Fault Detection by Backwards Analysis in Coloured Workflow Nets

Vasilii Ganishev[1], Olga Fengler[2] and Wolfgang Fengler[2]
[1]*Department of Applied Mathematics, National Research University "Moscow Power Engineering Institute", Moscow, Russia*
[2]*Faculty of Computer Science and Automation, Computer Architecture Group, TU Ilmenau, Ilmenau, Germany*

Keywords: Workflow Nets, Fault Detection, Backwards Analysis, Coloured Petri Nets.

Abstract: The increasing complexity of the business processes requires automated methods for trouble-shooting and debugging of the process model in operation. This paper proposes mechanism of fault detection based on the concept of backwards reachability for the coloured workflow nets. The formal verification methods defined for coloured Petri nets such as state space method or place invariants declaration suffer from fast-growing computational complexity. The article offers the set of firing rules for backwards token-play on coloured workflow net. This method helps to detect resource-related failures of the two most common types: "the missing arc" fault (the resource was not initialized) and "wrong expression fault" (there is no proper resource).

SCIENCE AND TECHNOLOGY PUBLICATIONS

1 INTRODUCTION

Rapid growth of business process integration and complexity causes the need for detailed modeling not only of the structure but also the behavior during execution. The workflow networks provide better functionality for modeling the execution of the process. But the interaction of the resource is not included in the classical concept. Looking at the behavior from the viewpoint of resources allows a more detailed description of the business process.

Creating a mechanism for error detection is a priority issue. If the model is too complicated, it is difficult to create error-free net. There are formal verification methods such as *state space method* (drawing the reachability graph) or *place invariants* (introduction of the special logical expressions for net marking). But they either suffer from rapid growth of computational complexity (state explosion in state space method) or are too complex to automate and need manual preparation (place invariants) (Jensen and Kristensen, 2009). The backwards analysis based on the theory of duality in Place/Transition nets (Lautenbach, 2003) proved to be a very powerful tool for fault detection and reachability analysis in Petri nets. The interpretations of this method have been already proposed for predicate/transition nets (Muller and Schnieder, 2007), probability propagation nets (Lautenbach and Susewind, 2012) and also BPMN dialect used in YAWL (Scharfe, 2013).

However, there are some papers proposing some variations of the backwards analysis for coloured Petri nets, for example (Bouali et al., 2009). Unfortunetely some faults cannot be detected by this approach, such as the "missing arc" fault (resource was not initialized) as it is shown in the work (Ganishev, 2013).

This paper provides a set of firing rules that adapts backwards reachability analysis to the concept of coloured Petri nets as well as coloured workflow nets. The detection of two types of the resource-related faults ("missing arc fault" and "wrong expression fault") will be shown in the example net. These faults are resource-related in the concept of coloured workflow nets, because the control flow itself has to satisfy the requirement of soundness. It makes the occurrence of such faults impossible there.

2 COLOURED WORKFLOW NETS

The classic concept of workflow nets does not support the distinction of the resources that are necessary in carrying out the process. For a detailed view of the processes and the modeling of the competition between resources the concept of coloured workflow nets has been proposed (van der Aalst et al., 2011).

In this paper we use the definition of coloured Petri net, which differs from classic one given in (Jensen, 1991), (Kristensen et al., 1998). Two types

Ganishev V., Fengler O. and Fengler W..
Fault Detection by Backwards Analysis in Coloured Workflow Nets.
DOI: 10.5220/0005513000880094
In *Proceedings of the 5th International Conference on Simulation and Modeling Methodologies, Technologies and Applications* (SIMULTECH-2015), pages 88-94
ISBN: 978-989-758-120-5
Copyright © 2015 SCITEPRESS (Science and Technology Publications, Lda.)

of capacity are introduced and the guard function is excluded from consideration.

