Soundness for Resource-Constrained Workflow Nets is Decidable

Natalia Sidorova and Christian Stahl

***Abstract*—We investigate the verification of the *soundness* property for workflow nets extended with resources, thereby considering the *most general* instance of soundness, which requires that for any number of instances, the workflow net has always the possibility to terminate, for a certain initial (finite) number of resource items per resource type; moreover, adding additional resources to a sound net does not violate the result. We prove that this problem is *decidable* by reducing it to a home-space problem, and we show how soundness can be decided by using the procedure for deciding a home-space property.**

***Index Terms*—Decidability, Petri nets, Workflow nets, Resource-constrained workflow nets, Soundness, Home-space**

I. Introduction

INFORMATION systems have become the backbone of most organizations. Processes form the core of most information systems [1]. They orchestrate people, information, and technology to deliver products. In this paper, we focus on *workflows*. A workflow refers to the automation of a process by an IT infrastructure, in whole or in part [2].

A workflow consists of a set of coordinated tasks describing the flow of work within the organization. The occurrence of those tasks may depend on *resources*, such as machines, manpower, and raw material. Often, several *cases* (i.e., instances) of a workflow may coexist, and they may all concurrently access the resources. Thus, the execution of a workflow can be seen as executing several threads of a piece of software.

A workflow forms a *parameterized system* with two parameters: the number k of cases and the vector R of resources, indicating a finite number of resources available for each resource type. Although we assume the workflow (and thus every case) to be finite state and also the number of resources available for each resource type to be finite, the total number of cases can be *unbounded*, and the number of resources available differs for each organization for the same workflow, making the analysis of such a system challenging.

One of the most established correctness criteria for workflows is the *soundness* property. In its most general form, soundness guarantees that for any number k of cases, there exists a number of resources of each type such that all cases have always the possibility to terminate. In addition, we require that adding resources to the workflow do not violate the result. As we restrict ourselves to *durable* resources in this paper, i.e. resources that can neither be created nor destroyed, soundness also ensures that the number of resources initially available remains invariant.

N. Sidorova and C. Stahl are with the Department of Mathematics and Computer Science, Technische Universiteit Eindhoven, The Netherlands (e-mail: {N.Sidorova,C.Stahl}@tue.nl).

Manuscript received ; revised .

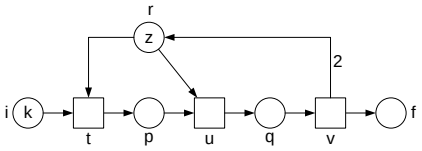

Fig. 1. An unsound resource-constrained workflow net.

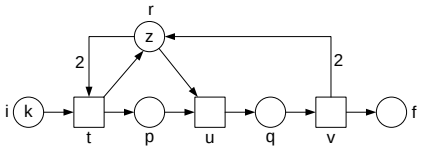

Fig. 2. A sound variant of the resource-constrained workflow net in Figure 1.

For the modeling of workflows, *workflow nets* have been established [3] and later also extended to deal with resources [4], [5]. To illustrate the model of workflow nets extended with resources and the soundness property, consider the simple example in Figure 1. Every case has to sequentially execute tasks t, u, and v. Each task is modeled as a transition in Figure 1. The tasks depend on one type of resources, modeled as tokens in place r. To execute task t, one resource is taken; u requires one resource, and v returns the two resources. A (fresh) case is modeled as a token in place i, a terminated case as a token in place f. The net in Figure 1 is unsound: For any number k of tokens in i (i.e., cases) and any number z of tokens in r (i.e., resources) such that $k = z$, it is always possible to reach a marking with k tokens in p and zero tokens in r. In this (nonfinal) marking, the net is stuck and, therefore, it is not sound. The cause of the deadlock in Figure 1 is that task t may be executed although there are not enough resources to continue with task u.

Figure 2 shows another resource-constrained workflow net, which is a slightly modification of Figure 1. In this net, task t takes two resources rather than one but returns one resource. This simple change guarantees that at least one instance has the possibility to execute task u because an instance that enters place p will always return one resource on r. As a consequence, for any number k of tokens in i and any number z of tokens in r, it is always possible to reach a marking with k tokens in f and z tokens in r. Thus, we conclude that Figure 2 is sound.

Verification of soundness has been addressed by several researchers. However, they reduce the complexity caused by the two sources of unboundedness—the number of cases and the addition of (arbitrary many) resources—by considering simpler instances of this problem. Van Hee et al. [5] restrict the number of resource types to one, and Barkaoui et al. [6]

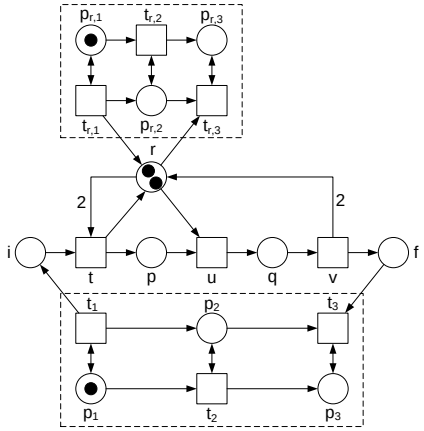

Fig. 3. Instantiation net [8] for Figure 2 in case of $z = 2$.

present solutions for restricted subclasses of workflow nets. Soundness of the net in Figure 2 can be proved using the technique in [5], for instance. In [5] remained still the open question whether soundness in this general setting is decidable, because the verification of parameterized systems is known to be undecidable in general [7].

