Deciding Behaviour Compatibility of Complex Correspondences between Process Models

Matthias Weidlich[1], Remco Dijkman[2], Mathias Weske[1]

[1] Hasso-Plattner-Institute, University of Potsdam, Germany
{matthias.weidlich,weske}@hpi.uni-potsdam.de
[2] Eindhoven University of Technology, The Netherlands
r.m.dijkman@tue.nl

Abstract. Compatibility of two process models can be verified using common notions of behaviour inheritance. However, these notions postulate 1:1 correspondences between activities of both models. This assumption is violated once activities from one model are refined or collapsed in the other model or in case there are groups of corresponding activities. Therefore, our work lifts the work on behaviour inheritance to the level of complex 1:n and n:m correspondences. Our contribution is (1) the definition of notions of behaviour compatibility for models that have complex correspondences and (2) a structural characterisation of these notions for sound free-choice process models that allows for computationally efficient reasoning. We show the applicability of our technique, by applying it in a case study in which we determine the compatibility between a set of reference process models and models that implement them.

1 Introduction

For two process models the compatibility of their behaviour can be verified, by determining that their behaviour is equivalent, modulo activities that have been added, removed, or refined. Compatibility verification is, for example, applied to determine whether a business process correctly implements the service that an organization provides to its clients, as it is specified by another (abstract) process (cf., [1]). As another example, compatibility verification is used to check whether a business process correctly implements a reference process (cf., [2, 3]).

Compatibility verification is based on correspondences that are defined between activities that are considered to be equivalent. While we assume these correspondences to be given, we discuss techniques for identifying correspondences when reviewing related work. For the case of elementary 1:1 correspondences between activities, common notions of behaviour inheritance [4, 5] can be applied to check for the absence of behavioural contradictions. These notions differ with respect to the treatment of activities that are without counterpart in the other model (i.e., added or removed). In the behavioural analysis, these transitions might either be *hidden* or *blocked*. If both models satisfy a certain behaviour equivalence, e.g., branching bisimulation or trace equivalence, once activities that are without any correspondence are hidden (blocked), we conclude on projection inheritance (protocol inheritance) [4].

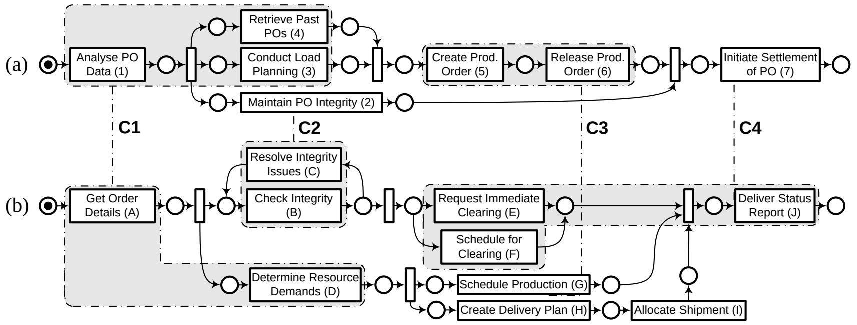
(a)
Analyse PO Data (1)
Retrieve Past POs (4)
Conduct Load Planning (3)
Maintain PO Integrity (2)
Create Prod. Order (5)
Release Prod. Order (6)
Initiate Settlement of PO (7)
C1
C2
C3
C4
(b)
Get Order Details (A)
Resolve Integrity Issues (C)
Check Integrity (B)
Request Immediate Clearing (E)
Schedule for Clearing (F)
Deliver Status Report (J)
Determine Resource Demands (D)
Schedule Production (G)
Create Delivery Plan (H)
Allocate Shipment (I)

Fig. 1. Two process models that illustrate an order processing, (a) is a reference model, (b) is a model customised for a specific organisation

In this paper, we solve the problem of checking compatibility in the presence of complex 1:n and n:m correspondences between two process models. We build upon the existing work on behaviour inheritance and lift it to the level of complex correspondences. Here, 1:n correspondences stem from activities from one model that are refined or collapsed in the other model. Moreover, n:m correspondences represent a relation between sets activities, for which there are no correspondences between one of their activity subsets. That is due to differences in modularisation of functionality between two process models. Fig. 1 illustrates our setting by a reference model (a) and a customised process model (b), along with four correspondences. Apparently, model (b) is not a hierarchical refinement of model (a), such that we observe a non-trivial relation between both models. For instance, activities A and D of model (b) have been identified to correspond to activities 1, 3, and 4 in the reference model (a). We answer the question, whether these correspondences are compatible.

The contribution of this paper is twofold. First, we introduce the notions of projection and protocol compatibility of correspondences and, therefore, process models. To this end, we use trace equivalence as the underlying equivalence criterion. Albeit based on the ideas of behaviour inheritance, we speak of compatibility as the notions are not directed. Second, we show that for the class of sound free-choice process models, these notions of compatibility can be characterised structurally. Thus, our notions can be decided efficiently based on structural analysis. We also report on findings from a case study. Due to space limitations, all proofs can be found in [6].

The remainder of this paper is structured as follows. Section 2 gives preliminaries for our work in terms of a formal model. Section 3 elaborates on our notions of behaviour compatibility of correspondences. Subsequently, their structural characterisation is addressed in Section 4. Section 5 introduces our case study. Finally, we review related work in Section 6 and conclude in Section 7.

2 Preliminaries

Our investigations are based on workflow (WF-) nets [7], a class of Petri nets used for process modelling and analysis. Petri net based formalisations have been presented for (parts of) common process modelling languages, such as BPEL, BPMN, and UML activity diagrams (e.g., [8–10]).

We recall basic definitions according to [7, 11]. A *net* is a tuple $N = (P, T, F)$ with P and T as finite disjoint sets of places and transitions, and $F \subseteq (P \times T) \cup (T \times P)$ as the flow relation. Without stating it explicitly, we assume a net to be always defined as $N = (P, T, F)$. We write $X = (P \cup T)$ for all nodes. The transitive closure of F is denoted by F^+. For a node $x \in X$, its preset and postset are defined as $\bullet x := \{y \in X \mid (y, x) \in F\}$ and $x\bullet := \{y \in X \mid (x, y) \in F\}$, respectively. A tuple $N' = (P', T', F')$ is a *subnet* for a net $N = (P, T, F)$, if $P' \subseteq P$, $T' \subseteq T$, and $F' = F \cap ((P' \times T') \cup (T' \times P'))$. Note that a subnet is induced by a given subset of places or transitions, respectively. A net N is *free-choice*, iff $\forall\, p \in P$ with $|p\bullet| > 1$ holds $\bullet(p\bullet) = \{p\}$. A *workflow (WF-) net* is a net $N = (P, T, F)$, such that there is exactly one place $i \in P$ with $\bullet i = \emptyset$, exactly one place $o \in P$ with $o\bullet = \emptyset$, and $\forall\, x \in X\ [\, iF^+x \ \wedge\ xF^+o \,]$. A *path* of length $n \in \mathbb{N}$, $n > 1$, is a sequence $\pi : \{1, \ldots, n\} \mapsto X$, denoted by $\pi(x_1, x_n)$ or $\pi = x_1, \ldots, x_n$, which satisfies $((1, x_1), (2, x_2)), \ldots, ((n-1, x_{n-1}), (n, x_n)) \in F$. We write $t \in \pi$ if $(i, t) \in \pi$ for some $i \in \mathbb{N}$. A *subpath* π' of a path π is a subsequence. The set $\mathcal{P}_N$ contains all complete paths $\pi(i, o)$ of a WF-net N. A path $\pi = x_1, \ldots, x_n$ in a net $N = (P, T, F)$ can be restricted to its transitions yielding the path $\pi^T = x_1, x_3, \ldots, x_m$ (if $x_1 \in T$) or $\pi^T = x_2, x_4, \ldots, x_m$ (otherwise) with $m \in \{n-1, n\}$ and $x_m \in T$. The set $\mathcal{P}_N^T$ contains all complete paths restricted to their transitions of N. We write $\pi^T \subseteq T'$ if for all $(i, t) \in \pi$ it holds $t \in T' \subseteq T$.

