Isotactics as a Foundation for Alignment and Abstraction of Behavioral Models*

Artem Polyvyanyy[1], Matthias Weidlich[2], and Mathias Weske[3]

[1] Queensland University of Technology, Brisbane, Australia
artem.polyvyanyy@qut.edu.au
[2] Technion – Israel Institute of Technology, Haifa, Israel
weidlich@tx.technion.ac.il
[3] Hasso Plattner Institute at the University of Potsdam, Potsdam, Germany
mathias.weske@hpi.uni-potsdam.de

Abstract. There are many use cases in business process management that require the comparison of behavioral models. For instance, verifying equivalence is the basis for assessing whether a technical workflow correctly implements a business process, or whether a process realization conforms to a reference process. This paper proposes an equivalence relation for models that describe behaviors based on the concurrency semantics of net theory and for which an alignment relation has been defined. This equivalence, called isotactics, preserves the level of concurrency of aligned operations. Furthermore, we elaborate on the conditions under which an alignment relation can be classified as an abstraction. Finally, we show that alignment relations induced by structural refinements of behavioral models are indeed behavioral abstractions.

1 Introduction

Behavioral models can serve different purposes: communicating ideas, simulating systems, or defining precise execution instructions. Tailoring a model for a certain purpose leads to the existence of several "related" models of the same original. Each model shall be appropriate for its purpose. In business process management (BPM), behavioral models on the business level should, thus, concentrate on aspects that are important from a business perspective, while technical implementation aspects are disregarded. Technical models, in turn, need to describe activities required for implementation, such as data mapping or error handling.

Given a set of related models, it is often feasible to map semantically related, or *aligned*, (groups of) modeling constructs across models. Fig. 1 shows two aligned behavioral models captured using BPMN [1] language. Both models describe behaviors of performing "product at the market" research. Related groups of tasks are enclosed in the areas denoted by dotted borders and connected by dashed lines, e.g., task "*Study product*" in Fig. 1(a) is aligned with tasks "*Select product*" and "*Collect product info*" in Fig. 1(b) by the semantical concept α.

There are many use cases in BPM that require the comparison of aligned behavioral models. Verifying equivalence, for instance, is the basis for assessing whether a technical workflow correctly implements a business process, or whether a process realization conforms to a reference process [2]. Further, abstraction of

* This work was initiated while the first author was with Hasso Plattner Institute.

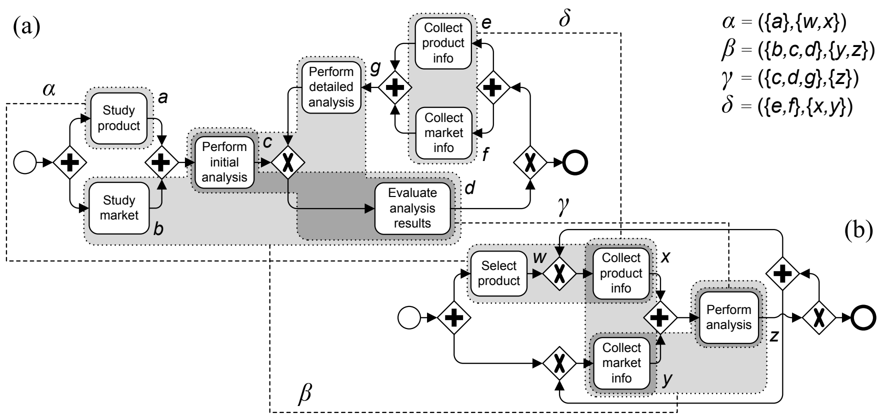


Fig. 1. Alignment of BPMN diagrams

behavioral models, i.e., an alignment that implies information loss from one model to another, plays an important role in managing model complexity [3]. Despite these observations, as of today, there is a lack of formal grounding for verifying behavior equivalence without imposing any assumption on the structure of an alignment relation. Note that in general groups of aligned modeling constructs may be of arbitrary size and can even overlap.

In this paper, we study models that describe behavior as a partially ordered, usually infinite, set (poset) of events. Here, an *event* is a phenomenon located at a single point in time [4]. For these models, we answer the question of *how to define an equivalence relation that preserves order and concurrency of event occurrences without imposing any assumptions on the structure of the alignment*. To answer this question, we introduce the notion of *isotactics*, which allows for comparing aligned behavioral models, very much like bisimulation [5,6] allows the comparison of non-aligned models. Formally, isotactics is implemented using the concept of a *tactic*, i.e., a poset of groups of events labeled with the same semantical concepts of the alignment relation. As such, our contribution is a first step toward a spectrum of equivalences for aligned behavioral models. Moreover, we show that common structural abstraction techniques for behavioral models [7,8] indeed preserve our new equivalence notion; however, we also show that isotactics makes the limitations of such structural approaches explicit. Note that the alignment construction, i.e., the discovery of semantically related constructs, is taken for granted; it can either be performed manually or automatically [9,10].

We proceed as follows: The next section presents preliminary notions. Section 3 is devoted to the discussion of alignment of behavioral models, which leads to the definition of isotactics. Then, Section 4 studies how this notion can aid in explaining the abstraction relation between behavioral models. Section 5 elaborates on the application of proposed notions. Finally, we draw conclusions.

2 Preliminaries

First, Section 2.1 discusses Petri nets – a formalism to which many languages for modeling behavior can be traced back [11]. Section 2.2 talks about causal nets – a way of representing concurrent runs of net systems.

2.1 Petri Nets

Petri nets are a well-known formalism for modeling behaviors.

Definition 1 (Petri net). A *Petri net*, or a *net*, $N = (P, T, F)$ has finite disjoint sets P of *places* and T of *transitions*, and the *flow* relation $F \subseteq (P \times T) \cup (T \times P)$.

For a *node* $x \in P \cup T$, $\bullet x = \{y \mid (y, x) \in F\}$ is the *preset*, and $x\bullet = \{y \mid (x, y) \in F\}$ is the *postset* of x. $Min(N)$ is the set of places of N with empty preset, i.e., $\{p \in P \mid \bullet p = \emptyset\}$. A node $x \in P \cup T$ is an *input* (*output*) of a node $y \in P \cup T$, iff $x \in \bullet y$ ($x \in y\bullet$). For $X \subseteq P \cup T$, let $\bullet X = \bigcup_{x \in X} \bullet x$ and $X\bullet = \bigcup_{x \in X} x\bullet$. For a binary relation R, we denote by R^+ (R^*) the transitive (and reflexive) closure of R.

In the graphical notation, places are represented by circles, transitions by rectangles, and flow relation by directed edges (see Fig. 2). Execution semantics of Petri nets is based on states and state transitions and best perceived as a "token game". The state of a net is represented by a *marking*, which describes a distribution of *tokens* on the net's places. Whether a transition is *enabled* at a marking depends on the tokens in its input places. An enabled transition can *occur*, which leads to a new marking of the net.

To formalize semantics, we identify the flow relation F with its characteristic function on the set $(P \times T) \cup (T \times P)$.

