Soundness and Separability of Workflow Nets in the Stepwise Refinement Approach

Kees van Hee, Natalia Sidorova, and Marc Voorhoeve

Eindhoven University of Technology
Department of Mathematics and Computer Science
P.O. Box 513, 5600 MB Eindhoven, The Netherlands
k.m.v.hee@tue.nl, n.sidorova@tue.nl, wsinmarc@win.tue.nl

Abstract. Workflow nets are recognized as a modelling paradigm for the business process modelling. We introduce and investigate several correctness notions for workflow nets, ranging from proper termination of cases to their mutual independence. We define refinement operators for nets and investigate preservation of correctness through these operators. This gives rise to a class of nets that are provably correct.

Keywords: Petri nets; workflow; modelling; verification; correctness; soundness; separability; serialisability.

1 Introduction

Petri nets are frequently used to model and analyse workflow processes in business process design (c.f. [1, 3]). The nets used in this area are appropriately called workflow nets (WF-nets). In software engineering, the same WF-nets can be used for modelling the life cycles of objects. A case (transaction, object) starts as a token in the initial place of the WF-net and after a series of steps this token evolves into a marking consisting possibly of several tokens. An important property is *proper completion*: from such a marking, it must be possible to reach the final marking of one token in the final place. This property is called *soundness*, c.f. [1, 4]. Soundness can be verified by more or less standard Petri net algorithms (e.g. coverability analysis).

In [4], it is argued that soundness alone is not compositional w.r.t. refinement; it is possible to refine a transition in a sound net with another sound net and obtain a non-sound result (see e.g. Figure 1 in Section 3). For this reason, soundness is considered in that work for free choice, safe and well-structured nets, and compositionality is proven for each of these classes.

In this paper we propose and investigate a generalization of the notion of soundness. We say that a workflow net is k-sound if any marking reached from k tokens in the initial place can reach the same k tokens in the final place. The original soundness becomes 1-soundness, and we propose to call workflow nets sound iff they are k-sound for each $k > 0$. A practical advantage of the new notion of soundness is introducing a possibility to avoid "earmarking" tokens to distinguish several cases processed in the net. Imagine processing n orders in the net. If the workflow net is 1-sound but not sound, every order has to be earmarked by adding a unique id-colour, thus guaranteeing a treatment of the order in isolation. If the net is sound, one can assure a proper completion of the task with k orders without introducing id's.

We show that the new notion of soundness is compositional w.r.t. refinement. Next, we prove several bisimilarity results, which allows to carry over temporal properties of nets to their refinements when the refinement is given by a sound (in the new sense) net. Unlike 1-soundness, no apparent verification algorithm for soundness exists, though we prove some classes of WF-nets to be sound.

Soundness is of course not the only correctness criterion. Analysis of a model can be done e.g. by proving temporal requirements specified in a temporal logic. It would be interesting to find a class of nets whose properties are the same for the WF-nets with removed earmarkings as for the original nets. The concept of serialisability in transaction processing [5] is based on the property that cases are independent of each other: the presence or absence of other cases does not influence the options for a specific case. This leads us to a similar concept of *serialisability* for WF-nets: the property that the set of traces of the WF-net with id-markings is equal to the set of traces of its abstraction. We show that state machines and cycle-free marked graphs are serialisable. On the negative side is the fact that serialisability is not a congruence w.r.t. refinement.

An attempt to soften the requirements results in a notion of *weak separability*: every marking reachable from the initial state with k tokens is representable as a sum of k markings each of which is reachable from a single initial token. Every serialisable net is clearly weakly separable. We show that weak separability together with 1-soundness imply soundness. Weak separability is a congruence w.r.t. place refinement, but not a congruence w.r.t. transition refinement. Looking for a compositional notion of separability, we come to a definition that is similar to serialisability, however, it does not require the trace equivalence between the net with id-tokens and its abstraction, but the equivalence of sets of Parikh vectors. One additional requirement turns this notion to the compositional one, that we call *split-separability*.

For business applications, weak separability is important because it formalizes the idea of independent cases: each marking is the sum of the markings of the individual cases and therefore all properties of the markings of a batch of cases are "cumulated properties" of the individual cases. The additional property of separability says that also the firings of a batch of cases is in fact the sum of the firings of the individual cases. If we associate to each firing the consumption of some resource, like money or energy, then separability implies that the consumption of the batch of cases equals the sum of the individual consumptions.

We prove state machines and cycle-free marked graphs to be sound, serialisable and split-separable. Combined with refinement, this fact gives rise to a class of WF-nets, that we call ST-nets, which are sound and split-separable by construction.

The rest of the paper is organized as follows. In Section 2, we sketch the basic definitions related to Petri nets and WF-nets. In Sections 3, we formulate the new notion of soundness and give weak bisimilarity results for sound refinements. In Section 4, we introduce and analyse the notions of soundness and separability. Afterwards, in Section 5 we define a class of separable by construction ST-nets and give a factorisation algorithm to invert refinement. We conclude in Section 6 with discussing the obtained results and directions for the future work.

2 Preliminaries

Let S be a set. A bag (multiset) m over S is a function $m : S \rightarrow \mathbb{N}$. We use $+$ and $-$ for the sum and the difference of two bags and $=, <, >, \leq, \geq$ for comparisons of bags, which are defined in a standard way, and overload the set notation, writing $\emptyset$ for the empty bag and $\in$ for the element inclusion. We list elements of bags between brackets, e.g. $m = [p^2, q]$ for a bag m with $m(p) = 2$, $m(q) = 1$, and $m(x) = 0$ for all $x \notin \{p, q\}$. The shorthand notation $k.m$ is used to denote the sum of k bags m.

For sequences of elements over a set T we use the following notation: The empty sequence is denoted with λ; a non-empty sequence can be given by listing its elements between angle brackets. The Parikh vector $\overrightarrow{\sigma} : T \longrightarrow \mathbb{N}$ of a sequence σ maps every element $t \in T$ to the number of occurrences of t in σ. $\overrightarrow{\sigma}(t)$ stands for the number of occurrences of t in σ. A concatenation of sequences σ_1, σ_2 is denoted with $\sigma_1\sigma_2$; $t\sigma$ and σt stand for the concatenation of t and sequence σ and vice versa. A projection of a sequence σ on elements of a set U (i.e. eliminating the elements from $T \setminus U$) is denoted as $\pi_U(\sigma)$. The shuffle $\sigma \| \gamma$ of two sequences is the set of sequences obtained by interleaving the elements of σ and γ; formally we have $\lambda \| \sigma = \sigma \| \lambda = \sigma$ and $a\sigma \| b\gamma = \{ax \mid x \in \sigma \| b\gamma\} \cup \{by \mid y \in a\sigma \| \gamma\}$.

Transition Systems A *transition system* is a tuple $E = \langle S, \mathit{Act}, T \rangle$ where S is a set of *states*, Act is a finite set of *action names* and $T \subseteq S \times \mathit{Act} \times S$ is a *transition relation*. A *process* is a pair $\langle E, s_0 \rangle$ where E is a transition system and $s_0 \in S$ an initial state.

We denote (s_1, a, s_2) from T as $s_1 \xrightarrow{a} s_2$, and we say that a leads from s_1 to s_2. For a sequence of transitions $\sigma = \langle t_1, \ldots, t_n \rangle$ we write $s_1 \xrightarrow{\sigma} s_2$ when $s_1 = s^0 \xrightarrow{t_1} s^1 \xrightarrow{t_2} \ldots \xrightarrow{t_n} s^n = s_2$, and $s_1 \xrightarrow{\sigma}$ when $s_1 \xrightarrow{\sigma} s_2$ for some s_2. In this case we say that σ is a trace of E. Finally, $s_1 \xrightarrow{*} s_2$ means that there exists a sequence of transitions $\sigma \in T^*$ such that $s_1 \xrightarrow{\sigma} s_2$. We use action label τ to denote silent actions and write $s_1 \Longrightarrow s_2$ when $s_1 = s_2$ or $s_1 \xrightarrow{\tau} \ldots \xrightarrow{\tau} s_2$. We write $s_1 \overset{a}{\Longrightarrow} s_2$ if $s_1 \Longrightarrow s_1' \xrightarrow{a} s_2' \Longrightarrow s_2$. To indicate that the step a is taken in the transition system E we write $s \xrightarrow{a}_E s'$, $s \overset{a}{\Longrightarrow}_E s'$ resp.

