Fachbereich 4: Informatik

Institute for IS Research

Institute for Web Science and Technologies

Probabilistic State Space Analysis for Inconsistency Measurement in Business Processes

Masterarbeit

zur Erlangung des Grades eines Master of Science (M.Sc.)
im Studiengang Informatik

vorgelegt von

Benedikt Knopp

Erstgutachter:	Prof. Dr. Matthias Thimm Institute for Web Science and Technologies
Zweitgutachter:	Dr. Carl Corea Institute for IS Research

Koblenz, im Oktober 2021

Erklärung

Hiermit bestätige ich, dass die vorliegende Arbeit von mir selbstständig verfasst wurde und ich keine anderen als die angegebenen Hilfsmittel – insbesondere keine im Quellenverzeichnis nicht benannten Internet-Quellen – benutzt habe und die Arbeit von mir vorher nicht in einem anderen Prüfungsverfahren eingereicht wurde.

	Ja	Nein
Mit der Einstellung dieser Arbeit in die Bibliothek bin ich einverstanden.	☒	□
Der Veröffentlichung dieser Arbeit im Internet stimme ich zu.	☒	□
Der Text dieser Arbeit ist unter einer Creative Commons Lizenz (CC BY-SA 4.0) verfügbar.	☒	□
Der Quellcode ist unter einer GNU General Public License (GPLv3) verfügbar.	□	☒
Die erhobenen Daten sind unter einer Creative Commons Lizenz (CC BY-SA 4.0) verfügbar.	□	☒

Koblenz, den 01.10.2021 .... Knopp

(Ort, Datum) (Unterschrift)

Zusammenfassung

In der heutigen Zeit gestaltet sich die Modellierung von Geschäftsprozessen aufgrund der Vielzahl möglicher Prozessverläufe und der Abhängigkeiten zwischen Prozessvariablen- und verläufen häufig als fehlerbehaftete Angelegenheit. Wenn die Korrektheit der modellierten Prozesse nicht gewährleistet werden kann, sollten Modellierer zumindest anhand einer Kennzahl das Risiko oder die Schwere der Modellierungsfehler einschätzen können sowie idealerweise auf diejeningen Elemente des Prozesses hingewiesen werden, welche die Fehler verursachen. In der vorliegenden Masterarbeit werden Werkzeuge ausgearbeitet um solche globalen Inkonsistenzen sowie die Verantwortlichkeiten einzelner Elemente zu quantifizieren. Dazu werden zunächst verschiedene Konzepte der Korrektheit von Geschäftsprozessen identifiziert, um anschließend pro Kriterium Kennzahlen für Inkonsistenz zu entwickeln. Diese Maße fußen auf einer probabilistischen Zustandsraumanalyse der Geschäftsprozesse, die Zustandsräume werden dabei als zeitdiskrete Markov-Ketten aufgefasst. Da im Lebenszyklus des kontemporären Geschäftsprozessmanagements die Modellierung von Prozessen häufig inkrementell verläuft, Modelle also schrittweise angepasst werden, bietet der probabilistische Ansatz ein Fundament für eine datengestützte Kalibierung der Kennzahlen anhand historischer Daten von Prozessverläufen.

Abstract

Contemporary business processes modelling is error-prone due to the variety of different process courses in large models and intricate dependencies among process variables as well as between process variables and process traces. If the correctness of these workflows cannot be guaranteed, a modeler might wish at least for an assessment of severity to which the correctness of the process model is hurt, and for elements of the process which are culpable for the violation to be pointed out. This master's thesis aims at providing means for such a quantitative assessment of inconsistencies. To this end, we firstly identify various characteristics that are commonly desired for workflows, and then propose means to evaluate degrees of conformance. We conduct a probabilistic analysis of the state spaces of workflows on which these means rely, namely by interpreting these state spaces as time-discrete Markov chains. Since the modelling of workflows is oftentimes conducted in an incremental fashion during the lifecycle of business process management, a probabilistic approach allows to calibrate our measures based on historical event data.

Table of Contents

List of Figures

1. Introduction

The field of *Business Process Management* (BPM) engages with the design, analysis and execution of business processes, or workflows, in organizations. A well-established tool (Weske, 2012) for the modelling of such workflows is the *Business Process Modelling Notation* (BPMN). With the help of BPMN, real-world processes can be represented graphically as a sequence flow of business activities and events, as illustrated in Fig. 1.