Definition 2 (Net semantics). Let $N = (P, T, F)$ be a net.

- $M : P \to \mathbb{N}_0$ is a *marking*, or a *state*, of N assigning each place $p \in P$ a number $M(p)$ of *tokens* in p; $\mathbb{N}_0$ denotes the set of all natural numbers including zero. With $[p]$, we denote the marking in which place p contains just one token and all other places contain no tokens. We identify M with the multiset containing $M(p)$ copies of p for every $p \in P$.
- For a transition $t \in T$ and a marking M of N, t is *enabled* at M, written $M[t\rangle$, iff $\forall p \in \bullet t : M(p) \geq 1$.
- If $t \in T$ is enabled at M, then t can *occur*, which leads to a new marking M' and the *step* $M[t\rangle M'$ of N with $M'(p) = M(p) - F(p, t) + F(t, p)$, $p \in P$.
- A *net system*, or a *system*, is a pair $S = (N, M_0)$, where M_0 is a marking of N. M_0 is called the *initial marking* of N.
- A sequence of transitions $\sigma = t_1 \dots t_n$, $n \in \mathbb{N}_0$, $t_i \in T$, $i \in 1 \dots n$, of net system $S = (N, M_0)$ is a *firing sequence* in S iff there exists a sequence of steps $(N, M_0)[t_1\rangle(N, M_1) \dots (N, M_{n-1})[t_n\rangle(N, M_n)$ which leads from marking M_0 to marking M_n via a (possibly empty) sequence of intermediate markings $M_1 \dots M_{n-1}$.
- For any two markings M and M' of N, M' is *reachable* from M, denoted by $M' \in [N, M\rangle$ iff there exists a *run* of N, i.e., there exists a firing sequence σ leading from M to M'.

In the following, we shall refer to the *natural marking* of a net N; the natural marking puts one token at every place from the set $Min(N)$ and no tokens elsewhere. In the graphical notation, it is accepted that tokens are drawn as black dots inside places. Fig. 2 shows two net systems (in natural initial markings); the systems correspond to the BPMN diagrams in Fig. 1. Transitions which correspond to tasks in BPMN diagrams are drawn as rectangles with labels inside; the labels are the short-names of tasks which appear next to each task in Fig. 1.

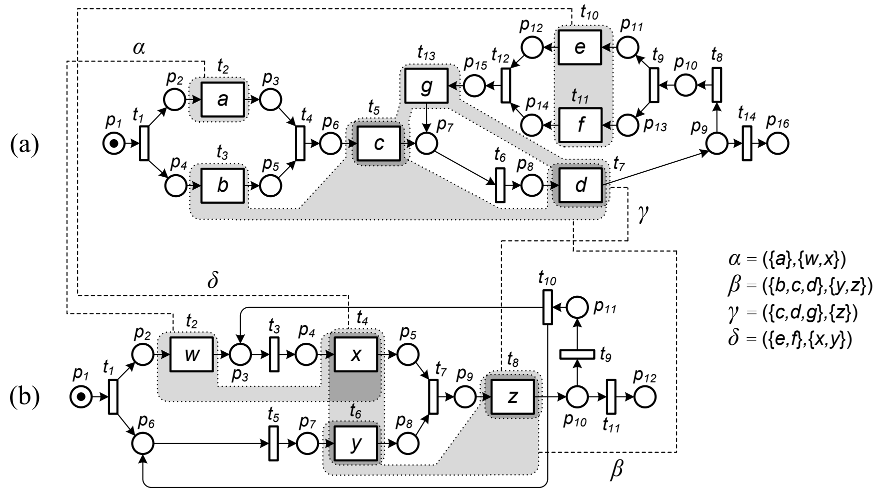


Fig. 2. Net systems that correspond to the BPMN diagrams in Fig. 1

2.2 Causal Nets and Processes

In this section, we present causal nets [4,12] and discuss how they can be used to capture *processes*, or *concurrent runs*, of net systems. Causal nets provide the foundation for general net theory [13].

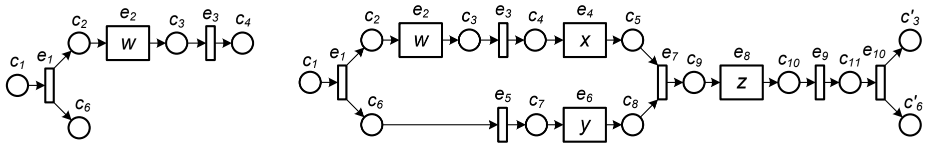
Definition 3 (Causal net). A net $N = (B, E, G)$ is a *causal* net, iff :
- for each $b \in B$ holds $|\bullet b| \leq 1$ and $|b\bullet| \leq 1$, and
- N is acyclic, i.e., G^+ is irreflexive.

Elements of E are called *events* and elements of B are called *conditions*. The events of causal nets are usually used to describe occurrences of "atomic events", e.g., occurrences of transitions of a net system. An occurrence of an event e is associated with a state in which all its preconditions ($\bullet e$) hold, and the effect of its occurrence is that all its preconditions cease to hold, and all its postconditions ($e\bullet$) begin to hold [4]. Given a causal net $N = (B, E, G)$, the concurrency relation of N is defined by $\|_N = ((B \cup E) \times (B \cup E)) \setminus (G^+ \cup (G^+)^{-1})$ (we omit the subscript if the context is clear). Note that $\|_N$ is symmetric and reflexive. Moreover, every two nodes of a causal net are either in the concurrency or in the (inverse) causal relation, where nodes x and y are causal if and only if xG^+y. The causal relation specifies a dependency between events of a causal net, such that if $e_1 G^+ e_2$, where $e_1, e_2 \in E$, then in the net system composed of the causal net and its natural initial marking, e_2 cannot occur without e_1 having priorly occurred.

A *process* of a net system is a causal net together with a mapping which allows interpreting the net as a concurrent run of the net system[4]. Prior to proceeding with defining the notion of a process, we present the notion of a *cut*. A *cut* of a causal net is the maximal co-set with respect to set inclusion, where a *co-set* is a set of pairwise concurrent conditions. A process is then defined as follows.

Definition 4 (Process). A *process* $\pi = (N_\pi, \rho)$ of a net system $S = (N, M_0)$, $N = (P, T, F)$, has a causal net $N_\pi = (B, E, G)$ and a function $\rho : B \cup E \to P \cup T$:

[4] Not to be confused with a business process or a process model.



(a) (b)

Fig. 3. Processes of the net system in Fig. 2(b)

- ∘ $\rho(B) \subseteq P,\ \rho(E) \subseteq T$ (ρ preserves the nature of nodes),
- ∘ $Min(N_\pi)$ is a cut, which corresponds to the initial marking M_0, that is $\forall\ p \in P : M_0(p) = |\rho^{-1}(p) \cap Min(N_\pi)|$ (π starts at M_0), and
- ∘ $\forall\ e \in E\ \forall\ p \in P : (F(p, \rho(e)) = |\rho^{-1}(p) \cap \bullet e|) \wedge (F(\rho(e), p) = |\rho^{-1}(p) \cap e\bullet|)$ (ρ respects the environment of transitions).