The *strong trace set* $\mathcal{ST}(E, s_0)$ of a process $\langle E, s_0 \rangle$ is defined as $\{\sigma \in \mathit{Act}^* \mid s_0 \xrightarrow{\sigma}\}$. Two processes are strongly trace equivalent iff their strong trace sets are equal. $\mathcal{ST}(E_1) = \mathcal{ST}(E_2)$. The *weak trace set* $\mathcal{T}(E, s_0)$ is defined as $\{\sigma \in (\mathit{Act} \setminus \{\tau\})^* \mid s_0 \overset{\sigma}{\Longrightarrow}\}$. Two processes are weakly trace equivalent iff their weak trace sets are equal.

Bisimulation Given two systems $N_1 = \langle S_1, Act, T_1 \rangle$ and $N_2 = \langle S_2, Act, T_2 \rangle$. A relation $R \subseteq S_1 \times S_2$ is a *simulation* iff for all $s_1 \in S_1, s_2 \in S_2$, $s_1 R s_2$ and $s_1 \xrightarrow{a} s_1'$ implies that there exists a transition $s_2 \xrightarrow{a} s_2'$ such that $s_1'\ R\ s_2'$. Relation R is a *bisimulation* [11] if R and R^{-1} are simulations.

Weak (bi)simulation is defined by copying the definitions for plain (bi)simulation and replacing $\xrightarrow{a}$ by $\xRightarrow{a}$ throughout. Two processes $\langle E, s \rangle$ and $\langle F, r \rangle$ are called *(weakly) bisimilar* iff there exists a (weak) bisimulation R such that $s\,R\,r$. We often add the adjective "strong" to non-weak simulation relations. Strong and weak bisimilarity are equivalence relations. [1]

Petri nets A *labelled Petri net* is a tuple $N = \langle S_N, T_N, F_N, l_N \rangle$, where:

- S_N and T_N are two disjoint non-empty finite sets of *places* and *transitions* respectively, the set $S_N \cup T_N$ are the *nodes* of N;
- F_N is a mapping $(S_N \times T_N) \cup (T_N \times S_N) \to \mathbb{N}$ which we call a *flow function*;
- $l_N : T_N \to Act$ labels each transition $t \in T_N$ with some action $l_N(t)$ from Act.

We assume that Act contains all transitions of all nets to be encountered. Unless stated otherwise, we assume that the labeling function maps a transition onto itself. If the identity function is not used, some transitions are labelled with the silent action τ.

We drop the N subscript whenever no ambiguity can arise and present nets with the usual graphical notation. A *path* of a net is a sequence $\langle x_1, \ldots, x_n \rangle$ of nodes such that $\forall\, i : 1 \leq i \leq n-1 : F(x_i, x_{i+1}) > 0$.

Markings are states (configurations) of a net. We consider a *marking* m of N as a bag over S and denote the set of all markings reachable in net N from marking m as $\mathcal{M}(N, m)$. The set of markings from which marking m can be reached is denoted as $\mathcal{S}(N, m)$.

Given a transition $t \in T$, the *preset* ${}^\bullet t$ and the *postset* $t^\bullet$ of t are the bags of places where every $p \in S$ occurs in ${}^\bullet t$ $F(p, t)$ times and in $t^\bullet$ $F(t, p)$ times. Analogously we write ${}^\bullet p, p^\bullet$ for pre- and postsets of places. To emphasize the fact that the preset/postset is considered within some net N, we write ${}^\bullet_N a, a^\bullet_N$. We overload this notation further allowing to apply preset and postset operations to a bag B of places/transitions, which is defined as the weighted sum of pre-/postsets of elements of B.

A transition $t \in T$ is *enabled* in marking m iff ${}^\bullet t \leq m$. An enabled transition t may fire, thus performing action $l(t)$. This results in a new marking m' defined by $m' \overset{\text{def}}{=} m - {}^\bullet t + t^\bullet$. For a firing sequence γ in a net N, we define ${}^\bullet_N \gamma$ and $\gamma^\bullet_N$ respectively as $\sum_{t \in \gamma} {}^\bullet_N t$ and $\sum_{t \in \gamma} t^\bullet_N$, which are the sums of all tokens consumed/produced during the firings of γ. So $m \xrightarrow{\gamma}_N (m + \gamma^\bullet_N - {}^\bullet_N \gamma)$.

[1] All systems proved to be weakly bisimilar in this paper are in fact branching bisimilar. This follows from Theorem 3.1 in [9]: a weak bisimulation where one of the related systems is τ-free is a branching bisimulation.

We interpret a Petri net N as a transition system/process where markings play the role of states, firings of the enabled transitions define the transition relation and the initial marking corresponds to the initial state. The notions of reachability, traces, simulation and bisimulation, etc. for Petri nets are inherited from the transition systems. When $m_N \, R \, m_M$ for some markings m_N, m_M and bisimulation R we say that (N, m_N) and (M, m_M) are bisimilar, written $(N, m_N) \sim (M, m_M)$.

Workflow Petri nets In this paper we primarily focus upon the *Workflow Petri nets (WF-nets)* [1]. As the name suggests, WF-nets are used to model the ordering of tasks in workflow processes. The initial and final nodes indicate respectively the initial and final states of cases flowing through the process.

Definition 1. *A Petri net N is a* Workflow net (WF-net) *iff:*

- *N has two special places (or transitions): i and f. Place (transition resp.) i is an initial place (transition): ${}^\bullet i = \emptyset$, and f is a final place (transition): $f^\bullet = \emptyset$.*
- *For any node $n \in (S \cup T)$ there exists a path from i to n and a path from n to f.*

We will call a WF-net sWF-net or tWF-net to indicate whether a WF-net has places or transitions as initial and final nodes. A tWF-net can be extended with an additional initial place and a terminal place up to an sWF-net.

3 Refinement and Soundness of Workflow Nets

When constructing models, the concept of refinement is very natural. A single task on a higher level can become a sequence of subtasks also involving choice and parallelism, i.e. it can be refined to a tWF-net. Similarly, being at some location (place of the net) resources (tokens) can undergo a number of operations, which can be reflected with a substitution of this place with an sWF-net. To build composed nets from WF-net components we will use two simple operations: Given two WF-nets L, M.

- *Place refinement* of a place $p \in S_L$ with sWF-net M yields a WF-net $N = L \otimes_p M$, built as follows: $p \in S_L$ is replaced in L by M; transitions from ${}^\bullet p$ become input transitions of the initial place of M and transitions from $p^\bullet$ become output transitions of the final place of M.
- *Transition refinement* of a transition $t \in T_L$ with tWF-net M yields a WF-net $N = L \otimes_t M$, built as follows: $t \in T_L$ is replaced by M; places from ${}^\bullet t$ become input places of the initial transition of M and places from $t^\bullet$ become output places of the final transition of M.

We consider transition and place refinements as basic techniques of our component-oriented design methodology. Note that the refinement of the initial

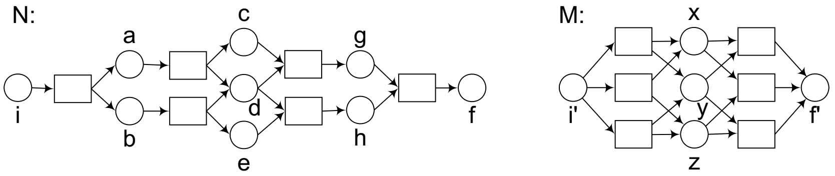

Fig. 1. Refining 1-sound nets

and final places are legitimate operations, and are in fact sequential compositions of nets.

The refinement operators satisfy the following trivial equations.

