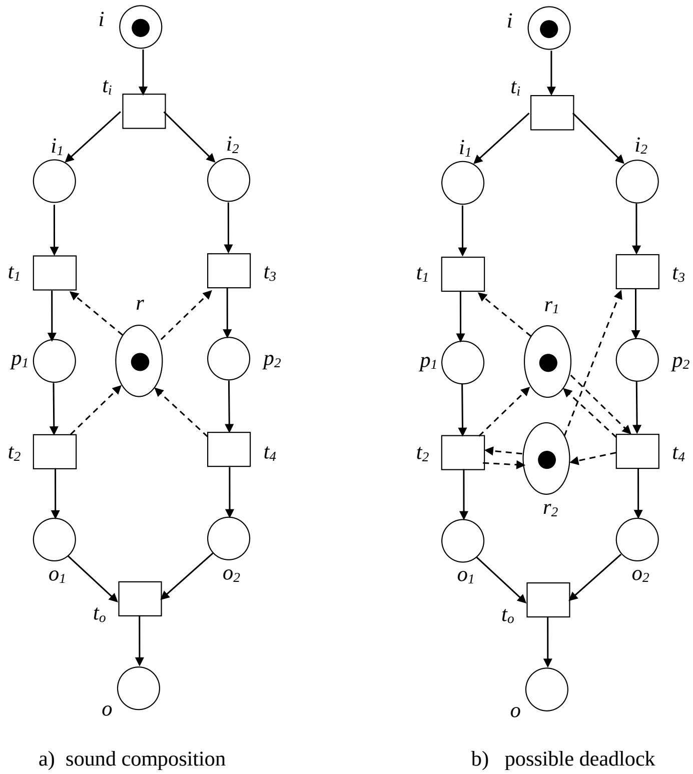
Soundness of a resource for a subnet does not necessarily imply it's soundness for a composition (as one would expect, taking into account the conservativeness of resource transformations in an RCWF-net). A parallelism may introduce additional deadlocks. However, a simple kind of additive closure exists:

Theorem 1: If N_1 and N_2 are sound then $N_1 \| N_2$ is sound and, moreover:

1) $r_1 \in \mathbf{res}(N_1), r_2 \in \mathbf{res}(N_2) \Rightarrow r_1 + r_2 \in \mathbf{res}(N_1 \| N_2)$;
2) $r \in \mathbf{res}(N_1 \| N_2) \Rightarrow \exists r_1 \in \mathbf{res}(N_1) : r \leq r_1$;
3) $r \in \mathbf{mres}(N_1 \| N_2) \Rightarrow \exists r_1 \in \mathbf{res}(N_1) : r \leq r_1$.

Proof: The soundness itself and the first statement follows from Proposition 2. Note that subnets N_1 and N_2 here work independently, without interferencing into each other's "part" of the common resource.

To prove the second statement we can take $r_1 = r$: since a resource is sound for a parallel composition, it properly supports system runs of the form $i|r \to i_1+i_2|r \to o_1+i_2|r \to o_1+o_2|r$.



a) sound composition

b) possible deadlock

Fig. 2. Two examples of RCWF-nets compositions

The third statement is a trivial consequence of the second one. ■

The first statement of Theorem 1 implies that

Corollary 2: If N_1 and N_2 are sound and $r_1 \in \mathbf{mres}(N_1), r_2 \in \mathbf{mres}(N_2)$, then there exists $r \in \mathbf{mres}(N_1 \| N_2)$ such that $r \leq r_1 + r_2$.

So, to find some minimal resource r one may search through a finite number of resources, less then or equal to $r_1 + r_2$. For every candidate $r' \leq r_1 + r_2$ the set $\mathcal{R}(N, i|r')$ is finite (Corollary 1) and can be constructed by a finite number of steps.

Note that we have not proven that this method of minimal sound resources computation allows to compute ALL elements of $\mathbf{mres}(N_1 \| N_2)$ (however, we believe it does). Nevertheless, the computed subset is always nonempty and nontrivial.

So, a problem of $\mathbf{mres}(N)$ calculation can be partially reduced to the same problem for subnets, composed in parallel. In most cases the process of decompositions ends with a purely sequentional workflows, which may have a very simple set of sound (and minimal sound) resources.