We refer to S as the *originative* system of π. A process π of S is *initial*, iff $E = \emptyset$.

Given a run of a net system, one can construct a unique process induced by the run (observe that the inverse does not hold). The starting point of the construction is a causal net composed of conditions that correspond to places from the initial marking of the net system and no events. The construction proceeds by stepwise appending events to the causal net. Each appended event corresponds to a transition in the run. Events are appended in the order in which corresponding transitions appear in the run. Each fresh event e which gets appended to the causal net and has corresponding transition t is appended together with output conditions, which must correspond to output places of t. Note that input conditions of e, which must correspond to input places of t, must be chosen from the conditions of the causal net with empty postsets.

Every process of a net system describes a family of runs together with information on concurrent nodes that participate in the runs. Processes of a net system can be in a prefix relation. Process π' *is an extension of* process π if during construction of π', induced by some run of the system, it is possible to observe π. Consequently, process π *is a prefix of* π'.

Definition 5 (Prefix of a process). Let $\pi = (N_\pi, \rho)$, $N_\pi = (B, E, G)$, be a process of a net system. Let c be a cut of N_π and let $c^{\downarrow}$ denote the set of nodes $\{x \in B \cup E \mid \exists\ y \in c : (x, y) \in G^*\}$. A process π_c *is a prefix of* π up to (and including) c, iff $\pi_c = ((B \cap c^{\downarrow}, E \cap c^{\downarrow}, G \cap (c^{\downarrow} \times c^{\downarrow})), \rho|_{c^{\downarrow}})$.

Fig. 3 shows two processes of the net system in Fig. 2(b), the process in Fig. 3(a) being a prefix of the one in Fig. 3(b). Here, event e_x corresponds to transition t_x of the originative system, i.e., $\rho(e_x) = t_x$ (for each event e_x), and condition c_y corresponds to place p_y of the originative system, i.e., $\rho(c_y) = p_y$ (for each condition c_y). Conditions c'_3 and c'_6 in Fig. 3(b) correspond to places p_3 and p_6 of the originative system, and represent second occurrences of respective places in the process. Both systems in Fig. 3 induce infinitely many processes.

3 Alignment

Alignment can be seen as the generic setting in which two models can be compared. We first reflect on the alignment of conceptual models in general in Section 3.1. Then, Section 3.2 turns the focus to net systems. Section 3.3 proposes the notion of isotactics to decide whether two aligned net systems show equivalent behaviors.

3.1 Alignment of Conceptual Models

The alignment of conceptual models has its roots in the field of data integration [14,15]. Despite terminological differences even in this field, cf., [16], a common interpretation defines an alignment as an association between semantically related entities of different models, e.g., between attributes of data schemas.

Following [15], an alignment consists of a set of *correspondences* between two models. Each correspondence relates two sets of entities of both models to each other. If both those sets are singletons, we speak of an elementary correspondence. Otherwise, the correspondence is called 1:n or n:m complex. The identification of correspondences, i.e., the construction of an alignment, is called *matching*.

A correspondence associates entities with each other, but does not define the semantics of this relation. Semantics is defined by extending an alignment toward a *mapping* comprising mapping expressions. Those are directed and define how the instances of entities of one model are transformed into instances of entities of another model. Consider a 2:1 complex correspondence between integer attributes of data schemas. A mapping expression may define the sum of two values from one schema as equivalent to a single value in the other schema.

Alignments and mappings of conceptual models may be checked for validity using a variety of properties. In the field of data integration, for instance, satisfiability and losslessness have been investigated [17]. The former holds for a mapping between two schemas if there is a pair of instances of either schema, i.e., a pair of data value tuples, that satisfies the constraints of the mapping. Losslessness relates to the result set that may be queried. If a mapping is lossless, all instances returned by a query on one schema have a counterpart in the other schema that is derived by the mapping.

3.2 Alignment of Behavioral Models

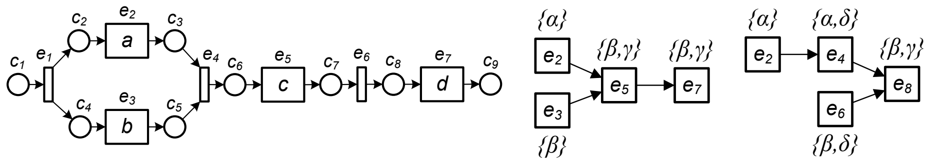
The notion of an alignment discussed for conceptual models in general can directly be applied to behavioral models, e.g., Petri net systems. Correspondences are then defined between sets of semantically related activities, i.e., transitions of different net systems. Formally, we capture such an alignment as follows.

Definition 6 (Alignment of net systems).
Let $S_1 = (N_1, M_1)$, $N_1 = (P_1, T_1, F_1)$, and $S_2 = (N_2, M_2)$, $N_2 = (P_2, T_2, F_2)$, be net systems. A set $\bowtie \subseteq \mathcal{P}_{\geq 1}(T_1) \times \mathcal{P}_{\geq 1}(T_2)$ is called an *alignment* of S_1 and S_2.[5]

Given an alignment $\bowtie$, we denote by $dom_{\bowtie}$ and $cod_{\bowtie}$ the domain and codomain of $\bowtie$, respectively. Recently, approaches for identifying correspondences between behavioral models have been presented [9,10]. Even though fully automatic identification of correspondences is hard to achieve, these works provide support for constructing an alignment in a semi-automated manner.

Again, correspondences between behavioral models capture only the relatedness, not the exact semantics in the sense of a mapping. That is, corresponding activities may not relate to the same real-world activities. For the models in Fig. 1, for instance, "*Study product*" may involve more than "*Select product*" and "*Collect product info*". Nevertheless, the activities are semantically related

[5] $\mathcal{P}_{\geq 1}(S)$ denotes the set of all non-empty subsets of a set S, including S itself.



(a) (b) (c)

Fig. 4. (a) A process of the net system in Fig. 2(a), (b) the set abstraction of the process in (a), and (c) the set abstraction of the process in Fig. 3(b)

and are considered to be equivalent for any analysis of the alignment. For a complex correspondence, sets of activities are considered to be equivalent. As such, analysis of an alignment is founded on these sets instead of single activities.

Before, we discussed properties of alignments in the field of data integration, i.e., satisfiability and losslessness. These properties may be translated into the domain of behavioral models. Satisfiability then requires the existence of a single process that is possible in two net systems after the corresponding transitions have been resolved. Apparently, this is a rather weak requirement. Drawing the analogy to behavioral models for losslessness yields a stricter criterion. It requires that the characteristics of all processes of one net system are preserved in the processes of the other net system once the correspondences have been resolved.

3.3 Isotactics of Aligned Behavioral Models

To decide if two net systems show equivalent behaviors under a given alignment, we rely on a comparison of their processes. Therefore, we first need to clarify how a single process is interpreted once we consider not only single events, but groups thereof as being semantically related. Given a process of a net system and subsets of its transitions, a set abstraction of the process captures its interpretation by relating to all events that represent occurrences of transitions from the subsets.