Lemma 2. *Let A, B, C be WF-nets, $a, c \in S_A \cup T_A$, $c \neq a$ and $b \in S_B \cup T_B$. If $A \otimes_a B$, $B \otimes_b C$ and $A \otimes_c C$ are defined, then $(A \otimes_a B) \otimes_b C$ and $A \otimes_a (B \otimes_b C)$ are defined and equal as well as $(A \otimes_a B) \otimes_c C$ and $(A \otimes_c C) \otimes_a B$ are.*

Soundness of WF-nets A natural requirement for WF-nets is that in any case the modelled process should be able to reach the end state, no matter what happens to it. This requirement has been called *soundness* by [1, 3]. That formulation of soundness does not combine with refinement, though. In Figure 1, WF-nets N, M are depicted that are sound according to the standard definition of soundness. However, the net $L = N \otimes_d M$ is not sound: $[i] \xrightarrow{*}_L [c, e, i'^2] \xrightarrow{*}_L [c, e, y^2, f'] \xrightarrow{*}_L [c, y^2, h]$. From this last state, no successor state can be reached: it is a deadlock containing nonterminal nodes. The reason is that net M terminates properly when started from $[i]$ but not from $[i^2]$.

This example shows that we need a stronger notion of soundness that would require a correct outcome of the WF-net work for initial markings with an arbitrary number of tokens in the initial place. For this reason, we generalize the soundness notion; the original soundness becomes 1-soundness according to the new definition.

Definition 3. *An sWF-net $N = \langle S, T, F, l \rangle$ with initial and final places i and f resp. is k*-sound *for $k \in \mathbb{N}$ iff $[f^k]$ is reachable from all markings m from $\mathcal{M}(N, [i^k])$.*
A tWF-net N with initial and final transitions t_i, t_f respectively is k-sound iff the sWF-net formed by adding to S_N places p_i, p_f with $^\bullet p_i = \emptyset, p_i^\bullet = [t_i], ^\bullet p_f = [t_f], p_f^\bullet = \emptyset$ is k-sound.
A WF-net is sound *iff it is k-sound for every natural k.*

Note that by the definition of soundness, $\mathcal{M}(N, [i^k]) \subseteq \mathcal{S}(N, [f^k])$ for any k-sound net N.

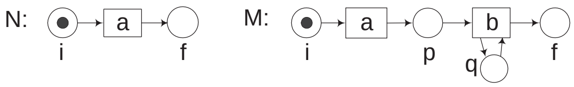

Fig. 2. Bisimilar Petri nets which are not WF-bisimilar nets

Bisimulation of WF-nets The notion of bisimulation for WF-nets must include the requirement of proper initialisation/termination. Consider e.g. nets N and M given in Figure 2. They are bisimilar Petri nets, however, N is sound while M has a deadlock and is not sound. We want to be able to transfer the conclusion about the soundness of a WF-net to all WF-bisimilar nets, therefore we do not consider nets N and M as WF-bisimilar.

Definition 4. *Given relation $R \subseteq (\mathbb{N} \to S_N) \times (\mathbb{N} \to S_N)$ on markings of sWF-nets N and M. R is a* WF-simulation *iff R is a simulation and*

$$(\forall\, k, x : [i_N^k]\, R\, x : x = [i_M^k]) \textit{ and } (\forall\, k, x : [f_N^k]\, R\, x : x = [f_M^k]).$$

R is a weak WF-simulation *iff R is a weak simulation and*

$$(\forall\, k, x : [i_N^k]\, R\, x : [i_M^k] \Longrightarrow x) \textit{ and } (\forall\, k, x : [f_N^k]\, R\, x : x \Longrightarrow [f_M^k]).$$

A strong/weak WF-simulation between tWF-nets N, M is a relation that can be extended (by adding pairs of markings) to become a strong/weak WF-simulation between the sWF-nets $\overline{N}, \overline{M}$ obtained by adding initial and terminal places to N, M respectively.

R is a strong/weak WF-bisimulation *iff R and R^{-1} are strong/weak WF-simulations. We will say that the WF-nets N and M are* strongly/weakly WF-bisimilar *iff there exists a strong/weak WF-bisimulation R between N and M such that $\forall\, k :: [i_N^k]\, R\, [i_M^k] \wedge [f_N^k]\, R\, [f_M^k]$.*

It is easy to show that WF-bisimilarity is an equivalence relation. Moreover, the following property holds:

Lemma 5. *Let N, M be WF-bisimilar WF-nets and N is sound. Then M is sound as well.*

Soundness and bisimulation of refinements We prove that refinement with sound nets yields weakly WF-bisimilar nets.

Theorem 6. *Let M be a sound sWF-net with all transitions τ-labelled, L be a net with a place $p \in S_L$ and $N = L \otimes_p M$. Then L and N are weakly WF-bisimilar.*

Proof. Let $R = \{(m + [p^\ell], m + x) \mid m \in (S_L \setminus \{p\}) \to \mathbb{N}\} \wedge x \in \mathcal{S}(M, [f_M^\ell])$. We prove that this relation is a weak WF-bisimulation. Note that $[i_L^k]\, R\, [i_N^k]$ and

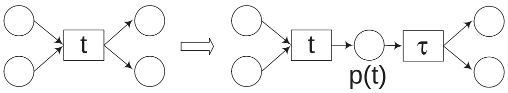

Fig. 3. Split refinement of transition t

$[f_L^k]\, R\, [f_N^k]$. Also, the "WF" requirement is satisfied, so it is sufficient to prove that (1) R is a weak simulation and (2) R^{-1} is a weak simulation.

(1): Suppose $(m + [p^\ell])\, R\, (m + x)$ and $(m + [p^\ell]) \xrightarrow{a}_L (m' + [p^r])$ with $m' \in (S_L \setminus \{p\}) \to \mathbb{N}$. If the transition a does not affect p-tokens, we have $r = \ell$ and $(m + x) \xrightarrow{a}_N (m' + x)$, with $(m' + [p^r])\, R\, (m' + x)$. Now suppose the transition consumes s and produces t p-tokens, so $s \leq \ell$ and $r = \ell - s + t$. Since $x \in \mathcal{S}(M, [i_M^\ell])$, $x \xrightarrow{*}_M [f_M^\ell]$, and thus $x \xrightarrow{*}_N [f_M^\ell]$. Since the transitions of M are τ-labelled, we have $(m + x) \Longrightarrow_N (m + [f_M^\ell]) \xrightarrow{a}_N (m' + [f_M^{\ell - s}] + [i_M^t])$, so $(m + x) \overset{a}{\Longrightarrow}_N (m' + y)$ with $y \in \mathcal{S}(M, [f_M^r])$ (due to the soundness of M), so $(m' + [p^r])\, R\, (m' + y)$.

(2): Suppose $(m + [p^\ell])\, R\, (m + x)$ and $(m + x) \xrightarrow{a}_N (m' + y)$. In case that $a \in T_M$, we have $m' = m$, $(m + x) \Longrightarrow_N (m + y)$ and $y \in \mathcal{S}(M, [f_M^\ell])$ due to the soundness of M, and so $(m + [p^\ell])\, R\, (m' + y)$. If $a \in T_N$, we set $F_L(a, p) = r$, $F_L(p, a) = s$. Thus by the construction of N, $x \geq [f_L^s]$ and $y = x - [f_L^s] + [i_L^r]$. Also, $(m + [p^\ell]) \xrightarrow{a}_L (m' + [p^{\ell - s + r}])$. Since M is sound and $x \in \mathcal{S}(M, [f_M^\ell])$, we have $y \in \mathcal{S}(M, [f_M^{\ell - s + r}])$, so $(m' + [p^{\ell - s + r}])\, R\, (m' + y)$. □

We prove another bisimulation result for transition refinement with sound nets. First, we introduce a simple transition refinement —*split refinement* (see Fig. 3): transition t is replaced with the tWF-net Σ_t with places $\{p_t\}$ and transitions $\{i_t, f_t\}$ such that ${}^\bullet i_t = f_t^\bullet = \emptyset$, $i_t^\bullet = {}^\bullet f_t = [p_t]$. In this section, we suppose that i_t has label t and f_t has label τ.