A coloured Petri net (CPN) formally is a tuple CPN = (P, T, F, C, K, M, m0) satisfying the requirements below:

- P is a finite set of places:

$$P \neq \emptyset; \tag{1}$$

- T is a finite set of transitions:

$$T \neq \emptyset, \tag{2}$$

$$P \cap T = \emptyset; \tag{3}$$

- F is an arc expression function. F models the relationship between states and transitions:

$$F \subseteq T \times P \cup P \times T, \tag{4}$$

$$F \neq \emptyset; \tag{5}$$

Elements: (p,t) is a precondition and (t,p) is a postcondition;

- C is a set of (logical) colours (types):

$$C_{\mathbb{N}_0} \subseteq \mathbb{N}_0 \times C, \tag{6}$$

$$C^{\infty}_{\mathbb{N}_0} \subseteq (\mathbb{N}_0 + \infty) \times C; \tag{7}$$

- K is a capacity, the maximal possible number of tokens with one colour in place:

$$P \to C^{\infty}; \tag{8}$$

K_s is a total capacity:

$$P \to \mathbb{N}^{\infty}; \tag{9}$$

- M is a multiplicity of arc:

$$F \to \mathbb{N}; \tag{10}$$

- m0 is an initial marking:

$$P \to C^{\infty}_{0}; \tag{11}$$

$$0 \leq m0 \leq k(p); \tag{12}$$

The definitions of enabled transitions, reachability and fire rules used in this paper are the same as the classic ones (Jensen, 1991).

By introducing this definition of CPN to the classic concept of workflow nets come out the coloured workflow nets (CWN) similar to the concept proposed (van der Aalst et al., 2005). A CWF notation covers the control-flow perspective, the resource perspective and the data/case perspective. *"A token in a place of type Case refers to a case and some or all of its attributes. Tokens in a place of type Resource represent resources. Places of type CxR hold tokens that refer to both a case and a resource"* (van der Aalst et al., 2005). However we will treat CxR places as Case places in this paper, because they show the same behavior during the execution of the process. These nets satisfy the following constraints (van der Aalst et al., 2005):

1. A CWN where all places of type Resource are removed should correspond to a Sound Workflow Net (sound WF-net).

2. The expression of each arc contains only one appearance of each element of set C per conjunction for resource place. That means that only one token of each colour can be taken from the place or come to it.

3. The capacity of each resource place is one. That means that only one token of each colour allowed to remain in the place at the moment.

This extension does not break limitations of the classic concept (there should be no increase in the number of tokens) (van der Aalst et al., 2011), because at any time classic constraints could be reached by unfolding.

A coloured workflow net should also satisfy following limitations of classical concept (van der Aalst and van Hee, 2002):

1. Every transition is on the way to end place.

2. There has to be no dead transitions, that can't be enabled by execution.

Fig. 1 shows an example of a coloured workflow net. It models the actions of a worker, who needs a resource for his work. The resource should be released at the end of the work. Place "Start" is an initial place. The transition "Init" initializes the business process. Place "Ready" models the worker's readiness to start the work. The place "Res" models the resource place. The transition "Use" shows the capture of the resource by the worker. The place "Work" models the process of working. The transition "Free" models the termination of the work and the release of the resource. The place "Done" is the termination (end) place of the single instance of the business process. The transition "Finish" prepares the net for the next run. The place "End" shows the termination of the whole run of the business process. The transition "t*" provides liveness of the net.

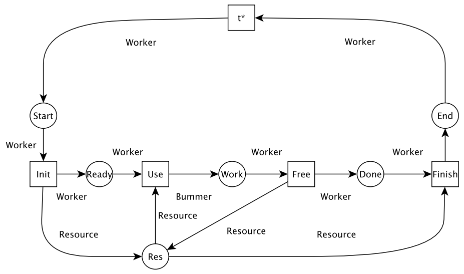

Figure 1: The example of the coloured workflow net.

3 FAULT DETECTION BY BACKWARDS ANALYSIS

The automatic detection of resource-related faults can help to manage resources in complex business models. Especially it may be very helpful in combination with auto-generated CWN or CPN from existing logs provided by business model, so called Play-In (van der Aalst, 2011). The methods for the backwards analysis in classic Petri nets concept or concepts of other High Level Petri nets cannot be used for the coloured Petri nets or workflow nets:

- in classic dual nets the transformation of the tokens when firing is not taken in account;
- in various types of higher Petri nets the transformation is modeled with the help of the activation vector (or activation function), but the concept of coloured Petri nets does not use activation vectors.