Definition 7 (Process set abstraction).
Let $S = (N, M_0)$, $N = (P, T, F)$, be a net system, $\pi = (N_\pi, \rho)$, $N_\pi = (B, E, G)$, be a process of S, and $\kappa \subseteq \mathcal{P}_{\geq 1}(T)$. The *set abstraction of* π *with respect to* κ, denoted by $\alpha_\kappa(\pi) = (H, \prec, \xi)$, is defined by the set of events $H = \{e \in E \mid \exists\, k \in \kappa : \rho(e) \in k\}$, the relation $\prec$, which is the restriction of the causal relation of N_π to H, and the function $\xi : H \to \mathcal{P}_{\geq 1}(\kappa)$ such that $\xi(e) = \{k \in \kappa \mid \rho(e) \in k\}$, $e \in H$.

Fig. 4(b) and Fig. 4(c) show set abstractions of the processes in Fig. 4(a) and Fig. 3(b), respectively. In both abstractions, we use the sets of transitions induced by the alignment depicted in Fig. 1 as κ. In the figures, boxes represent events whose corresponding transitions belong to at least one set in κ, i.e., they are visible with respect to κ; note that other events are considered to be silent for the purpose of alignment. Edges encode causal relations between events, e.g., $e_2 \prec e_5$ and $e_3 \prec e_5$ in Fig. 4(b). Finally, every event $e \in H$ gets labeled with a subset of κ; the subset is composed of elements of κ which contain the transition that corresponds to e, e.g., event e_5 in Fig. 4(b) corresponds to transition t_5 in Fig. 2(a), which is induced by task "*Perform initial analysis*" in Fig. 1 that participates in β and γ correspondences of the alignment. Essentially, a process set abstraction is an *elementary event structure* [4] composed of events that are visible as much as the alignment is concerned. In [4], the authors accept the equality of elementary event structures as an appropriate equivalence notion for causal nets; the claim is supported by proposing translations between both notations. We agree with this line of argument and accept two processes as equivalent if and only if their set abstractions are isomorphic, i.e., if and only if one can define a causality-preserving bijection between events.

As a next step, we relate process set abstractions to an alignment between net systems, that is, we decide whether an alignment between set abstractions of two processes of the net systems can be deduced from the given alignment between their transitions. This is the case if events in both set abstractions can be partitioned such that one can define a bijection relation between the partitions for which any two events taken from one abstraction and different parts of the partition, and any two events taken from the related parts of the partition of the other abstraction, are causally related in a similar way[6].

We refer to the partitions of events in set abstractions (see the discussion above) as process *tactics*. Let $(H_i, \prec_i, \xi_i)$ be a process set abstraction and let $h_1 \subseteq H_i$ and $h_2 \subseteq H_i$ be non-empty disjoint sets of events, then h_1 and h_2 are in *causal* relation, written $h_1 \prec_i h_2$, iff for every pair $(e_1, e_2) \in h_1 \times h_2$ holds $e_1 \prec_i e_2$; note that in the following we shall omit subscript i where the context is clear.

Definition 8 (Process tactic). Let $\alpha = (H, \prec, \xi)$ be the set abstraction of process π w.r.t. κ. A partition $\mathcal{H}$ of H is a *tactic* of α w.r.t κ iff :
- for every part $h \in \mathcal{H}$ there exists $k \in \kappa$ such that for every event $e \in h$ holds $k \in \xi(e)$, i.e., $\forall\ h \in \mathcal{H}\ \exists\ k \in \kappa\ \forall\ e \in h : k \in \xi(e)$ ($\mathcal{H}$ respects κ), and
- for every two parts $h_1, h_2 \in \mathcal{H}$, $h_1 \neq h_2$, holds either $h_1 \prec h_2$, or $h_2 \prec h_1$, or for each $(e_1, e_2) \in h_1 \times h_2$ holds $(e_1, e_2), (e_2, e_1) \notin \prec$ ($\mathcal{H}$ respects causality).

Every part of a tactic describes a *complex* event which stands for an occurrence of at least one and usually several semantically (by alignment) related transitions of the net system. A tactic, therefore, can be seen as a poset of complex events. Every set abstraction of a process has a *trivial* tactic, i.e., a tactic in which each of its parts is a singleton. Usually, a process set abstraction can be characterized by several tactics. Finally, aligned process set abstractions are defined as follows.

Definition 9 (Aligned process set abstractions).
Let π_1 and π_2 be processes of net systems S_1 and S_2, respectively. Let $\bowtie$ be an alignment of S_1 and S_2. Process set abstractions $\alpha_{dom_\bowtie}(\pi_1) = (H_1, \prec_1, \xi_1)$ and $\alpha_{cod_\bowtie}(\pi_2) = (H_2, \prec_2, \xi_2)$ are *aligned with respect to* $\bowtie$, denoted by $\alpha_{dom_\bowtie}(\pi_1) \bowtie \alpha_{cod_\bowtie}(\pi_2)$, iff there exist tactics $\mathcal{H}_1$ and $\mathcal{H}_2$ of π_1 and π_2, respectively, and a bijection $\chi : \mathcal{H}_1 \rightarrow \mathcal{H}_2$ such that: (i) for every $\chi(h_1) = h_2$, $h_1 \in \mathcal{H}_1$, there exists $(x, y) \in \bowtie$ such that $\forall e \in h_1 : x \in \xi_1(e)$ and $\forall e \in h_2 : y \in \xi_2(e)$ (χ respects alignment), and (ii) $\forall u, v \in \mathcal{H}_1 : u \prec_1 v \Leftrightarrow \chi(u) \prec_2 \chi(v)$ (χ respects causality).

We refer to χ as the alignment between tactics of process set abstractions. We say that processes are aligned if their abstractions are aligned. Apparently, the two set abstractions in Fig. 4(b) and Fig. 4(c) are not equivalent in the sense of [4],

[6] A partition of a set is a collection of disjoint subsets of the set whose union is the set.

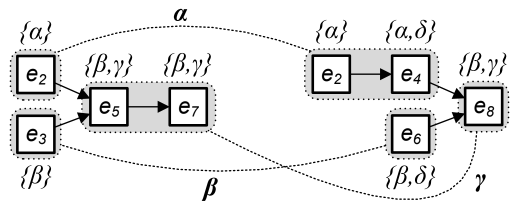


Fig. 5. Aligned process set abstractions

i.e., there exists no causality-preserving bijection between event sets. Nevertheless, one can rely on tactics to compare these set abstractions. Fig. 5 shows aligned process set abstractions from Fig. 4(b) and Fig. 4(c). In the figure, areas denoted by dotted borders with grey backgrounds define tactics (those which participate in the alignment). The dashed lines depict a bijection relation between the tactics and are labeled with semantical correspondences of the alignment from Fig. 1. Observe that the part $\{e_5, e_7\}$ of the tactic on the left can also be related to the part $\{e_8\}$ of the tactic on the right by using correspondence β; however, the existence of a correspondence is sufficient to decide for alignment.

