

Deciding Service Composition and Substitutability Using Extended Operating Guidelines

Christian Stahl^{*,a,b,1}, Karsten Wolf^c

^aHumboldt-Universität zu Berlin, Institut für Informatik, Unter den Linden 6, 10099 Berlin, Germany

^bDepartment of Mathematics and Computer Science, Technische Universiteit Eindhoven,
P.O. Box 513, 5600 MB Eindhoven, The Netherlands

^cUniversität Rostock, Institut für Informatik, 18051 Rostock, Germany

Abstract

We study the correct interaction between services using the following notion for correctness: there is no deadlock in the interaction of the services, and a given set of activities is not dead, that is, each activity in this set is executed in at least one run. The second condition has not been studied before.

An *operating guideline* of a service P is an operational characterization of all deadlock-free interacting partners of P . In this paper, we present an extension of the concept of an operating guideline to characterize all correctly interacting partners of a service P . This extension can be used for answering at least the following two questions. First, given a service R , does R interact correctly with P ? Second, given a service P' , can P be substituted by P' , that is, is every correctly interacting partner of P a correctly interacting partner of P' , too?

Key words:

Business process modeling and analysis, Process verification and validation, operating guidelines, substitutability, Petri nets

1. Introduction

One of the objectives of service-oriented computing (SOC) [1] is the modular structuring and loose coupling of interorganisational business processes. In this aspect, SOC meets the area of modeling and analysing workflows [2]. While SOC aims at composing complex business activities from more elementary ones (services), workflow modeling is (among others) concerned with the study of well-designed workflows and business processes. Central to the wellformedness of workflows is the concept of *soundness*. This property basically states that every process instance will terminate in a well-defined final state (i.e. it has no deadlocks and livelocks) while there are no useless (i.e. dead) activities. In the intersection of SOC and workflow modeling, we are thus interested in mechanisms for service composition (and related tasks such as discovery) which assure soundness in the overall system (e.g. a service orchestration).

Current approaches for matching and discovering services are incapable of asserting soundness in service discovery scenarios. Some approaches propose to compute and publish a public view P' of a provided service P [3, 4]. Then, a service requester R can check its composition $R \oplus P'$ to decide proper interaction. However, public view approaches do not explicitly state whether soundness of $P' \oplus R$ implies soundness of $P \oplus R$. Thus, existing public view approaches cannot be applied to obtain a globally sound system.

Other approaches suggest to compute an *operating guideline* OG_P for a given service P which represents all correctly interacting partners of P [5]. Then, a matching procedure between R and OG_P can be used for deciding whether $P \oplus R$ would interact correctly. Here, correctness refers to deadlock freedom so far.

*Corresponding author

Email addresses: stahl@informatik.hu-berlin.de (Christian Stahl), karsten.wolf@uni-rostock.de (Karsten Wolf)

¹Present address: TU Eindhoven; phone: +31 (0)40 247-5912; fax: +31 (0)40 246-3992

Deadlock freedom is a necessary but insufficient condition for soundness. In this paper, we extend the operating guideline approach by asserting—in addition to deadlock freedom—the absence of dead activities in the composed system. This is another necessary condition for soundness. The only remaining gap between our new approach and soundness is the possible existence of livelocks which are excluded by the soundness notion. For acyclic services, our approach already establishes soundness in the composed system since acyclic services cannot contain livelocks.

Another motivating scenario for our approach is inspired by [6]. In this article, all partners of a given service, which *enforce* or *exclude* certain behavioral patterns such as occurrences of activities, are characterized. This approach can be used, among others, for

- filtering of service registries for services that fit specific specifications (“enforce book”: I want to get a book selling service; “exclude credit card”: I do not want to pay by credit card),
- validating services by checking whether there exist partners that access certain features

Sometimes, enforcing some behavior is too strict. Consider an application for a credit with an online bank service. Of course, the client (service requester) wishes to have the activity “credit approved” executed in the service. However, there is hardly an online bank service where “credit approved” can be enforced by the client (which would mean that the user can always obtain a credit by just following a suitable communication pattern). There will rather be an internal decision based on which a credit is either approved or denied. In typical service models, the decision appears to the client as a nondeterministic choice. Thus, we need a weaker criterion that rules out at least all those services where “credit approved” is completely impossible. That is, R should match with OG_P if and only if it is at least *possible* to execute activity “credit approved” in the composition of the online bank service and the requester.

Formally, we want to compute a finite representation of the (generally infinite) set of all those partners R of a given service P where the composition $P \oplus R$ of both services is deadlock-free, and a certain set X of activities is not dead. For establishing soundness, this set X would be the set of all activities of P . In the online banking example, X would consist only of activity “credit approved”. We achieve this goal by extending the existing operating guideline approach with deadlock-free interaction.

Another important task is to decide whether or not service P can be substituted by a service P' . Obviously, this is the case if no partner R of P can distinguish between P and P' . As our substitutability notion compares the infinite sets of all partners R for P and P' , we present a decision algorithm that applies the (extended) concept of an operating guideline as a finite representation of these sets of services.

This paper is an extended version of [7] where the notion of an extended operating guideline has been introduced. In this paper, we extend the results of [7] by providing experimental results for computing extended operating guidelines and an algorithm to decide substitutability.

The paper is structured as follows. In Sect. 2 we recall open nets and operating guidelines. Next, in Sect. 3, we extend our notion of partners R for P to those partners R' where a certain set of places and transitions in $P \oplus R'$ is covered (i.e. each place can be marked and each transition is not dead). We show how to calculate a finite representation of all these partners by extending our notion of an operating guideline with a global constraint. An algorithm to decide substitutability for two services P and P' on the level of their operating guidelines with global constraint, is developed in Sect. 4. Section 5 presents related work and finally conclusions are drawn in Sect. 6.

2. Preliminaries

2.1. Open Nets

We assume the usual definition of a (place/transition) Petri net $N = [P, T, F]$ (see [8], for instance) and use standard notation to denote the preset and postset of a place or a transition: $\bullet x = \{y \mid [y, x] \in F\}$ and $x^\bullet = \{y \mid [x, y] \in F\}$.

Definition 1 (Open net). An *open net* $N = [P, T, F, I, O, m_0, \Omega]$ consists of a Petri net $[P, T, F]$ together with an *interface* defined as two disjoint sets $I \subseteq P$ of *input places* such that $\bullet p = \emptyset$ for any $p \in I$ and $O \subseteq P$ of *output places* such that $p^\bullet = \emptyset$ for any $p \in O$, a distinguished *initial marking* m_0 , and a set Ω of *final markings* such that no transition of N is enabled at any $m \in \Omega$.

We further require that in the initial and the final markings the interface places are not marked, i.e. $m \in \Omega \cup \{m_0\}$ implies $m(p) = 0$, for all $p \in I \cup O$.

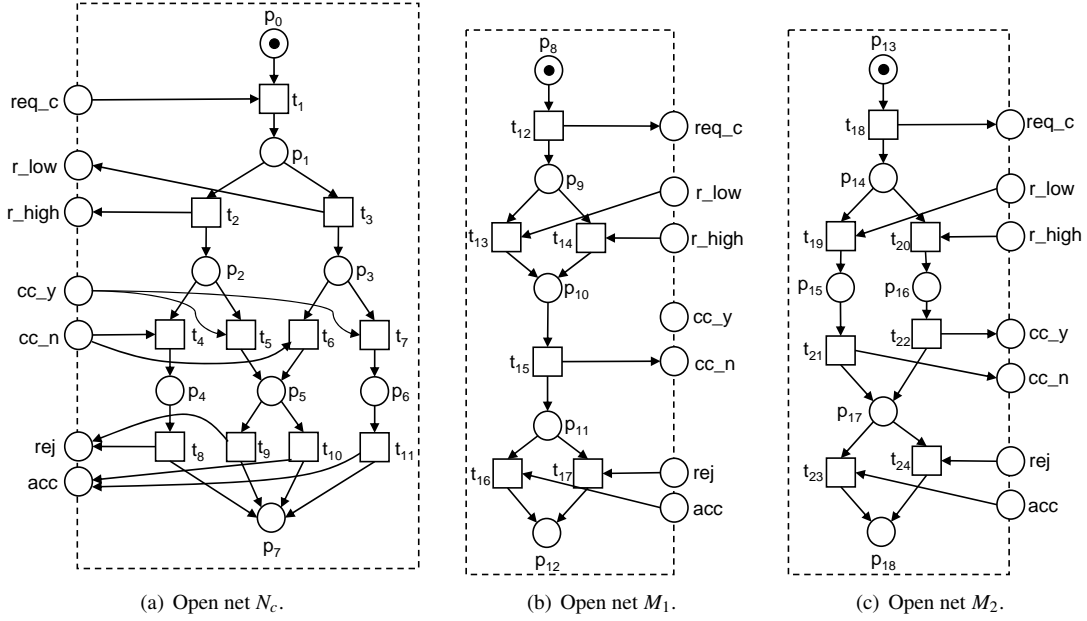


Figure 1: Credit approval process N_c and two strategies M_1 and M_2 .

We use indices to distinguish the constituents of different open nets (e. g. I_N refers to the set of input places of open net N).

The idea of extending Petri nets by an interface for asynchronous communication with an environment bases on the module concept for Petri nets which was proposed by Kindler [9].

The behavior of an open net is defined using the standard Petri net semantics [8], that is, a transition is enabled if each place of its preset holds a token. An enabled transition t can fire in a marking m by consuming tokens from the preset places and producing tokens on the postset places, yielding a marking m' . The firing of t is denoted by $m \xrightarrow{t} m'$ (a t -step), the transitive firing of a sequence of transitions is denoted by $m \xrightarrow{*} m'$. If $m = m_0$, then $m \xrightarrow{*} m'$ is a *run*.

In order to assign a reasonable meaning to *final* markings, we restrict our approach to such open nets where a marking in Ω does not enable any transition.

As an example, consider the open net N_c depicted in Fig. 1(a). The initial marking is $m_{0N_c} = [p_0]$ and the set of final markings is defined by $\Omega_{N_c} = \{[p_7]\}$. N_c has three input and four output places that are depicted on the dashed frame: $I_{N_c} = \{\text{req_c}, \text{cc_y}, \text{cc_n}\}$ and $O_{N_c} = \{\text{r_low}, \text{r_high}, \text{rej}, \text{acc}\}$. The open net models a credit approval process of an online banking service. After the customer has requested a credit (t_1), the bank decides whether the risk is high or low (t_2 and t_3). Then, the customer has to decide whether he accepts a credit control or not ($t_4 - t_7$). Based on this information the bank distinguishes three cases: If the risk is high and the customer does not accept a credit control, then the credit request is rejected (t_8). If there is only low risk and the customer accepts a credit control, then the request is accepted (t_{11}). In the third case, that is, if the risk is high and the customer accepts a credit control or the risk is low but the customer does not accept a credit control, the request is examined by an employee of the bank which is modeled by a nondeterministic choice (t_9 and t_{10}).