Recently, the authors in [8] tried to address the question whether soundness is decidable and presented a scheme for reducing this problem to a *home marking problem*. Unfortunately, the reduction scheme presented in [8] does not work. The error can easily be illustrated on the example in Figure 2. The proof idea in [8] is to construct a so-called case-resource instantiation net that instantiates a given net with an arbitrary number of instances and resources. Figure 3 illustrates the construction of a case-resource net from [8] applied to the net in Figure 2. The subnet on the bottom generates any number k of instances on the start place i by firing transition t_1. After firing t_2, this subnet can consume all terminated cases from place f by firing t_3. Likewise, the subnet at the top generates any number j of additional resources on the resource place r (by firing transition $t_{r,1}$). After firing $t_{r,2}$, the additional resources can be removed from place r by firing $t_{r,3}$.

Lemma 3.1(3) of [8] gives a false statement about the equivalence of the soundness problem of a resource-constrained workflow net to a home marking problem for its case-resource instantiation net. According to this lemma, the net in Figure 2 is sound if and only if for any reachable marking of the net in Figure 3, it is always possible to reach a marking with one token in p_3 and one token in place $p_{r,3}$ (i.e., this marking is a home-marking). However, the subnet at the bottom of Figure 3 might first put tokens on place r (i.e., it adds additional resources) and then steal those resources by firing transition $t_{r,3}$ before the running cases are terminated. In the example, we start with the marking $[p1, p_{r,1}, 2r]$ (i.e., one token on place p_1, one token on place $p_{r,1}$, and two tokens on r—the marking shown in Figure 3). Firing transition sequence $t_1\, t_1\, t_{r,1}\, t\, t\, t_{r,2}\, t_{r,3}$ creates two case instances, adds one resource, executes the task t of the workflow for both instances, and finally steals the only resource left available, thus yielding the marking $[p_1, p_{r,3}, 2p]$ in which the workflow net is deadlocked because no resource is available and, thus, transition u cannot be fired while both cases need a resource to proceed by firing u. One cannot easily repair the construction in Figure 3 and find a way to reduce the problem to the home marking problem. This would require to guarantee that transition $t_{r,3}$ can fire only if t_3 cannot become enabled (i.e., all cases have terminated and have been removed from f by firing t_3). This can, however, only be achieved by introducing an inhibitor arc for each resource place (i.e., an arc that tests whether a place contains zero tokens). The home-marking problem is, however, in general undecidable for Petri nets with inhibitor arcs [9], [10].

So the question whether soundness is decidable is still not answered. In this paper, we show that soundness verification for arbitrary resource-constrained workflow nets extended is decidable by reducing it to a *home-space property*. We also show how soundness can be decided by applying the decision procedure for deciding home-space properties.

Organization of the paper: We continue by providing the background in Section II. In Section III, we introduce our model of resource-constrained workflow nets—that is, workflow nets extended with resources. Next, in Section IV, we prove that soundness for resource-constrained workflow nets is decidable by reducing it to verifying a home-space property. We present an algorithm for deciding soundness in Section V. We discuss related work in Section VI and close with a conclusion.

II. Preliminaries

In this section, we provide the basic notations used in this paper, such as Petri nets and workflow nets.

A. Petri nets

Symbol $\mathbb{N}$ denotes the set of natural numbers, symbol $\mathbb{Z}$ the set of all integers, and symbol $\mathbb{Q}$ the set of rational numbers.

For two sets P and Q, let $P \uplus Q$ denote the disjoint union; writing $P \uplus Q$ expresses the implicit assumption that P and Q are disjoint. A *multiset* or bag m over P is a mapping $m : P \to \mathbb{N}$; for example, $[p_1, 2p_2]$ denotes a multiset m with $m(p_1) = 1$, $m(p_2) = 2$, and $m(p) = 0$ for $p \in P \setminus \{p_1, p_2\}$. We define $+$ for the sum and $-$ for the difference of two multisets and $=, <, >, \leq, \geq$ for comparison of multisets in the standard way. We overload the set notation, writing $\emptyset$ for the empty bag and $\in$ for the element inclusion. We canonically extend the notion of a multiset over P to supersets $Q \supseteq P$; that is, for a mapping $m : P \to \mathbb{N}$, we extend m to the multiset $m : Q \to \mathbb{N}$ so that for all $p \in Q \setminus P$, $m(p) = 0$. Analogously, a multiset can be restricted to a subset $Q \subseteq P$. For a mapping $m : P \to \mathbb{N}$, the *restriction* of m to the elements in Q is denoted by $m|_Q : Q \to \mathbb{N}$.

Definition 2.1 (Petri net): A *Petri net* $N = \langle P, T, F^+, F^- \rangle$ consists of

- a nonempty finite set P of *places*,
- a nonempty finite set T of *transitions* such that P and T are disjoint,
- a *mapping* $F^+ : (P \times T) \to \mathbb{N}$ from transitions to places, and
- a *mapping* $F^- : (P \times T) \to \mathbb{N}$ from places to transitions.

$\mathbf{C} = F^+ - F^-$ denotes the *incidence matrix* of N.

A *marking* $m : P \to \mathbb{N}$ is a distribution of tokens over the places. With m_0 we denote the initial marking of N, and (N, m_0) denotes a Petri net N with initial marking m_0. Depending on the context, we interpret a marking m of N either as a multiset over P or as a vector from $P \to \mathbb{N}$.

Graphically, a circle represents a place, a box represents a transition, and the directed arcs between places and transitions represent the flow relation. A marking is a distribution of tokens over the places. Graphically, a black dot represents a token.