VI. Soundness ensuring

In this section we consider a resource r, sound for both subnets but not sound for a parallel composition (like $r_1 + r_2$ in Fig. 2(b)). Note that such a resource always enables a non-empty set of "good" runs (at least two: $i|r \to i_1 + i_2|r \to o_1 + i_2|r \to o_1 + o_2|r$ and $i|r \to i_1 + i_2|r \to i_1 + o_2|r \to o_1 + o_2|r$). Hence a resource is not worthless and it would be interesting to develop some control policies or system transformations, preserving all "good" runs and disabling all "bad" ones (without increasing the initial resource).

So we consider both kinds of possible undesirable (not properly terminating) behaviors of a Petri net, namely, deadlocks and livelocks.

A reachable marking $c|r$ is a *deadlock state* iff $c \neq o$ and there is no transition $t \in T$ s.t. $c|r \xrightarrow{t} c'|r'$ for some c', r'.

A finite set L of reachable markings is a *livelock* iff

1) $|L| > 1$;
2) for any $c|r, c'|r' \in L$ there is a finite transition sequence $\sigma \in T^*$ s.t. $c|r \xrightarrow{\sigma} c'|r'$;
3) for any $c|r \in L$ and $t \in T$ s.t. $c|r \xrightarrow{t} c''|r''$ we have $c''|r'' \in L$.

A *livelock state* is a state that belongs to some livelock.

Note that by definition $o|r \notin L$ for any r.

By $D(N, i|r)$ we denote a set of all deadlock and livelock states of a marked RCWF-net $(N, i|r)$.

Theorem 2: If $N = N_1 \| N_2$ and $r \in \mathbf{res}(N_1) \cap \mathbf{res}(N_2)$ then $(N, i|r)$ is bounded (i.e. $\mathcal{R}(N, i|r)$ is finite).

Proof: From the second statement of Fact 2 the sets of control markings are finite for both N_1 and N_2. Obviously, the set of reachable control markings of N is a subset of a product of these two finite sets, hence it is also finite.

Now consider markings from $\mathcal{R}(N, i|r)$. Assume the converse — this set is infinite. Hence from the boundedness of the control subnet there exists some control cycle, strictly increasing the resource: $i|r \to c_1 + c_2|r' \xrightarrow{\sigma} c_1 + c_2|r' + r''$ with $c_1 \in \mathcal{M}((P_c)_1), c_2 \in \mathcal{M}((P_c)_2), \sigma \in T^*$ and $r'' \neq \emptyset$.

Recall that $T = T_1 \cup T_2$ and denote by σ_1 and σ_2 the largest subsequences of σ s.t. $\sigma_1 \in (T_1)^*$ and $\sigma_2 \in (T_2)^*$. Obviously, σ_1 and σ_2 are control cycles in N_1 and N_2 respectively.

From Proposition 1 neither σ_1 nor σ_2 can change the resource, hence their composition also cannot do this — a contradiction. ■

Since $D(N, i|r) \subseteq \mathcal{R}(N, i|r)$ we have:

Corollary 3: If $N = N_1 \| N_2$ and $r \in \mathbf{res}(N_1) \cap \mathbf{res}(N_2)$ then $D(N, i|r)$ is finite.

So the set of deadlocks and livelocks is computable by a simple reachability set construction and search. A naive deadlock control policy would be to compute a set of all deadlocks/livelocks and all their predecessing states and to control them in run-time, not allowing a system to make the wrong "last step".

A. Safe nets

A rather interesting case are safe workflows, i.e. RCWF-nets with safe control subnets (where none of the control places can accumulate more than one token). This is not a strong restriction, because every bounded net is weakly bisimilar to some safe net. Note that the net in Fig. 2(b) is safe and still has a deadlock.

A safe RCWF-net has only *ordinary* control arcs: $F_c(x, y) \leq 1$ for any x and y.

We can apply a transformation, eliminating all deadlocks/livelocks in a safe net:

Definition 3: Let N_1 and N_2 be sound safe RCWF-nets with the same set P_r of resource places, and let $r \in \mathcal{M}(P_r)$ be a resource s.t. $r \in \mathbf{res}(N_1)$ and $r \in \mathbf{res}(N_2)$.