The $inner_N$ of an open net N defines the Petri net that results from removing the interface places and the adjacent arcs from N . Obviously, $inner_N$ and N coincide if N has an empty interface. The inner of N_c , $inner_{N_c}$, is the net inside the dashed frame in Fig. 1(a).

As a correctness criterion for an open net N we require the absence of deadlocks in N .

Definition 2 (Deadlock). Let N be an open net. A *deadlock* is a nonfinal marking in N that does not enable a transition. N is *deadlock-free* if no deadlock is reachable from the initial marking of N .

This definition of a deadlock differs from the standard definition in literature as we discriminate between terminating (final) states and non-terminating states (i.e. deadlocks).

Let M and N be open nets with pairwise disjoint constituents, except for the interfaces. This can be achieved easily by renaming. Then, M and N are *composable* if the input places of M are the output places of N and vice versa (i.e. $I_M = O_N$ and $O_M = I_N$). So a composition of open nets is an open net with empty interface. For markings $m_M \in M, m_N \in N$, their composition $m = m_M \oplus m_N$ is defined by $(m_M \oplus m_N)(p) = m_M(p) + m_N(p)$ (assuming $m_M(p) = 0$ for $p \notin P_M$ and $m_N(p) = 0$ for $p \notin P_N$). These considerations lead to the following definition of composition.

Definition 3 (Composition of open nets). Let M, N be composable open nets. Then, the *composition* of M and N is the open net $M \oplus N$ defined by $P = P_M \cup P_N, T = T_M \cup T_N, F = F_M \cup F_N, I = O = \emptyset, m_0 = m_{0M} \oplus m_{0N},$ and $\Omega = \{m_M \oplus m_N \mid m_M \in \Omega_M, m_N \in \Omega_N\}$.

Consider the two open nets M_1 and M_2 depicted in Fig. 1(b) and Fig. 1(c), respectively and assume $m_{0M_1} = [p_8], \Omega_{M_1} = \{[p_{12}]\}, m_{0M_2} = [p_{13}],$ and $\Omega_{M_2} = \{[p_{18}]\}$. Then, N_c and M_1 as well as N_c and M_2 are composable. Notice that place cc_y becomes internal in the composition $N_c \oplus M_1$, but it is never marked.

Clearly, we are mostly interested in composing open nets such that the composition is deadlock-free. To this end, we define the notion of a strategy.

Definition 4 (Strategy). An open net M is a *strategy* for an open net N if $M \oplus N$ is deadlock-free. $Strat(N)$ denotes the set of all strategies for N .

Both, $M_1 \oplus N_c$ and $M_2 \oplus N_c$, are deadlock-free and thus, M_1 and M_2 are strategies for N_c .

In order to simplify presentation, we assume that each transition of an open net is connected to at most one interface place. This assumption does, however, not restrict generality as every open net can be transformed into an equivalent one (w.r.t. *Strat*) that obeys this restriction [5].

2.2. Operating Guidelines

In the following we recapitulate our concept of an *operating guideline* [10, 5]. With the help of an operating guideline we are able to represent the set of all strategies M for an open net N in a compact way. Technically, an operating guideline is a special annotated automaton. An annotated automaton A^Φ consists of a finite deterministic service automaton A and a function Φ that assigns to each state q of A a Boolean formula $\Phi(q)$. A^Φ represents a set $Match(A^\Phi)$ of open nets and for each element $M \in Match(A^\Phi)$, we say that M *matches* with A^Φ . More precisely, A^Φ does not represent M but the *behavior* of M , that is, the reachability graph of $inner_M$ which can be also represented by a service automaton.

Definition 5 (Service automaton). $A = [Q, C, \delta, q_0, \Omega]$ is a *service automaton* iff Q is a nonempty finite set of *states*, C is a set of *labels*, $\delta \subseteq Q \times C \times Q$ is a *transition relation* such that every state $q \in Q$ is reachable from q_0 via transitive applications of δ , $q_0 \in Q$ is the *initial state*, and Ω is a set of *final states*.

A is *deterministic* if it has no internal (i.e. τ labeled) transitions and each state has at most one x -labeled outgoing transition.

In order to represent the behavior of an open net M by a service automaton, let Q, δ, q_0, Ω be the set of reachable markings, the transition relation, the initial marking and the set of final markings of $inner_M$, respectively. In addition, choose $C = I \cup O \cup \{\tau\}$.

Definition 6 (Annotated automaton). An *annotated automaton* $A^\Phi = [Q, C, \delta, q_0, \Omega, \Phi]$ consists of a deterministic service automaton $A = [Q, C, \delta, q_0, \Omega]$ and an *annotation function* Φ , where, for all $q \in Q$, $\Phi(q)$ is a Boolean formula over literals in $C \cup \{final\}$.

We use annotated automata to represent (the behavior of) a set of *open nets*. Therefore, we take an annotated automaton A^Φ with Boolean formulae over literals in C and a special literal *final* and define when a service described in terms of an open net M with the interface $I \cup O$ matches with A^Φ . Intuitively, M matches with A^Φ if (1) its *behavior* is simulated by A^Φ and (2) if a marking m of M is simulated by a state q of A^Φ , then the arcs leaving m — interpreted

as an assignment assigning *true* to the corresponding literals of the formula $\Phi(q)$ — satisfy $\Phi(q)$. For more details, we refer to [11, 5].

We continue by defining a weak simulation relation [12] between two service automata. Then we introduce the minimal weak simulation relation which we need to define matching of an open net with an annotated automaton.

Definition 7 (Weak simulation). Let P and R be service automata and let \hat{a} stand for τ^* if $a \in C$ is τ , and a otherwise. A binary relation $\varrho_{P,R} \subseteq Q_P \times Q_R$ is a *weak simulation relation* of P by R iff if $[q_P, q_R] \in \varrho_{P,R}$ and there is a transition $[q_P, a, q'_P] \in \delta_P$ in P , then there is a transition $[q_R, \hat{a}, q'_R] \in \delta_R$ in R and $[q'_P, q'_R] \in \varrho_{P,R}$.

R *weakly simulates* P if there is a weak simulation relation $\varrho_{P,R}$ of P by R such that $[q_{0_P}, q_{0_R}] \in \varrho_{P,R}$.

If P and R do not contain τ transitions, then $\varrho_{P,R}$ is a *simulation relation*. There may exist several (weak) simulation relations between two service automata. Throughout this paper, we shall *always* confine to a particular one that we call the *minimal* (weak) simulation relation. It is well defined for the case where the second service automaton is deterministic.

Definition 8 (Minimal (weak) simulation). Let P and R be service automata, and R be deterministic. Let R (weakly) simulate P . The *minimal (weak) simulation relation*, $\varrho_{P,R}$, of P by R is defined inductively by

Base: $[q_{0_P}, q_{0_R}] \in \varrho_{P,R}$.

Step: If $[q_P, q_R] \in \varrho_{P,R}$ and $q_P \xrightarrow{a} q'_P$, then $[q'_P, q'_R] \in \varrho_{P,R}$ for the q'_R holding $q_R \xrightarrow{a} q'_R$.

Note that q'_R exists since R (weakly) simulates P and is uniquely determined since R is deterministic.

Proposition 1 (Facts about $\varrho_{P,R}$). 1. $\varrho_{P,R}$ is a (weak) simulation relation.

2. For every (weak) simulation relation $\varrho \subseteq Q_P \times Q_R$, $\varrho_{P,R} \subseteq \varrho$.

The first fact holds as we define $\varrho_{P,R}$ along the lines of the definition of simulation. The second fact is easy to verify as we insert elements to $\varrho_{P,R}$ only if they are ultimately required in order to establish a (weak) simulation relation.

Now we can define matching of an open net M with A^Φ . We could do so by considering the behavior of M described as a service automaton. Instead, we define matching of the reachability graph of $inner_N$ with A^Φ because this makes the technicalities and proofs in the next section easier to understand.

Definition 9 (Matching with A^Φ). Let M be an open net and let Y be the set of all reachable markings of the Petri net $M^* = inner_M$. Let $A^\Phi = [Q, C, \delta, q_0, \Omega, \Phi]$ be an annotated automaton with $C = I_M \cup O_M \cup \{final\}$. Then M *matches* with A^Φ iff there is a weak simulation relation $\varrho \subseteq Y \times Q$ of M^* by A^Φ inductively defined by

1. $[m_{0_M}, q_0] \in \varrho$;
2. If t is an internal transition of M (i.e. t is not connected to any interface place), $m, m' \in Y$, and $m \xrightarrow{t} m'$, then $[m, q] \in \varrho$ implies $[m', q] \in \varrho$;
3. If t is a receiving transition of M with $c \in I_M$, $c \in \bullet t$, $m, m' \in Y$, and $[m + [c]] \xrightarrow{t} m'$, then $[m, q] \in \varrho$ implies $[m', q'] \in \varrho$ for some q' with $[q, c, q'] \in \delta$;
4. If t is a sending transition of M with $c \in O_M$, $c \in t^\bullet$, $m, m' \in Y$, and $m \xrightarrow{t} [m' + [c]]$, then $[m, q] \in \varrho$ implies $[m', q'] \in \varrho$ for some q' with $[q, c, q'] \in \delta$;
5. For all $m \in Y$, at least one of the following properties holds:
 - An internal transition t is enabled at m ; or,
 - for all q such that $[m, q] \in \varrho$, $\Phi(q)$ evaluates to *true* for the following assignment β :
 - $\beta(c) = true$ if $c \in O_M$ and there is a transition t with $c \in t^\bullet$ that is enabled at m ;
 - $\beta(c) = true$ if $c \in I_M$ and there is a transition t with $c \in \bullet t$ that is enabled at $[m + [c]]$;
 - $\beta(c) = true$ if $c = final$ and $m \in \Omega_M$;
 - $\beta(c) = false$, otherwise.

Let $Match(A^\Phi)$ denote the set of all M such that M matches A^Φ .