Having defined the alignment of processes, we are able to define when the behavior of one net system can be mirrored by another net system under a given alignment. A system *covers the tactic* of another system if every process of the former system has a corresponding process in the latter system which mimics the behavior once abstractions have been applied. Formally, we capture this by a set that comprises pairs of aligned processes from both systems and require that the set is closed under process extensions. Note that the style of the next definition is inspired by the definitions of concurrent bisimulations in [18].

Definition 10 (Tactic coverage).
Let $\bowtie$ be an alignment of net systems S_1 and S_2. S_2 *covers the tactic of* S_1 *with respect to* $\bowtie$, denoted by $S_1 \leqslant_{\bowtie} S_2$, iff there exists a set $\mathcal{I} \subseteq \{(\pi_1, \pi_2)\}$ such that:
(i) π_1 is a process of S_1 and π_2 is a process of S_2.
(ii) If π_0^1 and π_0^2 are the initial processes of S_1 and S_2, respectively, $(\pi_0^1, \pi_0^2) \in \mathcal{I}$.
(iii) If $(\pi_1, \pi_2) \in \mathcal{I}$, then $\alpha_{dom_{\bowtie}}(\pi_1) \bowtie \alpha_{cod_{\bowtie}}(\pi_2)$ holds.
(iv) For each $(\pi_1, \pi_2) \in \mathcal{I}$ holds that if π_1' is an extension of π_1 then there exists $(\pi_1', \pi_2') \in \mathcal{I}$ where π_2' is an extension of π_2.
(v) For each $(\pi_1, \pi_2) \in \mathcal{I}$ holds that if π_1' is an extension of π_1 then for each π_2' extension of π_2 such that $\alpha_{dom_{\bowtie}}(\pi_1') \bowtie \alpha_{cod_{\bowtie}}(\pi_2')$ holds $(\pi_1', \pi_2') \in \mathcal{I}$.

We shall denote with $\mathcal{I}_{\bowtie}$ the set of process pairs used to decide $S_1 \leqslant_{\bowtie} S_2$. If each of two aligned net systems covers the tactic of the other one with respect to the alignment, we refer to the systems as *isotactic* with respect to the alignment.

Definition 11 (Isotactic net systems).
Let $\bowtie$ be an alignment of net systems S_1 and S_2. S_1 *and* S_2 *have equal tactics, or are isotactic, with respect to* $\bowtie$, denoted by $S_1 \doteqdot_{\bowtie} S_2$, iff $S_1 \leqslant_{\bowtie} S_2$ and $S_2 \leqslant_{\bowtie^{-1}} S_1$.

One can check that systems from Fig. 2 are isotactic with respect to the alignment proposed in Fig. 1. In net systems which are isotactic with respect to an alignment relation, it holds that for every process that one can observe in one net system one can also observe a process in the other net system so that the set abstractions of these processes with respect to the domain and the codomain of the alignment relation, respectively, are aligned (and vice versa). Intuitively, an alignment of process set abstractions denotes the equivalence of the processes with respect to complex events induced by the alignment and must be closed under process extensions. Consequently, isotactics preserves the order of occurrence for groups of transitions of net systems that are related by the alignment relation, as well as concurrent enabling of transitions in these groups. To define these properties, we need a relation to capture concurrent enabling of transitions in a net system (not to be confused with the concurrency relation for causal nets). For a net system $S = (N, M)$, $N = (P, T, F)$, the *transition concurrency relation* of S, denoted by $\|_S$, contains all pairs of transitions $(t_1, t_2) \in T \times T$ for which there exists a marking $M' \in [N, M\rangle$, such that $\bullet t_1 \uplus \bullet t_2 \subseteq M'$.

Theorem 1. *Let $S_1 = (N_1, M_1)$ and $S_2 = (N_2, M_2)$ be net systems and $\bowtie$ be an alignment of S_1 and S_2, s.t. $S_1 \leqslant_{\bowtie} S_2$ holds. Let $\alpha_\triangleright, \beta_\triangleright \in dom_\bowtie$ and let $t^1_\alpha \in \alpha_\triangleright$ and $t^1_\beta \in \beta_\triangleright$ be transitions of N_1 s.t. $t^1_\alpha \notin \beta_\triangleright$, $t^1_\beta \notin \alpha_\triangleright$, and for every $\gamma_\triangleright \in dom_\bowtie \setminus \{\alpha_\triangleright, \beta_\triangleright\}$ holds $\{t^1_\alpha, t^1_\beta\} \cap \gamma_\triangleright = \emptyset$. Then, the following properties hold:*

(1) If there exists a firing sequence $\sigma_1 = t^1_1 \dots t^1_\alpha \dots t^1_\beta$ in S_1, then there exists a firing sequence $\sigma_2 = t^2_1 \dots t^2_\alpha \dots t^2_\beta$ in S_2 s.t. there is $\alpha_\triangleleft, \beta_\triangleleft \in cod_\bowtie$ for which holds $(\alpha_\triangleright, \alpha_\triangleleft), (\beta_\triangleright, \beta_\triangleleft) \in \bowtie$, $t^2_\alpha \in \alpha_\triangleleft$, and $t^2_\beta \in \beta_\triangleleft$.

(2) If $t^1_\alpha \|_{S_1} t^1_\beta$, then there exist transitions t^2_α, t^2_β of N_2 s.t. $t^2_\alpha \|_{S_2} t^2_\beta$ and there is $\alpha_\triangleleft, \beta_\triangleleft \in cod_\bowtie$ for which holds $(\alpha_\triangleright, \alpha_\triangleleft), (\beta_\triangleright, \beta_\triangleleft) \in \bowtie$, $t^2_\alpha \in \alpha_\triangleleft$, and $t^2_\beta \in \beta_\triangleleft$.

Proof. Let $\pi_1 = (N_{\pi_1}, \rho_1)$, $N_{\pi_1} = (B_1, E_1, G_1)$, be a process of S_1 such that $e^1_\alpha, e^1_\beta \in E_1$, where $\rho_1(e^1_\alpha) = t^1_\alpha$ and $\rho_1(e^1_\beta) = t^1_\beta$, and: (1) ρ_1 is a bijection between events in E_1 and transitions in σ_1, (2) $e^1_\alpha \|_{N_{\pi_1}} e^1_\beta$. Since $S_1 \leqslant_\bowtie S_2$, there is a process $\pi_2 = (N_{\pi_2}, \rho_2)$, $N_{\pi_2} = (B_2, E_2, G_2)$, of S_2, such that $\alpha_{dom_\bowtie}(\pi_1) \bowtie \alpha_{cod_\bowtie}(\pi_2)$ with set abstractions $\alpha_{dom_\bowtie}(\pi_1) = (H_1, \prec_1, \xi_1)$ and $\alpha_{cod_\bowtie}(\pi_2) = (H_2, \prec_2, \xi_2)$. Moreover, there exist tactics $\mathcal{H}_1$ and $\mathcal{H}_2$ of events in $\alpha_{dom_\bowtie}(\pi_1)$ and $\alpha_{cod_\bowtie}(\pi_2)$, respectively, and a bijection $\chi : \mathcal{H}_1 \to \mathcal{H}_2$ which respects alignment and causality, cf., Definition 9. Let $h^1_\alpha, h^1_\beta \in \mathcal{H}_1$ be such that $e^1_\alpha \in h^1_\alpha$ and $e^1_\beta \in h^1_\beta$. Let $h^2_\alpha, h^2_\beta \in \mathcal{H}_2$ be such that $\chi(h^1_\alpha) = h^2_\alpha$ and $\chi(h^1_\beta) = h^2_\beta$. Let $e^2_\alpha \in h^2_\alpha$ and $e^2_\beta \in h^2_\beta$ be events of H_2. It holds that $h^1_\alpha \neq h^1_\beta$ due to the fact that χ preserves alignment and there exists no $\delta_\triangleright \in dom_\bowtie$ that contains t^1_α and t^1_β, i.e., $\nexists\, \delta_\triangleright \in dom_\bowtie : \{t^1_\alpha, t^1_\beta\} \subseteq \delta_\triangleright$.