3.1 Double-marked Coloured Workflow Net

The concept of the double-marked Petri nets was first introduced by Kurt Lautenbach (?) based on the theory of the dual spaces in order to do diagnoses in Petri nets. First approaches were proposed for place/transition nets (Lautenbach, 2003) and then applies to some other classes of High Level Petri nets (Muller and Schnieder, 2007), (Lautenbach and Susewind, 2012).

The algorithm behind this approach can be described as follows:

- In forward firing (normal case) the net works as classic coloured net.
- If the net is *dead*, but the terminal state is not yet reached, the end place (or places) will be marked with so called initial T-marking (transition-marking). Now places fire rather than transitions, and the net works backwards.

One element of the CPN, the set of colours cannot be simply dualized as in other classes of High Level Petri nets (Muller and Schnieder, 2007). This leads to the inability to use the theory of the dual spaces directly.

The main point of this paper is the introduction of *the neutral element*, that helps to avoid this limitation of CPN, and special firing rules to deal with it.

Formally a double-marked coloured Petri net is a tuple DMCPN = (P, T, C^*, K, M, m_0), where all elements except C^* and m_0 are defined as in the original concept of coloured Petri nets. C^* and m_0 are defined as follows:

- C^* is a set of (logical) colours (types) plus neutral element '*'
- m_0 is a tuple of the initial marking. $m_0 = \{ m_{P_0}, m_{T_0} \}$, where m_{P_0} is the initial place-marking and m_{T_0} is the initial transition-marking. At the beginning of the backwards analysis the transitions to the end places are marked with neutral element.

By dividing a set of places into two subsets (Case places and Resource places) and introducing transition t^* we come to informal definition of double-marked coloured workflow net (DMCWN).

3.2 Firing Rules

Following firing rules are required for DMCWN operation. They represent the adaptation of the set of rules in (Scharfe, 2013) for the concept of CWN and they consider the neutral element.

There are two types of firing rules with neutral element: α-firing rules (all firing places are not marked) and β-firing rules (some firing places are marked).

Simple α*-firing Rule*
Let the transition t be marked with a neutral element. The place which has only one pre-transition is *enabled for* α*-firing rule* if there is a neutral token in the pre-transition and there is no P-token in this place. By firing the post-transition of te marked place receives a neutral token and pre-transition loses its marking.

This rule can be illustrated as follows (Fig. 2, Fig. 3).

Special α*-firing Rules*
Special α-firing rules are the adaptation of Special α-firing rules in classic Petri nets for coloured nets.

Special α*-firing Rule 1*
Let the transition t be marked with a neutral element. If there are two or more places that are enabled with

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

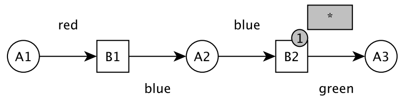

Figure 2: Simple α-firing rule.

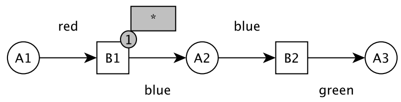

Figure 3: Simple α-firing rule after firing.

actual T-marking and all these places have identical or different post-transitions, then all places fire simultaneously. By firing every post-transition gets a neutral token and the pre-transition loses its marking. It could be shown with the following formula.
enabled:

$$\begin{aligned} \exists t:\ m_T(t) = '*';\ \exists p_j, j \in (0,n), n \geq 2,\\ (t,p_j) \in F\ \wedge\ (p_j,t_i) \in F, i \geq 1 \end{aligned} \tag{13}$$

firing:

$$m_T(t) = '' \ \wedge \forall i m_T(t_i) = '*'. \tag{14}$$

This rule can be illustrated as follows (Fig. 4, Fig. 5)

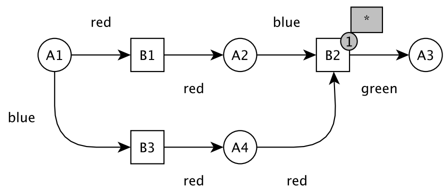

Figure 4: Special α-firing rule 1 before firing.