We define semantics for a WF-net $N = (P, T, F)$ with initial place i and final place o according to [7]. $M : P \mapsto \mathbb{N}$ is a *marking* of N, $\mathbb{M}$ is the set of all markings. For a place $p \in P$, M_p is the marking that puts a token on p and no token elsewhere. For a transition $t \in T$, M_t is the marking that puts a token on every place $p \in \bullet t$ and no token elsewhere. For a WF-net, M_i is the initial, M_o the final marking. $M(p)$ returns the number of tokens in p, if $p \in dom(M)$. Moreover, for two markings $M, M' \in \mathbb{M}$, $M \geq M'$ if $M(p) \geq M'(p)$ for all $p \in P$. For any transition $t \in T$ and any marking $M \in \mathbb{M}$, t is *enabled* in M, denoted by $(N, M)[t\rangle$, iff $\forall\, p \in \bullet t\ [\, M(p) \geq 1 \,]$. Marking M' is reached from M in N by *firing* of t, denoted by $(N, M)[t\rangle(N, M')$, such that $M' = M - \bullet t + t\bullet$. A *firing sequence* of length $n \in \mathbb{N}$ is a sequence $\sigma : \{1, \ldots, n\} \mapsto T$. For $\sigma = \{(1, t_x), \ldots, (n, t_y)\}$, we also write $\sigma = t_1, \ldots, t_n$. We write $t \in \sigma$ if $(i, t) \in \sigma$ for some $i \in \mathbb{N}$, and $\sigma \subseteq T'$ if for all $(i, t) \in \sigma$ it holds $t \in T' \subseteq T$. Any firing sequence σ with $(N, M_i)[\sigma\rangle(N, M_o)$ is a *complete trace* (or trace). The set of all complete traces of N is the *language* of N, denoted by $\mathcal{L}_N$. A *subtrace* σ' of a trace σ is a subsequence of σ. A marking $M' \in \mathbb{M}$ is reachable from $M \in \mathbb{M}$ in N, denoted by $M' \in [N, M\rangle$, if there exists a firing sequence σ, such that $(N, M)[\sigma\rangle(N, M')$.

We also recall the *soundness* criterion, which requires WF-nets (1) to always terminate, and (2) to have no dead transitions (proper termination is implied for WF-nets) [12]. A WF-net N is *live*, if for every marking $M \in [N, M_i\rangle$ and $t \in T$, there is a marking $M' \in [N, M\rangle$ such that $(N, M')[t\rangle$. A WF-net N is *bounded*, iff the set $[N, M_i\rangle$ is finite. A WF-net N with $N = (P, T, F)$ is *sound*, iff the short-circuit net N', $N' = (P, T \cup \{t_c\}, F \cup \{(o, t_c), (t_c, i)\})$, is live and bounded.

3 Behaviour Compatibility of Correspondences

This section introduces behaviour compatibility for correspondences between WF-nets. We use WF-nets as behavioural models due to our focus on process models. It is worth to mention though, that all concepts can be lifted to general Petri nets or even state transitions systems in a straightforward manner. First, Section 3.1 clarifies the notion of a correspondence. Second, Section 3.2 elaborates on a partitioning of traces that is imposed by these correspondences. Third, Section 3.3 introduces two kinds of behaviour compatibility for a pair of correspondences. Finally, Section 3.4 elaborates on how to decide behaviour compatibility.

3.1 Correspondences between WF-nets

In general, a correspondence between two WF-nets is defined by two sets of transitions of the WF-nets. Following on the classification of correspondences between data schemata or ontologies [13], we speak of elementary or complex correspondences depending on the cardinality of the associated set of transitions.

Definition 1 (Correspondence). *Let* $N = (P, T, F)$ *and* $N' = (P', T', F')$ *be two WF-nets. The* correspondence relation $\equiv \subseteq \wp(T) \times \wp(T')$ *associates corresponding sets of transitions of both nets to each other. Let* $T_1 \subseteq T$ *and* $T_2 \subseteq T'$*. If* $T_1 \equiv T_2$ *then* (T_1, T_2) *is referred to as a* correspondence. (T_1, T_2) *is called* elementary*, iff* $|T_1| = |T_2| = 1$*, and* complex *otherwise.*

In the remainder of this paper, we assume all correspondences to be non-overlapping. That is, two correspondences $C = (T_1, T_3)$ and $C' = (T_2, T_4)$ must not share any transition in any of the WF-nets, i.e., $T_1 \cap T_2 = \emptyset$ and $T_3 \cap T_4 = \emptyset$. Overlapping correspondences raise various questions regarding their intended meaning. Assume two correspondences are defined as $C = (\{a\}, \{x, y\})$ and $C' = (\{b\}, \{y\})$. Then, the occurrence of the two transitions x and y in one model might correspond to both, the occurrence of transition a only, or the occurrence of both transitions, a and b, in the other model. Hence, the inherent semantic ambiguity of overlapping correspondences has to be addressed as a prerequisite for any behavioural analysis.

In our example in Fig. 1, for instance, $C1$ would be classified as a complex 3:2 correspondence, while $C3$ is a 2:1 correspondence. Note that the correspondences depicted in Fig. 1 are all non-overlapping.

After having defined the notion of a correspondence, it is worth to mention that such a correspondence induces certain semantics. In terms of trace semantics, the transitions that are part of a correspondence occur in dedicated subtraces of the net. Thus, a correspondence induces a relation between subtraces of the one net and subtraces of the other net. For instance, correspondence $C1$ in Fig. 1 relates the subtraces $\langle 1, 3, 4\rangle$ and $\langle 1, 4, 3\rangle$ in model (a) to the subtrace $\langle A, D\rangle$ in model (b). For $C2$, in turn, there is a relation between the subtrace $\langle 2\rangle$ in model (a) and an infinite set of traces in model (b), e.g., $\langle B\rangle$ and $\langle B, C, B\rangle$.