(1) It holds that either $e^1_\alpha\ G^+_1\ e^1_\beta$ or $e^1_\alpha \|_{N_{\pi_1}} e^1_\beta$. Since $h^1_\alpha \neq h^1_\beta$ and χ preserves causality, it holds that either $e^2_\alpha\ G^+_2\ e^2_\beta$ or $e^2_\alpha \|_{N_{\pi_2}} e^2_\beta$. Hence, there is a firing sequence in S_2 in which transition $\rho_2(e^2_\alpha)$ fires before transition $\rho_2(e^2_\beta)$.

(2) Since $h^1_\alpha \neq h^1_\beta$ and χ preserves causality, it holds that $e^2_\alpha \|_{N_{\pi_2}} e^2_\beta$. Hence, it holds that $\rho_2(e^2_\alpha) \|_{S_2} \rho_2(e^2_\beta)$. □

Based on Theorem 1, we say that isotactics is (1) *order preserving* and (2) *concurrency preserving*. For instance, the order of t_3 and t_{13} and the concurrency of t_2 and t_3 from the system in Fig. 2(a) is preserved in the system in Fig. 2(b).

4 Abstraction

Abstraction can be seen as a special case of alignment if certain properties are satisfied. Next, we elaborate on these properties and define abstraction using the notion of tactic coverage. Again, we first reflect on the abstraction of conceptual models in Section 4.1 before we turn the focus to behavioral models in Section 4.2.

4.1 Abstraction of Conceptual Models

Abstraction is at the core of model creation, which comprises the mapping and reducing the entities of a problem domain for a certain purpose [19]. Abstraction is not limited to the process of creating a model for an (existing or non-existing) real world entity, though. The abstracted original may be a model as well. Entities of one model are then mapped to a more abstract model representing a reduced representation of the former. Abstraction of a model, thus, yields a second model that is aligned with the original model.

Abstraction of conceptual models relies on two elementary operations, aggregation and elimination. Aggregation refers to grouping entities that are semantically related. They have a joint representation in the abstract model. As such, aggregation leads to complex correspondences between the original model and the abstract model. Elimination, in turn, refers to the act of omitting entities. Certain entities of a model may be without counterpart in the abstract model. Eliminated entities are not part of any correspondence between the original model and the abstract model. Along these lines, abstraction of conceptual models is, for instance, the basis of the superclass concept in object oriented modeling.

Against this background, one may decide whether two conceptual models are related by abstraction based on an alignment between them. That is the case if one model can be derived from the other model by eliminating all entities that are not part of any correspondence and by aggregating the remaining entities according to the correspondences of the alignment. Since information loss is the desired outcome of abstraction, aggregation must not increase the number of entities represented in the model.

Note that the notion of specialization can be seen as the reverse operation for abstraction. Specialization relies on extension and refinement, the former being the reverse of elimination, the latter being the reverse of aggregation.

4.2 Abstraction of Behavioral Models

As for conceptual models in general, abstraction of behavioral models relies on the aggregation and elimination of model entities. In net systems, these entities are interpreted as poset of events, i.e., process runs. Thus, the abstraction of behavior is the abstraction of processes that may be eliminated or aggregated.

Consider two net systems and an alignment between them. To determine whether both models are related by abstraction, we check whether all differences in their processes are caused by elimination and aggregation from one system to the other. Elimination of a process means that a process of one system must have a corresponding process with the same tactic in the other system, but the reverse is not required to hold. Further, the notion of aligned process set abstractions, cf., Definition 9, enables us to consider the aggregation of processes. In fact, the partitioning of process set abstractions allows the definition of different aggregations. Since we require the existence of matching tactics, we actually require the existence of some valid aggregation operation.

Definition 12 (Abstraction of net systems).
Let S_1 and S_2 be net systems. An alignment $\bowtie$ of S_1 and S_2 is an *abstraction* iff $S_1 \leqslant_{\bowtie} S_2$ and the aggregation predicate $agg_{\bowtie}$ holds. S_1 is called an *abstract version* of S_2 with respect to abstraction $\bowtie$.

- We refer to $\bowtie$ as *meta abstraction* if $agg_{\bowtie}$ holds when $\forall(x,y) \in \bowtie : |x| \leq |y|$.
- We refer to $\bowtie$ as *instance abstraction* if $agg_{\bowtie}$ holds when for every $(\pi_1, \pi_2) \in \mathcal{I}_{\bowtie}$ there exists an alignment χ between some tactics of process set abstractions of π_1 and π_2 such that for all $(x,y) \in \chi$ holds $|x| \leq |y|$.

In Definition 12, elimination is captured by the concept of tactic coverage, i.e., an original net system describes all, and usually more, tactics than its abstract version. We propose to parameterize the abstraction notion by using different aggregation predicates. The role of an abstraction predicate is to define the semantics of the aggregation operation. Intuitively, it must reflect the aggregation of process related information. In this paper, we offer two aggregation predicates. The meta abstraction relies on the aggregation of sets of aligned transitions in net systems. In this case, one can argue that an original net system uses modeling constructs of higher granularity than its abstract version [20]. Alternatively, the instance abstraction ensures that set abstractions of processes taken from the set of pairs used to decide on tactic coverage – the elimination feature of abstraction – can be aligned in such a way that the aligned parts show the decrease in behavioral information, i.e., the sizes of parts decrease in size. The instance abstraction, therefore, ensures aggregation on the instance level. We foresee that new aggregation predicates will evolve to complement the above two.