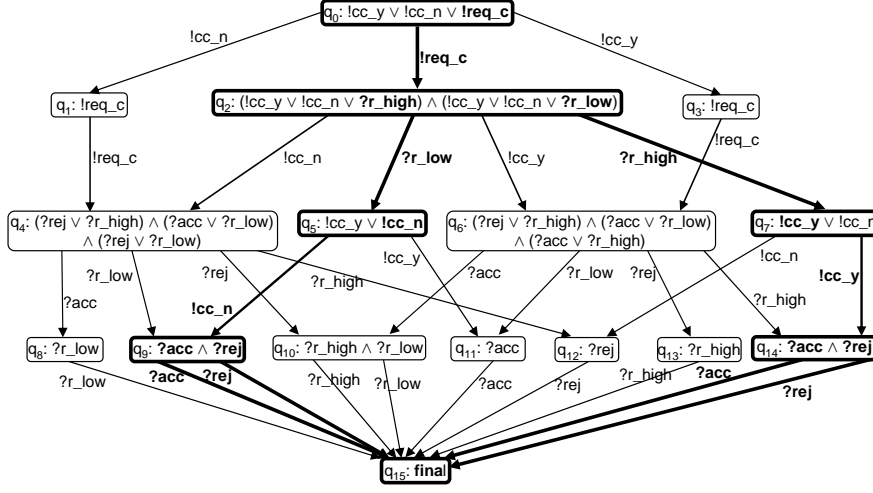


Figure 2: The operating guideline OG_{N_c} of the credit approval process N_c depicted in Fig. 1(a). For better readability, we add a leading “!” (“?”) to a literal x in the graphics of an OG_N if x is an output (input) place of a strategy M for N . Bold font illustrates the matching relation between $inner_{M_2}$ and OG_{N_c} .

In the formal definition, the assignment β used for evaluating an annotation represents transitions t of M that leave the considered marking m of M^* .

An operating guideline OG_N of an open net N is a special annotated automaton, such that an open net M matches with OG_N if and only if M is a strategy for N .

Definition 10 (Operating guideline). An annotated automaton is an *operating guideline* OG_N of an open net N iff $Strat(N) = Match(OG_N)$.

Figure 2 depicts the operating guideline OG_{N_c} for the credit approval process N_c in Fig. 1(a). It consists of 16 nodes and 31 edges and was calculated by our tool Fiona [13]. In the initial state q_0 , the annotation is $!lcc_y \vee !lcc_n \vee !req_c$ reflecting the possible choices of a strategy M for N_c . More precisely, M must be able to send at least one (expressed by the disjunction) of the three messages cc_y , cc_n , and req_c in its initial state. In contrast, annotation $?acc \wedge ?rej$ in state q_{14} reflects the fact that M being in marking m with $[m, q_{14}] \in \mathcal{Q}$ must be able to receive message acc and message rej . The two open nets M_1 and M_2 fulfil the requirements of Definition 9 and thus match with OG_{N_c} . For M_2 , this is illustrated in Fig. 2 using bold font.

3. Covering Open Net Nodes

The notion of soundness guarantees (among others) the absence of dead transitions in a workflow net. In this section, this idea is adapted to open nets. For an open net N and a set $X \subseteq P_N \cup T_N$ of open net nodes, we will characterize strategies M for N with X is *covered* in the composition $M \oplus N$. Here, to cover a place p means that p can be marked in some reachable marking while to cover a transition t means that t is not dead. Such a strategy M is then called a *Cover_X-strategy* for N . Clearly, if X contains all transitions of N , our coverability notion for open nets coincides with soundness, except for the fact that the composition may contain livelocks.

The motivation for dealing with *Cover_X-strategies* is to figure out if some functionality of a service (i.e. some communication patterns), for example a credit approval, can in principle be used by other services. We further show how to calculate a finite representation of all *Cover_X-strategies* for N by extending operating guidelines with a global constraint.

3.1. Deciding Coverability of Open Net Nodes

In this section, we show how a strategy M for N can be discovered as a *Cover_X-strategy* by just considering the operating guideline of N . In order to define our notion of *Cover_X-strategies*, we need to define what it means to cover an open net node.

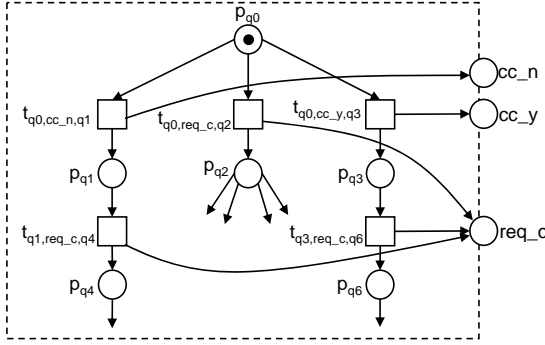


Figure 3: The initial part of the most permissive strategy MP_{N_c} for N_c which has been constructed according to Definition 13.

Definition 11 (Cover a place/transition). Let $N = [P, T, F, I, O, m_0, \Omega]$ be a deadlock-free open net with empty interface ($I = O = \emptyset$), and let $X \subseteq P \cup T$, $p \in P$, and $t \in T$. N covers X iff for all $p \in X \cap P$ (for all $t \in X \cap T$) there exists a run of N that includes a marking m with $m(p) \geq 1$ (a t -step).

Notice that if N covers two nodes, there is not necessarily a run in which both nodes are covered. In the example, transitions $t_1 - t_4$, t_6 , and $t_8 - t_{10}$ are covered in $M_1 \oplus N_c$ and transitions $t_1 - t_3$, t_5 , t_6 , t_9 , and t_{10} are covered in $M_2 \oplus N_c$. The following definition canonically extends strategies to strategies that cover a set X of open net nodes.

Definition 12 ($Cover_X$ -strategy). Let M be a strategy for an open net N , and let $X \subseteq P_N \cup T_N$. M is a $Cover_X$ -strategy for N iff X is covered in $M \oplus N$. With $Strat_{Cover_X}(N)$ we denote the set of all $Cover_X$ -strategies for N .

For N_c let $X = \{acc\}$ be given. That means, we are interested whether a credit approval is possible. Then, M_1 and M_2 are $Cover_X$ -strategies for N_c . Let $X = \{t_5, t_6\}$, that is, we are interested whether it is possible that a credit request has to be examined by an employee if the customer is not fixed in his credit control decision. Then M_2 is a $Cover_X$ -strategy for N_c , but M_1 is not (because transition t_5 cannot be enabled in $M_1 \oplus N$).

By Definition 12, every $Cover_X$ -strategy for N is also a strategy for N . Obviously, covering open net nodes restricts the set of strategies for N . Thus, we conclude $Strat_{Cover_X}(N) \subseteq Strat(N)$.

In the remainder of this section, we will define some notions and prove some properties of operating guidelines. Based on these properties, we can prove a criterion to decide whether an open net M is a $Cover_X$ -strategy for N . We start with the definition of the most permissive strategy for N . This strategy has the least restrictions of all strategies. Thus, the state space of its inner corresponds exactly to the transition system of the underlying automaton of OG_N .

Definition 13 (Most permissive strategy). Let $OG_N = [Q, C, \delta, q_0, \Omega, \Phi]$. The most permissive strategy for N is the open net $MP_N = [P, T, F, I, O, m_0, \Omega]$ whose behavior corresponds exactly to the transition system $[Q, C, \delta, q_0, \Omega]$ with $P = Q \cup C$, $T = \{t_{q_1,c,q_2} \mid [q_1, c, q_2] \in \delta, \text{ with } q_1, q_2 \in Q, c \in C\}$,

$$F = \{[q_1, t_{q_1,c,q_2}], [t_{q_1,c,q_2}, q_2] \mid [q_1, c, q_2] \in \delta\} \cup \begin{cases} [c, t_{q_1,c,q_2}], & \text{if } c \in I; \\ [t_{q_1,c,q_2}, c], & \text{if } c \in O. \end{cases}$$

$I = O_N$, $O = I_N$, and $m_0 = q_0$.

The resulting open net MP is a state machine. Figure 3 illustrates the construction of the most permissive strategy MP_{N_c} of the operating guideline OG_{N_c} depicted in Fig. 2.

Using the following corollary, we prove that the most permissive strategy MP for N is indeed a strategy for N .

Corollary 1. *The most permissive strategy MP for N is a strategy for N .*

For the proof of this corollary, we rely on a fact about operating guidelines as constructed in [5].

Proposition 2 ([5]). *For every operating guideline $OG = [Q, C, \delta, q_0, \Omega, \Phi]$ (of some service N) and all $q \in Q$, the formula $\Phi(q)$*

1. *uses only literals c where there is some $q' \in Q$ with $[q, c, q'] \in \delta$, and*

2. is satisfied for the assignment assigning true to all literals in $\Phi(q)$.

PROOF (OF COROLLARY 1). Let $OG = [Q, C, \delta, q_0, \Omega, \Phi]$. We construct open net MP as described in Definition 13. Let m_{q_0} be the initial marking of MP . By induction, it can be shown that, for all $q \in Q$, m_q is reached by Definition 9, with $[m_q, q] \in \mathcal{Q}_{MP, OG}$.

As there is a transition for each $[q, c, q'] \in \delta$, we can derive from Proposition 2 that all annotations evaluate to true when MP is evaluated according to Definition 9. So MP matches with OG_N and thus MP is a strategy for N . \square

The next definition establishes a connection between markings of an open net N and the inner of a strategy $M \in \text{Strat}(N)$. If $inner_M$ is in a marking m , then $K(m)$ (the knowledge that $inner_M$ has about N) is the set of markings of N that N might be in while $inner_M$ is in marking m .

Definition 14 (Knowledge). Let M be a strategy for an open net N . Let $Mark_{M^*}$ and $Mark_N$ denote the set of all reachable markings of $inner_M$ and N , respectively. Let further m_M denote a marking of M and m_{M^*} denote its restriction to places in $inner_M$. The knowledge $K : Mark_{M^*} \rightarrow \mathcal{P}(Mark_N)$ that $inner_{MP}$ has about the possible markings of N in marking m_{M^*} is defined by $K(m_{M^*}) = \{m_N \mid (m_M \oplus m_N) \text{ is reachable from } (m_{0M} \oplus m_{0N})\}$.