Let $N = N_1 \| N_2 = (P_c, P_r, T, F_c, F_r, i, o)$ be a not-(r)-sound parallel composition of N_1 and N_2.

By $D_c(N, i|r)$ we denote the set of all different control parts of elements of $D(N, i|r)$, and let $Z = |D_c(N, i|r)|$ (obviously, we have $Z > 0$). And $D_r(N, i|r)$ we denote the set of critical resources, i.e. all resource places, which do not contain any tokens for each state of $D(N, i|r)$,

A net $(N_a, i|r + v)$, where $N_a = (P_c, P_r \cup V, T, F_c, F_r \cup F_{in} \cup F_{out}, i, o)$, is called a *controlled system* of $(N, i|r)$ iff

- $V = \{v_k | k \in \overline{1, Z}\}$ is a set of additional "holding" places, with their number equal to the number of possible control deadlocks/livelocks in the net N;
- F_{in} are input holding arcs such that $F_{in} = \{(t, v_k) \mid F_c(t, p) = 1$ for some $p \in d_k$, where d_k is a k^{th} element of $D_r(N, i|r)\}$;
- F_{out} are output holding arcs such that $F_{out} = \{(v_k, t) \mid F_c(p, t) = 1$ for some $p \in d_k$, where d_k is a k^{th} element of $D_r(N, i|r)\}$;
- $v = (|d_1| - 1)v_1 + (|d_2| - 1)v_2 + \ldots + (|d_Z| - 1)v_Z$, where d_k is a k^{th} element of $D_c(N, i|r)$.

Main idea to start with is that we have to avoid the consequent triggering of transitions leading to deadlock/livelock places. Thus for every element of $D(N, i|r)$ we include into the net a holding (restraining) place which resource will allow triggering of transitions leading to just one control place among them. Note that the holder should not prevent triggering transitions not consuming critical resources (since making it empty does not cause deadlock). Resource should be put back to holding place right after token leaves deadlock place. Implementation of this idea based on the net from Fig. 2 is shown on Fig. 3.

Note that the initial resource $(r + v)$ is actually not an increased original initial resource r, since v is built over the new (additional) set of resource places.

Theorem 3: Let N_1 and N_2 be sound safe RCWF-nets with the same set P_r of resource places, and let $r \in \mathcal{M}(P_r)$ be a resource s.t. $r \in \mathbf{res}(N_1)$ and $r \in \mathbf{res}(N_2)$.

Let $N = N_1 \| N_2 = (P_c, P_r, T, F_c, F_r, i, o)$ be a not-(r)-sound parallel composition of N_1 and N_2.

Let net $(N_a, i|r+v)$ be the controlled system of the marked net $(N, i|r)$. Then $(N_a, i|r + v)$ is $(r + v)$*–sound.*

Proof: Obviously, none of the original deadlocks/livelocks is reachable (by construction).

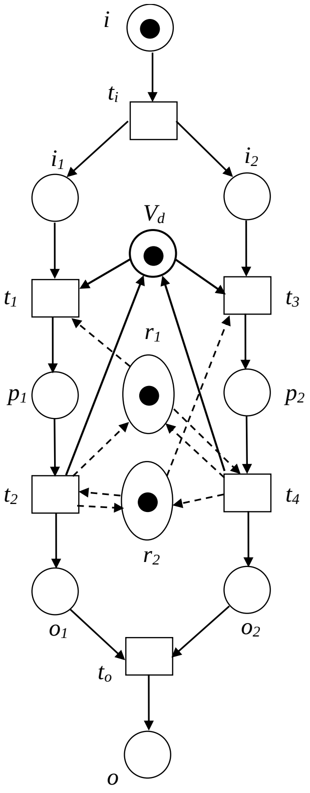


Fig. 3. Examples of RCWF-nets compositions with deadlock control

Now we need to prove that no new deadlocks/livelocks are introduced.