For a transition $t \in T$, we define the *preset* ${}^\bullet t$ and the *postset* $t^\bullet$ of t as the multisets of places where every $p \in P$ occurs $F^-(p,t)$ times in ${}^\bullet t$ and $F^+(p,t)$ times in $t^\bullet$. Analogously, we define for a place $p \in P$ its preset ${}^\bullet p$ and its postset $p^\bullet$. We also lift pre- and postsets to sets of places and of transitions. A place p is a *source* place if ${}^\bullet p = \emptyset$ and a *sink* place if $p^\bullet = \emptyset$.

The *behavior* of a Petri net N relies on the marking of N and the marking changes by the firings of transitions of N. A transition $t \in T$ is *enabled* at a marking m, denoted by $m \xrightarrow{t}$, if ${}^\bullet t \leq m$. If t is enabled at m, it can *fire*, thereby changing the marking m to a marking $m' = m - {}^\bullet t + t^\bullet$. The firing of t is denoted by $m \xrightarrow{t} m'$; that is, t is enabled at m and firing it results in m'.

The behavior of N can be extended to sequences: $m_1 \xrightarrow{t_1} \ldots \xrightarrow{t_{k-1}} m_k$ is a *run* of N if for all $0 < i < k$, $m_i \xrightarrow{t_i} m_{i+1}$. A marking m' is *reachable from* a marking m if there exists a (possibly empty) run $m_1 \xrightarrow{t_1} \ldots \xrightarrow{t_{k-1}} m_k$ with $m = m_1$ and $m' = m_k$; for $v = t_1 \ldots t_k$, we also write $m \xrightarrow{v} m'$. Marking m' is *reachable* if $m_0 = m$. The set $\mathcal{R}(m)$ represents the set of all markings of N that are reachable from m. If not clear from the context, we use $\to_N$ instead of $\to$ to emphasize that we consider the behavior of N.

We shall also use the exchange lemma [11] providing a condition under which the order of transitions in a transition sequence can be exchanged.

Proposition 2.1 ([11]): Let U and V be disjoint subsets of transitions of a Petri net N satisfying ${}^\bullet U \cap V^\bullet = \emptyset$. Let $A \subseteq U \uplus V$, and let $\sigma \in A^*$ be a sequence of transitions. Then, $m \xrightarrow{\sigma} m'$ in N implies $m \xrightarrow{\sigma|_U \sigma|_V} m'$ in N.

A marking m is a *home-marking* if from every reachable marking we can reach m. A set *HS* of markings of N is a *home-space* if for every reachable marking m, there exists a marking $m' \in HS$ such that m' is reachable from m.

A *place invariant* is a row vector $I : P \to \mathbb{Q}$ such that $I \cdot \mathbf{C} = 0$. When talking about invariants, we consider markings as *vectors*.

B. Workflow nets

A workflow refers to the automation of processes by an IT infrastructure, in whole or in part [2]. Workflows are *case*-based; that is, every piece of work is executed for a specific case. One can think of a case as a workflow instance, such as a mortgage, an insurance claim, or a purchase order. Each case is handled individually according to the workflow definition. The workflow definition specifies which tasks need to be executed for a case and in what order. The order, in which tasks are executed, is determined by conditions specifying dependencies between tasks.

We can model a workflow definition as a Petri net, thereby modeling tasks by transitions and conditions by places; the state of a case is captured by a marking of the net. The assumption that a typical workflow has a well-defined starting point and a well-defined ending point imposes syntactic restrictions on Petri nets that resulted in the following definition of a workflow net [12].

Definition 2.2 (WF-net): A Petri net $N = \langle P, T, F^+, F^- \rangle$ is a *workflow net* (WF-net) if it has a single source place i, a single sink place f, and every place and every transition is on a path from i to f.

In the first instance, researchers were interested in workflow correctness with respect to a single case. One of the most established correctness properties of WF-nets is *soundness*, as introduced by Van der Aalst [3] in the context of one case. Soundness guarantees that the workflow has always the possibility to terminate. Later on, multi-instance behavior attracted researchers' attention, where WF-nets are considered as parameterized systems modeling the processing of batches of tasks, as introduced in [13]. While in classical workflows, cases are considered to be independent and the modeling of multiple cases in one workflow net requires the introduction of id tokens, in batch workflows cases are considered to be undistinguishable and mixable (e.g., it does not matter which bicycle gets which wheel) and, as a consequence, cases are modeled with undistinguishable black tokens. Under certain conditions on the workflow structure, called *separability*, the behavior of the workflow net with undistinguishable cases (black tokens) is equivalent (up to trace equivalence) to the behavior of the workflow net with id tokens [13]–[15]. Moreover, every net with id tokens can be transformed into an up-to-bisimulation-equivalent net with black tokens only [13], [16].

Capturing the correctness notion for batch workflow nets requires the use of the generalized notion of soundness, as proposed in [13].

Definition 2.3 (soundness of a WF-net): Let $k \in \mathbb{N}$ and N be a WF-net.

- N is *k-sound* if, for every marking m reachable from marking $k[i]$, we can reach marking $k[f]$.
- WF-net N is *sound* if it is k-sound for all $k \in \mathbb{N}$.

The next definition gives a structural requirement for the correct design of a workflow. *Nonredundancy* of a place $p \in P$ guarantees that p can potentially be marked with a token in some reachable marking.

Definition 2.4: Let $N = \langle P, T, F^+, F^- \rangle$ be a WF-net. A place $p \in P$ is *nonredundant* if there exist $k \in \mathbb{N}$ and $m \in \mathbb{N}^P$ such that $k[i] \xrightarrow{*} m \wedge p \in m$.