As an example, for the most permissive strategy MP_{N_c} for N_c (see Fig. 3), we have the following knowledge values: $K([p_{q0}]) = \{[p_0]\}$, $K([p_{q1}]) = \{[p_0, cc_n]\}$, $K([p_{q2}]) = \{[p_0, req_c], [p_1], [p_2, r_high], [p_3, r_low]\}$, $K([p_{q3}]) = \{[p_0, cc_y]\}$, $K([p_{q4}]) = \{[p_0, cc_n, req_c], [p_1, cc_n], [p_2, cc_n, r_high], [p_3, cc_n, r_low], [p_4, r_high], [p_5, r_low], [p_7, r_high, rej], [p_7, r_low, acc], [p_7, r_low, rej]\}$.

The weak simulation relation ϱ used in Definition 9 actually establishes a relation between the knowledge values of the involved states. As the following proposition states, $[m, q] \in \varrho$ implies that $K(m) \supseteq K(m_q)$, where m_q is the marking in the most permissive strategy that corresponds to state q of an operating guideline.

Proposition 3 ([5]). Let M be a strategy for N and MP be the most permissive strategy for N . Let m_q denote the marking in $inner_{MP}$ that corresponds to state $q \in Q$ in OG_N (i.e. $[m_q, q] \in \mathcal{Q}_{MP, OG_N}$). Let m be reachable in $inner_M$. Then $K(m) = \bigcup_{q: [m, q] \in \mathcal{Q}_{MP, OG_N}} K(m_q)$.

The matching relation ϱ_{M, OG_N} relates a marking m of $inner_M$ to a (possible) set of states q of OG_N . Therefore, the knowledge that $inner_M$ has about the possible markings of N in m is equivalent to the union of the knowledge values of all markings m_q of $inner_{MP}$ with $[m_q, q] \in \mathcal{Q}_{MP, OG_N}$.

The notion of knowledge can be applied to the operating guideline OG_N of N . As every marking m_q in $inner_{MP}$ corresponds to a state q of OG_N , the knowledge OG_N has about N in q is equivalent to the knowledge $inner_{MP}$ has about N in m_q .

Definition 15 (Knowledge in OG). For an open net N let MP be the most permissive strategy for N and $OG_N = [Q, C, \delta, q_0, \Omega, \Phi]$. Let $Mark_N$ denote the set of markings of N and m_q be a marking of $inner_{MP}$. The knowledge $K : Q \rightarrow \mathcal{P}(Mark_N)$ that OG_N has about the possible markings of N in state $q \in Q$ is defined by $K(q) = K(m_q)$.

The following theorem presents a way to decide, on the basis of an operating guideline, whether a strategy M for N is also a $Cover_X$ -strategy for N .

Theorem 1 (Place/Transition coverability). Let M be a strategy for open net N . A place $p \in P_N$ (a transition $t \in T_N$) is covered in $M \oplus N$ iff there is a state $q \in Q$ of OG_N , a marking m_M in $inner_M$, and a marking $m_N \in K(q)$ with $[m_M, q] \in \mathcal{Q}_{M, OG_N}$, and $m_N(p) \geq 1$ (t is enabled in m_N).

PROOF. We present the proof for the case of a covered transition only. The case of a covered place is analogous.

(\Rightarrow) Let N , OG_N , and $M \in \text{Strat}(N)$ be given and let transition t be covered in $M \oplus N$. Then, according to Definition 11, there is a run $m_{0M \oplus N} \xrightarrow{t_1} \dots \xrightarrow{t_n} m_{M \oplus N} \xrightarrow{t} m'_{M \oplus N}$ in $M \oplus N$, $m'_{M \oplus N}(p) \geq 1$. Let m_M and m_N be the restrictions of marking $m_{M \oplus N}$ to places in $inner_M$ and N , respectively. As t is a transition of N , t is enabled in m_N as well. By Definition 14, we have $m_N \in K(m_M)$. By Proposition 3, there must be a state q in OG_N where $m_N \in K(q)$ and hence the implication of this theorem holds.

(\Leftarrow) Let N be an open net and $OG_N = [Q, C, \delta, q_0, \Phi]$. Let M be a strategy for N . Since M is a strategy for N , there is a weak simulation relation ϱ_{M, OG_N} of markings in $inner_M$ by states in Q . Let m_M, q , and m_N be as assumed. Thus, $[m_M, q] \in \varrho_{M, OG_N}$, $m_N \in K(q)$, and t is enabled in m_N . From Proposition 3 follows $m_N \in K(m_M)$. Consequently, there is a run in $M \oplus N$ that reaches $m_M \oplus m_N$ which can be extended by an occurrence of t since activation of t in m_N implies activation of t in $m_M \oplus m_N$. Since every run in $M \oplus N$ is deadlock-free (follows from M being a strategy for N), we can conclude that the considered run is deadlock-free, too. So there exists a deadlock-free run in $M \oplus N$ where t is covered and hence the replication of this theorem holds. \square

The value of Theorem 1 is that it gives us a criterion to check whether an open net node is covered or not. A place p of N is covered by a strategy for N if there is a state q in OG_N and the knowledge in q contains a marking of N where p is marked. A transition t of N is covered by a strategy for N if there is a state q and the knowledge in q contains a marking m of N where t is enabled. That way, it is easily possible to annotate each state q of OG_N with all places and transitions which are covered in q . This can be done during the calculation of the operating guideline.

As an example, based on the knowledge values $K([p_{q_0}]) - K([p_{q_4}])$ we presented above we can derive the following sets of nodes of N_c that are covered in states $q_0 - q_4$ of OG_{N_c} : $q_0 : \{p_0\}$; $q_1 : \{p_0, cc_n\}$; $q_2 : \{p_0 - p_3, req_c, r_high, r_low, t_1 - t_3\}$; $q_3 : \{p_0, cc_y\}$; $q_4 : \{p_0 - p_5, p_7, cc_n, cc_y, req_c, r_high, r_low, acc, rej, t_1 - t_4, t_6, t_8 - t_{10}\}$.

3.2. A Finite Representation of all $Cover_X$ -Strategies

In this section, we introduce a notion of an operating guideline with a global constraint as a finite representation of all $Cover_X$ -strategies for N . We further present an algorithm for deciding whether an open net M matches with such an operating guideline.

Consider again our running example N_c in Fig. 1(a). Suppose we want to cover $X = \{acc\}$ in N_c . We have $[acc] \in K(q_4), K(q_6), K(q_9), K(q_{11}), K(q_{14})$. So according to Theorem 1, a strategy M for N_c is a $Cover_X$ -strategy for N_c if it has at least a marking m_{acc} of $inner_M$ that matches with q_4, q_6, q_9, q_{11} , or q_{14} . As a second example assume $X = \{t_5, t_6\}$. In that case we have $[t_5] \in K(q_6), K(q_{14})$ and $[t_6] \in K(q_4), K(q_9)$. So M is a $Cover_X$ -strategy for N_c if it has at least a marking m_{t_5} of $inner_M$ that matches with q_6 or q_{14} and it has a marking m_{t_6} of $inner_M$ that matches with q_4 or q_9 .

The examples illustrate that is in general not possible to express the constraints for covering open net nodes in the shape of local annotations in each state of the operating guideline. Consequently, the present concept of an annotated automaton fails at representing all $Cover_X$ -strategies for N . To overcome this problem, we propose another representation of all $Cover_X$ -strategies for N that takes the non-locality of covering open net nodes into account. To this end, we will slightly enhance the concept of an operating guideline.

Consider again the example above. Since OG_{N_c} (see Fig. 2) represents all strategies and every $Cover_X$ -strategy for N_c is a strategy for N_c , we have to restrict OG_{N_c} to $Cover_X$ -strategies. This can be achieved by a *global constraint* specifying that, for every open net node $x \in X$ to be covered, at least one state q in OG_{N_c} with $x \in K(q)$ must be present in the matching relation between OG_{N_c} and a $Cover_X$ -strategy. This constraint can be expressed as a Boolean formula ψ_X .

In the following, we formalize annotated automata enhanced with a global constraint and define the matching relation between an open net and such an annotated automaton.

Definition 16 (Annotated automaton with global constraint). Let $A^\Phi = [Q, C, \delta, q_0, \Omega, \Phi]$ be an annotated automaton and ψ be a Boolean formula with propositions taken from the set Q . Then, $A^{\Phi, \psi} = [A^\Phi, \psi]$ is an *annotated automaton with global constraint* ψ .

As an example for a global constraint to OG_{N_c} , consider $\psi = (q_6 \vee q_{14}) \wedge (q_4 \vee q_9)$. This formula is satisfied if and only if true is assigned to sufficiently many states to cover the set $X = \{t_5, t_6\}$.

Enhancing an annotated automaton A^Φ with a global constraint ψ makes it necessary to redefine the matching relation of an open net M with an annotated automaton. M matches with an annotated automaton with global constraint $A^{\Phi, \psi}$ if it matches with A^Φ , and in addition satisfies ψ .

Definition 17 (Matching with $A^{\Phi, \psi}$). Let M be an open net, and let $A^{\Phi, \psi}$ be an annotated automaton A^Φ with global constraint ψ . M matches with $A^{\Phi, \psi}$ iff M matches with A^Φ using relation ϱ and ψ evaluates to true in the assignment $\gamma_M : Q_A \rightarrow \{true, false\}$ where $\gamma_M(q) = true$ iff there is a marking m of M such that $[m, q] \in \varrho$.

Finally, we are ready to construct the operating guideline with global constraint $OG_{\psi_X}(N)$ of an open net N as a representation of the set $Strat_{Cover_X}(N)$ of all $Cover_X$ -strategies for N .

Definition 18 (Global constraint for covering X). Let N be an open net and OG_N an operating guideline of N . Let $X \subseteq P_N \cup T_N$. For a place $p \in P$, let $p \sim q$ iff there is an $m \in K(q)$ where $m(p) > 0$. For a transition $t \in T$, let $t \sim q$ iff there is an $m \in K(q)$ where t is enabled. Then $\psi_X = \bigwedge_{x \in X} \bigvee_{q: x \sim q} q$.
 $OG_{\psi_X}(N) = [OG_N, \psi_X]$ defines an *operating guideline with global constraint* of N .

As a direct consequence of Theorem 1, we obtain the main result of this section; that is, $OG_{\psi_X}(N)$ represents all $Cover_X$ -strategies for N .

Theorem 2. M is a $Cover_X$ -strategy for N iff M matches with $OG_{\psi_X}(N)$.

PROOF. (\Rightarrow) Let M be a $Cover_X$ -strategy for N and let $OG_{\psi_X}(N)$ be an operating guideline with global constraint of N . We show M matches with $OG_{\psi_X}(N)$.

