Workflow Model Compositions Preserving Relaxed Soundness

Juliane Siegeris (born Dehnert)[1] and Armin Zimmermann[2]

[1] Fraunhofer ISST Berlin, juliane.siegeris@isst.fraunhofer.de
[2] Technische Universität Berlin, azi@cs.tu-berlin.de

Keywords: Workflow, Composition, Inter-organizational Workflow, Validation, Petri nets

Abstract. Very often, e.g. in the context of inter-organizational Workflow or web services, it is necessary to merge existing business process descriptions. It is clear that correctness criteria valid for the single process descriptions should remain valid also for the combined model. However, looking at the popular soundness criterion this can not always be guaranteed. In this paper various composition alternatives are summarized and their ability to preserve relaxed soundness (in contrast to soundness) is investigated.

1 Introduction

Process-aware information systems are an important aid in the design, improvement and execution of complex business processes. An important support for the modeling of complex business processes is provided by composition techniques. There are different scenarios for their application. The first are modular modeling and the combination of workflow patterns or building blocks. Other scenarios fall in the context of inter-organizational workflows or web services. Here it is essential to combine existing process descriptions on the basis of information exchange. Division of labor in general requires workflow composition, also inside one organization. Efficient use of available resources is an issue here.

The significance of composition within workflow modeling is reflected in the literature by numerous related publications, see e.g. [KMR00,AH02,AHT02] [HB03,CWBH+03]. Different composition variants are described and the result is checked for structural and behavioral properties. So far, the focus was put on the soundness property, i.e liveness and boundedness of the composed process model.

The aim of this paper is to analyze a list of significant composition techniques in terms of WF-nets and to check whether the composition of relaxed sound models is again relaxed sound. We will see that in comparison to soundness, relaxed soundness is preserved by additional composition techniques.

For the modeling we refer to WF-nets, a variant of Petri nets, that have been successfully used for the description and analysis of business processes. Their formal and furthermore operational semantics allows to use the process model as input for a workflow engine directly. In order to do this, the process description should be sound [Aal98]. *Soundness* guarantees that there are no dead tasks and that the process will always terminate properly, i.e. achieve the required result. *Relaxed soundness* has been proposed as a weaker property than soundness, thus allowing more workflow structures. In a relaxed sound WF-net, not all but only so many execution sequences must terminate properly, that every transition (task) is visited at least once. In [DZ04,DvdA04] it was shown how methods from Petri net controller synthesis can be applied to transform a relaxed sound and bounded WF-net into a sound model. Advantages and disadvantages w.r.t. the modeling and analysis of workflows are discussed for the two mentioned properties and well-structuredness in [DZ05].

The aim of this paper is to emphasize the benefit of relaxed soundness against the background of composition. Therefore, we will investigate a list of significant composition techniques and check whether relaxed soundness is preserved. The following techniques are covered:

- Refinement of a task by another workflow (or *subcontracting* in an interorganizational workflow): An atomic task is split into substeps that are described by another workflow in a hierarchical fashion. This technique goes back to [Val79] and was redefined for WF-nets in [AH02].
- Combinations of workflows as a whole: the simplest case is *chained execution* or sequential, but other options include iterative, parallel, alternative, discriminative (race condition), and multi merge composition, see e.g. [AH02,HB03,CWBH+03].
- Client-server-like asynchronous composition with information exchange during concurrent execution (similar to *loosely coupled*). Parts of the workflow are executed concurrently after the invocation of a service, and arbitrary information exchange may take place between the partners during the service execution, see e.g. [Aal99,KMR00,AHT02,Mar05].
- Parallel composition with mutual use of restricted resources or *capacity sharing*: Two or more workflows operate distributed and need to be synchronized because of common resources.

We will prove that all of the above composition techniques in fact preserve relaxed soundness. This is important because it guarantees that by starting with simple relaxed sound building blocks and combining them following the given composition rules, ill-formed workflows are avoided. The resulting complex WF-nets can thus be made sound automatically following [DZ04,DvdA04], and executed on a workflow management system afterwards.

The remainder of the paper is structured as follows. Necessary basics are briefly revisited first. The main part of the paper recalls the mentioned set of composition techniques in terms of WF-nets and provides proofs for the fact that the composition types preserve relaxed soundness. Some concluding remarks are given finally.

2 Preliminaries

For the modeling of business processes we refer to Place/Transition nets[3] and use the more specific class of WF-nets as introduced in [Aal98]. A WF-net (P, T, F) is briefly characterized by a source place ($^\bullet i = \emptyset$) and a sink place o ($o^\bullet = \emptyset$). Furthermore, it must hold that for any node $n \in (P \cup T)$ there is a path from i to n and from n to o. This ensures that every task (transition) or condition (place) contributes to the processing of cases.

Considering the behavior of a WF-net, we will always investigate the life-cycle of a single case, thus consider systems where initially only the source place i is marked ($M_i(i) = 1$ and for all $p \in P \setminus \{i\} : M_i(p) = 0$). Figure 1 (i) shows a simple WF-net with two parallel threads.