5 Application of Isotactics

This section elaborates on the application of isotactics. In particular, we focus on existing techniques for implementing the abstraction of behavioral models with a structural approach. These techniques implement transformations that are defined structurally, but motivated by behavioral characteristics. It is often assumed that abstraction operations should be *order preserving*, see [7,8]. However, there has been a lack of a precise definition of what constitutes order preservation in a general setting, i.e., without imposing any assumptions on the relation between activities of the original model and its abstract version. Based on isotactics, we provided such a definition, cf., Theorem 1. We see that common approaches to structural abstraction respect the presented abstraction notion. However, we also show that those structural approaches are limited in their expressiveness. There exist behavioral models that show an order preserving abstraction based on isotactics, but they cannot be derived from each other using the existing structural techniques. For instance, the triconnected abstraction of behavioral models is based on fragments obtained by applying the triconnected decomposition of a graph derived from the model [8]. These fragments are single-entry-single-exit (SESE) and form a containment hierarchy, which is leveraged for abstraction. In an abstraction step, the smallest (in the number of edges) SESE fragment that contains all irrelevant constructs (for the purpose of the model) gets replaced with a fresh activity. The latter represents the whole SESE fragment of a given detailed model in its abstract version.

The intuition behind the triconnected abstraction can be transferred to net systems. For a net $N = (P, T, F)$, a SESE fragment is given as a subnet $N' = (P', T', F')$ with $P' \subseteq P$, $T' \subseteq T$, and $F' = F \cap ((P' \times T') \cup (T' \times P'))$. In case of a special class of net systems, called WF-systems [21], one can efficiently compute all SESE subnets of a given WF-system by using the technique described in [22]. Finally, a (triconnected) *SESE abstraction* step is realized by replacing a SESE subnet of a net system with a single transition; note that we require that no place of the subnet contains a token.

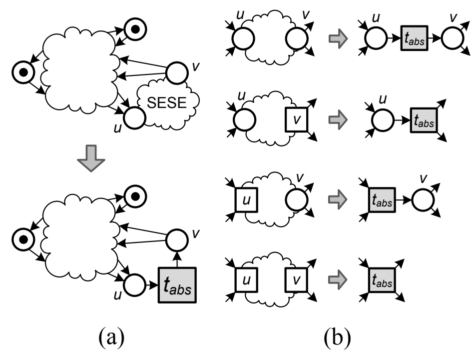


Fig. 6. SESE abstraction of net systems

Fig. 6 explains the SESE abstraction. Fig. 6(a) shows the general idea. Here, a SESE subnet of the original net system with entry u and exit v (top) gets replaced by transition t_{abs} in its abstract version (bottom). Depending on types of entry and exit nodes, we distinguish four abstraction operations, see Fig. 6(b). Every SESE abstraction operation induces an alignment relation between the original and the abstraction result; the set of transitions of the SESE fragment can be put into a correspondence with the abstract transition t_{abs}, whereas all other transitions are related by elementary correspondence with their copies in the resulting net system. Intuitively, the obtained alignment reflects some behavioral relation between the systems. Formally, this relation can be characterized using abstraction as introduced in Section 4.2.

Indeed, two net systems S_1 and S_2, where S_1 is safe and live [23] and S_2 is obtained from S_1 by means of a SESE abstraction operation, are in the meta abstraction relation as well as in the instance abstraction relation. A formal proof of this statement is beyond the scope of this paper. Nevertheless, one can trivially conclude that for every process of the original net system which contains events that represent transitions from the SESE subnet, there exists a process in the abstract net system which contains an event which represents an abstract transition t_{abs} such that set abstractions of these processes can be aligned. Safeness, liveness, as well as the absence of tokens at places of the SESE subnets, are required to ensure that the occurrence of transitions in the subnet has the same effect for the surrounding net as firing the abstract transition.

Besides the possibility to characterize the behavioral relation between net systems derived from each other by structural transformations, isotactics also makes the limitations of these techniques explicit. Consider two aligned models in Fig. 7. Fig. 7(a) shows the original model, whereas Fig. 7(b) proposes its abstract version. Both models show meta abstraction and instance abstraction. Apparently, the model in Fig. 7(b) cannot be derived from the original by means of SESE abstraction operations. The smallest SESE fragment which contains any subset of at least two tasks of the model in Fig. 7(a) is the fragment with entry u and exit v, which would imply the aggregation of the whole model into a single task; note that at least two tasks are required to trigger a SESE abstraction operation leading to a structural change in the abstract model. Nevertheless, the model in Fig. 7(a) covers the tactic of the model in Fig. 7(b) and satisfies aggregation predicates; both on the level of alignment of tasks and on the level of aligned tactics employed to decide on tactic coverage.

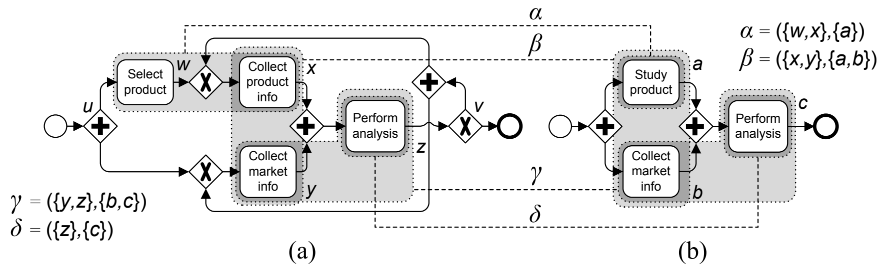


Fig. 7. Abstraction of BPMN diagrams: (a) original and (b) its abstract version

6 Related Work

Sequential equivalences have been classified in the linear-time branching-time spectrum by the seminal work of van Glabbeek, for concrete behavioral models [24] and for those with silent steps [25]. Bisimulation [26], which requires the ability of two models to simulate each other, is commonly seen as the upper bound of this spectrum. It was advocated that equivalences of this spectrum shall be applied for comparing behavioral models in BPM [27]. Several of the aforementioned equivalences can be lifted to non-sequential models as investigated in this paper. Isotactics is particularly inspired by notions of concurrent bisimulation as defined in [18]. A survey of equivalences for net systems under sequential and non-sequential semantics can be found in [28]. All those equivalences have in common that they assume models to be defined on the same level of granularity. As such, they are applicable only if an alignment is functional and injective, i.e., built of non-overlapping 1:1 correspondences. There have been only a few attempts to lift equivalences to a more general setting. In [29], trace partitioning is proposed to decide trace equivalence for non-overlapping complex n:m correspondences. A similar idea was followed for the comparison of state transition systems under non-overlapping complex correspondences between transitions [30]. We go beyond these results by grounding isotactics on concurrency semantics, thus preserving the level of concurrency. Also, our notion is more generic than those presented in [29,30], since it is applicable for overlapping correspondences.

The question of how to cope with elements of behavioral models that are not part of any correspondence has been addressed by behavior inheritance [31]. It proposes to rely on hiding (assign a silent label to transitions) and blocking (remove transitions) before bisimulation is assessed. This way, many use cases for the comparison of business process models can be addressed by verifying standard equivalences, cf., [2]. This work is orthogonal to the question of complex correspondences. Hiding and blocking may be applied before isotactics is verified.

Behavioral abstraction and refinement techniques typically aim at preserving behavioral properties, but are defined on the structural level. Behavioral abstraction was approached, e.g., with predefined patterns [32] and structural decomposition [8]. There also exist different sets of reduction rules for Petri nets [33,34]. For the reverse operation, different refinement operators have been proposed [35]. Such refinements replace a transition or place with a subnet that is embedded into the original net [23]. The notion of isotactics is not limited to hierarchical abstraction and refinement as implemented by structural techniques.

It makes the limitations of structural transformations explicit and opens the space for transformations that are directly grounded in the behavior.

7 Conclusion

We proposed the notion of isotactics – an equivalence relation for behavioral models that are based on concurrency semantics and for which an alignment relation has been defined. With respect to existing equivalence notions, isotactics stands out for two reasons: First, it does not impose any assumptions on the alignment relation. Second, it preserves the level of concurrency of aligned transitions, whereas existing work focuses on sequential semantics.

Given its broad applicability, isotactics can be used to solve a variety of issues in BPM. For instance, when a technical process model is changed, one can determine whether the modified model is still isotactic to a respective business-level model. If this is not the case, changes may be implemented accordingly. In many cases, however, both models will still be isotactic, so that no modifications will be required. In this way, isotactics can support consistent model evolution and improve the quality of the process landscape. The proposed characterization of abstraction defines a space for novel abstraction techniques. We showed that structural transformations for behavioral abstraction are limited. The notion of isotactics-based abstraction, thus, provides the foundation for techniques that are directly grounded in the behavior.

Isotactics, as proposed in this work, is the first step toward a spectrum of equivalences. Exploring this spectrum, e.g., by taking the branching structure into account, along with results on the computational complexity of deciding isotactics, are further directions for future work.

Acknowledgments. This research is partly funded by the ARC Linkage Project LP110100252 "Facilitating Business Process Standardisation and Reuse".

References

1. OMG: Business Process Model and Notation (BPMN), Version 2.0 (January 2011)
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. Polyvyanyy, A., Smirnov, S., Weske, M.: Business Process Model Abstraction. In: Handbook on Business Process Management 1. Springer (2010) 149–166
4. Nielsen, M., Plotkin, G.D., Winskel, G.: Petri nets, event structures and domains, Part I. Theoretical Computer Science (TCS) **13** (1981) 85–108
5. Milner, R.: A Calculus of Communicating Systems. Volume 92 of LNCS., (1980)
6. Park, D.M.R.: Concurrency and automata on infinite sequences. In: Theoretical Computer Science. Volume 104 of LNCS., Springer (1981) 167–183
7. Liu, D.R., Shen, M.: Workflow modeling for virtual processes: An order-preserving process-view approach. Information Systems (IS) **28**(6) (2003) 505–532
8. Polyvyanyy, A., Smirnov, S., Weske, M.: The triconnected abstraction of process models. In: BPM. Volume 5701 of LNCS., Springer (2009) 229–244
9. Dijkman, R.M., Dumas, M., García-Bañuelos, L., Käärik, R.: Aligning business process models. In: EDOC, IEEE CS (2009) 45–53
10. Weidlich, M., Dijkman, R.M., Mendling, J.: The ICoP framework: Identification of correspondences between process models. In: CAiSE. Volume 6051 of LNCS., Springer (2010) 483–498

11. Lohmann, N., Verbeek, E., Dijkman, R.M.: Petri net transformations for business processes – a survey. TOPNOC **2** (2009) 46–63
12. Goltz, U., Reisig, W.: The non-sequential behavior of Petri nets. Information and Control **57**(2/3) (1983) 125–147
13. Petri, C.A.: Non-Sequential Processes. GMD ISF. Gesellschaft für Mathematik und Datenverarbeitung (1977)
14. Rahm, E., Bernstein, P.A.: A survey of approaches to automatic schema matching. VLDB J. **10**(4) (2001) 334–350
15. Euzenat, J., Shvaiko, P.: Ontology matching. Springer, Heidelberg (DE) (2007)
16. Noy, N.F., Klein, M.C.A.: Ontology evolution: Not the same as schema evolution. Knowl. Inf. Syst. **6**(4) (2004) 428–440
17. Rull, G., Farré, C., Teniente, E., Urpí, T.: Validation of mappings between schemas. Data Knowl. Eng. **66**(3) (2008) 414–437
18. Best, E., Devillers, R.R., Kiehn, A., Pomello, L.: Concurrent bisimulations in Petri nets. Acta Informatica (ACTA) **28**(3) (1991) 231–264
19. Kühne, T.: Matters of (meta-)modeling. Softw. and Syst. Mod. **5**(4) (2006) 369–385
20. Holschke, O., Rake, J., Levina, O.: Granularity as a cognitive factor in the effectiveness of business process model reuse. In: BPM. Volume 5701 of LNCS., Springer (2009) 245–260
21. van der Aalst, W.M.P.: The application of Petri nets to workflow management. Journal of Circuits, Systems, and Computers (JCSC) **8**(1) (1998) 21–66
22. Polyvyanyy, A., Vanhatalo, J., Völzer, H.: Simplified computation and generalization of the refined process structure tree. In: WS-FM. Volume 6551 of LNCS., Springer (2010) 25–41
23. Murata, T.: Petri nets: Properties, analysis and applications. Proceedings of the IEEE **77**(4) (1989) 541–580
24. van Glabbeek, R.J.: The linear time-branching time spectrum (extended abstract). In: CONCUR. Volume 458 of LNCS., Springer (1990) 278–297
25. van Glabbeek, R.J.: The linear time - branching time spectrum II. In: CONCUR. Volume 715 of LNCS., Springer (1993) 66–81
26. van Glabbeek, R.J., Weijland, W.P.: Branching time and abstraction in bisimulation semantics. J. ACM **43**(3) (1996) 555–600
27. Hidders, J., Dumas, M., van der Aalst, W.M.P., ter Hofstede, A.H.M., Verelst, J.: When are two workflows the same? In: CATS. Volume 41 of CRPIT., (2005) 3–11
28. Pomello, L., Rozenberg, G., Simone, C.: A survey of equivalence notions for net based systems. In: APN. Volume 609 of LNCS., Springer (1992) 410–472
29. Weidlich, M., Dijkman, R.M., Weske, M.: Deciding behaviour compatibility of complex correspondences between process models. In: BPM. Volume 6336 of LNCS., Springer (2010) 78–94
30. Weidlich, M., Dijkman, R.M., Weske, M.: Behaviour equivalence and compatibility of business process models with complex correspondences. The Computer Journal (CJ) (2012) In press.
31. Basten, T., van der Aalst, W.M.P.: Inheritance of behavior. J. Log. Algebr. Program. **47**(2) (2001) 47–145
32. Polyvyanyy, A., Smirnov, S., Weske, M.: Process model abstraction: A slider approach. In: EDOC, IEEE CS (2008) 325–331
33. Berthelot, G.: Checking properties of nets using transformation. In: ATPN. Volume 222 of LNCS., Springer (1986) 19–40
34. Desel, J., Esparza, J.: Free Choice Petri Nets. Cambridge University Press (1995)
35. Brauer, W., Gold, R., Vogler, W.: A survey of behaviour and equivalence preserving refinements of Petri nets. In: ATPN. Volume 483 of LNCS., Springer (1989) 1–46