Since M is a $Cover_X$ -strategy for N , M is a strategy for N and thus matches with OG_N (i.e. $OG_{\psi_X}(N)$ without constraint ψ_X). Furthermore, every $x \in X$ is covered in $M \oplus N$ (by Definition 12). From Theorem 1 follows that, for all $x \in X$, there is a state $q \in Q$ of OG_N , a marking m_M in $inner_M$, and a marking $m_N \in K(q)$ with $[m_M, q] \in \mathcal{Q}_{M, OG_N}$, and x is marked/enabled in m_N . Thus, for each disjunction of ψ_X , γ_M assigns true to at least one state $q \in Q$. Consequently, M satisfies ψ_X and we conclude from Definition 17, M matches with $OG_{\psi_X}(N)$.

(\Leftarrow) Let M match with $OG_{\psi_X}(N)$. We show M is a $Cover_X$ -strategy for N .

Since M matches with $OG_{\psi_X}(N)$ we know by Definition 17 that M matches with OG_N . Thus, M is a strategy for N . Furthermore, M satisfies ψ_X (follows also from Definition 17). So we conclude, for each disjunction of ψ_X , γ_M assigns true to at least one state $q \in Q$. So for all $x \in X$, there is a state q with $x \sim q$. By Theorem 1 we conclude that all $x \in X$ are covered in $M \oplus N$ and therefore M is a $Cover_X$ -strategy for N . \square

The operating guideline representing all $Cover_X$ -strategies for N_c with $X = \{t_5, t_6\}$ is the operating guideline $OG_{\psi_X}(N_c) = [OG_{N_c}, \psi_X]$ where $\psi = (q_6 \vee q_{14}) \wedge (q_4 \vee q_9)$ as stated above. If we consider again open nets M_1 and M_2 (which are both strategies for N_c), then we get that M_2 matches with $OG_{\psi_X}(N_c)$ and it is hence a $Cover_X$ -strategy for N_c . In contrast, M_1 does not match with $OG_{\psi_X}(N_c)$, because it does not satisfy the global constraint. More precisely, there is no marking in $inner_{M_1}$ that matches with any of the nodes q_6 and q_{14} .

As another example, let $X = \{t_1, \dots, t_{11}\}$, meaning all transitions of N_c should not be dead in $M \oplus N$. Then, $OG_{\psi_X}(N_c)$ has (after minimization) the global constraint $\psi_X = (q_2 \vee q_4 \vee q_6) \wedge (q_4 \vee q_{12}) \wedge (q_6 \vee q_{11}) \wedge (q_4 \vee q_6 \vee q_9 \vee q_{14})$.

3.3. Case Study

The results presented in this section have been implemented in our analysis tool Fiona² [13]. Among others Fiona can be used to read an open net, calculate its operating guideline (with global constraint) and check whether an open net matches with an operating guideline (with global constraint).

Table 1 provides the results of a small case study including seven services (specified as open nets). “Purchase Order” and “Travel Service 1” are realistic services taken from the WS-BPEL specification [14]. “Travel Service 2” is a modification of “Travel Service 1”. “Olive Oil Ordering” [15] and “Help Desk Service Request” (from the Oracle BPEL Process Manager) are also WS-BPEL processes. The “Beverage Machine” is taken from [5] and “Book Shop” is a modified process provided by a German consulting company. The WS-BPEL processes have been translated into open nets using the compiler BPEL2oWFN³ [13].

For each service, we calculated its operating guideline with global constraint. Thereby we covered all open net nodes. Table 1 provides the size of the open net (number of places, input places, output places and transitions), the size of the operating guideline (number of nodes and edges), the size of the global constraint (number of disjunctions—the global constraint is a conjunction of disjunctions—and number of literals appearing in the constraint), and the time to calculate the respective operating guideline with global constraint.

²available at <http://www.service-technology.org/fiona>

³available at <http://www.service-technology.org/bpel2owfn>

Table 1: Experimental results running Fiona. All experiments were taken on a Intel Pentium M processor with 1.6 GHz and 1 GB RAM running Windows XP.

Service	Open net				OG		Constraint		time (sec)
	$ P $	$ I $	$ O $	$ T $	$ Q $	$ \delta $	$ \vee $	$ \psi $	
Purchase Order	22	4	6	7	168	548	4	103	3.5
Travel Service 1	15	4	4	5	56	140	3	24	0.6
Travel Service 2	22	6	6	9	288	1 008	5	160	17.6
Help Desk Service Request	15	4	4	8	16	30	4	9	0.3
Olive Oil Ordering	12	3	3	6	16	27	4	9	0.1
Beverage Machine	12	4	3	8	11	24	3	7	0.1
Book Shop	23	6	6	11	88	228	5	40	8.4

The case study illustrates that the operating guidelines of these services were calculated in reasonable time and the size of the global constraints is feasible. Reducing the number of open net nodes to be covered does not affect the above listed computation times but may increase the size of the global constraint (i.e. $|\psi|$) up to a factor of three which is still feasible.

3.4. Discussion

In the following we will compare (ordinary) operating guidelines and operating guidelines with global constraint. We further discuss some complexity issues.

Comparing an operating guideline OG_N for N and an operating guideline with global constraint $OG_{\psi_X}(N)$ for N , we identify that both operating guidelines have the same underlying automaton (i.e. the most permissive strategy). This is caused by the fact that each $Cover_X$ -strategy for N is also a strategy for N . Furthermore, if the most permissive strategy for N is not a $Cover_X$ -strategy for N , then the set of $Cover_X$ -strategies is empty.

Computing OG_N is proportional in time to the product of the number of states of N and an over-approximation of its most permissive strategy [11]. For $OG_{\psi_X}(N)$ the time complexity does not change, because all information necessary for annotating the states $q \in Q$ with the nodes of N and setting up the global constraint have to be computed for OG_N anyway. In order to increase efficiency, it is sufficient to annotate each state q only with open net nodes of X .

The space complexity of OG_N is proportional to the product of the number of states of N and its most permissive strategy [11]. If we compute $OG_{\psi_X}(N)$, then this complexity increases due to ψ_X . The global constraint is a conjunction of at most $|X|$ disjunctions where each disjunction may consist of at most $|Q|$ literals. Hence, the size of the global constraint is at most $O(|X| \cdot |Q|)$. The case study in Sect. 3.3 suggests that the size of ψ_X will be much smaller in practice.

Although the time and space complexity of OG_N is high, experimental results have shown that the calculation of OG_N is feasible in practical applications both for time and space (see [5], for instance). Based on the complexity considerations for $OG_{\psi_X}(N)$ and the outcome of our case study in Sect. 3.3 we conclude that the calculation of $OG_{\psi_X}(N)$ will be feasible in practical applications, too.

Matching an open net M with OG_N is proportional in time to the number of states in $M \oplus N$ [11]. If we match M with $OG_{\psi_X}(N)$, we additionally have to check whether the global constraint is satisfied by the assignment γ_M . This can be done in linear time w.r.t. the size of the constraint.

As the space complexity and the matching complexity for the proposed notion of operating guidelines with global constraint only marginally increase in comparison with ordinary operating guidelines and as a consequence of our case study in Sect. 3.3, we conclude that this novel notion is a well-suited instrument for checking compatibility of services.

4. Verifying Accordance Under Coverability

In the previous section we developed a notion of an operating guideline with global constraint as a finite representation of all $Cover_X$ -strategies for an open net N . Such an operating guideline can support service discovery since matching of an open net with such an operating guideline can be decided rather efficiently. This section is devoted to

another application of operating guidelines with global constraint. Suppose we want to substitute N by another open net N' such that (1) no environment of N is affected and (2) a certain set of activities of N' is covered. We refer to this substitutability notion as *accordance* and show that accordance between N and N' can be decided by the help of their corresponding operating guidelines with global constraint.

Throughout this section we will use service automata in the definitions and proofs instead of open nets because we do not need to consider markings (as in the previous section). The reason is that the information needed to cover open net nodes is already encoded in the matching (Definition 17) and the global constraint (Definition 18). As a benefit, the use of service automata will cause less formalities. Switching between the two formalisms, service automata and open nets, is possible as we do not lose information w.r.t. *Strat* [5, 16].

We continue by recalling known results for accordance. Then we extend these results and establish theoretical results which we need for supporting our algorithms. Finally, we explain the algorithmic solution and establish its correctness.

4.1. Existing Results for Accordance

Given a service automaton P , it might be necessary to change or add some functionality of P by substituting it by a new version P' . With accordance, we demand that this substitution must not affect any client of P : every current client of P has to be supported by P' as well. Because we assume that P does not know each client that uses P , P' must support each *potential* client of P , i.e. all elements in $Strat(P)$. Applications for accordance include the upgrade of a web shop and implementing a public view of a service in the setting of a service contract [17, 3]. This motivates the following notion of accordance between service automata P and P' .

Definition 19 (Accordance [16]). Let P and P' be service automata with equal interfaces (i.e. $C_P = C_{P'}$). P' *substitutes P under accordance* (short: P' accords with P) iff $Strat(P) \subseteq Strat(P')$.

Accordance guarantees that every client of P is also a client of P' . In order to decide accordance of P and P' , we need to compare $Strat(P)$ and $Strat(P')$. The problem is that the set $Strat$ may correspond to an infinite set of open nets. With the operating guidelines of P and P' we have, however, a finite representation of $Strat(P)$ and $Strat(P')$ that can be used to decide accordance.

Theorem 3 (Deciding accordance [16]). Let P and P' be service automata and let OG_P and $OG_{P'}$ be the corresponding operating guidelines. Then, P' accords with P iff

1. There is a simulation relation $\varrho_{OG_P, OG_{P'}}$ of OG_P by $OG_{P'}$.
2. For all $[q_P, q_{P'}] \in \varrho_{OG_P, OG_{P'}}$, the formula $\Phi_{OG_P}(q_P) \Rightarrow \Phi_{OG_{P'}}(q_{P'})$ is a tautology.

The value of this theorem is that accordance can be checked independently of the clients that P cooperates with and only P and P' have to be known to decide accordance. Next, we show how these results can be extended to decide accordance in the context of coverability and extended operating guidelines.

4.2. Theoretical Results for Accordance Under Coverability

In this section, we consider accordance in the context of coverability. Let for the rest of this section be P and P' service automata with equal interfaces and covering sets X and Y . Let $[OG_P, \psi_X]$ and $[OG_{P'}, \psi_Y]$ be the corresponding operating guidelines with global constraint. In this setting, accordance of P and P' should guarantee that every $Cover_X$ -strategy for P is a $Cover_Y$ -strategy for P' as well.