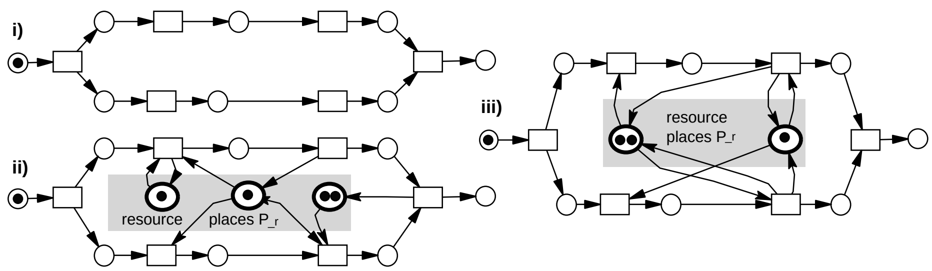

Fig. 1. A standard WF-net (i) and two resource constrained derivatives (ii) and (iii)

In the standard definition of WF-nets, resources are not explicitly characterized. According to [vHSSV05] we extend the notion of WF-nets to include information about the use of resources in the model. A resource belongs to a type. For every type a new place is introduced in the net, where resource tokens are located when they are available. The resources become part of the case-modeling tokens when they are occupied. Resources are assumed to be durable, i.e. they are used (blocked) and released later on. Resources are never created nor destroyed.

Definition 1 (Resource constrained WF-net). *A WF-net $PN = (P, T, F)$ becomes a resource constrained WF-net (P_{rc}, T, F_{rc}) by enhancing the set of places P with the set P_r of resource places ($P_{rc} = P \cup P_r, P \cap P_r = \emptyset$) and the flow relation F by corresponding arcs F_r ($F \cup F_r, F_r \subseteq (P_r \times T) \cup (T \times P_r)$).*

A standard WF-net can thus be interpreted as a special case of a resource-constrained one, where $P_r = \emptyset$. Different examples, illustrating the use of resources, are provided in the resource constrained WF-nets of Figure 1 (i-iii).

[3] An introduction to Place/Transition nets is e.g. given in [DR98], where the concepts of pre- and postset $^\bullet x$ and $x^\bullet$, marking M, firing rule and firing sequence are defined among others.

Resources are neither created nor destroyed during the processing. Therefore they are part of the initial marking M_i of the corresponding resource-constrained WF-system, and their initial number is specified by $R : P_r \longrightarrow \mathbb{N}$. Formally, M_i is defined as:

$$\forall p \in P \cup P_r : M_i(p) = \begin{cases} 1 & \text{if } p = i \\ R(p) & \text{if } p \in P_r \\ 0 & \text{otherwise} \end{cases}$$

An important property in the context of workflow management is soundness [Aal98]. A WF-net is sound if termination in a final marking M_f is always possible. Furthermore, there are no dead transitions and neither deadlocks nor live-locks.

This definition was enhanced for resource constrained WF-nets with multiple cases (k-soundness [vHSSV05]). In this paper we only consider single cases (the special case of 1-soundness), for which the definition reads as below. For notational convenience we introduce a *final marking* M_f such that

$$\forall p \in P \cup P_r : M_f(p) = \begin{cases} 1 & \text{if } p = o \\ R(p) & \text{if } p \in P_r \\ 0 & \text{otherwise} \end{cases}$$

Definition 2 (Soundness of resource constrained WF-nets). *A resource constrained WF-net PN with input place i is sound for some $R \in \mathbb{N}^{P_r}$ iff*

1. *For every state M reachable from state M_i it holds that the number of tokens in each resource place is less than or equal to its initial number: $\forall M \in R_{PN}(M_i), \forall p \in P_r : M(p) \leq R(p)$ (resources are durable).*
2. *For every state M reachable from state M_i, there is a firing sequence leading from state M to state M_f: $\forall M : (M_i \xrightarrow{*} M) \Rightarrow (M \xrightarrow{*} M_f)$ (proper termination).*
3. *In addition to [vHSSV05] we require that there are no dead transitions in PN: $\forall t \in T \; \exists M, M' : (M_i \xrightarrow{*} M \xrightarrow{t} M')$.*

Enhancing the definition of a sound firing sequence accordingly, we get

Definition 3 (Sound firing sequence). *Let PN be a resource-constrained WF-net initially marked with M_i. A firing sequence σ is sound iff it leads from M_i to M_f and does not violate the durability property: $M_i \xrightarrow{\sigma} M_f \wedge \forall M \in Visited_{PN}(M_i, \sigma)$ [4], $\forall p \in P_r : M(p) \leq R(p)$.*

The set of sound firing sequences of a WF-net PN with initial marking M_i is denoted by Σ^{sound}_{PN,M_i} in the following. If the initial marking is implicitly clear, we just write Σ^{sound}_{PN}.

In a sound WF-net all firing sequences beginning in M_i can be continued until M_f (i.e. terminate properly), resulting in a sound firing sequence. The resource

[4] With $Visited_{PN}(M, \sigma)$ we denote the set of markings visited during a firing sequence $\sigma = t_1, t_2, \ldots, t_n$ starting in M

constrained WF-nets of Figure 1 are all sound in the shown initial marking. However, if the resource places of net (iii) are initially only marked with one token, soundness of the corresponding WF-system would be violated.

Another important property for the modeling of business processes is *relaxed soundness*. A WF-system is relaxed sound iff each transition is contained in at least one sound firing sequence of the system.

Definition 4 (Relaxed soundness of resource constrained WF-nets). *A process specified by a (resource-constrained) WF-system* (PN, M_i) *is relaxed sound iff every transition of* PN *is contained in a sound firing sequence:* $\forall t \in T\, \exists \sigma \in \sigma_{sound}(PN, M_i) : t \in \sigma$.

Relaxed soundness poses weaker requirements to a process description than soundness. In contrast to a sound WF-net, a relaxed sound WF-net may have firing sequences which do not terminate properly. These firing sequences possibly deadlock in a marking other than M_f or do not terminate at all (livelock). Consider again the resource constrained WF-net from Figure 1 (iii). The net is relaxed sound, also if there is initially only one token per resource place.