3.2 Trace Partitioning based on Correspondences

In the previous section, we argued that a correspondence between sets of transitions induces semantics in terms of subtraces of two nets that are considered to correspond to each other against the background of the alignment. Therefore, for two correspondences, the constraints between the respective subtraces of both correspondences imposed by one net, should hold for the respective subtraces in the other net as well. Here, constraints refer to the observable order and number of occurrences of such subtraces in all complete traces of a net.

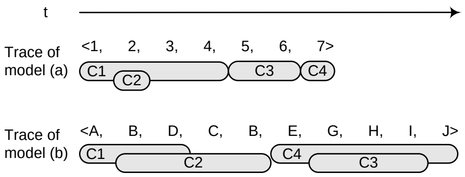

Fig. 2. Exemplary traces of the models of Fig. 1 along with their relation to correspondences

We illustrate the relation between subtraces by means of the WF-nets of Fig. 1. Here, we see that the constraints for the subtraces relating to the correspondences $C1$ and $C3$ are equal. That is, any subtrace of model (a) built of transitions of $C1$, is followed by a subtrace comprising transitions of $C3$. In addition, both subtraces occur at most once. This also holds for the respective subtraces in model (b), as exemplified for a pair of traces in Fig. 2. For correspondences $C1$ and $C2$, the constraints imposed by both models are equal either. That is, in any trace of both nets, a transition belonging to $C1$ is observed first and might potentially be followed by transitions of $C1$ and $C2$. Note that the specific order of interleaving transitions of both correspondences is different though. For instance, in the subtrace $\langle A, B, D, C, B \rangle$ of model (b), two transitions of $C1$ are followed by a transition of $C2$. This is not possible in any subtrace of model (a), due to the different number of interleaving transitions of $C1$ and $C3$ in both nets, cf., Fig. 2. When focussing on correspondences $C3$ and $C4$, however, we detect differences in the imposed constraints. For instance, there is a trace in model (b), in which a transition of $C4$ occurs before any transition of $C3$, which yields a contradiction with the semantics of model (a).

These examples illustrate that the interleaving of transitions belonging to different correspondences has to be taken into account when assessing behaviour compatibility. We speak of two interleaving transitions, if for every trace in which they directly occur together, there is another trace that is equal to the first one besides a switched order of the interleaving transitions.

Definition 2 (Interleaving Transitions). *Let $N = (P, T, F)$ be a WF-net. Two transitions $(t_1, t_2) \in (T \times T)$ are* interleaving*, iff for each trace $\sigma_1 \in \mathcal{L}_N$ with $(i, t_1), (i + 1, t_2) \in \sigma_1$ for some $i \in \mathbb{N}$, there is a trace $\sigma_2 \in \mathcal{L}_N$ with $\sigma_2 = \{(i, t_2), (i + 1, t_1)\} \cup \{(j, t) \mid (j, t) \in \sigma_1 \ \wedge \ j \neq i \ \wedge \ j \neq i + 1\}$. Given two disjoint sets of transitions $T_1, T_2 \subseteq T$, the set $\iota_{(T_1,T_2)}(N) \subseteq T_1 \cup T_2$ contains all transitions that are part of an interleaving transition pair $(t_1, t_2) \in (T_1 \times T_2)$.*

For our example in Fig. 1, the set of interleaving transitions for the correspondences $C_1 = (T_1, T_3)$ and $C_2 = (T_2, T_4)$ with $T_1 = \{1, 3, 4\}$, $T_2 = \{2\}$, $T_3 = \{A, D\}$, and $T_4 = \{B, C\}$ are defined as $\iota_{(T_1,T_2)} = \{2, 3, 4\}$ for model (a) and $\iota_{(T_3,T_4)} = \{B, C, D\}$ for model (b).

Given two correspondences for a net, their dependencies can be assessed in a certain trace by partitioning the trace into subtraces that represent interleaving and non-interleaving parts of the correspondences.

Definition 3 (Partitioning of a Trace). *Let $N = (P, T, F)$ be a WF-net and $T_1, T_2 \subseteq T$ two disjoint sets of transitions. For any trace $\sigma \in \mathcal{L}_N$ the partitioning $\rho_{(T_1,T_2)}(\sigma)$ induced by T_1 and T_2 is a sequence of subtraces of maximal length $\rho_{(T_1,T_2)}(\sigma) = \sigma_1, \ldots, \sigma_n$ such that for any $i \in \mathbb{N}$ with $1 \leq i \leq n$ it holds either $\sigma_i \subseteq T_1 \setminus \iota_{(T_1,T_2)}(N)$, $\sigma_i \subseteq T_2 \setminus \iota_{(T_1,T_2)}(N)$, $\sigma_i \subseteq \iota_{(T_1,T_2)}(N)$, or $\sigma_i \subseteq T \setminus (T_1 \cup T_2)$.*

According to this definition, any transition that is part of a trace belongs to one of the four classes w.r.t. two sets of transitions. It is an interleaving transition, it belongs to one of the sets of transitions without being an interleaving transition, or it is not part of the two sets at all. The partitioning of traces for the models of our example is illustrated in Fig. 3 for three exemplary pairs of transition sets. As these sets are induced by the respective correspondences, we also speak of *transitions of correspondences* and name the sets accordingly. Note that all transitions that do not relate to the respective correspondences have been neglected. For correspondences $C1$ and $C3$, in all traces a subtrace comprising non-interleaving transitions of $C1$ is followed by a subtrace comprising non-interleaving transitions of $C3$ in both models, (a) and (b). Similarly, for correspondences $C1$ and $C2$, non-interleaving transitions of $C1$ are followed by a subtrace consisting of interleaving transitions of both correspondences in both models. In contrast, for correspondences $C3$ and $C4$, the contradicting constraints as discussed above are visible in the trace partitioning. Non-interleaving transitions of $C3$ are followed by non-interleaving transitions of $C4$ in model (a), whereas interleaving transitions of $C3$ and $C4$ are followed by non-interleaving transitions of $C4$ in model (b).