Definition 20 (Accordance under coverability). Let P and P' be as stated above. P' *accords with P under coverability* iff $Strat_{Cover_X}(P) \subseteq Strat_{Cover_Y}(P')$.

In the rest of this section we develop three conditions—similar to those of Theorem 3—that enable us to decide whether P' accords with P under coverability. To this end, we start with taking care of the structure and the local annotations of the operating guidelines with global constraints for P and P' . Then we concern with checking the global constraints.

Structure and local annotations

We show that the same conditions as in Theorem 3—simulation relation and implication of the local annotations—actually hold in presence of global constraints, i.e. the global constraints cannot compensate incompatible local constraints.

Lemma 1 (Structure). *If P' accords with P under coverability, then there is a simulation relation of OG_P by $OG_{P'}$.*

PROOF. The structure of OG_P is determined by the most permissive strategy MP_P of P (see Definition 13) which clearly satisfies the global constraint X as well. By Definition 17 (Matching), this implies the claimed simulation. \square

Lemma 2 (Local annotations). *If P' accords with P under coverability, then, for every $[q_P, q_{P'}] \in \varrho_{OG_P, OG_{P'}}$, the formula $\Phi_{OG_P}(q_P) \Rightarrow \Phi_{OG_{P'}}(q_{P'})$ is a tautology.*

PROOF. Let $[q_P, q_{P'}] \in \varrho_{OG_P, OG_{P'}}$ and consider the following modification MP'_P to the most permissive strategy MP_P for P . Remove all edges leaving q_P . For each assignment β to the propositions of $\Phi_{OG_P}(q_P)$, insert a new state q_β , a τ -edge from q_P to q_β and, for all literals a with $\beta(a) = \text{true}$, an edge from q_β to the former a -successor of q_P . If $\beta(\text{final}) = \text{true}$, make q_β a final state and remove all leaving edges labeled with a send event.

By construction, the obtained service automaton MP'_P is a $Cover_X$ -strategy for P as every inserted state has successors according to a satisfying assignment of $\Phi_{OG_P}(q_P)$. The global constraint X is satisfied as well since no state is removed from MP_P and no state becomes unreachable. It is also clear that, for every β satisfying $\Phi_{OG_P}(q_P)$, $[q_\beta, q_{P'}] \in \varrho_{MP'_P, OG_{P'}}$. Thus, since by accordance under coverability MP'_P has to be a $Cover_Y$ -strategy for P' , the edges leaving q_β (which make assignment β) need to satisfy $\Phi_{OG_{P'}}(q_{P'})$. Consequently, the claimed implication holds.

The removal of send edges from final states (in the final marking of an open net no transition must be enabled according to Definition 1 and hence the same holds for final states in service automata) does not harm this argument since send events always appear or-connected to the remaining formula. \square

Global constraint

Remember, we have to show that every $Cover_X$ -strategy for P is a $Cover_Y$ -strategy for P' as well. So far we have proven, if OG_P and $OG_{P'}$ satisfy the two conditions of Theorem 3, then every $Cover_X$ -strategy for P is a strategy for P' . It remains to show when a $Cover_X$ -strategy for P satisfies the global constraint ψ_Y of P' . Again, as the set of $Cover_X$ -strategies for P is infinite we are interested in a finite criterion. To this end, we aim at constructing an automaton S that satisfies the following condition: if there is a $Cover_X$ -strategy for P that violates ψ_Y of P' , then there is also a subautomaton of S (i.e. a subgraph of S that contains the initial state) which is a $Cover_X$ -strategy for P and violates ψ_Y . That way, we reduce an infinite check to a finite one.

Throughout this section, we assume that there is a simulation of OG_P by $OG_{P'}$ and local annotations imply each other as stated in Lemmas 1 and 2. Without this assumption, P' would not accord to P under coverability and there would be no reason for considering the global constraints.

The apparent candidate for the desired automaton S would be MP_P . However, as Fig. 4 explains, this does not work. In essence, the problem with MP_P is that there may be several $q_{P'}$ related to a single q_P . This problem can be avoided through constructing S differently. The idea is to use the simulation relation $\varrho_{OG_P, OG_{P'}}$ as the set of states. It then turns out that simulations between S and MP_P ($MP_{P'}$, resp.) are very regularly structured. We call the constructed service the *least common multiple* (*lcm*). This name is inspired by the observation that, if P is a simple loop of length m and P' a simple loop of length n , then S would have exactly $\text{lcm}(m, n)$ states.

Definition 21 (Least common multiple of service automata). Let P and P' be deterministic service automata. Let there be a simulation of P by P' . The *least common multiple* of P and P' , $\text{lcm}(P, P')$, is the service automaton defined by $\mathcal{Q}_{\text{lcm}(P, P')} = \varrho_{P, P'}$, $[q_P, q_{P'}] \xrightarrow{a} [q'_P, q'_{P'}]$ iff $q_P \xrightarrow{a} q'_P$ and $q_{P'} \xrightarrow{a} q'_{P'}$, $q_{0\text{lcm}(P, P')} = [q_{0P}, q_{0P'}]$, and $C_{\text{lcm}(P, P')} = C_P$.

In this definition, we did not specify final states. Final states shall be inserted when subautomata are considered. As we apply *lcm* to operating guidelines, the restriction to deterministic service automata does not harm. As an example, consider Fig. 4(c). In the example we have $\text{lcm}(2, 4) = 4$ states.

$\text{lcm}(P, P')$ has three important properties: First, q'_P and $q'_{P'}$ are uniquely determined by q_P , $q_{P'}$, and a . Second, $x_P \xrightarrow{a} x'_P$ implies $x_{P'} \xrightarrow{a} x'_{P'}$ since there is a simulation from P to P' . Third, both, P and P' , simulate $\text{lcm}(P, P')$. The

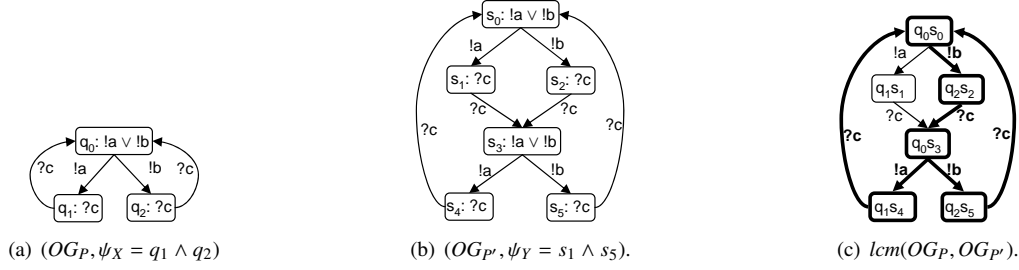


Figure 4: Motivation for lcm : Although every subautomaton of OG_P that satisfies Y (only the structure of OG_P) matches with $OG_{P'}$, $Strat_{Cover_X}(P) \not\subseteq Strat_{Cover_Y}(P')$. The (bold) subautomaton in (c) matches with (a) but violates ψ_Y in (b).

simulation of $lcm(P, P')$ by P is in fact a bisimulation, i.e. it is a simulation in both directions. Next, we show that the simulations $\mathcal{Q}_{lcm(P, P'), P}$, $\mathcal{Q}_{P, lcm(P, P')}$, and $\mathcal{Q}_{lcm(P, P'), P'}$ are minimal simulation relations.

Lemma 3. $\mathcal{Q}_{lcm(P, P'), P} = \{[q_P, q_{P'}], q_P \mid [q_P, q_{P'}] \in \mathcal{Q}_{lcm(P, P')}\}$, $\mathcal{Q}_{P, lcm(P, P')} = \{[q_P, [q_P, q_{P'}]] \mid [q_P, q_{P'}] \in \mathcal{Q}_{lcm(P, P')}\}$ and $\mathcal{Q}_{lcm(P, P'), P'} = \{[[q_P, q_{P'}], q_{P'}] \mid [q_P, q_{P'}] \in \mathcal{Q}_{lcm(P, P')}\}$ are minimal simulation relations.

PROOF. Let $[q_{0P}, q_{0P'}] \in \mathcal{Q}_{P, P'}$, $[q_{0lcm(P, P')}, q_{0P}] = [[q_{0P}, q_{0P'}], q_{0P}] \in \mathcal{Q}_{lcm(P, P'), P}$, $[q_{0P}, q_{0lcm(P, P')}] = [q_{0P}, [q_{0P}, q_{0P'}]] \in \mathcal{Q}_{P, lcm(P, P')}$, and $[q_{0lcm(P, P')}, q_{0P'}] = [[q_{0P}, q_{0P'}], q_{0P'}] \in \mathcal{Q}_{lcm(P, P'), P'}$.

Let $[q_P, q_{P'}] \xrightarrow{a} [q'_P, q'_{P'}]$. By construction of $lcm(P, P')$, $q_P \xrightarrow{a} q'_P$ and $q_{P'} \xrightarrow{a} q'_{P'}$, so $\mathcal{Q}_{lcm(P, P'), P}$ and $\mathcal{Q}_{lcm(P, P'), P'}$ are indeed simulation relations. Let, the other way round, $q_P \xrightarrow{a} q'_P$ and $[q_P, [q_P, q_{P'}]] \in \mathcal{Q}_{P, lcm(P, P')}$. Then, since there is a simulation of P by P' , there is a $q'_{P'}$ with $q_{P'} \xrightarrow{a} q'_{P'}$ which is uniquely determined since P' is deterministic. Consequently, $[q_P, q_{P'}] \xrightarrow{a} [q'_P, q'_{P'}]$ in $lcm(P, P')$ and thus $[q'_P, [q'_P, q'_{P'}]] \in \mathcal{Q}_{P, lcm(P, P')}$. Presence of $[[q_P, q_{P'}], q_P]$ indeed implies presence of $[q'_P, q'_{P'}]$ in the minimal simulation relation since P is deterministic, so there is no other choice to match a in P . Analogously, presence of $[q'_P, [q'_P, q'_{P'}]]$ and $[[q'_P, q'_{P'}], q'_{P'}]$ are required in the respective simulation relations. Consequently, all considered simulation relations are minimal. \square

Next, we show that there is a simulation of arbitrary $Cover_X$ -strategies for P by $lcm(OG_P, OG_{P'})$.

Lemma 4. Let $A \in Strat_{Cover_X}(P)$. Then there is a simulation of A by $lcm(OG_P, OG_{P'})$.

PROOF. As $A \in Strat_{Cover_X}(P)$, there is a simulation of A by OG_P (by Definition 17). By Lemma 3, there is a simulation of OG_P by $lcm(OG_P, OG_{P'})$. Since simulation is transitive, there is a simulation of A by $lcm(OG_P, OG_{P'})$. \square

Now we can show the main claim of this section. Assume there is an arbitrary $Cover_X$ -strategy for P , say A , which is not a $Cover_Y$ -strategy for P' . Using the simulation of A by $lcm(OG_P, OG_{P'})$, we construct a subautomaton of $lcm(OG_P, OG_{P'})$ which is a $Cover_X$ -strategy for P as well, and again no $Cover_Y$ -strategy for P' . We call the constructed subautomaton the *image* $Im(A)$ of A in $lcm(OG_P, OG_{P'})$.

Definition 22 (Image). Let A be a $Cover_X$ -strategy for P . Then the *image* of A , $Im(A)$, is defined by $\mathcal{Q}_{Im(A)} = \{[q_P, q_{P'}] \mid \exists q_A : [q_A, [q_P, q_{P'}]] \in \mathcal{Q}_{A, lcm(OG_P, OG_{P'})}\}$, $C_{Im(A)} = C_A$, $\delta_{Im(A)} = \delta_{lcm(OG_P, OG_{P'})}|_{\mathcal{Q}_{Im(A)}}$, $q_{0Im(A)} = q_{0lcm(OG_P, OG_{P'})}$, and $\Omega_{Im(A)} = \{q \in \mathcal{Q}_{Im(A)} \mid \text{no send event is activated in } q\}$.

$Im(A)$ restricts $lcm(P, P')$ to those states that are contained in the simulation relation of A by $lcm(P, P')$. Next we show that $Im(A)$ is a $Cover_X$ -strategy for P .

Lemma 5. If A is a $Cover_X$ -strategy for P , so is $Im(A)$.

PROOF. We have to show (1) OG_P simulates $Im(A)$, (2) $Im(A)$ satisfies the local annotations of OG_P , and (3) $Im(A)$ satisfies the global constraint ψ_X .

Ad (1). $Im(A)$ is (by Definition 22) a subautomaton of $lcm(OG_P, OG_{P'})$ which is bisimilar to P (see comment below Definition 21). Thus, a simulation of $Im(A)$ by OG_P clearly exists and the minimal one is the restriction of $\varrho_{lcm(OG_P, OG_{P'}), P}$ to $\varrho_{Im(A)}$.

Ad (2). We proceed by showing first that $[q_A, [q_P, q_{P'}]] \in \varrho_{A, Im(A)}$ implies that $[q_A, q_P] \in \varrho_{A, OG_P}$ (*). This is obviously true for the initial states. Let $[q_A, [q_P, q_{P'}]] \in \varrho_{A, Im(A)}$, $[q_A, q_P] \in \varrho_{A, OG_P}$ and $q_A \xrightarrow{a} q'_A$. Since both, OG_P and $Im(A)$, simulate A , there exist q'_P and $[q''_P, q''_{P'}]$ such that $q_P \xrightarrow{a} q'_P$ and $[q_P, q_{P'}] \xrightarrow{a} [q''_P, q''_{P'}]$ are in the respective systems. This implies $[q'_A, q'_P] \in \varrho_{A, OG_P}$ and $[q_A, [q''_P, q''_{P'}]] \in \varrho_{A, Im(A)}$. By construction of $lcm(OG_P, OG_{P'})$, $q''_P = q'_P$, so (*) holds.

Now we can show that $Im(A)$ satisfies the local annotations of OG_P . Let $[[q_P, q_{P'}], q_P] \in \varrho_{Im(A), OG_P}$. By construction of $Im(A)$, there is a state q_A of A where $[q_A, [q_P, q_{P'}]] \in \varrho_{A, Im(A)}$. With (*), $[q_A, q_P] \in \varrho_{A, OG_P}$. Since A is a strategy for P , the edge labels leaving q_A , together with the “final” status of q_A , form a satisfying assignment $\Phi_{OG_P}(q_P)$. Since $\varrho_{A, Im(A)}$ is a simulation, the edge labels leaving $[q_P, q_{P'}]$ include the ones that leave q_A . The “final” proposition is true in $[q_P, q_{P'}]$ unless there are “send” events leaving $[q_P, q_{P'}]$. We distinguish two cases. If there is a send event leaving $[q_P, q_{P'}]$, a single send event is sufficient to make $\Phi_{OG_P}(q_P)$ true. In the other case, the assignment used for checking $\Phi_{OG_P}(q_P)$ in $[q_P, q_{P'}]$ assigns true to at least those propositions where the assignment used in q_A is true. Since $\Phi_{OG_P}(q_P)$ is monotonous, $\Phi_{OG_P}(q_P)$ is true in $[q_P, q_{P'}]$.

Ad (3). By (*), ϱ_{A, OG_P} and $\varrho_{Im(A), OG_P}$ touch the same states of OG_P . Consequently, the value of ψ_X is the same for both A and $Im(A)$. Since A is a $Cover_X$ -strategy for P , ψ_X evaluates to true. \square

Now we prove that $Im(A)$ is not a $Cover_Y$ -strategy for P' if A is none. Remember that we assume throughout this section that local annotations accord to each other. Thus, A must satisfy local annotations of $OG_{P'}$. It is consequently sufficient to show that, if A violates the global constraint ψ_Y of $OG_{P'}$, so does $Im(A)$. Using monotonicity of ψ_Y , the following lemma is sufficient for establishing this claim. It proves that A touches at least as many states in $OG_{P'}$ as $Im(A)$. Thus, if $Im(A)$ satisfies ψ_Y so does A .

Lemma 6. *For every state $q_{P'}$ of $OG_{P'}$, if there is a state $q_{Im(A)}$ of $Im(A)$ such that $[q_{Im(A)}, q_{P'}] \in \varrho_{Im(A), OG_{P'}}$, then there is a state q_A such that $[q_A, q_{P'}] \in \varrho_{A, OG_{P'}}$.*

PROOF. We argue by induction on the construction of $\varrho_{A, Im(A)} = \varrho_{A, lcm(OG_P, OG_{P'})}$. By construction of $Im(A)$, the second components of $\varrho_{A, Im(A)}$ cover all states of $Im(A)$. For a state $[q_P, q_{P'}] \in Im(A)$, the fact that $\varrho_{Im(A), OG_{P'}}$ (see Lemma 3) guarantees that $[[q_P, q_{P'}], q'_{P'}] \in \varrho_{Im(A), OG_{P'}}$ implies that $q_{P'} = q'_{P'}$. Thus, it remains to show that, for each $[q_A, [q_P, q_{P'}]] \in \varrho_{A, Im(A)}$, $[q_A, q_{P'}] \in \varrho_{A, OG_{P'}}$. This is obviously true for the initial element $[q_{0A}, [q_{0OG_P}, q_{0OG_{P'}}]]$. Assume that $[q_A, [q_P, q_{P'}]] \in \varrho_{A, Im(A)}$ and $[q_A, q_{P'}] \in \varrho_{A, OG_{P'}}$. Let $[q'_A, [q'_P, q'_{P'}]]$ be the member of $\varrho_{A, Im(A)}$ that is inserted due to $[q_A, [q_P, q_{P'}]]$ and event a . Then clearly $q_A \xrightarrow{a} q'_A$ and $[q_P, q_{P'}] \xrightarrow{a} [q'_P, q'_{P'}]$ which implies (by construction of $lcm(OG_P, OG_{P'})$) $q_{P'} \xrightarrow{a} q'_{P'}$ in $OG_{P'}$. Consequently, $[q'_A, q'_{P'}] \in \varrho_{A, OG_{P'}}$. \square

With the help of these lemmas we can finally characterize the accordance check in the context of coverability.

Theorem 4 (Checking accordance under coverability). *Let P and P' be service automata. $Strat_{Cover_X}(P) \subseteq Strat_{Cover_Y}(P')$ iff*

1. *There is a simulation relation $\varrho_{OG_P, OG_{P'}}$ of OG_P by $OG_{P'}$.*
2. *For all $[q_P, q_{P'}] \in \varrho_{OG_P, OG_{P'}}$, the formula $\Phi(q_P) \Rightarrow \Phi(q_{P'})$ is a tautology.*
3. *For all subautomata S of $lcm(OG_P, OG_{P'})$ which are in $Strat_{Cover_X}(P)$, ψ_Y is satisfied.*

PROOF. (\Rightarrow) Assume $Strat_{Cover_X}(P) \subseteq Strat_{Cover_Y}(P')$. Then Lemma 1 and 2 establish the first two claims while the third one is evident from the assumption.

(\Leftarrow) Assume that all three claims hold. Let A be in $Strat_{Cover_X}(P)$. Then the first and second claims guarantee that there is a simulation of A by $OG_{P'}$ and all local annotations are satisfied. The global constraint is satisfied as well as otherwise Lemma 6 would state that the third assumption were invalid. \square

The operating guidelines in Fig. 4 satisfy the first two criteria of Theorem 4. To verify the third criterion of Theorem 4, we need to check nine subautomata. As the subautomaton depicted in Fig. 4(c) is in $Strat_{Cover_X}(P)$ but it violates ψ_Y , we conclude that P' does not accord with P under coverability.

4.3. An Algorithm for Checking Accordance Under Coverability

The service automaton $lcm(OG_P, OG_{P'})$ is finite and thus it contains only finitely many subautomata. Consequently, Theorem 4 already reduces the accordance problem to a finite check. Nevertheless, we propose to implement the verification in a slightly different manner. We start with checking the first two conditions of Theorem 4. This yields, in particular, the simulation relation $\rho_{OG_P, OG_{P'}}$ which we need for building $lcm(OG_P, OG_{P'})$. Then, we propose to iterate through all maximal assignments that violate the global constraint ψ_Y of $OG_{P'}$. For each such assignment, proceed as follows.

1. Remove all states $[q_P, q_{P'}]$ from $lcm(OG_P, OG_{P'})$ where false is assigned to $q_{P'}$ (by the maximal assignment).
2. Iteratively, remove all states $[q_P, q_{P'}]$ from $lcm(OG_P, OG_{P'})$ where the local annotation of q_P in OG_P is violated.
3. For the remaining subautomaton, evaluate the global constraint ψ_X of OG_P . If it evaluates to true, exit with result “not accordant”, otherwise continue with the next assignment.

This procedure is justified with the monotonicity of the global constraints. The subautomaton constructed in the second step, is the largest subautomaton of $lcm(OG_P, OG_{P'})$ that satisfies the local annotations of OG_P and violates the global constraint Y of $OG_{P'}$ with some assignment less or equal to the considered one. If this one violates X , every other subautomaton does, too.

Consider again the example in Fig. 4: (1) We have only two maximal assignments for ψ_Y assigning false to s_1 (the first) and assigning false to s_5 (the second). For the first maximal assignment we remove state q_1s_1 together with their adjacent edges yielding the subautomaton depicted bold in Fig. 4(c). The removal does not violate the local annotations in OG_P (2) and the resulting subautomaton satisfies ψ_X and thus we exit with “not accordant” (3). Note that starting with the second maximal assignment yields a subautomaton that satisfies ψ_X and hence proves non-accordance as well.

At this stage, we cannot provide experimental results. However, we believe—confirmed by our case study in Sect. 3.3—that global constraints tend not to be too large for practical processes, so the number of assignments to be considered should be tractable. The construction in the second step of the proposed procedure is part of the standard algorithm for computing most permissive strategies in Fiona and is known to be quite efficient. Consequently, we believe that the verification of accordance under coverability could be tractable in realistic cases even if worst case complexity is beyond the limits of theoretical tractability.

5. Related Work

The work presented in this paper is mainly inspired by the notion of soundness for workflow nets [2]. Soundness guarantees the absence of deadlocks, livelocks, and dead transitions. In this paper, we adopt the idea of soundness to our service model open nets. Given an open net N , we are interested in all open nets M such that the composition $M \oplus N$ is deadlock-free and certain places and transitions of N are covered. Here, cover means that these places can be marked and the transitions are not dead in $M \oplus N$. So far in contrast to soundness, our approach is limited in the sense that the composed system may contain livelocks.

The notion of relaxed soundness [18] ensures that for each transition t of N , there is a run that enables t and can be carried forward to the final state. Our approach is stricter than relaxed soundness, because we require that the composition $M \oplus N$ is deadlock-free that is, every run is deadlock-free.

There is also some relation to the research fields testing and computer-aided verification. In these fields testing/checking the coverage of certain activities is also a known and important sanity check, see [19], for instance.

Besides the relation to (relaxed) soundness, covering open net nodes can also be seen as a behavioral constraint for services. In [6] the authors introduced two kinds of behavioral constraints: to *enforce* and to *exclude* a set of open net nodes. A strategy M for N enforces (excludes) a transition t of N if every (no) run in $M \oplus N$ includes a t -step. Covering a transition t is thus equivalent to *not exclude* t . However, cover cannot be expressed by the approach proposed in [6], because the authors model a constraint as a constraint open net C and compose C and N .

The notion of accordance has been proposed in [17] and an algorithm for checking accordance in case of ordinary operating guidelines in [16]. Accordance is a refinement relation between two services. Similar refinement relations have been published in [20, 21, 22]. Projection inheritance in [20] preserves soundness but it is strictly coarser than accordance as it does not allow to reorder sending messages, for instance. The notions in [21, 22] are stricter than

accordance because they exclude deadlocks *and* livelocks. However, they do not consider covering of activities. For a detailed comparison to (ordinary) accordance we refer to [16].

The idea of constructing MP'_p in the proof of Lemma 2 has been adopted by the maximal strategy in [23]. Finally, the idea of using annotated automata as a representation of a set of automata has been first published in [24].

6. Conclusion

We proposed an approach to guarantee correct interaction between services, where correctness refers to deadlock freedom in the interaction of services and the coverability of certain activities. Our approach is inspired by the notion of soundness on the one hand and by the work on behavioral constraints for services on the other hand. We have shown that with the operating guideline of a service P we can decide if a partner service R is designed in a way such that a given set of activities of P can be covered in the composition of P and R . We further presented a finite representation of all partner services R by extending our notion of an operating guideline with a global constraint.

The advantage of the proposed approach is that the global constraint can be calculated based on the information that is already present when calculating the operating guideline. In addition, the global constraint does only marginally increase the complexity of matching a service with the operating guideline with global constraint. Experimental results have confirmed these considerations. Thus, operating guidelines with global constraint seem to be a well-suited instrument for deciding questions related to service composition.

In addition, we have investigated the problem when a service P can be substituted by another service P' such that any correctly interacting partner of P is not affected by this substitution and a that certain set of activities of P' will be covered. Since the presented decision algorithm applies the concept of an operating guideline with global constraint we can decide accordance under coverability even for the infinite sets of all *Cover*-strategies for P and P' .

In ongoing work, we plan to deal with a stricter correctness criterion that also excludes livelocks. That way, we can close the gap to soundness. Furthermore, we are working on an implementation of the presented algorithm for deciding accordance under coverability.

Acknowledgements

The authors wish like to thank Robert Danitz and Janine Ott for their work on the implementation of Fiona. Christian Stahl is funded by the DFG project “Substitutability of Services” (RE 834/16-1). Karsten Wolf is supported by the DFG within grant “Operating Guidelines for Services” (WO 1466/8-1).

References

- [1] M. P. Papazoglou, *Web Services: Principles and Technology*, Pearson - Prentice Hall, Essex, 2007.
- [2] W. M. P. v. d. Aalst, The Application of Petri Nets to Workflow Management, *The Journal of Circuits, Systems and Computers* 8 (1) (1998) 21–66.
- [3] W. M. P. v. d. Aalst, M. Weske, The P2P approach to Interorganizational Workflows, in: K. R. Dittrich, A. Geppert, M. C. Norrie (Eds.), *CAiSE 2001*, Vol. 2068 of LNCS, Springer-Verlag, 2001, pp. 140–156.
- [4] F. Leymann, D. Roller, M.-T. Schmidt, Web services and business process management., *IBM Systems Journal* 41 (2) (2002) 198–211.
- [5] N. Lohmann, P. Massuthe, K. Wolf, Operating guidelines for finite-state services, in: J. Kleijn, A. Yakovlev (Eds.), *ICATPN 2007*, Vol. 4546 of LNCS, Springer-Verlag, 2007, pp. 321–341.
- [6] N. Lohmann, P. Massuthe, K. Wolf, Behavioral constraints for services, in: G. Alonso, P. Dadam, M. Rosemann (Eds.), *BPM 2007*, Vol. 4714 of LNCS, Springer-Verlag, 2007, pp. 271–287.
- [7] C. Stahl, K. Wolf, Covering Places and Transitions in Open Nets, in: M. Dumas, M. Reichert (Eds.), *BPM 2008*, Vol. 5240 of LNCS, Springer-Verlag, 2008, pp. 116–131.
- [8] W. Reisig, *Petri Nets*, EATCS Monographs on Theoretical Computer Science Edition, Springer, 1985.
- [9] E. Kindler, A compositional partial order semantics for Petri net components, in: *ICATPN 1997*, Vol. 1248 of LNCS, Springer-Verlag, 1997, pp. 235–252.
- [10] P. Massuthe, K. Schmidt, Operating Guidelines – an Automata-Theoretic Foundation for the Service-Oriented Architecture, in: K. Cai, A. Ohnishi, M. Lau (Eds.), *QSIC 2005*, IEEE Computer Society, 2005, pp. 452–457.
- [11] P. Massuthe, K. Wolf, An Algorithm for Matching Non-deterministic Services with Operating Guidelines, *International Journal of Business Process Integration and Management (IJBPIIM)* 2 (2) (2007) 81–90.
- [12] R. Milner, *Communication and Concurrency*, Prentice-Hall, Inc., 1989.
- [13] N. Lohmann, P. Massuthe, C. Stahl, D. Weinberg, Analyzing Interacting BPEL Processes, in: S. Dustdar, J. Fiadeiro, A. Sheth (Eds.), *BPM 2006*, Vol. 4102 of LNCS, Springer-Verlag, 2006, pp. 17–32.
- [14] A. Alves et al., *Web Services Business Process Execution Language Version 2.0*, OASIS Standard, 11 April 2007, OASIS (Apr. 2007).

- [15] J. Arias-Fisteus, L. S. Fernández, C. D. Kloos, Applying model checking to bpel4ws business collaborations, in: H. Haddad, L. M. Liebrock, A. Omicini, R. L. Wainwright (Eds.), SAC 2005, ACM, 2005, pp. 826–830.
- [16] C. Stahl, P. Massuthe, J. Bretschneider, Deciding Substitutability of Services with Operating Guidelines, Informatik-Berichte 222, Humboldt-Universität zu Berlin, accepted for a journal (Apr. 2008).
- [17] W. M. P. v. d. Aalst, N. Lohmann, P. Massuthe, C. Stahl, K. Wolf, From public views to private views – correctness-by-design for services, in: M. Dumas, R. Heckel (Eds.), WS-FM 2007, Vol. 4937 of LNCS, Springer-Verlag, 2008, pp. 139–153.
- [18] J. Dehnert, W. M. P. v. d. Aalst, Bridging the gap between business models and workflow specifications., Int. J. Cooperative Inf. Syst. 13 (3) (2004) 289–332.
- [19] O. Kupferman, Sanity Checks in Formal Verification, in: C. Baier, H. Hermanns (Eds.), CONCUR 2006, Vol. 4137 of LNCS, Springer-Verlag, 2006, pp. 37–51.
- [20] W. M. P. v. d. Aalst, T. Basten, Inheritance of Workflows: An Approach to Tackling Problems Related to Change, Theor. Comput. Sci. 270 (1-2) (2002) 125–203.
- [21] M. Bravetti, G. Zavattaro, Contract Based Multi-party Service Composition, in: F. Arbab, M. Sirjani (Eds.), FSEN 2007, Vol. 4767 of LNCS, Springer-Verlag, 2007, pp. 207–222.
- [22] G. Castagna, N. Gesbert, L. Padovani, A Theory of Contracts for Web Services, in: G. C. Necula, P. Wadler (Eds.), POPL 2008, ACM, 2008, pp. 261–272.
- [23] A. Mooij, M. Voorhoeve, Proof techniques for adapter generation, in: R. Bruni, K. Wolf (Eds.), WS-FM 2008, LNCS, Springer-Verlag, 2008, to appear.
- [24] A. Wombacher, P. Fankhauser, B. Mahleko, E. J. Neuhold, Matchmaking for business processes based on choreographies, Int. J. Web Service Res. 1 (4) (2004) 14–32.