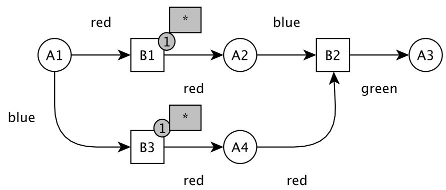

Figure 5: Special α-firing rule 1 after firing.

Special α-firing Rule 2
Let the transition t be marked with a neutral element. If there are two or more places, which have t as the pre-transition, and all these places have identical or different post-transitions, a total firing is not possible. Then the enabled places fire and are marked with symbol '*'. They should not fire again. By firing the post-transition gets a neutral token. When all places have fired, then t loses its marking and each symbol '*' from places should be removed. It could be shown with the following formula:

enabled:

$$\begin{aligned} \exists t:\ m_T(t) = '*';\ \exists p_j, j \in (0,n), n \geq 2,\\ (t,p_j) \in F\ \wedge\ (p_j,t_i) \in F, i \geq 1;\\ \exists k: k > 0\ (t,p_k) \in F\ \wedge\ (p_k,t_k) \in F; \end{aligned} \tag{15}$$

firing:

$$m_T(t) = '';\ \wedge \forall k\, m_T(t_k) = '*'\ \wedge\ m_P(p_k) = '*'; \tag{16}$$

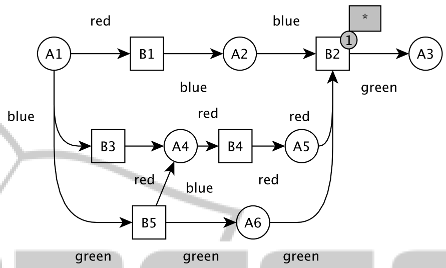

Figure 6: Special α-firing rule 2 before firing.

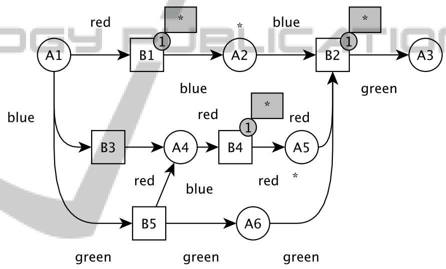

Figure 7: Special α-firing rule 2 after partial firing.

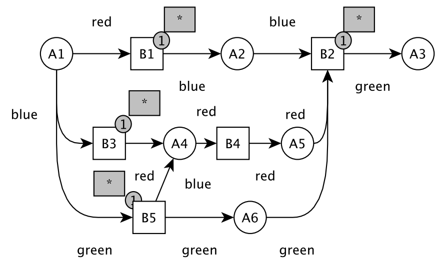

Figure 8: Special α-firing rule 2 after total firing.

If all have fired:

$$\forall j m_P(p_j) = '*' \Rightarrow m_P(p_j) = ''\ . \tag{17}$$

This rule can be illustrated as follows (Fig. 6, Fig. 7, Fig. 8)
β-firing Rule
This rule applies, if the application of α-firing rules is not possible.

Let the transition t be marked with a neutral element. If there is a place p, which has t as the pre-transition, but it contains P-token, then the transition

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

t changes its marking on the expression of the arc that goes from p to t. Now the transition t can not change its marking.
enabled:

$$\exists\, t:\ m_T(t) = '*';\ \exists p(t,p) \in F \wedge (p,t^*) \in F,\ m_P(p)\ \neq\ '';\qquad (18)$$

firing:

$$m_T(t) = (t,p). \qquad (19)$$

This rule can be illustrated as follows (Fig. 9, Fig. 10)

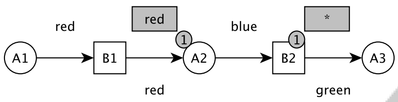

Figure 9: Special β-firing rule before firing.