Lemma 7. *Let N be a WF-net with $t \in T_N$ and $M = N \otimes_t \Sigma_t$. Then*

$$R \overset{def}{=} \{(m, m + [p_t^k] - k.t_N^\bullet) \mid m \in (S_N \to \mathbb{N}) \wedge m \geq k.t_N^\bullet\}$$

is a weak WF-bisimulation for N and M.

Proof. Consider some markings m, μ such that $m\, R\, \mu$, say $\mu = m + [p_t^k] - k.t_N^\bullet$. Suppose $m \xrightarrow{u}_N m'$. Then $\mu \Longrightarrow_M m \xrightarrow{u}_M m'$, so $m \overset{u}{\Longrightarrow}_M m'$. Clearly $m'\, R\, m'$. Now suppose $\mu \xrightarrow{u}_M \mu'$. If the transition that fired is i_t, then $\mu' = \mu - {}^\bullet_M t + [p_t]$, so $m \geq {}^\bullet_M t$, so $m \xrightarrow{u}_N m'$ with $m' = m - {}^\bullet_M t + t_M^\bullet$ and $m'\, R\, \mu'$. If that transition is f_t, then $u = \tau$ and $\mu' = \mu - [p_t] + t_M^\bullet$, so taking $m' = m$, we have $m \Longrightarrow_N m'$ and $m'\, R\, \mu'$. In all other cases, the transition that fired was u and ${}^\bullet_N u = {}^\bullet_M u$ and $u_N^\bullet = u_M^\bullet$. Since $\mu \geq {}^\bullet_N u$ and $\mu' = \mu - {}^\bullet_N u + u_N^\bullet$, we can take $m' = m - {}^\bullet_M u + u_M^\bullet$ and have $m \overset{u}{\Longrightarrow}_N m'$. This covers all cases. Clearly, R satisfies the additional requirements of a WF-bisimulation. □

Theorem 8. *Let L be a net with $t \in T_L$ and M a sound tWF-net with all transitions except i_M labelled with τ. Then L and $N = L \otimes_t M$ are weakly WF-bisimilar.*

Proof. Let $\overline{M}$ be the extension of M with the initial and final places and all transitions relabelled with τ. By Lemma 2, $(L \otimes_t \Sigma_t) \otimes_{p_t} \overline{M} = ((L \otimes_t M) \otimes_{i_M} \Sigma_{i_M}) \otimes_{f_M} \Sigma_{f_M}$. Thus, by Lemma 7, $L \otimes_t M$ is weakly WF-bisimilar to $(L \otimes_t \Sigma_t) \otimes_{p_t} \overline{M}$, which by Theorem 6 is weakly WF-bisimilar to $L \otimes_t \Sigma_t$ and by Lemma 7 is weakly WF-bisimilar to L. □

The theorems on weakly bisimilar refinements can be applied to yield soundness preservation.

Theorem 9. *Let L, M be sound WF-nets with $n \in S_L \cup T_L$ such that $N = L \otimes_n M$ is defined. Then N is sound.*

Proof. If $n \in S_L$, we use Theorem 6 after relabelling transitions of M with τ. L and N are weakly WF-bisimilar. Note that the relabelling does not influence the soundness. Suppose $[i_N^k] \xrightarrow{*} m$ within N, then there exists a state m' of L with $m\, R\, m'$ such that $[i_L^k] \xrightarrow{*} m'$. Since $m' \xrightarrow{*} [f_L^k]$ within L, there exists a state μ of N such that $m \xrightarrow{*} \mu$ and $\mu\ R\, [f_L^k]$. By the definition of WF-bisimulation we have $\mu = [f_N^k]$. If $n \in T_L$, we use Theorem 8 similarly. □

4 Separability

With introducing the new notion of soundness, we extended the applicability of WF-nets for the compositional design process. However, soundness is not the only criterium for the correctness of the behaviour. In general, one looks for a Petri net model that meets its specification given e.g. by a temporal logic formula; note that the behaviour should satisfy the specification whatever number of tokens is chosen to be placed into the initial place of the WF-net. The challenge is to reduce the number of cases to be considered, when possible.

In this section we introduce a notion of *separability*, a behavioural property stating that the behaviour of a WF-net with k initial tokens can be seen in some sense as a combination of the behaviours of k copies of the net each of which has one initial token.

4.1 Workflow nets with id-tokens and serialisability

In this subsection, we extend the semantics of labelled Petri nets by introducing *id-tokens*: we consider a token as a pair (p, a), where p is a place and $a \in Id$ is an identifier (a primitive sort of a colour). We assume Id to be a countable set. A transition $t \in T$ is *enabled* in an id-marking m iff there exists $a \in Id$ such that m contains tokens ${}^\bullet t$ with identifier a. A firing of t results in consuming these tokens and producing tokens with identifier a to $t^\bullet$. To make it clear whether the firing happens in a classical Petri net or in a net with id-tokens, we write $\xmapsto{t}$

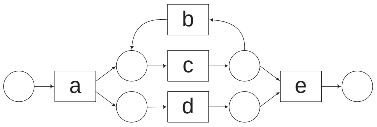

Fig. 4. Serialisable net with id-tokens that is not bisimilar to its abstraction

for firings in nets with id-tokens. Later on, we will use the extended semantics when working with id-tokens, and the standard semantics for classical tokens.

Though being a very simple sort of coloured nets, WF-nets with id-tokens are often expressive enough to reflect the essence of a modelled business process taking care of separating different cases which are processed in the net concurrently.

A net with id-tokens can be abstracted into a classical labelled Petri net in a natural way by removing the token id's. We denote the abstraction function as α and a marking obtained from a coloured marking m as $\alpha(m)$ resp. It is easy to see that the obtained net is a sound abstraction of the original net, i.e. it shows more behaviour (see [6] for the definition of sound abstraction):

Lemma 10. *Let N be a Petri net and m its id-marking. Then there exists a simulation relation between (N, m) and $(N, \alpha(m))$.*

Proof. It is trivial to show that $R = \{(m, \alpha(m) \mid m \in \mathcal{M}(N)\}$ is a simulation relation. □

There is no simulation between (N, m) and $(N, \alpha(m))$ in general (consider e.g. the trace *adcbace* for the WF-net in Figure 4).

Still, it would be interesting to see whether there exists a class of nets whose behaviour is *trace equivalent* to the behaviour of nets with id-tokens. For this purpose, we introduce a notion of *serialisability.*

Definition 11. *An sWF-net N is* serialisable *iff for any $k \in \mathbb{N}$, any firing sequence σ such that $[i^k] \xrightarrow{\sigma}$ there exist firing sequences $\sigma_1, \ldots, \sigma_k$ such that $[i] \xrightarrow{\sigma_1}, \ldots, [i] \xrightarrow{\sigma_k}$ and $\sigma \in (\sigma_1 \| \ldots \| \sigma_k)$.*

Theorem 12. *Let N be an sWF-net. Then N is serialisable iff for any id-marking M such that $\alpha(M) = [i^k]$ for some $k \geq 0$, we have $\{\sigma \mid [i^k] \xrightarrow{\sigma}_N\} = \{\sigma \mid M \overset{\sigma}{\longmapsto}_N\}$.*

Proof. ($\Rightarrow$): Let N be a serialisable net, $M = \sum_j [(i, c_j)]$ and k be given as specified. By Lemma 10 every trace of (N, M) is a trace of $(N, [i^k])$. We only have to prove that every trace of $(N, [i^k])$ is a trace of (N, M). Let σ be a trace of $(N, [i^k])$. Due to the serialisability of N there exist $\sigma_1, \ldots, \sigma_k$ such that $[i] \xrightarrow{\sigma_1}, \ldots, [i] \xrightarrow{\sigma_k}$ and $\sigma \in (\sigma_1 \| \ldots \| \sigma_k)$. Then we have $[(i, c_j)] \overset{\sigma_j}{\longmapsto}$ (since all the tokens produced and consumed in the firings of σ_j have colour c_j). Hence, $\sum_j [(i, c_j)] \overset{\sigma}{\longmapsto}$ and σ is a trace of (N, M).