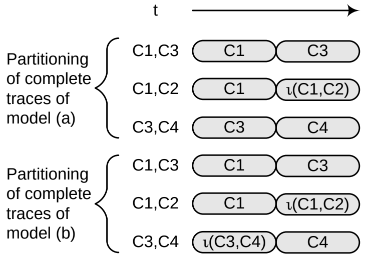
t
Partitioning of complete traces of model (a)
C1,C3
C1
C3
C1,C2
C1
ι(C1,C2)
C3,C4
C3
C4
Partitioning of complete traces of model (b)
C1,C3
C1
C3
C1,C2
C1
ι(C1,C2)
C3,C4
ι(C3,C4)
C4

Fig. 3. Partitioning of traces of the models in Fig. 1

3.3 Notions of Behaviour Compatibility

Based on a trace partitioning induced by the transitions of two correspondences, we define two notions of compatibility. Informally, both notions require that for each trace of the one net, there is a trace in the other net that shows the same partitioning in interleaving and non-interleaving subtraces of the transitions of the respective correspondences. Still, we distinguish two ways of coping with transitions that are not part of any correspondence. They might be hidden or blocked, cf., [4, 5]. Following on the notions of projection inheritance (hiding) and protocol inheritance (blocking), this distinction leads to the notions of projection and protocol compatibility of correspondences. First and foremost, we define projection compatibility for correspondences. It uses the notion of a trace projection. Given a WF-net $N = (P, T, F)$, a set of transitions $H \subseteq T$, and a trace $\sigma \in \mathcal{L}_N$, the set $H_{\sigma|j} = \{(x, t) \in \sigma \mid x < j \wedge t \in H\}$ denotes the occurrences of transitions of H in σ up to index $j \in \mathbb{N}$. Based thereon, we define the projection $\tau_H(\sigma)$ for a trace $\sigma \in \mathcal{L}_N$ of length n induced by H as $\tau_H(\sigma) = \bigcup_{i=0}^{|H_{\sigma|n}|}(i, t_i)$ with $t_i \in H$, such that $\exists\, j \in \mathbb{N}\; [\, (j, t_i) \in \sigma \wedge i = |H_{\sigma|j}|\,]$. Informally, the projected trace $\tau_H(\sigma)$ is derived by taking all transitions in H from σ.

Definition 4 (Projection Compatibility). *Let* $N = (P, T, F)$ *and* $N' = (P', T', F')$ *be WF-nets, and* $C_1 = (T_1, T_3)$*,* $C_2 = (T_2, T_4)$ *two correspondences.*

- C_1 *and* C_2 *are* projection compatible from N to N'*, iff for any trace* $\sigma \in \mathcal{L}_N$*, there is a trace* $\sigma' \in \mathcal{L}_{N'}$*, such that for the partitioned projections* $\rho_{(T_1,T_2)}(\tau_{(T_1 \cup T_2)}(\sigma)) = \sigma_1, \ldots, \sigma_n$ *and* $\rho_{(T_3,T_4)}(\tau_{(T_3 \cup T_4)}(\sigma')) = \sigma'_1, \ldots, \sigma'_m$ *it holds* $n = m$ *and for all* $i \in \mathbb{N}$ *with* $0 \leq i \leq n$*:*
  - $\sigma_i \subseteq (T_1 \setminus \iota_{(T_1,T_2)}(N)) \Rightarrow \sigma'_i \subseteq (T_3 \setminus \iota_{(T_3,T_4)}(N'))$.
  - $\sigma_i \subseteq (T_2 \setminus \iota_{(T_1,T_2)}(N)) \Rightarrow \sigma'_i \subseteq (T_4 \setminus \iota_{(T_3,T_4)}(N'))$.
  - $\sigma_i \subseteq \iota_{(T_1,T_2)}(N) \Rightarrow \sigma'_i \subseteq \iota_{(T_3,T_4)}(N')$.
- C_1 *and* C_2 *are* projection compatible*, iff they are projection compatible in either direction.*

Projection compatibility of two correspondences between two nets means that every complete trace in one net has a corresponding complete trace in the other net, which shows the same partitioning w.r.t. the two correspondences. We see that projection compatibility can be decided for two correspondences in isolation, i.e., independent of other correspondences. That is due to the fact that any transitions not belonging to the respective correspondences are projected before comparing the partitioning of traces. In contrast, protocol compatibility of two correspondences has to be decided always against the background of an alignment, i.e., a set of correspondences. Following on the approach introduced for protocol inheritance in [4], we use an encapsulation operator δ_H that creates the subnet induced by a set of transitions $H \subseteq T$ from a net $N = (P, T, F)$, such that $\delta_H(N) = (P, H, F_H)$. Encapsulation of a WF-net might yield a net that is not a WF-net anymore. Therefore, we also define the normalisation operator η_N that creates the workflow subnet of a subnet N_1 of a WF-net N, such that $\eta_N(N_1) = (P_\eta, T_\eta, F_\eta)$ with $P_\eta = P_1 \setminus X_r$ and $T_\eta = T_1 \setminus X_r$ and $X_r = \{x \in X_1 \mid i \not{F}_1^+ x \vee x \not{F}_1^+ o\}$ (i and o being the initial and final place of N). Normalisation yields the empty net, if there is no workflow subnet.

Definition 5 (Protocol Compatibility). *Let* $N = (P, T, F)$ *and* $N' = (P', T', F')$ *be WF-nets and* $\equiv$ *a correspondence relation between them. Let* $T_\equiv \subseteq T$ *and* $T'_\equiv \subseteq T'$ *be the transitions of both nets that are part of any correspondence, and* $E = \eta_N(\delta_{T_\equiv}(N))$ *and* $E' = \eta_{N'}(\delta_{T'_\equiv}(N'))$ *the normalised encapsulated nets. Let* $C_1 = (T_1, T_3)$ *and* $C_2 = (T_2, T_4)$ *be two correspondences.*

- C_1 *and* C_2 *are* protocol compatible from N to N'*, iff* E *and* E' *are WF-nets and* $C1$ *and* $C2$ *are projection compatible from* E *to* E'*.*
- C_1 *and* C_2 *are* protocol compatible*, iff they are protocol compatible in either direction.*

Protocol compatibility of correspondences between two nets captures that every complete trace of the one net has a corresponding complete trace in the other net, which shows the same partitioning w.r.t. the two correspondences once transitions that are not aligned are removed. Thus, protocol compatibility is traced back to projection compatibility of the normalised encapsulated nets that contain only aligned transitions. However, it is important to notice that both notions are orthogonal. That is, correspondences between two nets might show solely projection compatibility, but not protocol compatibility, and vice versa.

Regarding the WF-nets in Fig. 1, we conclude that, for instance, correspondences $C1$ and $C2$ are projection compatible, whereas correspondences $C1$ and $C4$ are not due to the interleaving of their transitions in model (b) (transitions D, E, and F), which is not possible in model (a), cf., Fig. 3. Direct application of the protocol compatibility criterion to our example yields a negative result, as both models contain no-operation (NOP) transitions that only realise the splitting and merging of control flow. These transitions are not part of the encapsulated nets, such that normalisation yields an empty net and there is no completed trace from the initial to the final marking in both nets. Still, these NOP transitions can be neglected in a way that they are part of the encapsulated nets. Then, for instance, encapsulation removes transitions H and I of model (b). As this does not change the observable behaviour of the remaining transitions, correspondences $C1$ and $C2$ are also protocol compatible, whereas correspondences $C1$ and $C4$ are not, owing to the aforementioned issues.

So far, we discussed the compatibility of a pair of correspondences in isolation. Both notions can be lifted from a pair of correspondences to a set of correspondences between two nets in a straightforward manner.