Consider some deadlock/livelock $c_1 + c_2|r$ of the original net. From the third statement of Fact 2 and the safety property none of the control states of N_1, except c_1 can have the same or larger total number of tokens in all places of c_1 (similarly for the net N_2 and places of c_2). Hence $|c_1+c_2|-1 = |c_1|+|c_2|-1$ tokens is enough for all control states except this particular deadlock/livelock — hence the corresponding holding place would not introduce any undesirable restriction. ■

VII. Conclusion

We presented two methods of deadlock/livelock avoidance for a restricted resource. The first one can be applied for any pair of sound workflows, but requires specific run-time control, not incorporated into the net itself. The second one uses the original Petri net structure, but is applicable to the safe nets only. The proposed technique is similar to a technique, studied in the area of Flexible Manufacturing Systems (see [9] for a classical result). However, the key difference is the possibility of parallel behaviours in subnets (in FMS each subnet is a simple sequential automaton).

Further research will consider the application of holding places techique to the general case of RCWF-nets. As it can be seen from some preliminary counterexamples, the method would possibly require some additional modifications.

The problem of exact $\mathbf{mres}(N)$ calculation is still open but it will be studied in the further research. We believe that our approach can be applied here, at least for large nontrivial subclasses of RCWF-nets. For example, we plan to study structured workflows [2], [8], that can be obtained from primitives by a set of algebraic operations, such as parallel and sequential compositions. Another interesting method of sound resource compositions was presented in [11] — based on algebraic expressions over multisets.

Our method is not implemented in applications. However, we believe it can be used as a verification and/or optimization tool in workflow management systems. Basically, it may enable the elimination of specific deadlocks and livelocks, induced by incorrect (unverified) parallel composition of submodules.

References

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

[2] W.M.P. van der Aalst, K.M. van Hee. *Workflow Management: Models, Methods and Systems*, MIT Press, 2002.

[3] W.M.P. van der Aalst, K.M. van Hee, A.H.M. Hofstede, N. Sidorova, H.M.W. Verbeek, M. Voorhoeve, M.T. Wynn. Soundness of workflow nets: classification, decidability, and analysis, *Form. Asp. of Comp.*, 23(3):333–363, Springer, 2011.

[4] K. Barkaoui, L. Petrucci. Structural Analysis of Workflow Nets with Shared Resources. In *Proc. Workflow Management: Net-based Concepts, Models, Techniques and Tools (WFM98)*, volume 98/7 of *Computing Science Reports*, pages 82–95, Eidhoven University of Technology, 1998.

[5] K. Barkaoui, R. Ben Ayed, Z. Sbaï. Workflow Soundness Verification based on Structure Theory of Petri Nets. *International Journal of Computing and Information Sciences*, Vol. 5, No. 1, 2007. P.51–61.

[6] V. A. Bashkin, I. A. Lomazova. Resource equivalence in workflow nets. In *Proc. Concurrency, Specification and Programming, 2006*, volume 1, pages 80–91. Berlin, Humboldt Universitat zu Berlin, 2006.

[7] V.A. Bashkin, I.A. Lomazova. Soundness of Workflow Nets with an Unbounded Resource is Decidable *Joint Proc. of Petri Nets and Software Engineering (PNSE'13) and Modeling and Business Environments (ModBE'13). Milano, 2013.* Vol. 989 of CEUR. 2013. P. 61–75.

[8] P. Chrząstowski-Wachtel. Sound Markings in Structured Nets. In *Proc. Concurrency, Specification and Programming, 2005*, pages 71–85. Warsaw, Warsaw University, 2005.

[9] J. Ezpeleta, J.-M. Colom, J. Martinez. A Petri Net Based Deadlock Prevention Policy for Flexible Manufacturing Systems. *IEEE Transactions on Robotics and Automation*, 11(2), 1995. P.173–184.

[10] K. van Hee, A. Serebrenik, N. Sidorova, M. Voorhoeve. Soundness of Resource-Constrained Workflow Nets. In *Proc. ICATPN 2005*, volume 3536 of *Lecture Notes in Computer Science*, pages 250–267. Springer, 2005.

[11] I.A. Lomazova, I.V. Romanov. Analyzing Compatibility of Services via Resource Conformance. *Fundamenta Informaticae*, Vol. 128, No. 1–2, 2013. P.129–141.

[12] N. Sidorova, C. Stahl. Soundness for resource-contrained workflow nets is decidable. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 43(3), 2013. P.724–729.