Example 2.1: Consider the Petri net in Figure and ignore place r and its adjacent arcs. The resulting Petri net is a workflow net. The net is sound and every place is nonredundant.

III. Resource-constraint workflow nets

Workflow nets specify the handling of tasks within an organization, but they do not model resources necessary for the execution. In other words, a workflow net models the process perspective of a workflow while abstracting from resources. However, it is known that excluding resources from the model can lead to wrong verification results (see [17], for instance). To overcome this, we extend workflow nets with resource information. The resulting model are *resource-constrained workflow nets*, which have been introduced in [4], [5].

A resource belongs to a *type*; thus, we model each resource type as a place. Each token in such a place models an available resource of the respective resource type. Resources become part of a case when they are occupied. In this paper, we assume that resources are *durable*; that is, they can neither be created nor destroyed. Resources are claimed during the execution of a case and then released. By abstracting from the resource places and its adjacent arcs, we obtain the WF-net to which we refer as the *production net*.

Definition 3.1 (RCWF-net): A Petri net $N = \langle P_p \uplus P_r, T, F_p^+ \uplus F_r^+, F_p^- \uplus F_r^- \rangle$ is a *resource-constrained workflow net* (RCWF-net) if

- $N_p = \langle P_p, T, F_p^+, F_p^- \rangle$ is a WF-net, the *production net* of N;
- P_p is the set of *production places*, and P_r is the set of *resource places*;
- $F_r^+ : (P_r \times T) \to \mathbb{N}$ maps transitions to resource places; and
- $F_r^- : (P_r \times T) \to \mathbb{N}$ maps resource places to transitions.

The initial marking $m_0 = k[i] + R$ of an RCWF-net N consists of a number $k \in \mathbb{N}$ tokens in place i, specifying the number of cases in the workflow that are concurrently executed, and an initial marking for the set P_r of resources places, denoted as a resource vector $R \in \mathbb{N}^{P_r}$.

Example 3.1: Figures 1 and 2 show two RCWF-nets, each containing a single resource place r. Removing this place and its adjacent arcs yields the respective production net. The initial marking of both RCWF-nets is $m_0 = k[i] + z[r]$; that is, there are k tokens on the initial place and z tokens on resource place r.

We adapt the definition of soundness for WF-nets to RCWF-nets. Soundness of an RCWF-net N guarantees that the underlying production net of N is sound; that is, also in the presence of resources, a case has always the possibility to terminate. In addition, we put two conditions on the resources: First, we require that all resources that are initially available are again available after all cases are terminated. Second, we also require that at any reachable marking, the number of available resources does not increase the number of initially available resources. These two criteria are a consequence of our restriction to durable resources, because they ensure that no resources are created or removed.

Definition 3.2 (soundness of an RCWF-net): Let N be an RCWF-net.

- N is *(k, R)-sound* for some $k \in \mathbb{N}, R \in \mathbb{N}^{P_r}$ if for all $m \in \mathcal{R}(k[i] + R) : m \xrightarrow{*} (k[f] + R) \wedge m|_{P_r} \leq R$.
- N is *sound* if there exists $R_0 \in \mathbb{N}^{P_r}$ such that, for all $k \in \mathbb{N}, R \in \mathbb{N}^{P_r}$ with $R \geq R_0$, N is (k, R)-sound. In this case we also say that N is sound for R_0.

Example 3.2: The RCWF-net in Figure 1 is, for example, $1, z$-sound for all $z \geq 2$, but it is not sound. In contrast, the RCWF-net in Figure 2 is sound for $z[r]$ with $z \geq 2$.

We now recapitulate three necessary conditions for soundness taken from [18]. The first condition ensures that no resource tokens can be created; that is, if N initially contains R tokens on its resource places, then every reachable marking has a resource vector $R' \leq R$. The second condition states that there exists a place invariant for places i and f, guaranteeing that the number of instances remains constant. Likewise, the third condition requires that, for every resource place, there exists a place invariant, guaranteeing that the number of resources remains constant.

Proposition 3.1 ([18]): For any sound RCWF-net N without redundant places in its production net, we have

1) $\forall x \in \mathbb{Z}^T : (\mathbf{C} \cdot x)|_{P_p \setminus \{i\}} \geq 0$ implies $(\mathbf{C} \cdot x)|_{P_r} \leq 0$.
2) There exists a place invariant I_p such that $I_p(i) = I_p(f) = 1$ and, for all $r \in P_r$, $I_p(r) = 0$.
3) For each $r \in P_r$, there exists a place invariant I_r satisfying $I_r(i) = I_r(f) = 0, I_r(r) = 1$, and $\forall r' \in P_r \setminus \{r\} : I_r(r') = 0$.

Moreover, due to parametrization of the number of resources in the definition of soundness for RCWF-nets, soundness of an RCWF-net implies soundness of its production net:

Proposition 3.2 ([18]): Let N be a sound RCWF-net. Then its production net N_p is sound too.

RCWF-nets satisfying the properties in Proposition 3.1 and having a sound production net can be unsound only if they contain a deadlock or a livelock due to a lack of resources during the production process.

Example 3.3: For the RCWF-net in Figure 2, $i + p + q + f$ is an invariant in the production net according to Proposition 3.1(2) and $r + p + 2q$ an invariant for resource place r according to Proposition 3.1(3). Furthermore, the production net is sound. In contrast, the RCWF-net in Figure 1 has the same invariants and a sound production net, but it can deadlock due to the lack of resources, as shown in the introduction, and is therefore not sound.