($\Leftarrow$): Now assume we have the described property for N and we have to prove that N is serialisable. Let $[i^k] \xrightarrow{\sigma}$ for some k. Consider a marking $M = \sum_j [(i, c_j)]$ with $j \neq l \Rightarrow c_j \neq c_l$ for all $j, l \in \{1, \ldots, k\}$. Since σ is also a trace of (N, M) and all tokens of M have different colours, we can split σ according to the colours of firings into $\sigma_1, \ldots, \sigma_k$ such that $\sigma \in (\sigma_1 \| \ldots \| \sigma_k)$. We have $[(i, c_j)] \xmapsto{\sigma}_j$ for all $j \in \{1, \ldots, k\}$. Hence, $[i] \xrightarrow{\sigma}_j$ for all $j \in \{1, \ldots, k\}$. So N is serialisable. □

Trace equivalence between the nets with id-tokens and their abstractions for serialisable nets allows to perform the verification of trace properties, e.g. LTL-properties, on the abstractions of the nets, thus simplifying the verification task. The same holds for some problems of the performance analysis. If one associate time or price to every transition of the net independent of token's id's, then the analysis results obtained with an abstracted net hold for the original net as well.

4.2 Serialisable subclasses of WF-nets

In this subsection we consider two subclasses of WF-nets which we prove to be serialisable.

Definition 13. *Let $N = \langle S, T, F \rangle$ be a Petri net. N is a* state machine *(SM) iff* $\forall\, t \in T :| {}^\bullet t \mid \leq 1 \wedge \mid t^\bullet \mid \leq 1$.

Definition 14. *Let $N = \langle S, T, F \rangle$ be a Petri net. N is a* marked graph *(MG) iff* $\forall\, p \in S :| {}^\bullet p \mid \leq 1 \wedge \mid p^\bullet \mid \leq 1$.

Marked graphs are dual to state machines in the graph-theoretic sense and from the modelling point of view. State machines can represent conflicts by a place with several output transitions, but they can not represent concurrency and synchronization. Marked graphs, on the other hand, can represent concurrency and synchronization, but cannot model conflicts or data-dependent decisions.

We will refer to the marked graph tWF-nets and state machine sWF-nets as MGWF-nets and SMWF-nets respectively.

Theorem 15. *All SMWF-nets are sound and serialisable.*

Proof. Let N be an SMWF-net and $[i^k] \xrightarrow{\sigma} m$. We shall prove by induction on the length of σ that σ can be serialized. The case $\sigma = \epsilon$ is trivial, so let the statement hold for σ', $[i^k] \xrightarrow{\sigma'} m'$ and we prove the statement for $\sigma = \sigma' t$. By the induction hypothesis, σ' can be serialized into $\sigma'_1, \ldots, \sigma'_k$ such that $[i] \xrightarrow{\sigma'_j} m_j$. Since ${}^\bullet t \leq m'$ and $\mid {}^\bullet t \mid \leq 1$, there exists an m_l such that $m_l \geq {}^\bullet t$. Thus $[i] \xrightarrow{\sigma_l t}$ and σ can be serialized. So N is serialisable.

In an SMWF-net, $\mid {}^\bullet t \mid = \mid t^\bullet \mid = 1$ for any transition t, meaning that the number of tokens in the marking cannot change with any firing. So only markings of the form $\sum_{1 \leq j \leq k} [p_j]$ are reachable from $[i^k]$. By the definition of WF-nets, every place lies on a path from i to f. Since N is an SMWF-net, the existence of a path from place p_j to place f is equivalent to $[p_j] \xrightarrow{*}_N [f]$. Hence, $\sum_{1 \leq j \leq k} [p_j] \xrightarrow{*} [f^k]$ and N is sound. □

Cycle-free MGWF-nets are sound and serialisable. The proof depends upon the following lemma.

Lemma 16. *Let N be a cycle-free MGWF-net with transitions $t, u \in T_N$ and a place $a \in S_N$ such that there exist paths in N from t to a and from a to u and $\emptyset \xrightarrow{\sigma}_N m$ for some σ, m. Then we have $m(a) \leq \overrightarrow{\sigma}(t) - \overrightarrow{\sigma}(u)$.*

Proof. Note that N is a tWF-net, so traces σ with $\emptyset \xrightarrow{\sigma}_N$ must start with a firing of the initial transition i. Since N is cycle-free, the existence of a path between nodes implies that these nodes are different. We use induction on the length of the path from t to u. For the path of length 2, i.e. $a \in t^\bullet$ and $a \in {}^\bullet u$, the proof is immediate. If the path is longer, we can e.g. find $b \in S_N, v \in T_N$ such that there exists a path $tbv \ldots a \ldots u$. By the induction hypothesis, $m(a) \leq \overrightarrow{\sigma}(v) - \overrightarrow{\sigma}(u)$ and $m(b) \leq \overrightarrow{\sigma}(t) - \overrightarrow{\sigma}(v)$. Hence, $m(a) + m(b) \leq \overrightarrow{\sigma}(t) - \overrightarrow{\sigma}(u)$, so $m(a) \leq \overrightarrow{\sigma}(t) - \overrightarrow{\sigma}(u)$ □

As a corollary, if σ contains one firing of the initial transition i and a firing of a transition t, the marking m with $\emptyset \xrightarrow{\sigma} m$ satisfies $m(p) = 0$ for any place p on a path between i and t. We can now prove our theorem.

Theorem 17. *All cycle-free MGWF-nets are sound and serialisable.*

Proof. Let $\leq$ be the following partial order on sequences: $\sigma \geq \rho$ iff $\forall\, t :: \overrightarrow{\sigma}(t) \geq \overrightarrow{\rho}(t)$. We use induction like in the proof of Theorem 15, strengthening the induction hypothesis: σ can be serialized into $\sigma_1, \ldots, \sigma_k$ in such a way that $\sigma_1 \geq \ldots \geq \sigma_k$. So let N be the extension of an MGWF-net with the initial an terminal places and suppose $[i^k] \xrightarrow{\sigma t}$. By the induction hypothesis on σ we have a decreasing serialization $\sigma_1, \ldots, \sigma_k$ with $[i] \xrightarrow{\sigma_j}$ and $\sigma \in \sigma_1 \| \ldots \| \sigma_k$. Let m be a marking such that $[i^k] \xrightarrow{\sigma} m$ and m_j such that $[i] \xrightarrow{\sigma_j} m_j$ for all $j \in \{1, \ldots, k\}$. From Lemma 16, we know that σt has at most k occurrences of t, so σ has at most $k-1$ occurrences of t. Due to the ordering of σ_j's, we can conclude that σ_k does not contain t. Let $n \leq k$ be the smallest index such that σ_n does not contain t. We have $m_j \cap {}^\bullet t = \emptyset$ for $j < n$ by the corollary of Lemma 16. Moreover, for $j \geq n$ we have $m_j \cap {}^\bullet t \leq m_n \cap {}^\bullet t$, since there exist traces γ_j not containing t such that $\sigma_j \gamma_j = \sigma_n$. Since N does not contain multiple edges, we deduce from ${}^\bullet t \leq m = \sum_j m_j$ that ${}^\bullet t \leq m_n$. So we can serialize σt into traces $\sigma_1, \ldots, \sigma_{n-1}, \sigma_n t, \sigma_{n+1}, \ldots, \sigma_k$. Since σ_{n-1} contained t, we have $\sigma_{n-1} \geq \sigma_n t$, completing the induction step. Again, 1-soundness follows from the WF-net property. □

Serialisability is not compositional As it normally happens with the notions based on the traces of the systems, the serialisability is not compositional. Figure 5 shows a net obtained as a place refinement of a marked graph with a state machine, which are both serialisable as we know from the theorems proven above. The trace *aecabf* of the refined net cannot be serialized.

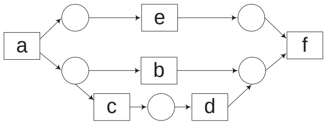

Fig. 5. Not serialisable net

4.3 Weak separability

Our next approach is to look at the markings of the net only:

Definition 18. *An sWF-net N is* weakly separable *iff for any $k \in \mathbb{N}$ and any marking m, $[i^k] \xrightarrow{*} m$ implies that there exist markings $m_1, \ldots, m_k$ such that $m = m_1 + \ldots + m_k$ and $[i] \xrightarrow{*} m_j$ for $j = 1, \ldots, k$.*
We say that a tWF-net N is weakly separable iff the sWF-net obtained by adding a place with the outgoing arc to the initial transition of N and a place with ingoing arc from the final transition of N is weakly separable.

Property 19. Serialisability implies weak separability.

Proof. If $[i^k] \xrightarrow{*} m$, there exists a σ such that $[i^k] \xrightarrow{\sigma} m$, so there exist $\sigma_1, \ldots, \sigma_k, m_1, \ldots, m_k$ such that $[i] \xrightarrow{\sigma_1} m_1, \ldots [i] \xrightarrow{\sigma_k} m_k$ and $\sigma \in (\sigma_1 \| \ldots, \|\sigma_k)$. Clearly, $m = m_1 + \ldots + m_k$. □

Requirements that weak separability puts on a net are essentially weaker than the ones of serialisability, which also means that we loose some options for analysis on the class of weakly separable nets in comparison to the serialisable nets. However, weak separability is sufficient to reduce the problem of soundness to 1-soundness:

Theorem 20. *Let N be a weakly separable and 1-sound net. Then N is sound.*

Proof. Consider a marking m reachable from $[i^k]$ where k is an arbitrary positive natural number. Since N is weakly separable, there exist $m_1, \ldots, m_k$ such that $m = m_1 + \ldots + m_k$ and $[i] \xrightarrow{*} m_1, \ldots, [i] \xrightarrow{*} m_k$. Since N is 1-sound, $m_1 \xrightarrow{*} [f], \ldots, m_k \xrightarrow{*} [f]$, which means that $m \xrightarrow{*} [f^k]$. So N is sound. □

A legitimate question would be whether weak separability implies soundness even without additional requirements. The answer to this question is negative: Figure 6 gives a weakly separable net which is not sound, and moreover not 1-sound.

Corollary 21. *The class of all weakly separable nets* is not a subclass *of all sound nets.*

The reverse is also not true: Figure 7 shows a sound free-choice[2] net (see [8]) which is not separable.

[2] N is a *free-choice Petri net* iff $\forall\, t_1, t_2 \in T$, ${}^\bullet t_1 \cap {}^\bullet t_2 \neq \emptyset$ implies ${}^\bullet t_1 = {}^\bullet t_2$.

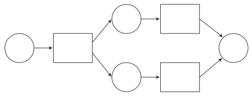

Fig. 6. A weakly separable net that is not 1-sound

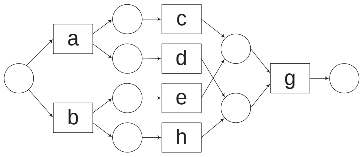

Fig. 7. A sound net that is not weakly separable

Corollary 22. *The class of all sound free-choice nets* is not a subclass *of all weakly separable nets.*

Thus, the notion of separability is in some sense orthogonal to the notion of soundness.

Like soundness and unlike serialisability, weak separability is a congruence with respect to the place refinement:

Theorem 23. *Let L, M be weakly separable WF-nets, moreover M is a sound sWF-net and $p \in P_L$. Then the net $N = L \otimes_p M$ is weakly separable.*

Proof. We may assume L to be an sWF-net since a tWF-net could be transformed to the sWF-net just by adding initial and final places. Let i, f, i_M, f_M be respectively the initial and final places of L and M. We shall prove that N is weakly separable.

Let $[i^k] \xrightarrow{*}_N m$. Then there is a trace σ such that $[i^k] \xrightarrow{\sigma}_N m$. As the nodes of L and M are disjoint, m can be represented as $m_L + m_M$ for some m_L, m_M where m_L is a marking over $P_L \setminus \{p\}$ and m_M is a marking over P_M. Similarly, trace σ can be projected into two traces σ_L, σ_M such that $\sigma_L \in T_L^*$, $\sigma_M \in T_M^*$. Note that $\sigma \in \sigma_L \| \sigma_M$.

Since $(N, [i^k])$ and $(L, [i^k])$ are weakly bisimilar (Theorem 6), σ_L is a trace of $(L, [i^k])$: $[i^k] \xrightarrow{\sigma_L}_L \overline{m}_L$. Due to the weak separability of L, $\overline{m}_L$ can be split into a sum $\overline{m}_{L,1} + \ldots + \overline{m}_{L,k}$, such that $[i] \xrightarrow{*}_L \overline{m}_{L,1}, \ldots, [i] \xrightarrow{*}_L \overline{m}_{L,k}$. Due to the soundness of M, we can prove by induction on the length of σ that $\overline{m}_L = m_L + [p^n]$ and $[i_M^n] \xrightarrow{*} m_M$ for some n. Due to the weak separability of M, m_M can be split into a sum $m_{M,1} + \ldots + m_{M,n}$, such that $[i] \xrightarrow{*}_M m_{M,1}, \ldots, [i] \xrightarrow{*}_M m_{M,n}$. Now we choose an arbitrary bijective function that maps every occurrence of p in each of $\overline{m}_{L,i}$ to a $m_{M,j}$, replace every p by $m_{M,j}$ according to the chosen mapping and thereby get the splitting of m we are looking for. □

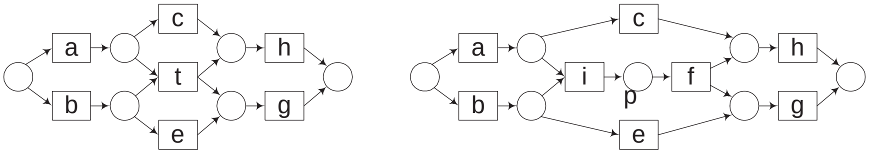

Fig. 8. Not weakly separable transition refinement of a weakly separable net

Applying transition refinement in the same way does not necessarily result in a weakly separable net. Figure 8 gives a weakly separable net and a refinement of this net where transition t is substituted with a sound weakly separable tWF-net. The resulting net is not weakly separable: marking $[p]$ is reachable from the initial marking $[i^2]$, however, it cannot be split into a sum of two markings reachable from $[i]$. Note that the net is nevertheless *sound*!

4.4 Separability

Finally, we shall try and introduce a notion stronger than weak separability but not as restrictive as serialisability. Moreover, we shall look for a subclass of nets where this notion is compositional w.r.t. refinements.

Definition 24. *An sWF-net N is* separable *iff for any $k \in \mathbb{N}$, any firing sequence σ such that $[i^k] \xrightarrow{\sigma}$, there exist firing sequences $\sigma_1, \ldots, \sigma_k$ such that $[i] \xrightarrow{\sigma_1}, \ldots, [i] \xrightarrow{\sigma_k}$ and $\overrightarrow{\sigma} = \overrightarrow{\sigma_1} + \ldots + \overrightarrow{\sigma_k}$.*

The following properties follow immediately from the corresponding definitions:

Property 25. (1) Serialisability implies separability. (2) Separability implies weak separability.

Note that the class of serialisable nets is strictly included in the class of separable nets: a not serialisable WF-net from Figure 5 *is separable*: e.g. the problematic trace $aecabf$ can now be separated into $aebf$ and ac.

For business applications of WF-nets, separability can be used to provide cost-effective management by simplifying the cost analysis. If costs are associated to every transition firing, the total cost of processing of k orders given by a trace in a WF-net is equal to the sum of costs of processing of k individual orders, each given by a trace with 1 initial token.

Unlike serialisability, separability turns out to be a congruence w.r.t. the place refinement operation:

Theorem 26. *Let L, M be separable WF-nets. If $p \in P_L$ and M is a sound sWF-net then $L \otimes_p M$ is separable.*

Proof. We may assume that L is an sWF-net; if not, we extend it. Let $N = L \otimes_p M$ and write $i_L = i_N$ as i. Assume $p \neq i$. Suppose $[i^k] \xrightarrow{\sigma}_N m$ for some m. We shall construct $\sigma_1, \ldots, \sigma_k$ such that $\vec{\sigma} = \sum_{1 \leq j \leq k} \vec{\sigma}_j$, where $[i] \xrightarrow{\sigma_j}_N$ for all j.

Let $\ell = \sigma^\bullet_N(p)$, $n = {}^\bullet_N\sigma(p)$. We define γ, ρ as the projections of σ on T_L, T_M respectively, so $\vec{\sigma} = \vec{\gamma} + \vec{\rho}$. Due to the existence of a weak WF-bisimulation between N and L, we have $[i^k] \xrightarrow{\gamma}_L$. The separability of L implies then the existence of $\gamma_1, \ldots, \gamma_k$ such that $\vec{\gamma} = \sum_j \vec{\gamma}_j$ with $[i] \xrightarrow{\gamma_j}_L$ for all j. Let $\ell_j = \gamma_j{}^\bullet_L(p)$, $n_j = {}^\bullet_L\gamma_j(p)$ for all j. Since $\gamma^\bullet_N(i_M) = \gamma^\bullet_L(p) = \ell$ and likewise ${}^\bullet_N\gamma(f_M) = n$, we have $\sum_{1 \leq j \leq k} \ell_j = \ell$ and $\sum_{1 \leq j \leq k} n_j = n$. We have $[i_M^\ell] \xrightarrow{\rho}_M m'$ with $m'(f_M) = n$. Due to the separability of M, we can find $\rho_1, \ldots, \rho_\ell$ such that $[i_M] \xrightarrow{\rho_j}_M m'_j$ with n of the traces ρ_j complete (i.e. $[i_M] \xrightarrow{\rho_j} [f_M]$ and $\vec{\rho} = \sum_{1 \leq j \leq \ell} \vec{\rho}_j$). Since $\sum_j \ell_j = \ell, \sum_j n_j = n$, we can partition the ρ_j's into disjoint sets $R_1, \ldots, R_k$ with respectively $\ell_1, \ldots, \ell_k$ elements such that m_j of the traces in R_j are complete for each j. We construct σ_j with $[i] \xrightarrow{\sigma_j}_N$ by adding transitions t from γ_j one by one preceded by ${}^\bullet_L t(p)$ completed traces from R_j. Since ${}^\bullet_L\gamma_j(p) = n_j$, all the completed traces in R_j are used in this process. We add the incomplete traces in any sequential order at the end. This we do for $1 \leq j \leq k$. These σ_j satisfy the requirement. If $p = i$ we can copy the above proof, setting $\ell = k$ and $\ell_j = 1$ for all j. □

Transition refinement is still a problem: Figure 9 shows a refinement of a separable net which yields an inseparable net: the trace *abicg* cannot be separated. Therefore, we constrict the class of separable nets in the following way:

Definition 27. *An sWF-net N is* split-separable *iff $\mathcal{S}(N) = (\ldots(N \otimes_{t1} \Sigma_{t1}) \otimes_{t2} \ldots) \otimes_{tn} \Sigma_{tn}$, $T_N = \{t_1, \ldots, t_n\}$ (the net obtained by applying the split-refinement to every transition of N), is separable.*

Due to Lemma 2, the order of split-refinements in the above definition is not important.

Property 28. Split-separability implies separability.

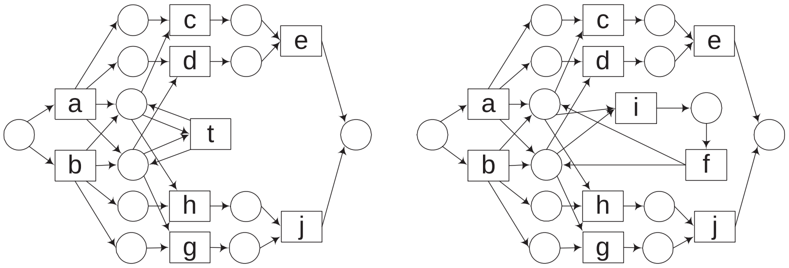

Fig. 9. Not separable transition refinement of a separable net

Proof. Let L be a split-separable net and $N = \mathcal{S}(L)$ its split-refinement, so N is separable. We shall prove that L is separable. We label i_t with t and f_t with τ for every transition $t \in T_L$. Then N and L are weakly bisimilar. Now let σ be a trace of L, $[i^k] \xrightarrow{\sigma}_L$ and σ' some corresponding trace of N, $[i^k] \xrightarrow{\sigma'}_N$. σ' can be separated into $\sigma'_1, \ldots, \sigma'_k$, $[i] \xrightarrow{\sigma'_1}_N, \ldots, [i] \xrightarrow{\sigma'_k}_N$ and $\overrightarrow{\sigma'} = \overrightarrow{\sigma'_1} + \ldots + \overrightarrow{\sigma'_k}$. Due to bisimilarity, $\sigma_1, \ldots, \sigma_k$ obtained from $\sigma'_1, \ldots, \sigma'_k$ by replacing every i_t with t and removing all f_t's are the traces of L and $\overrightarrow{\sigma} = \overrightarrow{\sigma_1} + \ldots + \overrightarrow{\sigma_k}$. So L is separable. □

Lemma 29. *Let L be a split-separable net and $N = L \otimes_t \Sigma_t$ for some $t \in T_L$. Then N is split-separable.*

Lemma 30. *Any split refinement of a split-separable net is split-separable.*

Proof. Notice that the net $\mathcal{S}(\mathcal{S}(N))$ can be obtained from $\mathcal{S}(N)$ by place refinement. Since $\mathcal{S}(N)$ is separable, $\mathcal{S}(\mathcal{S}(N))$ is also separable (Theorem 26). □

Theorem 31. *Let L, M be split-separable WF-nets. (1) If $p \in P_L$ and M is a sound sWF-net then $L \otimes_p M$ is split-separable. (2) If $t \in T_L$ and M is a sound tWF-net then $L \otimes_t M$ is split-separable.*

Proof. (1) Let $N = L \otimes_p M$. We have to prove that N is separable, i.e. $N' = \mathcal{S}(N)$ is separable. Since L and M are split-separable, $L' = \mathcal{S}(L)$ and $M' = \mathcal{S}(M)$ are separable too. Due to Lemma 2, $\mathcal{S}(N) = \mathcal{S}(L) \otimes_p \mathcal{S}(M)$. Hence, by Theorem 26, $\mathcal{S}(N)$ is separable.
(2) By Lemma 29, $L' = L \otimes_t \Sigma_t$ is split-separable. Now construct M' by adding initial and final places to M (M' is split-separable as well) and consider $N' = L' \otimes_{p_t} M'$. N' is split-separable due to part (1) of this theorem. Label i_M and f_t in N' to τ, then N' is weakly bisimilar to N, where all labels are visible. So N is split-separable too. □

Theorem 32. *SMWF-nets and acyclic MGWF-nets are split-separable.*

Proof. Since SMWF-nets and acyclic MGWF-nets are serialisable, they are also separable. Now notice that the classes of SMWF-nets and acyclic MGWF-nets are closed under the split-refinement operation, hence, these nets are split-separable. □

5 ST-nets

It is hard to find algorithms that check soundness and/or separability for an arbitrary WF-net, but we can define classes of nets that are sound and (split) separable by construction. One such class, called ST-nets, is treated in this section. These nets are constructed from state machines and marked graphs by means of refinement. In many cases, modelling problems can be solved by (provably correct) ST-nets.

Algorithm 36 (CheckST(N)).

```
$\mathit{\Delta} := \mathit{CompDistEnd}(N)$;                              compute distances to the end node
$X := S_N \cup T_N \setminus \{f_N\}$;                                     initialise search for $x$
while $X \neq \emptyset$ do                                                search loop
  pick $x \in X$;                                                          pick a candidate
  $M := \mathit{FindFactor}(N, x, \mathit{\Delta})$;                       search for a factor
  if $M \neq S_N \cup T_N \wedge \mathit{CheckSMMG}(M)$                    SM/MG factor found
  then return($\mathit{CheckST}(\mathit{Quotient}(N, M))$)                 recursive call
  else $X := X \setminus \{x\}$                                            continue search
od;                                                                        No smaller SM/MG factor found
return($\mathit{CheckSMMG}(N)$)
```

Algorithm 37 (FindFactor($N, x, \mathit{\Delta}$)).

```
$X = x^{\bullet}$;                                                         initialise possible internal nodes
$Y = \emptyset$;                                                           initialise possible end nodes
while $X^{\bullet} \not\subseteq (X \cup Y) \vee {}^{\bullet}X \not\subseteq (X \cup \{x\})$ do    stop when augmentation stabilises
  $X := X \cup X^{\bullet} \cup ({}^{\bullet}X \setminus \{x\}) \cup {}^{\bullet}Y$;               augmentation step
  $Y := \{y \in X \mid \mathit{\Delta}(y) = \mathit{Min}_{z \in X}\, \mathit{\Delta}(z)\}$;         compute candidates for $y$
  if $\exists f : Y = \{y\} \wedge x \neq y \wedge \mathit{type}(x) = \mathit{type}(y)$             test for candidate $y$
  then $X := X \setminus Y$ else $Y := \emptyset$ fi                       adjust $X$, $Y$
od;                                                                        augmentation stable
return($X \cup Y \cup \{x\}$)                                              $S_N \cup T_N$ if unsuccessful
```

Fig. 10. Factorization algorithm

Definition 33. *The set $\mathcal{N}$ of ST-nets is the smallest set of nets N defined as follows:*
– *if N is an acyclic MGWF-net, then $N \in \mathcal{N}$;*
– *if N is an SMWF-net, then $N \in \mathcal{N}$;*
– *if $N \in \mathcal{N}, s \in S_N$ and $M \in \mathcal{N}$ is an sWF-net, then $N \otimes_s M \in \mathcal{N}$;*
– *if $N \in \mathcal{N}, t \in T_N$ and $M \in \mathcal{N}$ is a tWF-net, then $N \otimes_t M \in \mathcal{N}$.*

Property 34. Let N be an ST-net, $N = L \otimes_n M$ for some WF-net L, $n \in S_L \cup T_L$ and ST-net M. Then L is an ST-net as well.

Theorem 35. *All ST-nets are sound and split-separable.*

Proof. Follows immediately from Theorems 15, 17, 9, 32 and 31. □

Algorithm 36 checks whether a given WF-net N is an ST net. It looks for a subnet of N, which is an STWF- or MGWF-net, i.e. $N = L \otimes_n M$ for some node n of a WF-net L. We call such a net M a *factor* of N and L the *quotient*. By Definition 33 and Property 34, N is an ST-net iff L is an ST-net. So the algorithm proceeds recursively with checking whether L is an ST-net. There exist various ways to speed up the algorithm but we choose the given presentation for the sake of simplicity.

The algorithm starts by computing the distance function $\Delta : (S_N \cup T_N) \to \mathbb{N}$ that gives the length of the shortest path of a node x to f_N. Then we pick up an arbitrary node $x \neq f_N$ and compute the smallest SM/MG factor (if any) with initial node x, with following Algorithm 37. Note that a factor of N with initial node x and terminal node y corresponds to a set S of nodes containing x, y and all successors of nodes $n \in S\backslash\{y\}$ and predecessors of nodes $n \in S\backslash\{x\}$, i.e. such that ${}^{\bullet}(S\backslash\{x\}) \subseteq S$ and $(S\backslash\{y\})^{\bullet} \subseteq S$. This observation allows us to compute the smallest such S by successive augmentation, starting with the set S containing all nodes from $x^{\bullet}$. The candidate for being the terminal node y in each augmentation step is the node that is nearest to the end node f_N, which is the reason for calculating Δ. The algorithm uses the function *type* on nodes that returns either "place" or "transition". The minimal distance computation (*CompDistEnd*), SM/MG check (*CheckSMMG*) and quotient computation (*Quotient*) are trivial and have not been elaborated further.

6 Conclusion

In this paper we studied workflow nets that allow "batched" cases. This perspective led to a strengthened notion of soundness. Advantages of this notion is that sound in the new sense WF-nets can be used freely as components without restricting their use to e.g. safe nets. Bisimilarity results speed up verification of temporal properties for composite nets.

Comparison of 1-soundness and (strengthened) soundness led to the notion of separability: independency of individual cases within a batch. Weakly separable and 1-sound nets are (strongly) sound. We introduced a notion of split-separability and proved its compositionality w.r.t. refinement, allowing a hierarchical approach to modelling and validation. A particular application of this approach are the processes that can be modelled by ST-nets, which are "sound by construction" and split-separable.

Future work We investigated a strengthening of 1-soundness, though as argued in [7], 1-soundness is too strong a notion for some applications. It is interesting to investigate e.g. separable nets that are not fully sound.

Decidability and computability are an issue. Clearly, 1-soundness can be assessed by coverability analysis (c.f. [12]), but soundness and separability are a different matter. A decision algorithm for separability of 1-sound WF-nets can be found, but as yet not an efficient one. Soundness is probably undecidable in general, as well as separability, but it is still a question for further investigations.

Soundness and separability of communicating WF-subnets (c.f. [10]) will need extension of our class of operators that preserve soundness and separability. We intend to develop component-oriented strategies for connecting nets. Wider classes of nets than WF-nets can be considered as well. The use of net components with several entry and/or exit nodes enables a component-based modelling strategy that allows more freedom than refinement alone.

Acknowledgment We are grateful to the referees for their constructive remarks and suggestions.

References

1. W.M.P. van der Aalst. *Verification of Workflow Nets.* In Azéma, P. and Balbo, G., editors, *Proceedings ATPN '97*, LNCS 1248, Springer 1997.
2. W.M.P. van der Aalst, J. Desel and A. Oberweis, editors *Business Process Management, Models, Techniques and Empirical Studies.* LNCS 1806, Springer 1998.
3. W.M.P. van der Aalst and K.M. van Hee. *Workflow Management: models, methods and systems.* The MIT Press, 2000.
4. W.M.P. van der Aalst. *Workflow Verification: Finding Control-Flow Errors using Petri-net-based techniques.* In [2], pages 161-183.
5. S. Ceri and G. Pelagatti. *Distributed Databases: Principles and Systems.* McGraw-Hill 1984.
6. P. Cousot and R. Cousot. Abstract interpretation: A unified lattice model for static analysis of programs by construction or approximaton of fixpoints. In *Fourth Annual Symposium on Principles of Programming Languages (POPL) (Los Angeles, Ca)*, pages 238–252. ACM, January 1977.
7. J. Dehnert and P. Rittgen. *Relaxed Soundness of Business Processes.* In K.R. Dittrich, A. Geppert and M.C. Norrie, editors, *Proceedings CAISE '01*, LNCS 2068, pages 157-170, Springer 2001.
8. J. Desel and J. Esparza. *Free Choice Petri Nets.* Cambridge University Press, 1995.
9. R.J. van Glabbeek and R.P. Weijland. *Branching Time and Abstraction in Bisimulation Semantics (extended abstract).* In G.X. Ritter, editor, *Proceedings IFIP '89*, pages 613-618. North Holland 1989.
10. E. Kindler, A. Martens and W. Reisig. *Inter-operability of Workflow Applications: Local Criteria for Global Soundness.* In [2], pages 235-253.
11. R. Milner. Operational and algebraic semantics of concurrent processes. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science, vol. B*, chapter 19, pages 1201–1242. Elsevier Science, 1990.
12. H.M.W. Verbeek, T. Basten, and W.M.P. van der Aalst. Diagnosing workflow processes using Woflan. *The Computer Journal*, 44(4):246–279, 2001.