From the given definitions it can easily be seen that a *sound* WF-net (either resource constrained or not) will also necessarily be *relaxed sound*. Note that if there are no resource places, the definitions of soundness and relaxed soundness coincide with the classical soundness notion [Aal98] and the primary notion of relaxed soundness [DR01], respectively.

3 Composition Techniques

In this section, different composition techniques are considered and interpreted in terms of WF-nets. Moreover, it is shown that their application to relaxed sound WF-nets leads to composed models that are again relaxed sound. The results presented in the first two subsections mainly transfer well-known results to the class of relaxed sound WF-nets.

To start with, net composition via transition refinement is considered. This method was first introduced in [Val79], where it was used to enhance Petri nets by well-formed blocks. In [AH02] the method was adapted for WF-nets.

3.1 Composition via transition refinement

Two WF-nets are composed by replacing a transition of the first WF-net (A) by a transition-surrounded second WF-net (B). Figure 2 illustrates this kind of composition. It is easy to see that the resulting net is again a WF-net.

Refining a transition t of a sound WF-net A by a transition surrounded sound WF-net B, the composed WF-net is not necessarily sound. If the main WF-net (here WF-net A) is not safe, proper termination is not guaranteed. We refer again to Figure 2. The result of the illustrated composition is not sound[5]. This goes back to the fact that the refining WF-net becomes initiated with two tokens

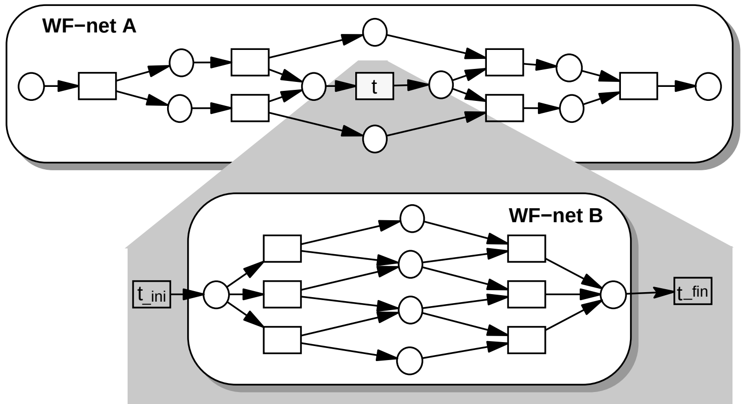

Fig. 2. Not sound WF-net composed by transition refinement

in i. This caused a deadlock, as B was in fact 1- but not 2-sound. However, if the two WF-nets are sound[6] and the main WF-net is additionally safe, the composed WF-net is always sound [AH02]. We will now investigate the property for relaxed sound WF-nets which are not necessarily safe.

Theorem 1. *When a transition t of a relaxed sound WF-net A is refined by a relaxed sound WF-net B, the resulting WF-net C is again relaxed sound.*

Proof. To prove that C is relaxed sound, we have to show that every transition t_i of C is contained in at least one sound firing sequence of C[7]. We construct a set of sound firing sequences of C as follows. First, all sound firing sequences of A that do not contain t are considered (this set may be empty). Second, we take all sound firing sequences of A that do contain t (there must be at least one of them) and replace t by one of the (always existing) sound firing sequences of B. Third, we select one of the sound firing sequences of A containing t, and form a set of new firing sequences by substituting t in it by elements of a set of sound firing sequences of B. This set is chosen such that all transitions of WF-net B are contained in it (which is always possible because B is relaxed sound). The union of these three sets is a set of sound firing sequences of C by construction. It remains to be shown that each transition of C is contained in at least one of them, which is obvious because A and B are relaxed sound and all their "local" sound firing sequences are contained in the constructed set. □

3.2 Combinations of workflows as a whole

Within this paragraph we consider purely structural composition techniques that define the interaction of two WF-nets A and B by the use of workflow pattern.

[5] Note, here the counterexample from [HSV03] was slightly changed, as the refining WF-net (here WF-net B) was primarily not sound.

[6] We again refer to the classical soundness definition here, i.e. 1-soundness

[7] Except for $t_i = t$, which is replaced by B.

The following basic and advanced pattern will be used: sequence, structured cycle, parallel split (AND-join), synchronization (AND-join), exclusive choice (XOR-split), simple merge (XOR-join) multiple choice (OR-split), synchronizing merge (OR-join), discriminator and multi merge.

Sequential composition of WF-nets One workflow process is enabled after the completion of the other. Within the proposed composition technique this was implemented linking two WF-nets with a transition connecting the sink of the first with the source of the second WF-net, cf. Figure 3 (i).

Iterative composition of WF-nets Two workflow processes can repeatedly be executed after one another, where the loop can be abandoned after termination of one of the processes. The composition technique, implementing this pattern of a structured cycle in terms of WF-nets, is provided in Figure 3 (ii).

Parallel composition of WF-nets Two workflow processes are routed in parallel. This composition technique was implemented accommodating the parallel split and the synchronization pattern as shown in Figure 3 (iii).

Alternative composition of WF-nets Two workflow process are activated alternatively. There are two implementations possible. Applying the basic WF-pattern exclusive choice and simple merge two WF-nets are composed, such that only one of them is executed, cf. Figure 3 (iv). The second possibility of conditional routing is implemented accommodating the advanced WF-pattern multiple choice and synchronizing merge. Here, the two WF-nets can be used either in parallel or alternatively, cf. Figure 4 (i).