IV. Decidability of soundness

Given an RCWF-net N together with a proper initial marking R_0 for the resource places (e.g., given by an oracle), we show that checking soundness of N reduces to deciding a home-space property for a modified version of N. As the latter problem is known to be decidable [19], we can conclude that checking soundness is decidable as well.

Soundness of an RCWF-net N requires that, for all $k \in \mathbb{N}, R \in \mathbb{N}^{P_r}$ with $R \geq R_0$, net N is (k, R)-sound. To cover different initial markings (i.e., the number of tokens in place i and in the resource places of N) in one Petri net, we modify N so that it can arbitrarily increase its initial marking on i and on the resource places. Figure 4 illustrates the construction. It is similar to a construction used by Juhas et al. in [16], which allows adding tokens on the initial place; in our construction,

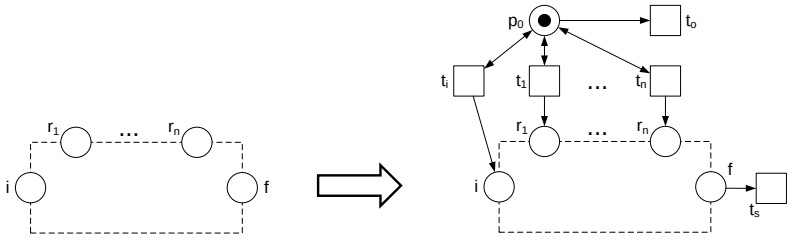

Fig. 4. Constructing the transformed RCWF-net.

we can add tokens to resource places as well. We refer to the resulting net as the *transformed RCWF-net* N' of N. It has an additional place p_0 which is initially marked with one token. The token in this place enables transitions t_i, $t_1, \ldots, t_{|P_r|}$, and t_o. A firing of transition t_i produces a token in the initial place i, thereby increasing the number of cases running in N; each transition $t_1, \ldots, t_{|P_r|}$ produces a token in the respective resource place. Firing transition t_o removes the token from place p_0; that is, after this transition has fired, the the number of tokens in i and the resource places cannot be increased any more. In addition, we add a transition t_s to N' that removes the tokens from f corresponding to the completed cases.

Definition 4.1 (transformed RCWF-net): The *transformed RCWF-net* N_{tr} of an RCWF-net $N = \langle P_p \uplus P_r, T, F_p^+ \uplus F_r^+, F_p^- \uplus F_r^- \rangle$ with n resource types (i.e., $|P_r| = n$) is the tuple $\langle P_{tr}, T_{tr}, F_{tr}^+, F_{tr}^- \rangle$ with

- $P_{tr} = P_p \uplus P_r \uplus \{p_0\}$,
- $T_{tr} = T \uplus \{t_i, t_o, t_s\} \uplus \{t_j \mid 1 \leq j \leq n\}$,
- For all $p \in P_p, t \in T$, $F_{tr}^+(p,t) = F_p^+(p,t)$ and $F_{tr}^-(p,t) = F_p^-(p,t)$;
  for all $p \in P_r, t \in T$, $F_{tr}^+(p,t) = F_r^+(p,t)$ and $F_{tr}^-(p,t) = F_r^-(p,t)$;
  ${}^\bullet p_0 = [t_i, t_1, \ldots, t_n]$;
  $p_0^\bullet = [t_o, t_i, t_1, \ldots, t_n]$;
  ${}^\bullet t_i = {}^\bullet t_1 = \ldots = {}^\bullet t_n = [p_0]$;
  $t_i^\bullet = [p_0, i]$ and $t_j^\bullet = [p_0, r_j]$, for all $1 \leq j \leq n$;
  ${}^\bullet t_s = [f]$ and $t_s^\bullet = \emptyset$;
  ${}^\bullet t_o = [p_0]$ and $t_o^\bullet = \emptyset$.

The rest of this section is devoted to the proof that soundness is decidable. To this end, we show that N is sound if and only if its transformed RCWF-net N_{tr} has a particular home-space property. For the implication, we show that every firing sequence in N_{tr} can be reshuffled using Proposition 2.1 such that the resulting sequence contains a transition sequence that can be executed in N. By the soundness of N, we can conclude the home-space property of N_{tr}.

Lemma 4.1: Let $N = \langle P_p \uplus P_r, T, F_p^+ \uplus F_r^+, F_p^- \uplus F_r^- \rangle$ be a sound RCWF-net. Then, the transformed RCWF-net N_{tr} of N, initialized with marking $m_0 = [p_0] + R_0$, has the home-space $HS = \{R \mid R \in \mathbb{N}^{P_r} : R \geq R_0\}$.

Proof: We have to show that for any marking $m \in \mathcal{R}_{N_{tr}}(m_0)$, there is a marking $m' \in HS$ such that $m \xrightarrow{*} m'$. Let σ be a transition sequence from the initial marking of N_{tr} to m. We use Proposition 2.1 to reshuffle the firing sequence σ. Observe that for $U = \{t_i, t_1, \ldots, t_n\}$, we have ${}^\bullet U = [p_0]$ and ${}^\bullet p_0 \cap T = \emptyset$; that is, p_0 does not belong to the postset of any subset of transitions of N. Thus we can move all the firings of $\{t_i, t_1, \ldots, t_n\}$ to the beginning of the firing sequence, obtaining a firing sequence $\sigma_1 = \sigma|_{\{t_i,t_1,\ldots,t_n\}}\sigma|_{T \uplus \{t_o,t_s\}}$ and, by Proposition 2.1, $m_0 \xrightarrow{\sigma_1} m$.