Definition 6 (Projection & Protocol Compatibility (Correspondence Relation)). *A correspondence relation between two nets is* projection (protocol) compatible*, iff all correspondences are pairwise projection (protocol) compatible.*

3.4 Decidability of Behaviour Compatibility

In the general case, behaviour compatibility of two correspondences between two WF-nets can be decided by state space exploration. Under the assumption of a finite state space, all state transitions relate to none of the correspondences, one of the correspondences, or the interleaving of both correspondences, respectively. Therefore, the trace partitioning (cf., Definition 3) is directly visible in the respective labelled transition system (LTS). Fig. 4 illustrates this dependency by the LTS of model (a) of Fig. 1. In the lower system, we highlighted the transitions that are related to correspondences $C_1 = (T_1, T_3)$ and $C_2 = (T_2, T_4)$. Each state transition that is part of $T_1 \cup T_2$ can be classified as belonging to one of the three sets, $\iota_{(T_1,T_2)}$, $T_1 \setminus \iota_{(T_1,T_2)}$, or $T_2 \setminus \iota_{(T_1,T_2)}$. Based thereon, complex state transitions are derived that represent the transition sequences of maximal length comprising solely transitions of one of the aforementioned three sets and transitions that are not in $T_1 \cup T_2$. The latter is illustrated in the lower LTS in Fig. 4, in which the transition $\iota_{(C_1,C_2)}$ contains solely interleaving transitions and transitions that are not related to $C1$ and $C2$.

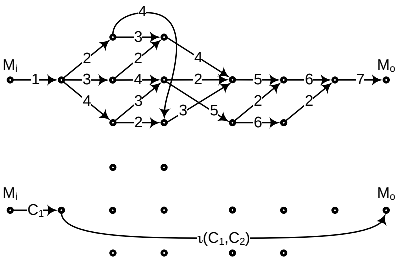

Fig. 4. LTS of model (a) of Fig. 1 along with transitions related to correspondences $C1$ and $C2$

Moreover, interleaving transitions can be characterised as being enabled concurrently in some marking or as not changing the marking when being fired.

Lemma 1. *Let $N = (P, T, F)$ be a WF-net. A pair of transitions $(t_1, t_2) \in (T \times T)$ is interleaving, iff there is a marking $M \in [N, M_i\rangle$ such that (1) $M \geq M_{t_1} + M_{t_2}$ and with $(N, M)[t_1\rangle(N, M_1)$ and $(N, M)[t_2\rangle(N, M_2)$ it holds $M_o \in [N, M_1\rangle$ and $M_o \in [N, M_2\rangle$, or (2) $(N, M)[t_1\rangle(N, M)$, $(N, M)[t_2\rangle(N, M)$, and $M_o \in [N, M\rangle$.*

Based thereon, we conclude decidability of behaviour compatibility of correspondences for nets with a finite state space. The proofs can be found in [6].

Theorem 1. *Given two bounded WF-nets and a set of correspondences, it is decidable whether two correspondences are projection or protocol compatible.*

Apparently, any approach of deciding behaviour compatibility based on state space exploration is computationally hard in the general case, due to the state explosion problem. The problem of whether two LTS show the same trace semantics is PSPACE-complete [15]. Hence, structural characterisations of behaviour compatibility for certain classes of nets are crucial for any real-world application.

4 A Structural Characterisation of Compatibility

This section shows that projection compatibility and, therefore, also protocol compatibility are decided efficiently for correspondences between sound free-choice WF-nets. That is due to the fact that for sound free-choice WF-nets, there is a tight coupling of syntax and semantics. First, Section 4.1 discusses the properties of sound free-choice WF-nets that are used in our approach. Second, Section 4.2 introduces the notion of path consistency of two correspondences between two WF-nets. Finally, Section 4.3 elaborates on how this structural characterisation is used to decide behaviour compatibility.

4.1 Properties of Sound Free-Choice WF-Nets

As mentioned above, sound free-choice WF-nets show a tight coupling of syntax and semantics. In particular, if N is sound and free-choice, the existence of a path $\pi(x, y)$ between places x and y implies the existence of a firing sequence containing all transitions on $\pi(x, y)$ (cf., Lemma 4.2 in [16]). Actually, this implication requires the marking M_y to be a home marking (a marking reachable from every marking that is reachable from the initial state). Still, the implication might be lifted to all home markings M_1 with $M_1(y) > 0$. Due to soundness of the net N, the short-circuit net N' is live and bounded, such that all markings $M \in [N, M_i\rangle$ are home markings in N'. Thus, all markings $M_1(y) > 0$ are reachable from markings $M_2(x) > 0$, if $M_1, M_2 \in [N', M_i\rangle$.

Another important property of sound free-choice nets is the possibility to compute the following two relations efficiently.

Concurrency Relation. The concurrency relation $|| \subseteq X \times X$ for the nodes X of a net N contains all pairs (x_1, x_2) such that $M \geq M_{x_1} + M_{x_2}$ for some reachable marking M. Thus, the concurrency relation identifies concurrently enabled transitions or marked places, respectively. Note that any sound free-choice net is also safe (cf., Lemma 1 in [17]). Thus, a single transition cannot be enabled concurrently with itself. According to [18], the concurrency relation can be determined in $O(n^3)$ time for live and bounded free-choice nets with n as the number of nodes of the net.

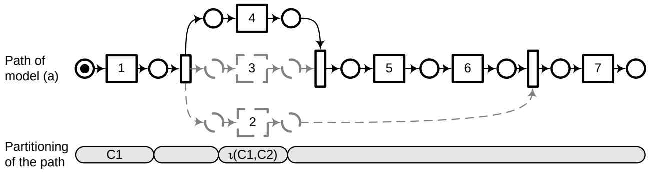

Fig. 5. A path of model (a) of Fig. 1 along with the partitioning induced by the non-interleaving and interleaving transitions of correspondences $C1$ and $C2$

Exclusiveness Relation. The exclusiveness relation $+ \subseteq T \times T$ for the transitions of a net N contains all pairs (t_1, t_2) that never occur together in a complete trace, i.e., for all complete traces $\sigma \in \mathcal{L}_N$ it holds $t_1 \in \sigma \Rightarrow t_2 \notin \sigma$ and $t_2 \in \sigma \Rightarrow t_1 \notin \sigma$. According to [19] (Lemma 3), the exclusiveness relation can be deduced from the concurrency relation and the transitive closure of the flow relation for sound free-choice nets. Based thereon, the exclusiveness relation can also be computed in $O(n^3)$ time with n as the number of nodes as detailed in [19]. The exclusiveness relation can be lifted from the transitions to all nodes of a net. Two places p_1 and p_2 are exclusive if there is no complete trace that visits two markings M_1 and M_2 with $M_1(p_1) > 0$ and $M_2(p_2) > 0$. Obviously, this information can be deduced directly from the exclusiveness of transitions.

In our example in Fig. 1, for instance, transitions D and E of model (b) are in the concurrency relation, while transitions E and F are exclusive.