Discriminative composition of WF-nets Two workflow processes are enabled at the same time. After the first terminates, subsequent tasks are activated. Termination of the second process is awaited but ignored, i.e. no subsequent tasks are triggered. This behavior was implemented using a parallel split and a discriminator pattern. Figure 4 (ii) illustrates this composition technique. The subsequent task is modeled by transition t. The privilege to activate the subsequent task is modeled by a semaphore, i.e. a resource place initially marked with one token.

Multi merge of WF-nets Two workflow processes are activated in parallel. If one of them terminates the subsequent task is activated. In contrary to the previous composition technique, the subsequent task is not activated once, but twice. In order to unify the two threads again, the proposed composition technique uses an exclusive choice and a synchronization pattern. Figure 4 (iii) illustrates this composition technique.

When applying the proposed set of composition techniques, it is guaranteed that the resulting net is always a WF-net, which can hence again be used for

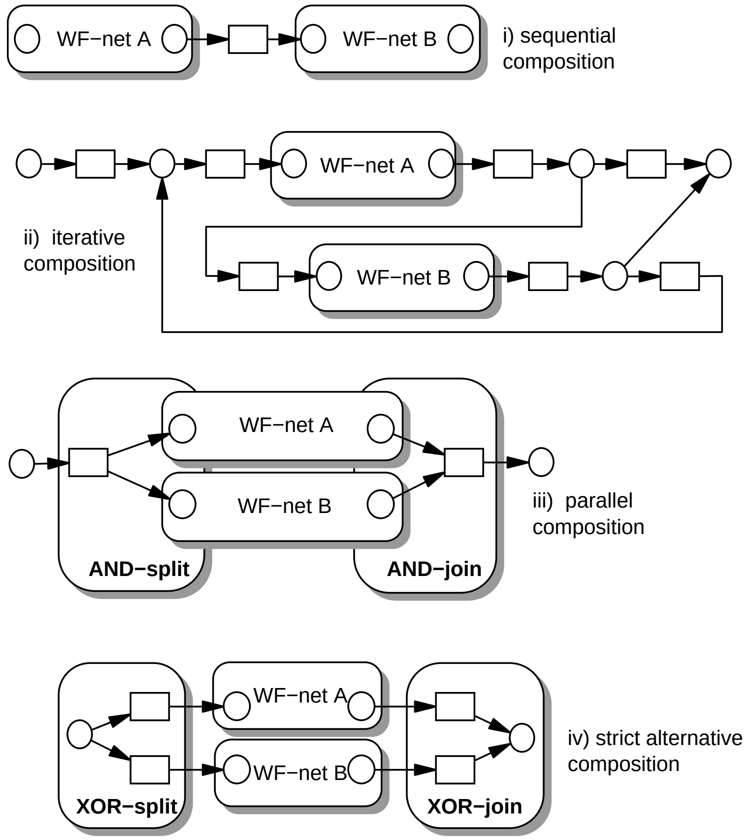

Fig. 3. Structural composition rules using basic WF-pattern

composition. This follows from the fact that the WF-nets are only composed via their source and there sink place. Note that the proposed composition techniques only represent a choice. There are other combinations of WF-pattern possible, providing meaningful compositions of two or more workflow processes.

We will now investigate whether the proposed composition techniques maintain relaxed soundness. Therefore we will prove the following statements.

1. A sequence of relaxed sound WF-nets is relaxed sound.
2. The result of the iterative composition of two relaxed sound WF-nets is again relaxed sound.
3. A parallel composition of relaxed sound WF-nets is relaxed sound.
4. An alternative composition of relaxed sound WF-nets is relaxed sound.
5. The proposed discriminative composition of two relaxed sound WF-nets yields again a relaxed sound WF-net.
6. The proposed multi merge composition of two relaxed sound WF-nets yields again a relaxed sound WF-net.