Next we apply Proposition 2.1 again, for σ_1, $V = \{t_s\}$ and U consisting of the rest of the transitions of N_{tr}. Clearly, since $t_s^\bullet = \emptyset$, the conditions of Proposition 2.1 are satisfied and we can delay all the firings of t_s until the end of the firing sequence σ, obtaining $\sigma_2 = \sigma|_{\{t_i,t_1,\ldots,t_n\}}\sigma|_{T \uplus \{t_o\}}\sigma|_{\{t_s\}}$ and, by Proposition 2.1, $m_0 \xrightarrow{\sigma_2} m$.

Finally, we apply Proposition 2.1 to $\sigma|_{T \uplus \{t_o\}}$. Since ${}^\bullet t_o = [p_0]$ and $p_0 \notin T^\bullet$, we can move firings of t_o to the beginning of the sequence. Thus, we obtain a firing sequence $\sigma' = \sigma|_{\{t_i,t_1,\ldots,t_n\}}\sigma|_{\{t_o\}}\sigma|_T\sigma|_{\{t_s\}}$.

The firing of $\sigma|_{\{t_i,t_1,\ldots,t_n\}}\sigma|_{\{t_o\}}$ in N_{tr} leads to a marking $m_1 = k[i] + R_0 + R$ for some $k \in \mathbb{N}, R \in \mathbb{N}^{P_r}$, which is a possible initial marking of N. Due to the construction of the transformed RCWF-net N_{tr}, $\sigma|_T$ is firable in (N, m_1), leading to the same marking in N and N_{tr}: $m_1 \xrightarrow{\sigma|_T} m_2$.

In N_{tr}, we have $m_2 \xrightarrow{\sigma|_{\{t_s\}}} m$. In N, due to its soundness, there is a firing sequence $\gamma \in T^*$ such that $m_2 \xrightarrow{\gamma} k[f] + R_0 + R$. Since ${}^\bullet t_s = [f]$ in N_{tr} while f is a sink place in N, we have $m_2 \xrightarrow{\sigma|_{\{t_s\}}} m \xrightarrow{\gamma} \ell[f] + R_0 + R$ with $\ell = k - |(\sigma|_{\{t_s\}})|$ in N_{tr}. We can conclude that $m \xrightarrow{\gamma \cdot (t_s)^\ell} R_0 + R$ and since $R_0 + R \in HS$, the home-space property is proven. Note that only transitions of $T \cup \{t_o\}$ are used to reach a home-space marking from an arbitrary reachable marking. ■

For the reverse implication of our decidability theorem, we must prove the converse of Lemma 4.1. Interestingly, this statement only holds if we add additional assumptions.

Lemma 4.2: Let N be an RCWF-net without redundant places in its production net, for which the three conditions of Proposition 3.1 hold, and N_{tr} be its transformed RCWF-net with home-space $HS = \{R \mid R \in \mathbb{N}^{P_r} : R \geq R_0\}$. Then N is sound.

Proof: We have to show that for all $m \in \mathcal{R}_N(k[i] + R)$ with $k \in \mathbb{N}, R \in \mathbb{N}^{P_r}, R \geq R_0$, there is a firing sequence $m \xrightarrow{*} k[f] + R$. Let σ be a transition sequence from the initial marking $m_0 = k[i] + R$ of N to m. Firing sequence $\sigma_0 = (t_i)^k (t_1)^{(R-R_0)(r_1)} \ldots (t_n)^{(R-R_0)(r_n)}$ leads to marking m_0 in (N_{tr}, R_0): $R_0 \xrightarrow{\sigma_0} m_0$ and σ is firable in (N_{tr}, m_0): $m_0 \xrightarrow{\sigma} m$.

Because of the home-space property, there exists a firing sequence $\gamma \in (T \uplus \{t_o\})^*$ (see Lemma 4.1) in N_{tr} such that $m \xrightarrow{\gamma} R_1$. Let $|(\gamma|_{\{t_o\}})| = \ell$. Applying Proposition 2.1 to γ and $V = \{t_o\}$, we obtain $m \xrightarrow{\gamma|_T} \ell[f] + R_1 \xrightarrow{(t_o)^\ell} R_1$.

Due to the construction of N_{tr}, $\gamma|_T$ is a firing sequence of N, too. Proposition 3.1(3) implies that $R_1 = R$, while Proposition 3.1(2) implies that $\ell = k$. Furthermore, Proposition 3.1(1) implies that every marking on the path from m to $\ell[f] + R_1$ does at most contain R resources. Thus $m \xrightarrow{\gamma|_T} k[f] + R$, implying that N is sound. ■

Given a resource vector R_0, we have a necessary and sufficient condition for soundness of N. Moreover, this condition reduces the soundness check for N to checking a home-space property of the transformed RCWF-net of N.

Theorem 4.1 (Soundness is decidable): Let N be an RCWF-net. Then, soundness of N is decidable.

Proof: First, remove the redundant production places from N. The obtained net N' has the same behavior as N, because the redundant places can never get a token, and therefore N' is sound if and only if N is sound. Then check whether the three properties of Proposition 3.1 hold for N'. If not, the net is not sound. If they do hold, by Lemmata 4.1 and 4.2, checking soundness of N' (and hence of N) reduces then to checking a home-space property of the transformed RCWF-net (N'_{tr}, R_0) of N_{tr}. The latter is decidable, as shown in [19]; thus, checking soundness of N is decidable, too. ■

V. An algorithm for deciding soundness

Using the construction from [19], we show that in order to check soundness it is sufficient to check proper termination for the set of *minimal reachable markings* containing at least one token on some production place.