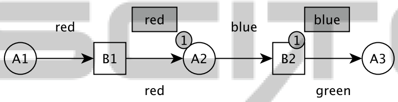

Figure 10: Special β-firing rule after firing.

β-firing Rule (case of multiple places)
Let the transition t be marked with a neutral element. If there are places $p_1,\dots,p_n$, which have t as the pre-transition, but k places contain P-tokens, then the transition t changes its marking on the expressions of the arcs that go from p_k to t. Each expression is separated from others with the symbol '/'. The places that contain no P-tokens fire with α-firing rules. Now the transition t may not change its marking.

enabled:

$$\exists t:\ m_T(t) = '*';\ \exists p_j, j \in (0,n), n \geq 2, (t,p_j) \in F\ \wedge\ (p_j,t_i) \in F, i \geq 1; \exists k: k > 0\ m_P(p_k) \neq\ '';\qquad (20)$$

firing:

$$m_T(t) = (t,p_{m1})/.../(t,p_{mk}), m_1 \dots m_k \in m \subseteq n\ \wedge\ m_T(t_d) = '*', d \in n\ \wedge\ d \notin m.\qquad (21)$$

This rule can be illustrated as follows (Fig. 11, Fig. 12)

4 EXAMPLES OF FAULT DETECTION

"Missing arc" Fault. Let us illustrate the functionality of backwards analysis in coloured workflow nets in following example (Fig. 13). It is the net from previous chapter (p. 2), but with following differences:

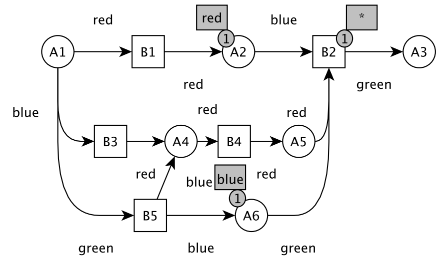

Figure 11: Special β-firing rule (case of multiple places) before firing.

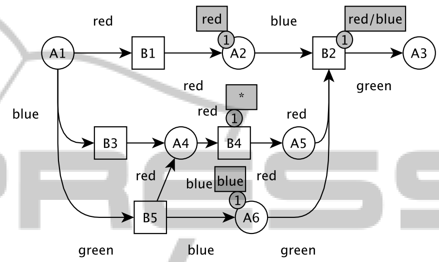

Figure 12: Special β-firing rule (case of multiple places) after firing.

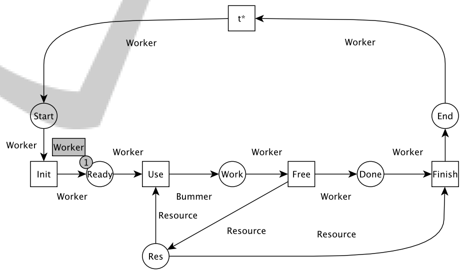

Figure 13: Example net for "Missing arc" fault. The dead net.

there is no arc from the transition "Init" to the place "Res" (the resource cannot be initialized).

When the place "Ready" is marked, then the transition "Use" cannot fire because the resource is required (Fig. 14). Then the transition "Finish" will be marked with the neutral element.

The place "Done" is enabled for special α-firing rule 2 ("Res" cannot fire, because it also needs token from "Use"). After that the place "Work" is enabled for simple α-firing rule 1. When the transition "Use" is marked, there is no place that is enabled for α-firing rules. But we can use β-firing rule for the place "Ready". It changes the marking in "Use" to *Worker*. After that the net should be analyzed (Fig. 14).