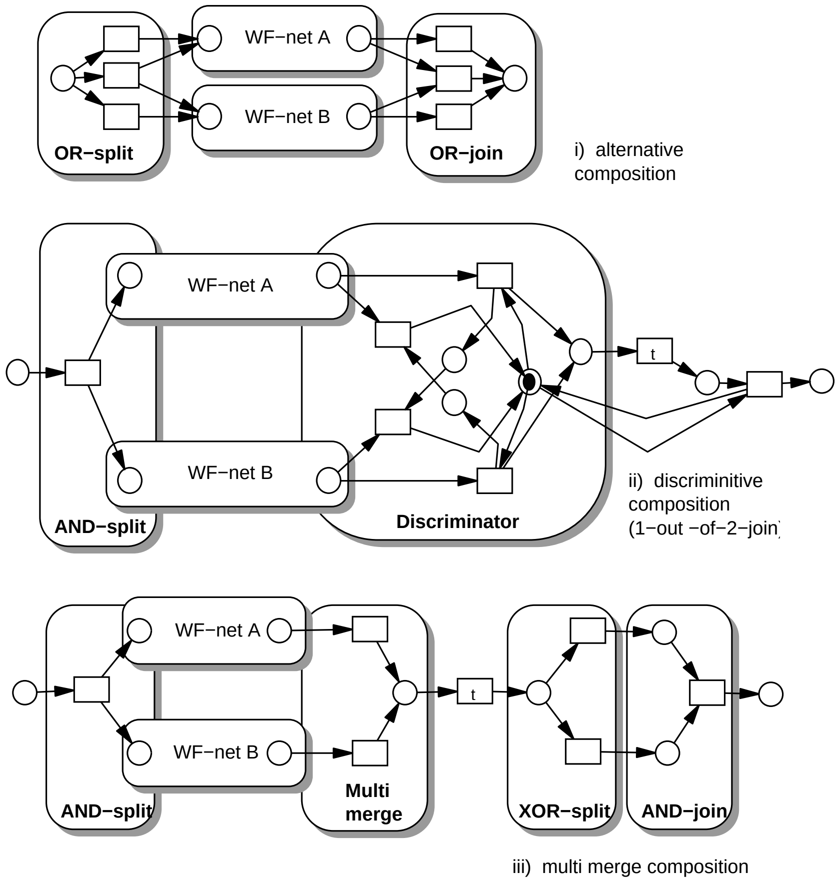

Fig. 4. Structural composition rules using advanced WF-pattern

Proof. The proof argumentation is for all statements the same. Replacing every placeholder for the WF-nets A and B in the composition rules with a single transition with one input and one output place, we gain a set of WF-nets. All these WF-nets are relaxed sound. We exploit the previous result, concluding that refining the transitions by relaxed sound WF-nets, the resulting nets are again relaxed sound. □

Note that this result cannot be transferred for soundness, as some of the gained WF-nets are not sound, namely the ones described in Figure 4 (i) and 4 (iii).

The following two composition techniques are somehow more complex than the previous ones. The difference is that the interaction of the two WF-nets now goes beyond the use of the source and the sink place but comprises additional elements.

3.3 Combination of WF-Nets due to information exchange

The fact that e.g. two organizations interact on some purpose is mostly reflected in the exchange of data or flow of information. In terms of WF-nets this is modeled by interface places. Typical examples include sending and reception of data or documents.

The corresponding composition technique assumes two independent WF-nets A and B, where B provides a service that A needs (client-server pattern). Figure 5 illustrates this type of combination. Some information must be passed between A and B to facilitate their interaction. Therefore the server WF-net B has to be invoked by a request, and an interface for the exchange of results and possible further data/information must be available.

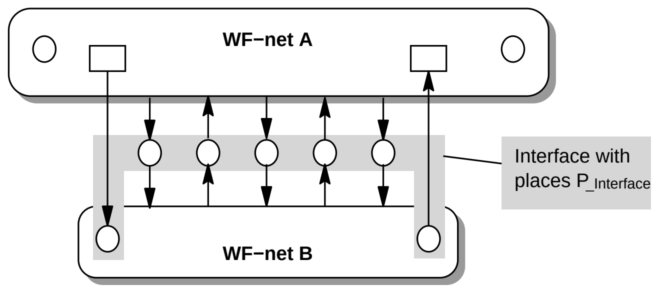

Fig. 5. Combination of WF-nets via interface places

This composition technique is similar to the approach proposed in [AHT02], where C-nets modeling the behavior of SW-components are composed to form complex architectures. Therefore a set of interface places was introduced, connecting transitions of the two WF-nets. This set is denoted by $P_{interface}$ in the following, and shown in the figure.

As in [AHT02] we assume the interaction to be always executed within the scope of the client (WF-net A). That is, the client starts the interaction (marks the initial place of B) and the server always reports back to the client when it finished the interaction (marks the final place of B). The combined workflow model C comprises the client model A, interface places $P_{interface}$, and server workflow B. Therefore, initial and final places of the client i_A and o_A are the respective places of the combined model. The initial and final place of the server are part of the interface ($i_B, o_B \in P_{interface}$).

There are two further assumptions on this composition type: First, every place of the interface connects exactly one pair of transitions: the introduced interface places thus have exactly one transition in their preset and one transition in their postset, out of which obviously exactly one belongs to each WF-net A or B. Formally, $\forall p \in P_{interface} : |{}^{\bullet}p| = |p^{\bullet}| = 1$, ${}^{\bullet}p \in T_A \Leftrightarrow p^{\bullet} \in T_B$ and vice versa. We denote by the set of synchronization transitions T_{sync} the ones that are connected to an interface place, $T_{sync} = \{t \in T_A \cup T_B | \exists p \in P_{interface} : t \in {}^{\bullet}p \cup p^{\bullet}\}$.

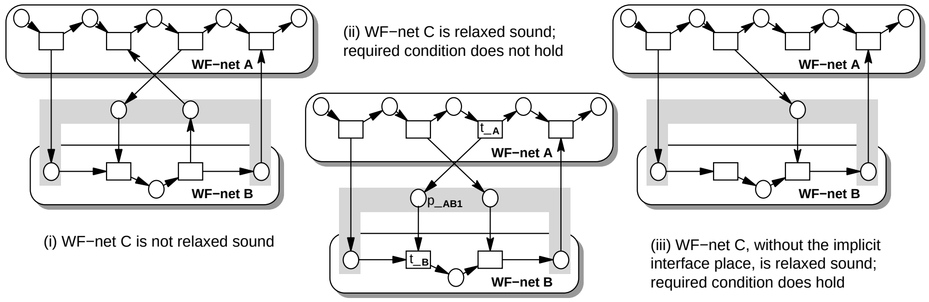

Fig. 6. Examples for the combination of WF-nets via interface places

Moreover, we require every synchronization transition to be connected to only one interface place $\forall t \in T_{sync} : |{}^\bullet t \cap P_{interface}| = |t^\bullet \cap P_{interface}| = 1$. There is thus a one-to-one correspondence between synchronization transitions in A and B, which is formally captured by relation $sync(t_1, t_2) \Leftrightarrow \exists p \in P_{interface} : t_1 \in {}^\bullet p, p \in {}^\bullet t_2 \vee t_2 \in {}^\bullet p, p \in {}^\bullet t_1$.

It has been shown in [AHT02] that the combined net C is again a WF-net. However, it is not clear whether the (relaxed) soundness of C follows from the soundness properties of A and B. In the general case (without further restrictions) the combination does not preserve soundness nor relaxed soundness which is illustrated in Figure 6 (i).

For sound WF-nets there are two alternative additional requirements that are sufficient conditions for a soundness-preserving composition of this type. It was shown in [AHT02] that the global model C is sound if the local workflow nets are branching bisimular. Its informal meaning for the workflow is that the behavior of A is not restricted by adding B and the interface. A structural property that is a sufficient condition which is simpler to check is a *request-response-pattern* defined in the same paper. However, it restricts the allowed interactions significantly.

We will show in the following that C is relaxed sound if A and B are, provided that there are pairs of sound firing sequences in A and B such that the synchronization transitions appear in the same order and multiplicity in them.

A minor additional requirement is an upper bound on the number of occurrences of every synchronization transition in any local firing sequence (in an isolated A and B). This is done only to prevent infinitely many invocations of B. Transitions other than the synchronizing ones may still occur infinitely often.

The idea behind the proof is to look at the local sound firing sequences of A that have some interaction with B, and to consider those that "match" some local firing sequence of B. Two firing sequences match if they describe an interleaving of transition firings that may be executed concurrently without a deadlock. The non-synchronization transitions are obviously not an issue here, we only have to consider the interactions between the two models. Each of the firing sequences in A and B can be executed locally until the next synchronization transition appears. Here come the structural restrictions into play: because of the one-to-one relationship between synchronization transitions in A and B, their sequence is defined by the way they are connected with an interface place. If we imagine all matching firing sequences constructed in this way, we only have to make sure that every transition of A and B appears in one of them to know that C is relaxed sound.

To improve readability of the following theorem we introduce the notion of an abstracted firing sequence to filter out non-synchronization transitions. A firing sequence $\sigma^{abstract}$ of a WF-net $PN = (P, T, F)$ w.r.t. the transition subset $T_{sync} \subseteq T$ is denoted as an abstraction of a firing sequence σ of PN iff $\sigma^{abstract}$ is derived from σ by deleting every occurrence of all $t \in T \setminus T_{sync}$.

We say that two abstracted firing sequences $\sigma_A^{abstract}$ and $\sigma_B^{abstract}$ of WF-nets A and B match if their lenghts are equal, $\left|\sigma_A^{abstract}\right| = \left|\sigma_B^{abstract}\right|$, and the transition steps are pairwise connected by interface places[8]: $\forall i \in 1, \ldots \left|\sigma_A^{abstract}\right| :$ $sync(\sigma_A^{abstract}[i], \sigma_B^{abstract}[i])$.

Theorem 2. *Let WF-net C be the composition of relaxed sound WF-nets A and B as described above, and $P_{interface}$ their set of interface places. Consider the two sets of all abstracted sound firing sequences for A and B, denoted by $\Sigma_A^{sound,abstract}$ and $\Sigma_B^{sound,abstract}$.*

The composed WF-net C is relaxed sound if every synchronization transition of A is contained in an abstracted sound firing sequence of A for which there is a matching abstracted sound firing sequence of B (and vice versa). Formally, $\forall t \in T_A \cap T_{sync} : \exists \sigma_A \in \Sigma_A^{sound,abstract}$ such that $t \in \sigma_A$ and $\exists \sigma_B \in \Sigma_B^{sound,abstract}$ with σ_A matching σ_B.

Proof. To prove that the composed WF-net C is relaxed sound, we have to show that there are sound firing sequences of C such that all transitions of C are contained in at least one of them. We consider two cases for transition $t \in T_A \cup T_B$:

1. There is no sound firing sequence containing t with a matching sequence: Thus there is no synchronization transition contained in the sound firing sequences visiting t, and hence B is not invoked; therefore $t \in T_A$. As WF-net A was relaxed sound, there is a sound firing sequence $\sigma \in \Sigma_A^{sound}$ containing t. The firing sequences visiting t are not influenced by the introduction of the additional interface places (otherwise t would have been part of such a related pair of firing sequences), concluding that σ must also be a sound firing sequence of the composed WF-net C.

2. There is a sound firing sequence containing t with a matching firing sequence: Assume w.l.o.g. that $t \in T_A$, and denote the sound firing sequence containing t by σ_A. We may then safely assume from the theorem that there is at least one sound firing sequence σ_B of B that matches σ_A.

It remains to be shown that t is contained in a sound firing sequence of C. Such a firing sequence is constructed by an interleaving of σ_A and σ_B with the following rules.

- In every step, select either σ_A or σ_B to be progressed, such that every transition firing follows the local sequence in A or B.

[8] $\sigma[i]$ denotes the i-th transition in the sequence

– Transitions from σ_B may only be selected in the time span between an invocation of B, i.e. when a token is added to $i_B \in P_{interface}$, until B has terminated, i.e. when a token is added to o_B.
– If at least one of the next transitions in the sequences σ_A and σ_B is not a synchronization transition, select it to be fired. This is always possible in any order because there are no synchronization dependencies.
– In the case that both next transitions are in T_{sync}, fire them one after the other in the sequence that is specified by their postset or preset relation with the connecting interface place. This ordering is unique because of the restrictions on the interface.
– Continue until both sequences σ_A and σ_B have been fully executed, which is the case when o_A is marked.

The local order of the transitions in σ_A and σ_B remains the same in the constructed firing sequence of C, and all dependencies between A and B are observed. The effect of the introduced interface places comes down to a synchronization of the connected transitions. Because σ_A and σ_B were sound, we can conclude that also their constructed interleaving is sound. □

Note that our additional requirement is much weaker than the one given in [AHT02]. It is in fact sufficient to require bisimulation only for a set of firing sequences covering all transitions, to ensure that a composition of relaxed sound WF-nets preserves this property. The consequence of the fewer restrictions is that WF-net A may not only postpone but possibly also restrict the behavior of WF-net B and vice versa.

Although the above requirement is sufficient for a preservation of relaxed soundness, there are other cases in which C is relaxed sound as well. Figure 6 (ii) shows an example. In the shown case the problem stems from an unnecessary synchronization between transitions t_A and t_B, which is overspecified because of their indirect causal dependency. Such cases can be easily detected and avoided based on the notion of *implicit places* [Ber87]. A place is implicit if its removal does not change the overall behavior, i.e. does not enable additional firing sequences. As a consequence, we remove all implicit places from the interface, which possibly extends the set of synchronization patterns for which the above proof applies.

Removing the implicit place p_{AB1} from Figure 6 (ii) leads to the model given in Figure 6 (iii) where our condition holds. It can therefore be concluded that the composed WF-net from Figure 6 (ii) is relaxed sound.

3.4 Parallel composition with mutual use of restricted resources

For this composition technique we explicitly refer to resource-constrained WF-nets. Remember that resources were typed via resource places. If two processes request the same type of resources it is useful to compose the two nets by merging the resource places.

In the presence of shared resources it has to be investigated whether there are any bad interactions, e.g. leading to a deadlock. Therefore the two nets are always composed in parallel, i.e. initiated at the same time. Figure 7 illustrates

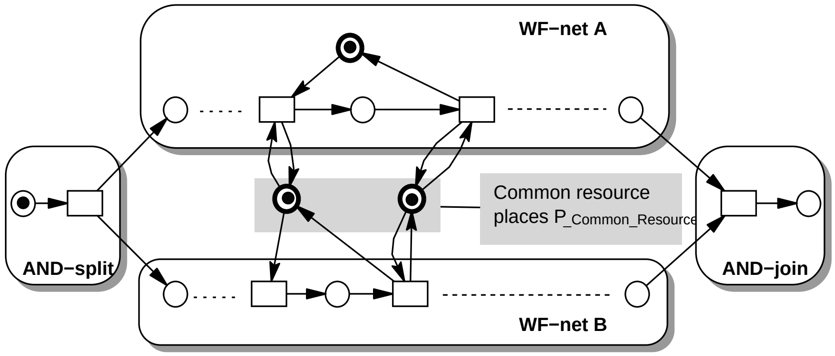

Fig. 7. Composition of WF-nets via common resource places

this kind of composition. It is obvious that the resulting net again fulfills the requirements of a WF-net.

Starting from two sound WF-nets this composition technique does not maintain soundness. A counterexample is given in Figure 8. Still, we will show that starting with relaxed sound WF-nets the resulting net is again relaxed sound.

Theorem 3. *Composing relaxed sound resource constrained WF-nets A and B at common resource places, the resulting WF-net C is relaxed sound.*

Proof. We only have to show that there are enough sound firing sequences in C such that all transitions of C are contained in at least one of them. We know the primary WF-nets contained enough sound firing sequences to cover the set T_A or T_B, respectively. These two nets are now composed in parallel. It is nevertheless possible to execute A completely first, and then B as a whole because of their relaxed soundness property. The resulting firing sequences are obviously sound sequences of C and cover all transitions of A and B by construction. □

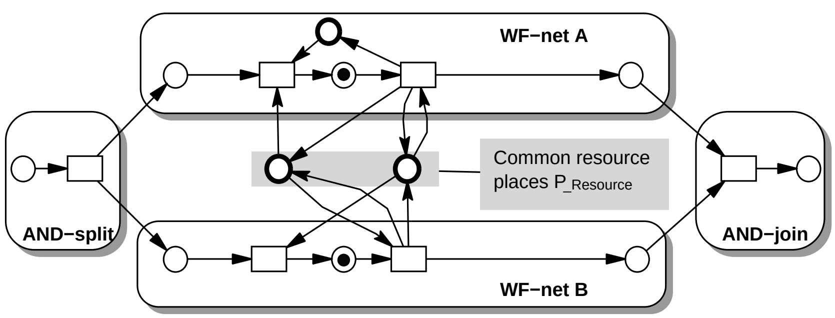

Fig. 8. Deadlock in a WF-net, composed by joining common resource places

4 Conclusion

This paper investigated whether typical composition techniques for Petri net workflow models preserve relaxed soundness. We have shown that (under additional restrictions in some cases) any two relaxed sound WF-nets can be composed, leading to a WF-net that is again relaxed sound. The application of previous results drawing on Petri net controller synthesis [DZ04,DvdA04] extend such a net to a sound one with an automated algorithm.

The presented results allow to construct WF-nets by combining basic patterns in a stepwise composition or hierarchical refinement approach. Any combination of the described compositions is possible in sequential steps. Such a composition always leads to a relaxed sound model if the initial building blocks were relaxed sound. The only restriction is that common resources and interface places may not be used at the same time for the proofs to hold. The controller generated by [DvdA04] guarantees a sound result to be derived from the final composition.

For the resource composition technique this means that the presented result is not as trivial as one may think from the proof. If the possibility of mutual waiting for the release of resources exist, it is not required to fully sequentialize the executions of A and B. Parts of the execution may allow interleaving without running into a deadlock. None of the possible concurrent behavior is deleted, because the controller algorithm always computes the maximally permissive behavior. Applying the algorithm to a relaxed sound model with shared resources thus results in scheduling resource accesses such that no deadlock will occur.

Although it is guaranteed that the composed WF-net is relaxed sound, it may be unbounded. That is the case if one of the initial WF-nets was unbounded; unboundedness is never introduced by the application of the presented rules. As the used controller algorithm only works on the basis of a finite reachability graph, it cannot be applied in these cases.

A side effect of the presented results is the following. Relaxed soundness can be shown in finite time if it holds, while the check for not being relaxed sound takes infinite time for unbounded nets [Deh03]. The set of compositions preserving relaxed soundness of this paper may offer a better possibility to check relaxed soundness for unbounded WF-nets. If subnets can be identified in a model such that it can be interpreted as the result of a composition, relaxed soundness has to be checked for the subnets only. The problem is thus cut back in size, which can be done repeatedly until a set of submodels is derived that are known to be relaxed sound.

References

[Aal98] W.M.P. van der Aalst. The Application of Petri Nets to Workflow Management. *The Journal of Circuits, Systems and Computers*, 8(1):21–66, 1998.

[Aal99] W.M.P. van der Aalst. Interorganizational Workflows: An Approach based on Message Sequence Charts and Petri Nets. *Systems Analysis - Modelling - Simulation*, 34(3):335–367, 1999.

[AH02] W.M.P. van der Aalst and K.M. van Hee. *Workflow Management: Models, Methods, and Systems*. MIT press, Cambridge, MA, 2002.

[AHT02] W.M.P. v.d. Aalst, K.M. van Hee, and R.A. v.d. Toorn. Component-based software architectures: a framework based on inheritance of behavior. *Science of Computer Programming*, 42(2–3):129–171, 2002.

[Ber87] G. Berthelot. Transformations and decompositions of nets. In G. Rozenberg, editor, *Advances in Petri Nets*, volume 266 of *LNCS*. 1987.

[CWBH+03] P. Chrzastowski-Wachtel, B. Benatallah, R. Hamadi, M. O'Dell, and A. Susanto. A Top-Down Petri Net-Based Approach for Dynamic Workflow Modeling. In W. van der Aalst, A. ter Hofstede, and M. Weske, editors, *Int. Conf. on BPM*, volume 2678 of *LNCS*, pages 336–353, 2003.

[Deh03] J. Dehnert. *A Methodology for Workflow Modeling - From business process modeling towards sound workflow specification*. PhD thesis, TU Berlin, 2003.

[DR98] J. Desel and W. Reisig. Place/Transition Petri Nets. volume 1491 of *LNCS*. Springer, 1998.

[DR01] J. Dehnert and P. Rittgen. Relaxed Soundness of Business Processes. In K.L. Dittrich, A. Geppert, and M.C. Norrie, editors, *Advanced Information System Engineering, CAISE 2001*, volume 2068 of *LNCS*, pages 157–170. Springer, 2001.

[DvdA04] J. Dehnert and W.M.P. van der Aalst. Bridging the Gap Between Business Models and Workflow Specifications. *Int. Journal of Cooperative Information Systems (IJCIS)*, 13(3):289–332, 2004.

[DZ04] J. Dehnert and A. Zimmermann. Making Workflow Models Sound Using Petri Net Controller Synthesis. In R. Meersman and Z. Tari et.al., editors, *Int. Conf. Cooperative Information Systems (CoopIS) 2004*, volume 3290 of *LNCS*, pages 139–154, Cyprus, 2004.

[DZ05] J. Dehnert and A. Zimmermann. On the Suitability of Correctness Criteria for Business Process Models. In W.M.P. van der Aalst and B. Benatallah et.al., editors, *Int. Conf. Business Process Management, BPM 2005*, volume 3649 of *LNCS*, pages 386–391, France, 2005.

[HB03] R. Hamadi and B. Benatallah. A Petri Net-based Model for Web Service Composition. In X. Zhou and K.-D. Schewe, editors, *14th Australasian Database Conference (ADC2003)*, volume 17 of *Conferences in Research and Practice in Information Technology*, Australia, 2003.

[HSV03] K.M. van Hee, N. Sidorova, and M. Voorhoeve. Soundness and Separability of Workflow Nets in the Stepwise Refinement Approach. In W.M.P. van der Aalst and E. Best, editors, *24th Int. Conf. on Application and Theory of Petri Nets*, LNCS, pages 337–356. Springer, 2003.

[KMR00] E. Kindler, A. Martens, and W. Reisig. Inter-Operability of Workflow Applications: Local Criteria for Global Soundness. In W.M.P. van der Aalst, J. Desel, and A. Oberweis, editors, *BPM: Models, Techniques, and Empirical Studies*, volume 1806 of *LNCS*, pages 235–253. Springer, 2000.

[Mar05] Martens, A. Analyzing web service based business processes. In M. Cerioli, editor, *8th Int. Conf. on Fundamental Approaches to Software Engineering (FASE 2005)*, volume 3442 of *LNCS*, pages 19–33. Springer Verlag, 2005.

[Val79] R. Valette. Analysis of Petri nets by stepwise refinements. *Journal of Computer and System Sciences*, 18:35–46, 1979.

[vHSSV05] K. M. van Hee, A. Serebrenik, N. Sidorova, and M. Voorhoeve. Soundness of resource-constrained workflow nets. In *ICATPN*, pages 250–267, 2005.