We first partition the set of the reachable markings into the set of resource markings and the markings containing tokens on production places; that is, $\mathcal{R}(N, R_0 + [p_0]) = R^{P_r} \uplus R^{P_p}$ with $R^{P_r} = \mathcal{R}(N, R_0 + [p_0]) \cap \mathbb{N}^{P_r}$ and $R^{P_p} = \mathcal{R}(N, R_0 + [p_0]) \setminus \mathbb{N}^{P_r}$. The home-space property holds for any $m \in R^{P_r}$. We shall show that the home-space property holds for R^{P_p} if and only if it holds for the set $R^{P_p}_{min}$ of minimal markings of R^{P_p}, which is defined as $R^{P_p}_{min} = \{m \mid m \in R^{P_p} \wedge m \neq \emptyset \wedge \nexists m' \in R^{P_p} : m' < m\}$.

Lemma 5.1: Let N be an RCWF-net and (N_{tr}, R_0) be its transformed RCWF-net. The home-space property holds for $R^{P_p}_{min}$ iff it holds for R^{P_p}.

Proof: If there is a marking $m \in R^{P_p}_{min}$ for which the home-space property does not hold, then the home-space property does not hold for $(N, R_0 + [p_0])$ because $R^{P_p}_{min} \subseteq R^{P_p} \subseteq \mathcal{R}(N, R_0 + [p_0])$.

Let the homes-space property hold for all markings $m \in R^{P_p}_{min}$. We partition R^{P_p} into subsets according to the number of tokens on the production places: $R^{P_p} = \biguplus_{i=1}^{\infty} R_i$, where $R_i = \{m \mid m \in R^{P_p} \wedge \sum_{p \in P_p} m(p) = i\}$, and we define $R_0 = R^{P_r}$.

We prove by induction on i that all $m \in \mathcal{R}(N, R_0 + [p_0])$ have the home-space property.

For $i = 0$ the induction hypothesis holds trivially.

Let it hold up to some $k \in \mathbb{N}$. Take $m \in R_{k+1}$. If $m \in R^{P_p}_{min}$, then the home-space property holds. If not, there is a marking $m_1 \in R^{P_p}_{min}$ such that $m_1 < m$. Since $m_1 \in R^{P_p}_{min}$, the home-space property holds for m_1: $m_1 \xrightarrow{\sigma} m_h$ for some $\sigma \in T^*$, $m_h \in R^{P_r}$.

Then $m \xrightarrow{\sigma} m_h + (m - m_1)$ and $m_h + (m - m_1) \in \mathcal{R}(N, R_0 + [p_0])$. Since $m_1 \in R^{P_p}_{min} (\subseteq R^{P_p})$, m_1 contains at least one token on some production place, and m_h does not contain tokens on the production places. Therefore, the number of tokens on the production places of $\sum_{p \in P_p}(m_h + (m - m_1))(p) < \sum_{p \in P_p} m(p)$. Therefore, the induction hypothesis holds for $(m_h + (m - m_1))(p)$ and thus $m \xrightarrow{\sigma} m_h + (m - m_1) \xrightarrow{*} m'_h$ for some $m'_h \in R^{P_r}$, which implies that the home-space property holds for m as well. ■

The problem remains is to efficiently compute the set of minimal markings of R^{P_p}.

Example 5.1: The minimal markings for Figure 1 are $R^{P_p}_{min} = \{[i, 2r], [2p], [q]\}$. It is easy to see that $[2p]$ is a deadlock, proving the net to be unsound. For Figure 2, we obtain $R^{P_p}_{min} = \{[i, 2r], [p, r], [q]\}$. As we can reach a final marking from all these markings, we conclude soundness.

VI. Related work

The verification of soundness for workflow nets (WF-nets) extended with resources has been investigated by many researchers. Extending workflows with resources resulted in the model of resource-constrained workflow nets (RCWF-nets) [4], [5]. To cope with the resource parameter, the notion of (generalized) soundness [20] for WF-nets had to be adapted [5]. We distinguish between approaches where the number of resources is assumed to be fixed and those approaches where it is variable, meaning, adding additional resources does not violate the soundness property.

Juhas et al. [16] present for a weaker problem instance where only the absence of deadlocks is considered a reduction to an ILP problem. Van Hee et al. [5] solve the problem instance of a variable number of resources for a single resource type. They transform the workflow part of the RCWF-net into a state machine and annotate the transitions of this net with the effect on the resource place. The algorithm is then based on place invariants. In [18], Van Hee et al. define four necessary criteria based on traps and siphons for analyzing the general instance of soundness of RCWF-nets.

Barkaoui et al. [6] investigate the verification of soundness for three restricted classes of RCWF-nets. In their setting, the verification of soundness boils down to checking boundedness and some structural property (i.e., commoner's property—every minimal siphon is trap controlled).

There also exist extensions of the temporal logics CTL and ATL to reason about resources [21], [22]. Although the problem instance considered in this paper can be expressed in terms of those logics, verification would require to check the system for all parameters.

VII. Conclusion

We have investigated the soundness property for resource-constrained workflow nets (RCWF-nets) in its most general form. An RCWF-net is sound if there exists a number R_0 of resources for each of its finitely many resources types such that for every finite number k of workflow cases and any greater number R of resources, it is always possible to reach a state where all k cases are terminated and the resources R

are available. We have proved soundness to be decidable by reducing the problem to checking a home-space property in (a modified version) of the net. In addition, we have shown that the reduction schema used to prove decidability in [8] is wrong.