4.2 Path Consistency of Correspondences

In order to reason on behaviour compatibility of two correspondences between two sound free-choice WF-nets, we assess their structural consistency. That is, the existence of certain paths in two process models is evaluated with respect to the correspondences. To this end, we define the partitioning of a path that is induced by two sets of transitions, similar to the partitioning of a trace presented in Section 3.2. Here, we consider solely the transitions of a path and neglect all places. Note that such a partitioning is grounded on the interleaving transitions of both sets. However, according to Lemma 1, the notion of two interleaving transitions can be traced back to their concurrent enabling (or a structural analysis of their pre- and postset, respectively), which, in turn, can be decided structurally for sound free-choice WF-nets, cf., Section 4.1.

Definition 7 (Partitioning of a Path). *Let $N = (P, T, F)$ be a WF-net and $T_1, T_2 \subseteq T$ two disjoint sets of transitions. For any path $\pi \in \mathcal{P}_N^T$ the partitioning $\rho_{(T_1,T_2)}(\pi)$ induced by T_1 and T_2 is a sequence of subpaths of maximal length $\rho_{(T_1,T_2)}(\pi) = \pi_1, \ldots, \pi_n$ such that for any $i \in \mathbb{N}$ with $1 \leq i \leq n$ it holds either $\pi_i \subseteq T_1 \setminus \iota_{(T_1,T_2)}(N)$, $\pi_i \subseteq T_2 \setminus \iota_{(T_1,T_2)}(N)$, $\pi_i \subseteq \iota_{(T_1,T_2)}(N)$, or $\pi_i \subseteq T \setminus (T_1 \cup T_2)$.*

Fig. 5 Illustrates the partitioning of an exemplary path of model (a) of our example with respect to correspondences $C1$ and $C2$. As mentioned before, transition 1 is a non-interleaving transition related to correspondence $C1$. Transition 4 is in the set of interleaving transitions of both correspondences. All other transitions on the highlighted path do not relate to any of the correspondences.

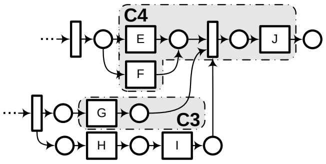

Fig. 6. Excerpt of model (b)

When comparing the partitioning of paths induced by two correspondences between two WF-nets, certain subpaths have to be neglected. That is, subpaths that represent a detour of a transition that is part of a correspondence are identified and removed from the net. Apparently, this reduction has to happen solely in case there is another transition of the correspondence that might be enabled concurrently. We illustrate the need for this kind of preprocessing with Fig. 6. It shows an excerpt of model (b) of our example. Assume that we investigate correspondences $C3$ and $C4$. Then, a path comprising transitions H and I would suggest that a non-interleaving transition related to $C4$ (transition J) can occur without any occurrence of an interleaving transition of both correspondences (transitions E, F, and G). Hence, the subpath comprising transitions H and I is removed by the preprocessing. Note that the preprocessing uses the concurrency relation and the exclusiveness relation, which can be derived from the net structure as discussed in Section 4.1.

Definition 8 (Preprocessing). *Let $N = (P, T, F)$ be a WF-net and $T_1, T_2 \subseteq T$ two disjoint sets of transitions. Let $X_{pp} \subseteq (X \setminus (T_1 \cup T_2))$ contain all nodes x for which there is a transition $t_1 \in T_1 \cup T_2$, such that $x || t_1$ and for all $t_2 \in T_1 \cup T_2$ with $t_1 || t_2$ it holds either $x || t_2$ or $x + t_2$. The* preprocessed *WF-net for N is a subnet $N' = (P', T', F')$ with $P' = P \setminus X_{pp}$ and $T' = T \setminus X_{pp}$.*

Once two WF-nets are preprocessed with respect to a pair of correspondences, their path consistency is assessed. Loosely spoken, path consistency implies that both nets show equal partitionings of paths from the initial to the final place regarding the correspondences when transitions not related to the correspondences are neglected. Similar to the projection for a trace (cf., Section 3.3), we define the projection of path as follows. Given a WF-net $N = (P, T, F)$, a set of transitions $H \subseteq T$, and a path $\pi \in \mathcal{P}_N^T$, the set $H_{\pi|j} = \{(x, t) \in \pi \mid x < j \wedge t \in H\}$ denotes the containment of transitions of H in π up to index $j \in \mathbb{N}$. Then, the projection $\tau_H(\pi)$ for a path $\pi \in \mathcal{P}_N^T$ of length n induced by H is defined as $\tau_H(\pi) = \bigcup_{i=0}^{|H_{\pi|n}|}(i, t)$ with $t \in H$, such that $\exists\, j \in \mathbb{N}\ [\ (j, t) \in \pi \wedge i = |H_{\pi|j}|\ ]$.

Definition 9 (Path Consistency of Correspondences). *Let $N = (P, T, F)$ and $N' = (P', T', F')$ be two WF-nets preprocessed with respect to two correspondences $C_1 = (T_1, T_3)$ and $C_2 = (T_2, T_4)$.*

- ○ *C_1 and C_2 are* path consistent from N to N', *iff for any path of transitions $\pi \in \mathcal{P}_N^T$, there is a path $\pi' \in \mathcal{P}_{N'}^{T'}$, such that for the partitioned projections $\rho_{(T_1,T_2)}(\tau_{(T_1 \cup T_2)}(\pi)) = \pi_1, \ldots, \pi_n$ and $\rho_{(T_3,T_4)}(\tau_{(T_3 \cup T_4)}(\pi')) = \pi'_1, \ldots, \pi'_m$ it holds $n = m$ and for all $i \in \mathbb{N}$ with $0 \leq i \leq n$:*
  - – $\pi_i \subseteq (T_1 \setminus \iota_{(T_1,T_2)}(N)) \Rightarrow \pi'_i \subseteq (T_3 \setminus \iota_{(T_3,T_4)}(N')$.
  - – $\pi_i \subseteq (T_2 \setminus \iota_{(T_1,T_2)}(N)) \Rightarrow \pi'_i \subseteq (T_4 \setminus \iota_{(T_3,T_4)}(N')$.
  - – $\pi_i \subseteq \iota_{(T_1,T_2)}(N) \Rightarrow \pi'_i \subseteq \iota_{(T_3,T_4)}(N')$.
- ○ *C_1 and C_2 are* path consistent, *iff they are path consistent in either direction.*

For our example setting in Fig. 1 and correspondences $C1$ and $C2$, we see that both models are path consistent. All paths from the initial to the final place in both models shows the same (projected) partitionings, i.e., a non-interleaving transition related to $C1$ is followed by an interleaving transition of both correspondences. In contrast, correspondences $C3$ and $C4$ are not path consistent. Even if both nets are preprocessed, for instance, model (a) contains a path in which non-interleaving transitions related to correspondence $C3$ (transitions 5 and 6) are followed by a non-interleaving transition related to $C2$ (transition 7). Such a path does not exit in model (b) as transitions E, F, and G are interleaving.