For the analysis of the fault the incidence matrix, the P- and T-reachability graphs are required. The incidence matrix looks as follows:

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

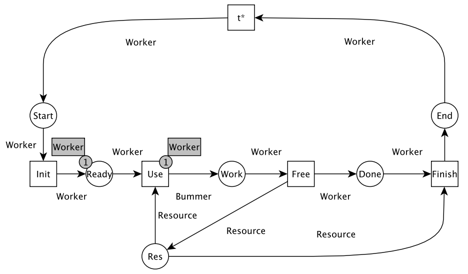

Figure 14: Example net for "Missing arc" fault. After backward analysis.

$$\begin{pmatrix} \text{-Worker} & 0 & 0 & 0 & \text{Worker} \\ \text{Worker} & \text{-Worker} & 0 & 0 & 0 \\ 0 & \text{-Resource} & \text{Resource} & 0 & 0 \\ 0 & \text{Worker} & \text{-Worker} & 0 & 0 \\ 0 & 0 & \text{Worker} & \text{-Worker} & 0 \\ 0 & 0 & 0 & \text{Worker} & \text{-Worker} \end{pmatrix}$$

The columns illustrate the transitions in order of firing. And the rows illustrate the places in order that they get marking.

The P- and T-reachability graphs look as follows (the order is the same as in the incidence matrix):

$$\begin{pmatrix} 0 & \text{Worker} & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\begin{pmatrix} 0 & \text{Worker} & 0 & 0 & 0 \end{pmatrix}$$

As we can see, the P-marking of the place "Ready" and T-marking of the transition "Use" are the same. But according to incidence matrix for firing this transition needs the token "Resource" from the place "Res" as well. To solve this problem the place "Res" should be initialized with this token.

"Wrong expression" Fault. The second example also repeats the example from previous chapter (p. 2), but now the expression of the arc from the transition "Use" to the place "Work" will be "Bummer".

When the places "Work" is marked, then the transition "Free" cannot fire because it needs a token "Worker" from "Work" (Fig. 15). Then the transition "Finish" will be marked with the neutral element.

The place "Done" is enabled for special α-firing rule 2 ("Res" cannot now fire, because it also needs token from "Use"). We can use β-firing rule for the place "Work". It changes the marking in "Free" to *Worker*. After that the net should be analyzed (Fig. 16).

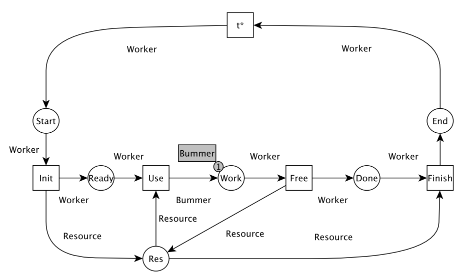

Figure 15: Example net for "Wrong expression" fault. Dead net.

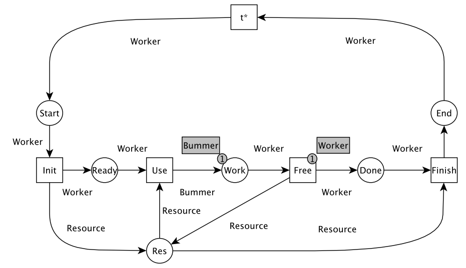

Figure 16: Example net for "Wrong expression" fault. After backward analysis.

The incidence matrix looks as follows (the difference with the previous example that it has initialization of "Res"):

$$\begin{pmatrix} \text{-Worker} & 0 & 0 & 0 & \text{Worker} \\ \text{Worker} & \text{-Worker} & 0 & 0 & 0 \\ \text{Resource} & \text{-Resource} & \text{Resource} & 0 & 0 \\ 0 & \text{Worker} & \text{-Worker} & 0 & 0 \\ 0 & 0 & \text{Worker} & \text{-Worker} & 0 \\ 0 & 0 & 0 & \text{Worker} & \text{-Worker} \end{pmatrix}$$

The P- and T-reachability graphs look as follows:

$$\begin{pmatrix} 0 & 0 & 0 & \text{Bummer} & 0 & 0 \end{pmatrix}$$

$$\begin{pmatrix} 0 & 0 & \text{Worker} & 0 & 0 \end{pmatrix}$$

As we can see, the P-marking of the place "Work" and T-marking of the transition "Free" are different. The incidence matrix shows that they are connected. To solve this problem one can either change the expression of the arc from "Work" to "Free" to "Bummer" or change the marking of the place "Work" to "Worker" (by changing the expression of arc from pre-transition).