Although soundness is decidable, there is so far no efficient decision algorithm because our proposed algorithm decides a home-space property, which requires a finite but (in general) too high number of reachability checks. Ongoing research is devoted to study more efficient algorithms. In addition, we keep as an open problem the calculation of the smallest number of resources R_0 for which soundness can be proved.

ACKNOWLEDGMENT

The research of Christian Stahl is supported by the NWO project "Behavior-Oriented Service Substitution."

REFERENCES

[1] M. Dumas, W. M. P. v. d. Aalst, and A. H. M. t. Hofstede, *Process-Aware Information Systems: Bridging People and Software through Process Technology*. Wiley, 2005.

[2] W. M. P. v. d. Aalst and K. M. v. Hee, *Workflow Management: Models, Methods, and Systems*. MIT press, Cambridge, MA, 2002.

[3] W. M. P. v. d. Aalst, "Verification of workflow nets," in *ICATPN 1997*, ser. Lecture Notes in Computer Science, P. Azéma and G. Balbo, Eds., vol. 1248. Springer, 1997, pp. 407–426.

[4] K. Barkaoui and L. Petrucci, "Structural Analysis of Workflow Nets with Shared Ressources," in *WFM 1998*, 1998, pp. 82–95.

[5] K. M. v. Hee, A. Serebrenik, N. Sidorova, and M. Voorhoeve, "Soundness of resource-constrained workflow nets," in *ICATPN 2005*, ser. Lecture Notes in Computer Science, G. Ciardo and P. Darondeau, Eds., vol. 3536. Springer, 2005, pp. 250–267.

[6] K. Barkaoui, R. Benayed, and Z. Sbai, "Workflow Soundness Verification Based on Structure Theory of Petri Nets," *International Journal of Computing & Information Sciences*, vol. 5, no. 1, pp. 51–62, 2007.

[7] K. R. Apt and D. Kozen, "Limits for automatic verification of finite-state concurrent systems," *Inf. Process. Lett.*, vol. 22, no. 6, pp. 307–309, 1986.

[8] F. L. Tiplea and C. Bocaneala, "Decidability results for soundness criteria of resource-constrained workflow nets," *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, vol. 42, no. 1, pp. 238–249, 2012.

[9] M. Hack, "Decidability questions for petri nets," Ph.D. dissertation, MIT, Cambridge, Mass, 1975.

[10] J. Esparza and M. Nielsen, "Decidability issues for petri nets - a survey," *Bulletin of the EATCS*, vol. 52, pp. 244–262, 1994.

[11] J. Desel and J. Esparza, *Free Choice Petri Nets*, ser. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, Cambridge, UK, 1995, vol. 40.

[12] W. M. P. v. d. Aalst, "The Application of Petri Nets to Workflow Management," *The Journal of Circuits, Systems and Computers*, vol. 8, no. 1, pp. 21–66, 1998.

[13] K. M. v. Hee, N. Sidorova, and M. Voorhoeve, "Soundness and separability of workflow nets in the stepwise refinement approach," in *ICATPN 2003*, ser. Lecture Notes in Computer Science, W. M. P. v. d. Aalst and E. Best, Eds., vol. 2679. Springer, 2003, pp. 337–356.

[14] E. Best, J. Esparza, H. Wimmel, and K. Wolf, "Separability in conflict-free petri nets," in *PSI 2006*, ser. Lecture Notes in Computer Science, I. Virbitskaite and A. Voronkov, Eds., vol. 4378. Springer, 2007, pp. 1–18.

[15] E. Best and P. Darondeau, "Separability in persistent petri nets," *Fundam. Inform.*, vol. 113, no. 3-4, pp. 179–203, 2011.

[16] G. Juhás, I. Kazlov, and A. Juhásová, "Instance deadlock: A mystery behind frozen programs," in *PETRI NETS 2010*, ser. Lecture Notes in Computer Science, J. Lilius and W. Penczek, Eds., vol. 6128. Springer, 2010, pp. 1–17.

[17] H. Foster, W. Emmerich, J. Kramer, J. Magee, D. S. Rosenblum, and S. Uchitel, "Model checking service compositions under resource constraints," in *ESEC/SIGSOFT FSE 2007*, I. Crnkovic and A. Bertolino, Eds. ACM, 2007, pp. 225–234.

[18] K. M. v. Hee, N. Sidorova, and M. Voorhoeve, "Resource-constrained workflow nets," *Fundam. Inform.*, vol. 71, no. 2-3, pp. 243–257, 2006.

[19] D. F. Escrig and C. Johnen, "Decidability of home space property," Université Paris-Sud, LRI report 503, 1989.

[20] K. M. v. Hee, N. Sidorova, and M. Voorhoeve, "Generalised soundness of workflow nets is decidable," in *ICATPN 2004*, ser. Lecture Notes in Computer Science, J. Cortadella and W. Reisig, Eds., vol. 3099. Springer, 2004, pp. 197–215.

[21] N. Bulling and B. Farwer, "Expressing properties of resource-bounded systems: The logics rtl* and rtl," in *CLIMA 2009*, ser. Lecture Notes in Computer Science, J. Dix, M. Fisher, and P. Novák, Eds., vol. 6214. Springer, 2010, pp. 22–45.

[22] N. Alechina, B. Logan, N. H. Nga, and A. Rakib, "Resource-bounded alternating-time temporal logic," in *AAMAS 2010*, W. v. d. Hoek, G. A. Kaminka, Y. Lespérance, M. Luck, and S. Sen, Eds. IFAAMAS, 2010, pp. 481–488.