4.3 Reasoning on Behaviour Compatibility

We illustrated the dependency between path consistency of a pair of correspondences and their behaviour compatibility using our example. In fact, we show that both notions coincide for sound free-choice WF-nets, see [6] for the proofs.

Theorem 2. *Let $N = (P,T,F)$ and $N' = (P',T',F')$ be two preprocessed sound free-choice WF-nets, and $C_1 = (T_1,T_3)$, $C_2 = (T_2,T_4)$ two correspondences. Then, path consistency and projection compatibility of C_1 and C_2 coincide.*

Based thereon, behaviour compatibility of correspondences between sound free-choice WF-nets can be decided efficiently.

Corollary 1. *The following problem can be solved in $O(n^3)$ time with n as the maximum of the number of nodes of both nets.*
For two correspondences between two sound free-choice WF-nets, to decide projection and protocol compatibility.

5 Evaluation

We evaluated our techniques for deciding behaviour compatibility, by applying them to a collection of similar model pairs, between which correspondences were already identified. For both notions of behaviour compatibility, we first identified incompatibilities for all pairs of models with respect to their correspondences. Second, we investigated the resulting incompatibilities, to determine whether they represent information that is useful to the designer.

The collection consisted of 10 pairs that were taken from Dutch municipalities. Each of these pairs represents a standard process [20] and an implementation of this standard process by a municipality. Each process model from the collection has, on average, 17.9 nodes, with a minimum of 11 nodes and a maximum of 69 nodes for a single process model. The average number of arcs pointing into or out of a single node is 1.2. In total there were 190 correspondences between the model pairs, 31 of which were complex. All models were available as (or could be transformed into) free-choice WF-nets. In addition, we verified that all models are sound, such that the structural characterisation of behaviour compatibility as introduced in Section 4 could be leveraged. With our implementation of this approach, the analysis of both, projection and protocol

Table 1. Overview on compatible and incompatible correspondences

Type of Comp.	Type of Correspondence Pair	Compatible	Incompatible
Projection Comp.	Elementary	83% (1732)	17% (354)
	Complex	30% (115)	70% (274)
Protocol Comp.	Elementary	38% (86)	62% (143)
	Complex	8% (7)	92% (86)

compatibility, was done in milliseconds for most model pairs. For two pairs of models, it took up to several seconds to decide compatibility.

Projection Compatibility. For all 190 correspondences in total 2475 combinations of correspondences were to be investigated for projection compatibility. Table 1 shows the number of projection compatible and incompatible pairs for the correspondences. We say that a pair of correspondences is elementary, if both correspondences are elementary; if one is complex, we say that the pair is complex. The table shows that most elementary correspondences are projection compatible, while most complex correspondences are not. This result is not surprising, because complex correspondences are more complicated than elementary correspondences and, therefore, it is harder to make them compatible.

As a second step, we randomly selected 25 elementary and 25 complex incompatibilities to investigate whether they represented information that is useful to the designer. This was indeed the case. However, when studying these incompatibilites in detail, we established that there existed overlap between them in the sense that they could be traced back to the same cause for incompatiblity. Specifically, there were 26 pairs that had overlap with another pair (of the 26). If we considered each correspondence pair only once, there were only 8 cases of incompatibility; the 'common' pairs caused 3.25 incompatibilities on average. This kind of redundancy in the information that is presented to the designer is undesirable and leads to the conclusion that incompatibilities can be presented to the designer in a more compact manner.

Protocol Compatibility. For all pairs of models, we also derived the encapsulated models in order to assess protocol compatibility. To this end, we removed solely transitions representing activities that are not part of any correspondence and neglected additional NOP transitions realising the splitting and merging of control flow. However, for four out of our 10 pairs of models, we observed that at least for one model encapsulation led to a net that could not be normalised into a WF-net. In these cases, encapsulation led to a disconnect of the initial and the final place, such that both places were no longer connected by any path. As these models describe processes that are bound to failure (they cannot complete properly), they could not be investigated any further. For the remaining four model pairs, the normalised encapsulated nets were sound, such that our structural characterisation of behaviour compatibility could be exploited. As illustrated in Table 1, the amount of compatible correspondences is much lower than for the case of projection compatibility. This is mainly due to activities that have been introduced as intermediate steps when implementing the standard process. Apparently, such deviations are not in line with protocol compatibility. Due to the freedom of the municipalities to deviate from the reference process in such a way, the notion of projection compatibility seems to be more appropriate than protocol compatibility for this use case.

6 Related Work

Our work is related to three streams of research, *matching of process models*, *model specialisation*, and *process model similarity*.

In order to assess behaviour consistency, we postulate the existence of correspondences between activities of two process models. In some use cases, these correspondences are given implicitly, e.g., when deriving a custom process model from a reference model. Still, other use cases might require the explicit definition of correspondences, such that automatic support for suggesting correspondences is needed. To this end, techniques based on structural analysis and natural language processing have been proposed in order to identify correspondences between single activities [21, 22]. Recently, the ICoP framework has been introduced, which aims also at the detection of complex correspondences [23]. In addition, techniques known from the field of schema and ontology matching [13, 24] can be applied to detect correspondences between process model elements.

A behavioural model can be specialised by *refinement* and *extension* [5]. Refinement refers to the definition of an activity or a set thereof in more detail. Extension, in turn, refers to the act of adding new activities. Both transformations might or might not preserve one of the well-known behaviour equivalences, see [25]. Behaviour consistent refinements have been investigated in detail for many formal models, such as process algebras and Petri nets [25–28]. See [29] for a thorough survey on Petri net refinements. Obviously, the work on model refinement and extension has a different focus than our work. We target at an assessment of correspondences between models for which the concrete specialisation relation is not known. Still, transformations that preserve the introduced notions of projection and protocol compatibility need to be investigated. For the existing notions of behaviour inheritance, projection and protocol inheritance, a set of four inheritance preserving model transformations has been presented in [4].

Behaviour compatibility is a boolean criterion based on a behaviour equivalence. Process models that are related by correspondences might also be analysed regarding their behavioural similarity. Recently, the question of how to quantify behavioural similarity has received much attention [30]. Process similarity can be assessed by using behavioural abstractions [31], relating similar (sub-) traces of two models to each other [32], or quantifying the degree of state-based simulation [33]. These approaches typically focus on the complete behaviour of two models. Therefore, additional effort might be required to give diagnostic information with respect to the correspondences for a similarity value below one.

7 Conclusion

In this paper, we addressed the question of how to decide on the compatibility of two business process models. To this end, correspondences between both models are assumed to exist, whereas we do not impose any restrictions on the type of correspondences that can exist. In particular, there might be complex 1:n and or even n:m correspondences between activities of both models. Building upon the existing work on behaviour inheritance, we introduced the notions of projection and protocol compatibility of correspondences between process models. They guarantee that correspondences do not induce behavioural contradictions in terms of trace semantics, once activities that are not part of any correspondence are hidden or blocked. Besides the definition of these notions, our contribution is a structural characterisation of both notions for a pair of correspondences between sound free-choice WF-nets. Based thereon, behaviour compatibility is decided in $O(n^3)$ time with n as the maximum of the number of nodes of both nets. As a proof of concept, we applied our technique to determine the compatibility between 10 reference process models and 10 models that implement them.

Clearly, our contribution is of relevance not only for the use case of customising reference models. The application of behaviour inheritance has been advocated to solve other problems, such as those related to *dynamic change*, *information management* [2], and *service-oriented design* [1]. Dynamic change addresses the question how to ensure behavioural consistency for running process instance once the respective process model is adapted. Information management refers to an aggregated view on multiple variants of a process model. Service-oriented design addresses the issue of designing a business process that correctly implements a service that an organisation provides to its clients, as it is specified by another (abstract) process. Our notions of behaviour compatibility allow for tackling these problems in a broader context by going beyond elementary 1:1 relations between activities when comparing model behaviour. Still, this requires further investigations on model transformations that preserve behaviour compatibility. Although a formal discussion is beyond the scope of this paper, the aforementioned four transformation rules that preserve projection (and partly protocol) inheritance [4] can be assumed to preserve our notions of behaviour compatibility as well. The presence of complex correspondences, however, opens the space for investigations on transformation rules that consider the partitioning of activities induced by such correspondences. In future work, stricter notions of behaviour equivalence, such as branching bisimulation [25], might also be applied as the grounding for behaviour compatibility. Finally, the application of our technique in a case study shows that many redundant incompatibilities are notified to the process designer. Consequently, in future work we aim at developing a technique that presents incompatibilities in a compact manner.

References

1. Dijkman, R., Dumas, M.: Service-oriented design: a multi-viewpoint approach. IJCIS **13**(4) (2004) 337–368
2. van der Aalst, W.M.P.: Inheritance of business processes: A journey visiting four notorious problems. In: Petri Net Technology for Communication-Based Systems. Volume 2472 of LNCS., Springer (2003) 383–408
3. Guth, V., Oberweis, A.: Delta-analysis of petri net based models for business processes. In: Proc. of the 3rd Int. Conf. on Applied Informatics. (1997) 23–32
4. Basten, T., Aalst, W.: Inheritance of Behavior. JLAP **47**(2) (2001) 47–145
5. Schrefl, M., Stumptner, M.: Behavior-consistent specialization of object life cycles. ACM Trans. Softw. Eng. Methodol. **11**(1) (2002) 92–148
6. Weidlich, M., Dijkman, R., Weske, M.: Deciding Behaviour Compatibility of Complex Correspondences between Process Models. Technical report 11-2010, Hasso Plattner Institute `http://bpt.hpi.uni-potsdam.de/pub/Public/MatthiasWeidlich/bc_r.pdf`
7. Aalst, W.: The application of Petri nets to workflow management. Journal of Circuits, Systems, and Computers **8**(1) (1998) 21–66

8. Lohmann, N.: A feature-complete Petri net semantics for WS-BPEL 2.0. In: WS-FM. Volume 4937 of LNCS. (2008) 77–91
9. Dijkman, R.M., Dumas, M., Ouyang, C.: Semantics and analysis of business process models in BPMN. Information & Software Technology **50**(12) (2008) 1281–1294
10. Eshuis, R., Wieringa, R.: Tool support for verifying UML activity diagrams. IEEE Trans. Software Eng. **30**(7) (2004) 437–447
11. Desel, J., Esparza, J.: Free Choice Petri Nets. Cambridge University Press (1995)
12. Aalst, W.: Verification of workflow nets. In: ICATPN. Volume 1248 of LNCS. (1997) 407–426
13. Euzenat, J., Shvaiko, P.: Ontology Matching. Springer-Verlag (2007)
14. Hack, M.: Decidability Questions for Petri Nets. PhD thesis, M.I.T. (1976)
15. Valmari, A.: The state explosion problem. In: Petri Nets. Volume 1491 of LNCS., Springer (1996) 429–528
16. Kiepuszewski, B., Hofstede, A., Aalst, W.: Fundamentals of control flow in workflows. Acta Inf. **39**(3) (2003) 143–209
17. Aalst, W.: Workflow verification: Finding control-flow errors using petri-net-based techniques. In: BPM. Volume 1806 of LNCS. (2000) 161–183
18. Kovalyov, A., Esparza, J.: A polynomial algorithm to compute the concurrency relation of free-choice signal transition graphs. In: WODES. The Institution of Electrical Engineers (1996) 1–6
19. Weidlich, M., Mendling, J., Weske, M.: Efficient Consistency Measurement based on Behavioural Profiles of Process Models. IEEE Trans. on Software Engineering, 2010. To appear.
20. Documentair structuurplan (Accessed: 20 February 2009) `http://model-dsp.nl/`.
21. Nejati, S., Sabetzadeh, M., Chechik, M., Easterbrook, S.M., Zave, P.: Matching and merging of statecharts specifications. In: ICSE, IEEE CS (2007) 54–64
22. Dijkman, R., Dumas, M., García-Bañuelos, L., Käärik, R.: Aligning business process models. In: EDOC. (2009) 45–53
23. Weidlich, M., Dijkman, R., Mendling, J.: The ICoP framework: Identification of correspondences between process models. In: CAiSE. Volume 6051 of LNCS., Springer (2010) 483–498
24. Rahm, E., Bernstein, P.A.: A survey of approaches to automatic schema matching. VLDB Journal **10**(4) (2001) 334–350
25. van Glabbeek, R.J., Goltz, U.: Refinement of actions and equivalence notions for concurrent systems. Acta Inf. **37**(4/5) (2001) 229–327
26. Aceto, L., Hennessy, M.: Adding action refinement to a finite process algebra. Inf. Comput. **115**(2) (1994) 179–247
27. Quartel, D., Pires, L.F., van Sinderen, M.: On architectural support for behavior refinement in distributed systems design. Journal of Integrated Design and Process Science **6**(1) (2002) 1–30
28. Vogler, W.: Behaviour preserving refinement of petri nets. In: WG. Volume 246 of LNCS., Springer (1986) 82–93
29. Brauer, W., Gold, R., Vogler, W.: A survey of behaviour and equivalence preserving refinements of petri nets. In: ICATPN. Volume 483 of LNCS., Springer (1989) 1–46
30. Dumas, M., García-Bañuelos, L., Dijkman, R.M.: Similarity search of business process models. IEEE Data Eng. Bull. **32**(3) (2009) 23–28
31. van Dongen, B.F., Dijkman, R.M., Mendling, J.: Measuring Similarity between Business Process Models. In: CAiSE. Volume 5074 of LNCS., Springer (2008) 450–464
32. Wombacher, A.: Evaluation of technical measures for workflow similarity based on a pilot study. In: OTM. Volume 4275 of LNCS., Springer (2006) 255–272
33. Sokolsky, O., Kannan, S., Lee, I.: Simulation-based graph similarity. In: TACAS. Volume 3920 of LNCS., Springer (2006) 426–440