5 CONCLUSIONS

The coloured workflow nets provide very powerful functionality for modeling and describing the behavior of business processes and resource management, but the debugging and resource-related fault detection in many cases has to be performed manually.

In this work the mechanism of the backwards analyses for automated troubleshooting and fault detection in coloured workflow net is proposed. The two sets of firing rules for the double-marked coloured workflow net were formulated and described. This method allows finding resource-related faults in the

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

business process with minimal amount of resources and time.

However the question about the efficiency and the computational complexity of the proposed concept remains open. Also a comparative study with existing approaches (Rabbi et al., 2010a), (Rabbi et al., 2010b) has to be done.

REFERENCES

Bouali, M., Barder, P., and Schon, W. (2009). Backward reachability analysis of colored petri nets. In *The European Safety and Reliability Conference (ESREL09)*, Prague, Czech Republic. Taylor and Francis Group.

Ganishev, V. (2013). Use of high level petri nets for modeling of adaptive workflow systems, in german: Anwendung von hoeheren erweiterten petri netzen fuer die modellierung von adaptiven workflow-systemen. Master's thesis, TU Ilmenau, Ilmenau, Germany.

Jensen, K. (1991). Coloured petri nets: A high level language for system design and analysis. In Rozenberg, G., editor, *Advances in Petri Nets 1990*, volume 483 of *Lecture Notes in Computer Science*, pages 342–416. Springer Berlin Heidelberg.

Jensen, K. and Kristensen, L. M. (2009). *Coloured Petri Nets*. Springer-Verlag Berlin Heidelberg, Cambridge, MA, USA.

Kristensen, L. M., Christensen, S., and Jensen, K. (1998). The practitioners guide to coloured petri nets. *International Journal on Software Tools for Technology Transfer*, 2(2):98–132.

Lautenbach, K. (2003). *Duality of Marked Place/Transition Nets*. University Koblenz-Landau, Koblenz, Germany.

Lautenbach, K. and Susewind, K. (2012). *Probability Propagation Nets and Duality*, volume 11 of *Arbeitsberichte des Fachbereichs Informatik*. University Koblenz-Landau, Koblenz, Germany.

Muller, J. and Schnieder, E. (2007). Duality in high level petri-nets - a basis to do diagnoses. In *Simulation Conference, 2007 Winter*, pages 629–636.

Rabbi, F., Wang, H., and MacCaull, W. (2010a). Compensable workflow nets. In Dong, J. and Zhu, H., editors, *Formal Methods and Software Engineering*, volume 6447 of *Lecture Notes in Computer Science*, pages 122–137. Springer Berlin Heidelberg.

Rabbi, F., Wang, H., and MacCaull, W. (2010b). Yawl2dve: An automated translator for workflow verification. In *Secure Software Integration and Reliability Improvement (SSIRI), 2010 Fourth International Conference on*, pages 53–59.

Scharfe, C. (2013). *Modeling of Adaptive Business Processes, in german: Modellierung von adaptiven Geschaeftsprozessen, Thesis*. PhD thesis, TU Ilmenau, Ilmenau, Germany.

van der Aalst, W. (2011). *Process Mining. Discovery, Conformance and Enhancement of Business Processes*. Springer-Verlag.

van der Aalst, W., Jrgensen, J., and Lassen, K. (2005). Lets go all the way: From requirements via colored workflow nets to a bpel implementation of a new bank system. In Meersman, R. and Tari, Z., editors, *On the Move to Meaningful Internet Systems 2005: CoopIS, DOA, and ODBASE*, volume 3760 of *Lecture Notes in Computer Science*, pages 22–39. Springer Berlin Heidelberg.

van der Aalst, W. and van Hee, K. (2002). *Workflow Management: Models, Methods, and Systems*. MIT Press, Cambridge, MA, USA.

van der Aalst, W., van Hee, K., ter Hofstede, A., Sidorova, N., Verbeek, H., Voorhoeve, M., and Wynn, M. (2011). Soundness of workflow nets: classification, decidability, and analysis. *Formal Aspects of Computing*, 23(3):333–363.

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS