

Computation of Behavioural Profiles of Processes Models

Matthias Weidlich¹, Jan Mendling², and Mathias Weske¹

¹ Hasso Plattner Institute, Potsdam, Germany
{matthias.weidlich,weske}@hpi.uni-potsdam.de
² Humboldt-Universität zu Berlin, Germany
jan.mendling@wiwi.hu-berlin.de

Abstract. In this report, we introduce the notion of a behavioural profile that captures the essential behavioural constraints of a process model. We also show, how these profiles can be calculated in $O(n^3)$ time for sound free-choice Petri nets with n places and transitions.

1 Introduction

In the context of process model alignment, the question of consistency between processes has been open for quite a while. While the well-known ‘Business-IT-Gap’ (cf. [1, 2]) is only the most prominent incarnation of this problem, the question of how to align models created for different purposes is fundamental. Clearly, there is a need for a consistency definition that is weaker than the existing equivalence criteria. This report introduces the notion of behavioural profiles. These profiles capture the essential behavioural constraints of a process model. Therefore, they are a means to assess consistency between process models in the context of model alignment. Weaker than trace equivalence, equivalence of two behavioural profiles, nevertheless, ensures preservation of all dependencies between corresponding activities. We also show that consistency can be checked for sound free-choice Petri nets in $O(n^3)$ time with n being the number of places and transitions.

The remainder of this report is structured as follows. Formal preliminaries are given in Section 2, while Section 3 introduces behavioural profiles. We review related work in Section 4 and conclude in Section 5.

2 Formal Foundation

For our investigations, we use workflow (WF-) nets [3] as our formal grounding. On the one hand, these nets have been applied for process modelling for over a decade [3]. On the other hand, Petri net based formalisations have been presented for (at least parts of) most common process modelling languages. For instance, BPEL has been formalised using open workflow nets [4] and a subset of BPMN is formalised using standard Petri nets [5]. The latter might be transformed into a WF-net based formalisation, if there is a dedicated set of end states. In the

context of the UML, semantics have also been defined for sequence diagrams based on M-nets [6]. As an M-net requires one dedicated initial marking and end marking, it might also be transformed into a WF-net.

Based on [3], we recall some basic definitions. 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. The set of all preceding nodes $x \in (P \cup T)$ is defined as $\bullet x := \{y \in (P \cup T) \mid (y, x) \in F\}$, whereas $x^\bullet := \{y \in (P \cup T) \mid (x, y) \in F\}$ denotes succeeding nodes. Further on, let F^+ be the transitive closure of F , i.e. $x F^+ y$ if there is a path from x to y .

The *marking* of a net is a bag over the set of places, i.e. it is a function from P to the natural numbers. We use the notation s_j for the following markings. If j is a place, s_j is the marking that puts a token in j with no tokens elsewhere. If j is a transition, s_j is the marking that puts one token in all places $\bullet j$. A transition $t \in T$ is *enabled* in a marking s , denoted by $(N, s)[t]$, iff $\bullet t \leq s$. *Firing* of t in s leads to a new marking $s' = s - \bullet t + t^\bullet$, denoted by $(N, s)[t](N, s')$. A *firing sequence* $\sigma = t_1, \dots, t_n$ with $t_j \in T$ is a sequence of transitions that can be fired in sequential order (refer to [3] for the formal definition). We write $t_j \in \sigma$, if t_j is an element of the firing sequence.

A *system* $S = (N, i)$ is given by a net N and an initial marking i . The set of all reachable markings of S is denoted by $[N, i]$. A *workflow net* is a net $N = (P, T, F)$ with a dedicated input place $i \in P$ (it is the only place p with $\bullet p = \emptyset$) and a dedicated output place o (it is the only place p with $p^\bullet = \emptyset$). In addition, the short-circuit net (derived by inserting a transition between the initial and the final place) $N' = (P, T \cup \{t_c\}, F \cup \{(o, t_c), (t_c, i)\})$ for N has to be strongly connected. A *workflow system* $S = (N, i)$ is a system with N being a workflow net.

For the remainder of this paper, we always assume all nets to be defined as $N = (P, T, F)$ without stating it explicitly. Workflow systems are means to model process models, therefore, we use the terms *activity* and *transition* synonymously.

3 Behavioural Profiles of Process Models

The behaviour of a process model can be traced back to characteristic relations that capture dedicated behavioural aspects. These relations, in turn, yield the *behavioural profile* of a process model. We dedicate Section 3.1 to a formal definition of these relations. Afterwards, we investigate the relation properties in Section 3.2 and show how they can be derived efficiently in Section 3.3. Finally, we explicate on the differences between equivalence of behavioural profiles and trace equivalence in Section 3.4.

3.1 Characteristic Relations

In general, we can distinguish three fundamental relations between activities of a process model. The execution of two activities might happen either in a *strict order*, *exclusively*, or *concurrently*. It is worth to mention that these relations,

however, state *potential* dependencies. That is, the actual execution of an activity is not explicitly enforced. With respect to our formal model, we capture these characteristic dependencies as relations between WF-net transitions that are identified based on the existence of a certain firing sequence.

Albeit inspired by the log ordering relations that have been proposed in the context of workflow mining [7], our relations are different. We base our definitions on the notion of an *indirect* weak order dependency, whereas the ordering relations in [7] are grounded on a *direct* sequential order. In the context of workflow mining, *causal* dependencies in the sense of one activity directly succeeding another are emphasized. As a result, for instance, the notion of exclusiveness is restricted to ‘pairs of transitions that never follow each other directly’ [7]. In contrast, we aim at capturing exclusiveness also for activities that might occur at different stages of a firing sequence.

All of our characteristic relations are grounded on the concept of weak order. Therefore, we first introduce the weak order dependency as an auxiliary relation.

Definition 1 (Weak Order Relation). *Let (N, i) be a WF-system. The weak order relation $\succ \subseteq T \times T$ contains all pairs (x, y) , such that there is a firing sequence $\sigma = t_1, \dots, t_n$ with $(N, [i])[\sigma]$, $j \in \{1, \dots, n - 1\}$, and $j < k \leq n$ for which holds $t_j = x$ and $t_k = y$.*

Based thereon, we define the first characteristic relation, that is strict order, between two transitions.

Definition 2 (Strict Order Relation). *Let (N, i) be a WF-system. The strict order relation $\rightsquigarrow \subseteq T \times T$ contains all pairs (x, y) with $x \succ y$ and $y \not\succeq x$.*

As mentioned before, the strict order relation enforces neither the occurrence of the first transition, nor of the second transition. Thus, for the two transitions in Fig. 1, it holds $A \rightsquigarrow B$. The second relation, which captures the exclusive occurrence of two transitions is captured as follows.

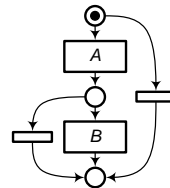


Fig. 1. *Strict order relation*

Definition 3 (Exclusiveness Relation). *Let (N, i) be a WF-system. The exclusiveness relation $+$ $\subseteq T \times T$ contains all pairs (x, y) with $x \not\succeq y$ and $y \not\succeq x$.*

In order to capture concurrency, we propose a definition that is based on the resulting firing sequences of concurrent execution under interleaving semantics. That is, the relation does not enforce *true* concurrency in the sense of two transitions that are enabled independent of each other. Instead, we consider the *observable* firing sequences.

Definition 4 (Observation Concurrency Relation). *Let (N, i) be a WF-system. The observation concurrency relation $\parallel \subseteq T \times T$ contains all pairs (x, y) with $x \succ y$ and $y \succ x$.*

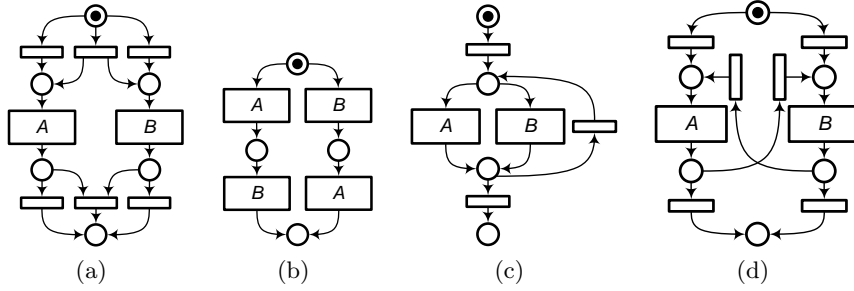


Fig. 2. Observation concurrent transitions

Again, the our notion of concurrent transitions does not enforce the occurrence of both transitions. For instance, two transitions that are preceded by an inclusive OR-split as illustrated in Fig. 2(a) (note that the net is not sound) will be in the observation concurrency relation, i.e. $A||B$. One might also think of concurrency based on the interleaving of transitions as depicted in Fig. 2(b). However, that, in turn, requires labelled transitions. That is, the definition of a WF-net has to contain a labelling function that assigns labels to transitions. As a result the occurrence of a label in a trace might result from firing of different transitions. However, such a labelling is not part of our formal model, which excludes the net in Fig. 2(b). Further on, concurrency might also result from choices between two transitions in cyclic structures. In Fig. 2(c) and Fig. 2(d), it also holds $A||B$, as the observable firing sequences indicate concurrency, even though there is no marking in which both transitions are enabled.

3.2 Properties of Characteristic Relations

Following on the approach introduced in [7] for ordering relations in execution logs, we can also show orthogonality for our characteristic relations.

Property 1. For any WF-system (N, i) holds that the characteristic relations \rightsquigarrow , $+$, and $||$ are mutually exclusive and \rightsquigarrow , \rightsquigarrow^{-1} , $+$, and $||$ partition $T \times T$.

Let $\succ^{-1} = \{(y, x) \in T \times T \mid x \succ y\}$ the inverse relation for \succ . Then the property can be verified by the definition of the characteristic relations as $\rightsquigarrow = \succ \setminus \succ^{-1}$, $\rightsquigarrow^{-1} = \succ^{-1} \setminus \succ$, $+$ $= (T \times T) \setminus (\succ \cup \succ^{-1})$, and $|| = \succ \cap \succ^{-1}$.

It is worth to mention that this property holds even, if a WF-system shows behavioural anomalies as deadlocks (cf. Fig. 3(a)) or livelocks (cf. Fig. 3(b)). Even in case the initial marking is a deadlock, i.e. there is not a single firing sequence, all transitions would be considered to be exclusive, that is $+$ $= (T \times T)$ owing to $\succ = \succ^{-1} = \emptyset$.

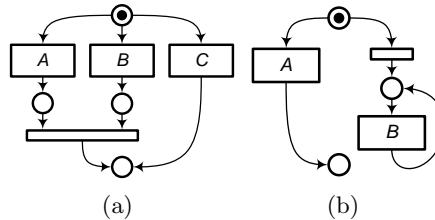


Fig. 3. Behavioural anomalies

In the previous section, we saw that cyclic structures have a substantial impact on the characteristic relations. For instance, two exclusive transitions inside a cycle are considered to be observation concurrent, which is depicted in Fig. 2(c) and Fig. 2(d). Therefore, we have to address the question of how these cycles are reflected in the behavioural profile, i.e. the characteristic relations.

Property 2. For any WF-system (N, i) holds that a transition $t \in T$ is either said to be exclusive to itself ($t + t$) or observation concurrent to itself ($t||t$).

In order to verify this property, we have to consider two cases, $(t, t) \in \succ$ and $(t, t) \notin \succ$. For the former it holds that $(t, t) \in \succ$ implies $(t, t) \in \succ^{-1}$, which yields $t||t$. For the latter, we have the opposite implication, i.e. $(t, t) \notin \succ$ implies $(t, t) \notin \succ^{-1}$ and, therefore, $t + t$. Consequently, the characteristic relations capture, whether a transition might occur at most once ($t + t$), or might be repeated due to a control flow cycle ($t||t$). For instance, $A + A$ and $B||B$ in Fig. 3(b).

3.3 Deriving the Characteristic Relations

The characteristic relations of the behavioural profile follow directly from the weak order relation, which requires the existence of a certain firing sequence. Such a firing sequence is one path in the reachability graph of the net. Therefore, the whole state space of the net has to be considered, which is well-known to require exponential space and time for arbitrary Petri nets [8].

However, for a dedicated class of systems, that is *sound free-choice* WF-systems, a different approach can be taken to determine the behavioural profile. Before we describe the approach, we recall basic definitions for this class of systems.

According to [9], a net $N = (P, T, F)$ is *free-choice*, if $(s, t) \in F$ implies $\bullet t \times s \bullet \subseteq F$ for every place s and transition t . A system (N, i) is free-choice, if N is free-choice. A system (N, i) is *live*, if for every reachable marking $s' \in [N, s]$ and $t \in T$, there is a marking $s'' \in [N, s']$ such that $(N, s'')[t]$. A system (N, i) is *bound*, iff the set of reachable markings $[N, i]$ is finite. It has been shown, that liveness and boundness is closely related to the *soundness* criterion, which requires a WF-system (1) to always terminate, and (2) to have no dead transitions (note that proper termination is implied for WF-systems) [10]. In fact, a net is sound, if and only if the corresponding short-circuit net is live and bound [10].

We start by deriving the observation concurrency relation. We first define an auxiliary relation that captures concurrent enabling of two transitions.

Definition 5 (True Concurrency Relation). *Let (N, i) be a WF-system. The true concurrency relation $||_t \subseteq T \times T$ contains all pairs (x, y) with $x \neq y$, such that there is a reachable marking $s \in [N, i]$ that enables both transitions concurrently, i.e. $s \geq s_x + s_y$.*

As mentioned above, observation concurrency does not enforce concurrent enabling of two transitions. In contrast, the true concurrency relation requires concurrent enabling, for instance $A||_t B$ in Fig. 2(a), but not in Fig. 2(c). Nevertheless, there is the following dependency between both relations.

Lemma 1. *For any free-choice system holds, every pair of transitions that is true concurrent is also observation concurrent.*

Proof. Let (N, i) be a free-choice system, $x, y \in T$, and $x \parallel_t y$. From the latter, we know that there is a marking $s \in [N, i)$ with $s \geq s_x + s_y$. Therefore, there are two possible firing sequences $(N, s)[xy)$ and $(N, s)[yx)$, which yields $x \succ y$ and $y \succ x$, in other words observation concurrency.

Further on, we can relate structural dependencies of transitions in sound free-choice nets to the characteristic relations. For (N, i) as a sound free-choice system, we say that two transitions $x, y \in T$ are *structurally concurrent*, if xF^+y and yF^+x . They are *structurally ordered*, if xF^+y and $yF^\neq x$, and *structurally exclusive*, if $xF^\neq y$ and $yF^\neq x$.

Lemma 2. *For any sound free-choice system holds, for every two transitions that are not true concurrent, observation concurrency and structural concurrency coincide.*

Proof. Let (N, i) be a sound free-choice system, $x, y \in T$, and $x \not\parallel_t y$.

\Rightarrow From $x \parallel y$ we know $x \succ y$. Thus, there exist two markings $s_1, s_2 \in [N, i)$ with $(N, s_1)[x)(N, s_2)$, due to the absence of dead transitions (soundness property).

Assume that $xF^\neq y$. The net is free-choice, that is for every two transitions $t, u \in T$ either $\bullet t \cap \bullet u = \emptyset$ or $\bullet t = \bullet u$. Therefore, all places x^\bullet cannot impact on the enabling of y and there must be a marking $s_3 \in [N, s_2)$ enabling y for which holds $s_3 \geq s_y + \sum_{p \in x^\bullet} (s_p)$. In other words, s_3 marks all places x^\bullet and enables y . Consequently, there must also be a marking s_4 from which s_3 is derived by firing of x , i.e. $(N, s_4)[x)(N, s_3)$. Then, $s_4 \geq s_x + s_y$, which yields a contradiction, as we required $x \not\parallel_t y$. Therefore, xF^+y and also yF^+x , due to the argument turned around.

\Leftarrow From xF^+y and yF^+x we know that, due to the soundness and the free-choice property, there must be a firing sequence containing both transitions in either order, i.e. $x \succ y$ and $y \succ x$. Thus, both transitions are observation concurrent.

Structural exclusiveness can also be related to the characteristic relations for sound free-choice systems.

Lemma 3. *For any sound free-choice system holds, for every two transitions that are not true concurrent, exclusiveness and structural exclusiveness coincide.*

Proof. Let (N, i) be a sound free-choice system, $x, y \in T$, and $x \not\parallel_t y$.

\Rightarrow From $x + y$ we know $x \not\neq y$. Assume that xF^+y . Then, there must be two markings $s_1, s_2 \in [N, i)$ with $(N, s_1)[x)(N, s_2)$, due to soundness of the net (x must not be a dead transition). Because of the free-choice structure and the absence of deadlocks and livelocks (soundness), xF^+y implies the existence of a marking $s_3 \in [N, s_2)$ with $s_3 \geq s_y$, i.e. y is enabled. That yields a contradiction with $x \not\neq y$. Therefore, we know that $xF^\neq y$. The same holds true for the argument turned around, which leads to $yF^\neq x$.

\Leftarrow We have $x \mathcal{P}^{\neq} y$. Assume that $x \succ y$. Thus, there are two markings $s_1, s_2 \in [N, i]$ with $(N, s_1)[x](N, s_2)$. As $x \mathcal{P}^{\neq} y$, all places x^\bullet cannot impact on the enabling of y , which requires the existence of a marking $s_3 \in [N, s_2]$ with $s_3 \geq s_y + \sum_{p \in x^\bullet} (s_p)$. Consequently, there is a marking s_4 from which s_3 is derived via firing of y , $(N, s_4)[y](N, s_3)$. That, $s_4 \geq s_x + s_y$ which is not in line with $x \not\parallel_t y$. Again, the argument can be turned around for the assumption of $y \succ x$. Consequently, $x + y$.

Finally, we are able to state that the characteristic relations can be derived from the true concurrency relation and the flow relation.

Theorem 1. *Given a sound free-choice system and its true concurrency relation, the behavioural profile can be calculated from the transitive closure of the flow relation.*

Proof. Observation concurrency is given by true concurrency and structural concurrency according to Lemma 1 and Lemma 2. Exclusiveness can be derived from the flow relation as well (Lemma 3). It remains to show how strict order can be derived from the transitive closure of the flow relation. Let (N, i) be a sound free-choice system, $x, y \in T$, $x F^+ y$, and $y \mathcal{P}^{\neq} x$. We know that there exist two markings $s_1, s_2 \in [N, i]$ with $(N, s_1)[x](N, s_2)$, due to soundness of the net (x must not be a dead transition). Under the free-choice property and the absence of deadlocks and livelocks (soundness), the existence of a marking $s_3 \in [N, s_2]$ with $s_3 \geq s_y$ follows from $x F^+ y$. Thus, $x \succ y$. Now, consider $y \mathcal{P}^{\neq} x$. If we assume $y \succ x$, we get that both transitions are required to be true concurrent, $y \parallel_t x$ (cf. \Leftarrow of the proof of Lemma 3). As the true concurrency relation is given, we can exclude these pairs of transitions, such that for all remaining pairs $y \mathcal{P}^{\neq} x$ implies $y \not\prec x$. Finally, from $x \succ y$ and $y \not\prec x$ follows $x \rightsquigarrow y$. This means that strict order can also be derived from the transitive closure of F .

Based thereon, the behavioural profile can be calculated very efficiently for sound free-choice nets.

Corollary 1. *Given a sound free-choice WF-system, the behavioural profile can be derived in $O(n^3)$ with n as the number of transitions and places of the net.*

Proof. For any free-choice net the true concurrency relation can be calculated in $O(n^3)$ with n as the number of transitions and places of the net according to [11]. Further on, the transitive closure for the flow relation is well-known to be computable in $O(n^3)$ as well [12]. According to Theorem 1 this suffices to derive the behavioural profile.

It is worth to mention that the requirements for the application of our approach can also be decided in polynomial time. The free-choice property can be decided solely based on the structure of the net, i.e. the flow relation. On the other hand, soundness can be traced back to liveness and boundness, which can also be decided in polynomial time for free-choice nets [10].

3.4 Behavioural Profiles vs. Trace Equivalence

After we introduced behavioural profiles, we investigate their relation to the notion of trace equivalence. In particular, two aspects that impact on the set of possible traces are not captured in the behavioural profile.

First, the weak order relation and, therefore, all characteristic relations, is based on the *existence* of a firing sequence containing certain transitions. Whether or not a transition is obligated to occur at a certain point is, however, not captured. Of course that might result in different traces.

Consider, for instance, transition A in Fig. 4. This transition might be skipped in the net on the left-hand side, which yields traces that commence with the occurrence of B or C . These traces are not possible in the net on the right-hand side, in which all traces start with the occurrence of A .

The second aspect that is not captured by the behavioural profile, are transition cardinalities in traces. Although the behavioural profile specifies whether a transition

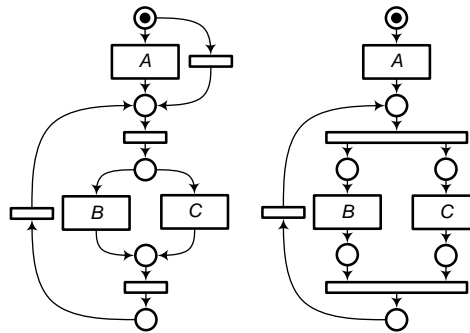


Fig. 4. Equal behavioural profiles, but not trace equivalent

might occur at most once or whether it might occur multiple times, dependencies between the occurrence cardinalities of two transitions are neglected. Again, this is illustrated by Fig. 4. In both nets, transitions B and C are part of a loop. However, in the net on the left-hand side, only one transition might occur per loop iteration, whereas the other net enforces the occurrence of both transitions per iteration. As a consequence, the latter allows solely for traces, in which at most two occurrences of B (or C , respectively) follow on each other directly.

Thus, the example in Fig. 4 illustrates that the equivalence of behavioural profiles does not imply trace equivalence. However, we observe the following.

Property 3. Every two WF-systems that are trace equivalent have the same behavioural profile.

This property follows directly from the definition of the characteristic relations. Two systems having the same set of traces will end up with the same weak order relation, as it requires the existence of a certain trace. That, in turn, results in equivalent characteristic relations.

4 Related Work

In this section we discuss three related areas, namely behavioural equivalence, behavioural inheritance, and correspondences between process models.

The consistency of process models closely relates to different notions of *behavioural equivalence* such as trace equivalence and bisimulation. These notions yield a true or false answer and can, therefore, not directly be applied if models overlap partially. Our notion of consistency that is based on behavioural profiles can be regarded as a relaxed version of trace equivalence that can also handle overlapping models. A good overview of various equivalence notions is presented in [13]. The strict true or false nature of these notions has been criticized in [14] who propose a similarity measure for process mining. As mentioned above, similar relations, but not exactly those of behavioural profiles, are used in process mining as input for constructing models [15]. Other work uses causal footprints as a behavioural abstraction for determining the similarity between process models [16].

Behaviour inheritance aims at applying the idea of inheritance known from static structures to behavioural descriptions. In [17], *protocol inheritance* and *projection inheritance* were introduced by Basten et al. as basic notions of behaviour inheritance based on labelled transition systems and branching bisimulation. A model inherits the behaviour of a parent model, if it shows the same external behaviour when all actions that are not part of the parent model are either *blocked* (protocol inheritance) or *hidden* (projection inheritance). Focussing on object life cycles, Schrefl and Stumptner build upon this work and further distinguished *weak invocation consistency* and *strong invocation consistency* [18].

For our approach we assume that *correspondences* of two process models have been identified and captured. The research area of schema integration, and in particular, schema matching investigates how such correspondences can be identified automatically. A good overview is provided in [19]. As discussed in the introduction, our notion of consistency can be related to desirable properties of schema mappings. An overview of such properties is given in [20]. For business process models, it is almost of equivalent importance to detect mismatches. The work presented in [21, 22] provides a systematic framework of diagnosis and resolution of such mismatches.

5 Conclusion

In this report we have addressed the research challenge of defining a notion of consistency between process models that is more relaxed than trace equivalence. Our contribution is the concept of a behavioural profile that captures the essential behavioural constraints of a process model. Such behavioural profiles are used for the definition of a formal notion of consistency. We proved that consistency can be checked in $O(n^3)$ time for sound free-choice Petri nets with n nodes.

References

1. Grover, V., Fiedler, K., Teng, J.: Exploring the Success of Information Technology Enabled Business Process Reengineering. *IEEE Transactions on Engineering Management* **41**(3) (August 1994) 276–284

2. Rolland, C., Prakash, N.: Bridging the Gap Between Organisational Needs and ERP Functionality. *Requirements Engineering* **5**(3) (October 2000) 180–193
3. van der Aalst, W.M.P.: The application of petri nets to workflow management. *Journal of Circuits, Systems, and Computers* **8**(1) (1998) 21–66
4. Lohmann, N.: A feature-complete Petri net semantics for WS-BPEL 2.0. In Dumas, M., Heckel, R., eds.: *Web Services and Formal Methods*, Forth International Workshop, WS-FM 2007 Brisbane, Australia, September 28-29, 2007, Proceedings. Volume 4937 of *Lecture Notes in Computer Science.*, Springer-Verlag (2008) 77–91
5. Dijkman, R., Dumas, M., Ouyang, C.: Semantics and Analysis of Business Process Models in BPMN. *Information and Software Technology (IST)* **50**(12) (2009) 1281–1294
6. Eichner, C., Fleischhack, H., Meyer, R., Schrimpf, U., Stehno, C.: Compositional semantics for uml 2.0 sequence diagrams using petri nets. In Prinz, A., Reed, R., Reed, J., eds.: *SDL Forum*. Volume 3530 of *Lecture Notes in Computer Science.*, Springer (2005) 133–148
7. van der Aalst, W.M.P., Weijters, T., Maruster, L.: Workflow mining: Discovering process models from event logs. *IEEE Trans. Knowl. Data Eng.* **16**(9) (2004) 1128–1142
8. Valmari, A.: The state explosion problem. In Reisig, W., Rozenberg, G., eds.: *Petri Nets*. Volume 1491 of *Lecture Notes in Computer Science.*, Springer (1996) 429–528
9. Desel, J., Esparza, J.: *Free choice Petri nets*. Cambridge University Press, New York, NY, USA (1995)
10. van der Aalst, W.M.P.: Verification of workflow nets. In Azéma, P., Balbo, G., eds.: *ICATPN*. Volume 1248 of *Lecture Notes in Computer Science.*, Springer (1997) 407–426
11. Kovalyov, A., Esparza, J.: A polynomial algorithm to compute the concurrency relation of free-choice signal transition graphs. In: *Prof. of the International Workshop on Discrete Event Systems, WODES'96*, Edinburgh, The Institution of Electrical Engineers (1996) 1–6
12. Warshall, S.: A theorem on boolean matrices. *Journal of the ACM (JACM)* **9**(1) (1962) 11–12
13. van Glabbeek, R.J., Goltz, U.: Refinement of actions and equivalence notions for concurrent systems. *Acta Inf.* **37**(4/5) (2001) 229–327
14. de Medeiros, A.K.A., van der Aalst, W.M.P., Weijters, A.J.M.M.: Quantifying process equivalence based on observed behavior. *Data Knowl. Eng.* **64**(1) (2008) 55–74
15. Aalst, W., Weijters, A., Maruster, L.: Workflow Mining: Discovering Process Models from Event Logs. *IEEE Transactions on Knowledge and Data Engineering* **16**(9) (2004) 1128–1142
16. van Dongen, B.F., Dijkman, R.M., Mendling, J.: Measuring Similarity between Business Process Models. In Bellahsene, Z., Léonard, M., eds.: *CAiSE*. Volume 5074 of *Lecture Notes in Computer Science.*, Springer (2008) 450–464
17. Basten, T., van der Aalst, W.M.P.: Inheritance of Behavior. *Journal of Logic and Algebraic Programming (JLAP)* **47**(2) (2001) 47–145
18. Schrefl, M., Stumptner, M.: Behavior-consistent specialization of object life cycles. *ACM Trans. Softw. Eng. Methodol.* **11**(1) (2002) 92–148
19. Rahm, E., Bernstein, P.A.: A survey of approaches to automatic schema matching. *VLDB Journal* **10**(4) (2001) 334–350
20. Rull, G., Farré, C., Teniente, E., Urpí, T.: Validation of mappings between schemas. *Data Knowl. Eng.* **66**(3) (2008) 414–437

21. Küster, J.M., Gerth, C., Förster, A., Engels, G.: Detecting and resolving process model differences in the absence of a change log. [23] 244–260
22. Dijkman, R.M.: Diagnosing differences between business process models. [23] 261–277
23. Dumas, M., Reichert, M., Shan, M.C., eds.: Business Process Management, 6th International Conference, BPM 2008, Milan, Italy, September 2-4, 2008. Proceedings. In Dumas, M., Reichert, M., Shan, M.C., eds.: BPM. Volume 5240 of Lecture Notes in Computer Science., Springer (2008)