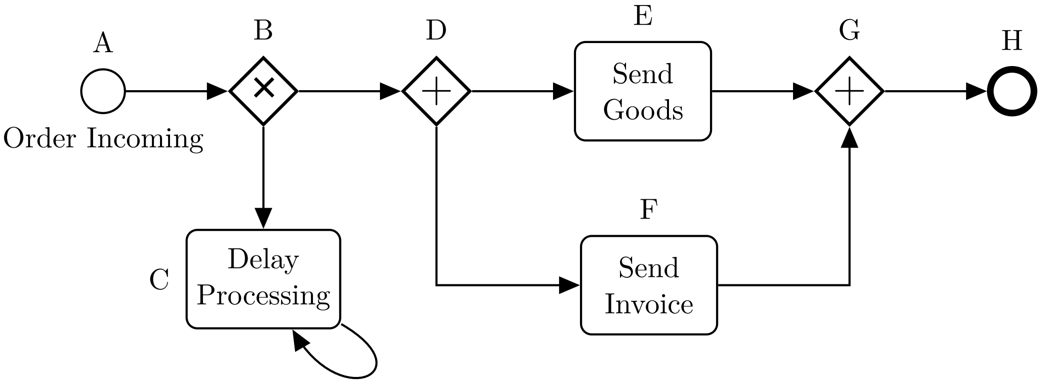


Figure 1: A sample BPMN model, describing the handling of orders in some commercial context, with events (A, H), activities (C, E, F), an exclusive (B) and inclusive gateways (D, G).

Modelling business processes may be subject to various errors. On the one hand, low-level errors may occur, such as a misuse of notation or violations of general workflow conventions, for example the need for a process to always terminate. The process fragment depicted in Fig. 1 exhibits such a weakness: If a process instance is routed at the exclusive gateway B in direction of the task *Delay Processing*, the process cannot reach the final event H and therefore never terminates. On the other hand, higher-level errors concerning the business domain may occur, for example non-compliance with external regulations, so-called *business rules*. Naturally, a human modeler may assess varying degrees of severity to different errors, for example depending on the importance of some violated business rule, or depending on the likelihood of an error to actually occur. The field of *inconsistency measurement* (Bertossi et al., 2005; Thimm, 2019) is engaged with developing means for such a qualitative assessment of contradictions in knowledge bases. In general, the term inconsistency may be defined for various underlying logical formalisms, and has been richly discussed in the scope of propositional logic (Thimm, 2019). Recent work (Corea and Delfmann, 2019; Corea and Thimm, 2020; Corea et al., 2019) has adapted and extended these methods of propositional logic for the area of business process management.

In this thesis, means are elaborated to measure inconsistency in process models. To this end, we utilize Petri nets to formalize execution semantics of processes and then explore the state space of these nets in a probabilistic fashion. In the following, the goal of this thesis and this basic methodology will be motivated by identifying gaps in current research.

1.1. Motivation

Two main aspects of the work in this thesis are to be substantiated. Firstly, we identify the need for a low-level, flow-centred way of analyzing errors in business process models. Secondly, we argue to embed these means into a probabilistic framework.

While working with business process models and business rules has been well-established in industry, this work turned out error-prone due to a collaborative and incremental modelling process as identified by Batoulis and Weske (2017). Particularly, the authors noted a high degree of inconsistency in industrial business rule bases. To tackle this issue, Corea and Thimm (2020) developed measures to quantify the degree of inconsistency in such rule bases. Besides that, Ciccio et al. (2017) and Corea and Delfmann (2019) identified a similar problem in the scope of process-mined declarative rule bases. All works analyze declarative process models. Opposed to declarative models, procedural models such as BPMN models describe workflows in an imperative manner by specifying an explicit execution order.

We argue that measuring inconsistencies in process models should not only capture declarative aspects, such as business rules and constraints, but also extend to procedural models. Rule bases serve the purpose of verifying actual procedural models as their counterpart. If some constraint is to be verified in a process model, questions arise whether, for instance, elements that are to be checked actually exist in the process model and if they are reachable in the course of a process execution. This depends on the sequence flow of the process model. Therefore, we propose to analyze inconsistencies in process models based on an analysis of the process state space. This analysis should be sensitive to particular elements, such that *culpabilities* for erroneous behaviour can be identified.

Secondly, we motivate to design such a measure in a probabilistic fashion. Naturally, re-occurring process executions exhibit a statistical model based on historical data. On the one hand, real-world processes may be non-deterministic due to human involvement. On the other hand, processes may be widely or fully automatized, and be therefore determined completely by process variables. We are encouraged by van der Aalst (2015) to utilize the growing availability of data, in particular event logs. Measurement which analyzes processes based on the reachability of process states could elegantly be adjusted by process data. Also, data-supplied measures could be easily integrated into the BPM Lifecycle (Weske, 2012). The BPM Lifecycle proceeds iteratively in four phases, as briefly depicted in Fig. 2. In the design phase, models are constructed based on organizational and technical requirements. In the configuration phase, these models are embedded into the organizational software

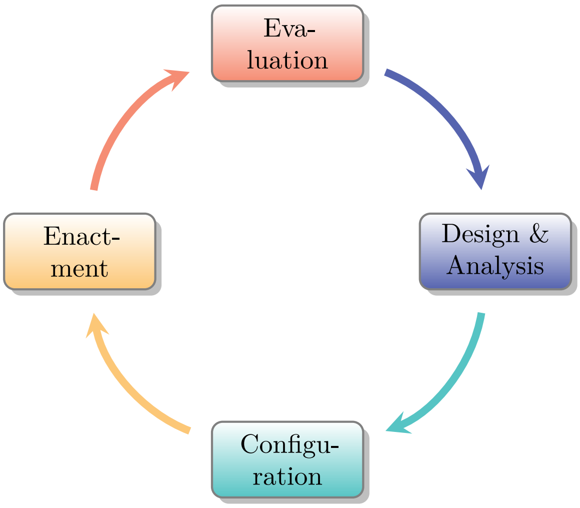


Figure 2: The BPM Lifecycle according to (Weske, 2012). During the (re-)design phase, modelers may receive data-driven support.

systems. Process instances are then executed in the enactment phase, and the resulting historical data may be evaluated for process enhancement. In a subsequent redesign, existing models may be adjusted based on that data (Lu et al., 2008). We argue that at this point, inconsistency measures adjusted according to the data could be deployed and give helpful assessments of the previous global and local process consistencies.

1.2. Research Aim

This thesis targets at supporting companies in modelling their workflows in a consistent way. To this end, we provide a quantitative assessment of the inconsistency degree in procedural models that is sensitive to individual element culpability. We focus on the aspect of consistency as opposed to compliance as a low-level view on technical correctness opposed to a high-level view on domain-specific correctness.

RA: Develop measures that suitably describe inconsistencies in workflow models and pinpoint culpable elements.

At this point, it still remains ambiguous firstly what formalism to use for defining procedural models, and then what the term inconsistency actually means for procedural models. Also, besides that question concerning the nature of the term inconsistency, questions arise concerning the nature of measures for that inconsistency. One should be clear about what desirable properties such a measure should have, for example, how a measure applied to some model behaves when adding inconsistent information (Unruh et al., 2021).

RQ1: How should inconsistency on workflow models be defined and how to measure it?

RQ2: What are desirable properties that inconsistency measures on workflow models should have?

As a formalism to capture workflow semantics, we will deploy Petri nets. Firstly, this is due to the well foundations Petri nets have for modelling workflows (Weske, 2012). Secondly, we remain independent from the top-level modelling language, since Petri nets can formalize a variety of such languages. However, BPMN is the de-facto industry standard and also highly expressive, for example, in modelling decision- and data-aware processes (Batoulis et al., 2017). Therefore, we made use of BPMN to create and investigate a running example. We are confident that a research leaned on conventional BPMN models will yield results that are generally applicable.

RQ3: To what extend can the newly defined measures capture the expressivity of business process modelling languages?

It is important to note that the extend of expressivity to be captured directly affects the development of the inconsistency measures. The more expressive a language, the more sophisticated the analytical means may need to be. For example, simple workflows may be formalized with simple Petri nets, but more recent works (de Leoni et al., 2018; Haarmann et al., 2018) deploy the more expressive Coloured Petri nets in order to describe data- and decision-awareness of processes. Therefore, techniques of a state-space analysis of such higher nets are available. However, in the scope of this thesis, one should take care not to adapt these techniques too quickly. Two considerations support this statement. Firstly, exploiting the state space of Petri nets is not a novel approach, but it is novel in the scope of inconsistency measurement. This also encourages to ask preliminary questions such as RQ2 to ascertain that practical measures will be safeguarded by a clear paradigm. Secondly, even simple Petri nets are considerably powerful - not Turing-complete, but Turing-complete with slight extensions (Zaitsev, 2014). Thus, we are confident that after developing and evaluating simple measures in-depth, these measures may be extended rather easily to apply to a more expressive formalism.
In the following, we infer concrete objectives from the research questions to embark upon.

RO1: Identify desirable low-level characteristics of workflows. (RQ1)

RO2: Identify desirable characteristics for inconsistency measures. (RQ2)

RO3: Develop methods to quantify workflows concerning these characteristics. (RQ1)

RO4: Assess the measures in the light of the postulates. (RQ1, RQ2)

RO5: Evaluate the measures in the light of (RQ3).

1.3. Structure of this Work

This thesis proceeds as follows. Firstly, the formalism of simple Petri nets will be introduced and linked to the application domain of business processes, yielding the so-called workflows. Secondly, the field of inconsistency measurement which provides concepts for quantitative assessments of knowledge bases will be introduced. These two preliminary aspects constitute Sec. 2. It is our goal to tie these fields together, since the application of inconsistency measurement in procedural languages is relatively new (Unruh et al., 2021). To this end, we develop a framework which allows us to analyze workflows in a probabilistic fashion, and also clarify terminological questions about what characterizes consistency or inconsistency in a workflow. On that basis, measures for inconsistencies will be proposed. This constitutes the content of Sec. 3. After that, we will delve into a more fine-grained assessment of workflows and develop means to pinpoint inconsistency culprits, in Sec. 4. Having those means ready, we will apply them to a complex business process in order to assess the means in the light of the research questions. After that, in Sec. 5, we regard work related to the matter of this thesis. In Sec. 6, we draw conclusions and give outlooks for possible future work.

2. Preliminaries

This section recalls the well-known Petri net formalism tracing back to (Petri, 1962), which will be built upon to conduct the intended measuring of inconsistency in business process models. Also, the general notion of Petri nets is narrowed down to the more specific use-case of workflows, or business processes. For these workflows, characteristics are introduced (Weske, 2012) which in practise are commonly considered to be desirable. Thereupon, because we would like to propose quantifications of these characteristics, we draw a bow to the field of inconsistency measurement.

2.1. Petri nets

In practise, business processes are not usually modeled with the help of bare low-level Petri nets. Instead, higher-level languages, especially the Business Process Modelling Notation (BPMN, (Weske, 2012)) and Event-Driven Process Chains (EPC, (Scheer et al., 2004)) are common. However, Petri nets can capture the execution semantics of these higher-level languages, even to the full extends with Coloured Petri Nets (CPNs, (Jensen, 1986). However, this thesis elaborates preliminaries to link procedural models to inconsistency measurement, so we firstly focus on simple, uncoloured nets. The definitions given in this section can be retraced for example in the work of Murata (1989).

Definition 2.1. (Petri net). A simple *Petri net* is a tuple $N = (P, T, A, W)$ with $P \cap T = \quad$ and $A \quad (P \times T) \quad (T \times P)$ where we call P the *places*, T the *transitions* and A the *arks* of the net. The *ark weighting* W is a function $W : A \to \mathsf{N}_+$. For each $a \quad P \quad T$, we define the *pre-set* $\bullet a$ and the *post-set* $a\bullet$.

$$\begin{aligned} \bullet a &= \{b \quad P \quad T \mid (b, a) \quad A\} \\ a\bullet &= \{b \quad P \quad T \mid (a, b) \quad A\} \end{aligned}$$

In the following, if the weight of an ark is not specified explicitly, it amounts to 1. The idea of the place-transition structure in Petri nets is to model discrete time-steps processes with the help of a token play. Places bear tokens, and transitions consume certain amounts of tokens in their pre-set to produce certain amounts of tokens in their post-set, where the amounts are prescribed by the ark weight function.

Definition 2.2. (Execution Semantics). A *marking* is a function $M : P \to \mathsf{N}$. We call a place $p \quad P$ *marked* if $M(p) > 0$. We call a transition $t \quad T$ *enabled* in N at M if every place p with $(p, t) \quad A$ is marked with $M(p) \geq W\big((p, t)\big)$. If t is enabled, the *firing* of t at M is a relation $M \overset{t}{\Rightarrow} M$ such that

$$M\ (p) = \begin{cases} M(p) - W\big((p, t)\big), & \text{if } (p, t) \quad A \\ M(p) + W\big((p, t)\big), & \text{if } (t, p) \quad A \\ M(p), & \text{otherwise} \end{cases}$$

If the transition itself is not of interest, we use the notation $M \Rightarrow M$ for brevity. A *firing sequence* is a tuple of markings $(M^0, ..., M^n)$ with $n \geq 1$, where for all $i = 0, ..., n-1$, $M^i \Rightarrow M^{i+1}$. We write $M^0 \Rightarrow M^n$ to denote that there exists a firing sequence $(M^0, ..., M^n)$.
A *marked Petri net* (N, M) is a Petri net N with a marking M over its places. To facilitate the notation, we fix the ordering of places $(p_0, ..., p_{|P|})$ and display markings in vector form $(M(p_0), ..., M(p_{|P|}))$ as elements in $\mathsf{N}^{1\times|P|}$.

Process courses can be modeled by means of transitions in between markings. To illustrate this, consider the abstract examplary BPMN model in Fig. 1. At this point, we do not go into the details of the BPMN semantics; however, a simple example with basic constructs of BPMN and a possible mapping to Petri nets (Dijkman et al., 2008)) are alluded in Fig. 3 to facilitate the understanding of the Petri net formalism in potential application areas.

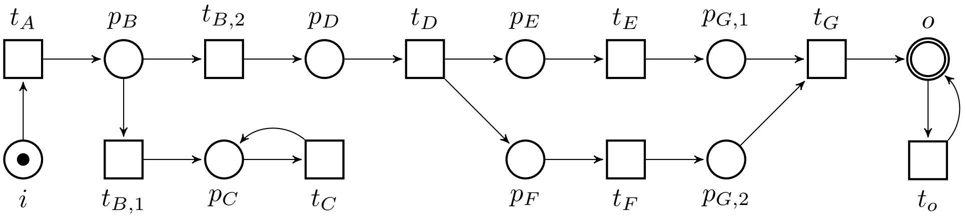
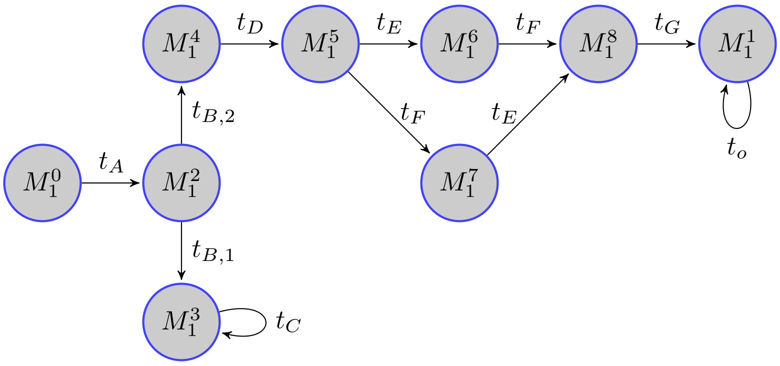


Figure 3: A Petri Net representation (N_1, M_1^0) of the BPMN model from Fig. 1, where $M_1^0(p_A) = 1$ and $M_1^0(p) = 0$ for all other places p. All arks have weight 1. The mark on place o indicates it as the *final place*, which is further explained in Def. 2.5.

The business process starts with the start event *Order Incoming* at A, the correspondent Petri net concept is that of an *initial place* (i). The course to the exclusive gateway B is formalized by a change of markings in the Petri net via firing transition t_A, which deducts the token in i and adds a token in p_B. The exclusive choice at B then to either proceed to the task *Delay Processing* at node C or to the parallel gateway D is left implicit in the Petri net, since both transitions $t_{B,1}$ and $t_{B,2}$ are enabled. If the process takes the turn to the task at C, the repeated execution of that task corresponds to the repeated firing of transition t_C in the Petri net. If the process takes the turn to the parallel gateway D, the parallelized execution of the tasks *Send Goods* at E and *Send Invoice* at F is modelled in the Petri net by t_D producing tokens in both p_E and p_F, leading to both t_E and t_F being enabled at the same time. The firing order of these transitions, aka the execution order of the tasks then is non-deterministic. However, the joining gateway G will only continue the process execution after both tasks are completed. In terms of the Petri net representation, transition t_G may fire only if the post-sets of t_E and t_F are filled. The process then terminates when reaching the end event H, which corresponds to a marking where exactly one token is in the net in o and firing transition t_o just fixes the final state of the process.



$$
\begin{array}{rcccccccccc}
 & & i & p_B & p_C & p_D & p_E & p_F & p_{G,1} & p_{G,2} & o \\
M_1^0 = & (& 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \ ) \\
\\
M_1^1 = & (& 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \ ) \\
M_1^2 = & (& 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \ ) \\
M_1^3 = & (& 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \ ) \\
M_1^4 = & (& 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \ ) \\
M_1^5 = & (& 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \ ) \\
M_1^6 = & (& 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \ ) \\
M_1^7 = & (& 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \ ) \\
M_1^8 = & (& 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \ )
\end{array}
$$

Figure 4: The reachability graph R_1 for the Petri net presented in Fig. 3 (above). The definition of the markings as token distributions over places (below).

This token play describes a transitioning in between possible markings of the Petri net. Given an initial marking M^0 with exactly one token in p_A, one can iteratively search for possible successor markings and thus explore the *state space* of the net, as depicted in Fig. 4 for the example. This state space may also be referred to as the *reachability graph* if the focus lays more on graph-theoretical aspects.

Definition 2.3. (Reachability Graph, State Space). For a marked Petri net (N, M^0), the *reachability graph* or *state space* over the net is the smallest set $\mathsf{R} = (\mathsf{M}, \mathsf{E})$ such that $M^0 \quad \mathsf{M}$ and, if $M \quad \mathsf{M}$ and $M \overset{t}{\Rightarrow} M$, then $M \quad \mathsf{M}$ and $(M, M\ ) \quad \mathsf{E}$. A marking $M \quad \mathsf{M}$ is called *reachable* if $M^0 \Rightarrow M$.

Definition 2.4. (Boundedness). A marked Petri net (P, T, A, W, M^0), with reachability graph $\mathsf{R} = (\mathsf{M}, \mathsf{E})$ is called bounded if and only if there is a $k \quad \mathsf{N}$ such that for all $p \quad P, M \quad \mathsf{M} : M(p) \leq k$.

Fig. 4 depicts the state space of the running example and defines what token distribution each of the markings describes. As shown in Fig. 4, we label each edge in the reachability graph (M, E) with the firing transition that leads to the respective change of markings.

2.2. Workflows and Soundness

While the concepts introduced in the previous section describe Petri nets in a very general fashion, the process depicted in Fig. 3 already captures certain aspects of the kind of processes that are in the focus of this thesis, that is, *workflows* or *business processes*. Also, the language constructs of BPMN reflect conventions to describe such workflows: For example, a workflow commonly features a designated start point and a designated end point, corresponding to a start event and an end event found in BPMN models. In the following, we define what a workflow is in the Petri net formalism. If not indicated otherwise, the discussion in the rest of this thesis is limited to nets that satisfy these workflow characteristics.

Definition 2.5. (Workflow, cf. van der Aalst et al. (2011), Weske (2012)). Let (N, M^0) be a marked Petri net with state space (M, E). We call the net a *workflow*, if

1. There is a designated place $i \quad P$, called the *initial place*, where $\bullet i =$ and for all other places p, $\bullet p =$ (single source).
2. There is a designated place $o \quad P$, called the *final place*, where $o\bullet = \{t_o\}$ and $t_o\bullet = \{o\}$ and for all other places p, $p\bullet =$ (single sink).
3. $M^0(i) = 1$ and $M^0(p) = 0$ for all places $p = i$ (initial marking), and $M^1 \quad \mathsf{M}$ with $M^1(o) = 1$ and $M^1(p) = 0$ for all places $p = o$ (final marking).
4. For all transitions $t \quad T$, $t\bullet =$ and $\bullet t =$.
5. The state space is *continuable* in the sense that for all $M \quad \mathsf{M}$, there is $M \quad \mathsf{M}$ with $(M, M\ ) \quad \mathsf{E}$.

This definition of a workflow is comparable to the definitions of van der Aalst et al. (2011) and Weske (2012). However, the classical definition requires also a *connectivity* property for nodes, that is, the property that every place and transition needs to be on a path between the initial place and the final place. We omit this property here because we will explicitly check the degree of non-connectivity, more exactly, of non-termination, in the context of inconsistency measurement. As a relaxed criterion, we introduce property (4). Also, we require the property of continuability (5) as a technical extension that asserts the correctness of the probabilistic analysis; this will become clearer in Sec. 3.1. As for the final marking M^1, that marking will be included in the following in all state spaces, even if it should not be reachable, because it plays a crucial role in our intended analysis of workflow correctness criteria.

Because of their deployment in industry, there is an economical interest in this correctness of workflows. The exact meaning of correctness certainly depends on the application scenario. However, this question has initiated a vivid academic discource (Batoulis et al., 2017; Dehnert and Rittgen, 2001; Martens, 2003; van der Aalst, 1997). One common characteristics is the demand on a workflow to always terminate (*option to complete*). This demand seems intuitively reasonable; a working process that is non-terminating may certainly be affected by some technical problem. Another property is the *dead-transition freeness*, that is, the demand that all transitions should be able to fire at some point in the workflow. This demand seems also reasonable, because a transition that never fires is either redundant, which should initiate a remodelling of the workflow in a reduced manner, or the transition is not redundant but another modelling error prevents that the transition can be reached. A third aspect is that of *fairness*, that is, the scale to which different transitions fire in proportion to each other. More on fairness and a proposition to assess the degree of (un-)fairness will be discussed in Sec. 3.4. Firstly, we sum up the first two properties in the classical notion of *soundness*.

Definition 2.6. (Soundness of Workflows (van der Aalst et al., 2011; Weske, 2012)). Let (N, M^0) with $N = (P, T, F)$ a workflow with state space $\mathsf{R} = (\mathsf{M} , \mathsf{E})$. The workflow is called *sound* if and only if there is

1. The Option to Complete: For every reachable state M, there is a firing sequence leading from M to the final marking where there is exactly one token in o.

$$\begin{aligned} M \quad & \mathsf{M} : M^0 \Rightarrow M \\ = \quad & M \Rightarrow M^1 \end{aligned}$$

2. Proper Completion: If a token is in o, there is exactly one token in o and all other places are empty.

$$M \quad \mathsf{M} : M(o) > 0 \;=\; \quad M(o) = 1 \text{ and } M(p) = 0 \quad p \quad P \setminus \{o\}$$

3. Freeness from Dead Transitions: Each transition can be activated in some firing sequence starting from the initial marking.

$$t \quad T \quad M, M \quad \mathsf{M} : M^0 \Rightarrow M \overset{t}{\Rightarrow} M$$

To facilitate working with the soundness criteria, we make use of the following result from van der Aalst et al. (2011).

Lemma 2.1. If a workflow has the option to complete (1), then proper completion is also satisfied (2).

Proof. Comparable to the proof from van der Aalst. If there is improper completion, i.e. a token in the final place while another token is still at another place, the case that only one token is in o and no other token in the net cannot occur: The token in the final place does not vanish and the remaining token either stays in its place or produces tokens by firing transitions, because firing a transition produces tokens (cf. Def. 2.5, (4)). □

A structural property similar, but not equivalent to the dead transition freeness (3), is the so-called *liveness*.

Definition 2.7. (Liveness, (van der Aalst, 1997)). A workflow is called *live*, if and only if at every reachable state, every transition can be fired in some firing sequence, that is:

$$t \quad T, M \quad \mathsf{M} : M^0 \Rightarrow M \;=\; \quad M \,, M \; : M \Rightarrow M \overset{t}{\Rightarrow} M$$

In other words, freeness from dead transition means that each transition can be fired in some firing sequence starting from the initial marking, while liveness means that this should hold not only for the initial marking, but for every reachable state. Verifying liveness can be a strong means for veryfing workflow correctness in general, as we will further discuss in Sec. 3.3.

2.3. Inconsistency Measurement

As a framework in which to embed the quantitative assessment of workflow correctness which we intend to propose, we access the field of inconsistency measurement (Grant, 1978; Thimm, 2019). This area is engaged with quantifying degrees of conflicts, or contradictions, within knowledge bases in logics of various kinds, such as propositional logic or first-order-logic. Measures were proposed that deploy various mathematical concepts, for an overview see for example Thimm (2017). A common denominator of all measures is the idea of mapping a knowledge base to a non-negative real number.

Definition 2.8. Let K be a knowledge base. An inconsistency measure on K is a function

$$\mathsf{I} : \mathsf{K} \rightarrow \mathsf{R}_{\geq 0}$$

Here, the knowledge base K is some set of formula over a signature in some logic. However, also elements of a procedural language such as Petri nets may be interpreted as a knowledge base. A preliminary work for this in the inconsistency measurement area was elaborated by Unruh et al. (2021). Vital for the decision on how to define an inconsistency measure is the question which properties a measure should hold, such that different measures may be comparable to one another. For this purpose, rationality postulates are proposed, for example (Hunter and Konieczny, 2006):

- **Consistency**: $\mathsf{I}(\mathsf{K}) = 0$ if and only if K is consistent, i.e. K does not contain a contradiction in the sense of the respective logic.

- **Normalization**: $0 \leq I(K) \leq 1$, or more generally, the values are bounded. This makes nets of different sizes comparable to each other.
- **Monotony**: $K \quad K \; = \quad I(K) \leq I(K\ )$, i.e. more information leads to more or equal inconsistency.
- **Free Formula Independence**: If α is a free formula in K, then $I(K \quad \{\alpha\}) = I(K)$, i.e. formulae not entangled in contradictions have no impact on the inconsistency value.
- **Dominance**: If α 0 and $\alpha \quad \beta$, then $I(K \quad \{\alpha\}) \geq I(K \quad \{\beta\})$, i.e. formulae that are logically stronger potentially cause more inconsistency.

In the scope of this thesis, we raise the question what postulates should be satisfied for the inconsistency measures on workflows, and more primarily, what meaningful postulates may be defined. Unruh et al. (2021) proposed such postulates for Petri nets in general. For this thesis, the narrowing on a subset of these Petri nets, namely workflows, should be regarded. Also, this thesis differs in its probabilistic methodology and in its method of dynamically regarding state transitions, as opposed to the static net analysis of Unruh et al. All in all, we need to discuss in how far workflows and their state spaces can be regarded as knowledge bases in the classical sense, in order to properly define postulates such as the monotony criterion. In the beginning of the next section, the need for such a discussion will become more apparent after specifying the probabilistic methodology.

3. Inconsistency Measures on Workflows

In the previous section, workflows were defined as special Petri nets, forming a subset of these nets which capture the execution semantics of business processes as deployed in business practise. This thesis is intended as a contribution to an analysis on such workflows based on inconsistency measures. In this section, propositions are made on how exactly this analysis may be conducted. Before specifying the actual measures, we will specify as the backbone of these measures the details of our intended probabilistic approach.

3.1. Probabilistic State Space Analysis

We interpret workflows as time-discrete processes which in particular may be non-deterministic in the courses that they take. We furthermore assume that the choices of these courses can be described by some probability function. In practise, one might define such a function based on historical data accumulated after observing the process. In the basic case that no such data is available, one might naively assume that the probabilities of different courses are equally distributed.

Definition 3.1. (Probability Functions). On a state space $\mathsf{R} = (\mathsf{M}, \mathsf{E})$ over a Petri net (P, T, A), a *probability function* is a mapping $\eta : \mathsf{M} \times T \times \mathsf{M} \to [0,1]$ with $\eta((M, t, M\ )) > 0 \quad = \qquad M \xrightarrow{t} M$ and $\sum_{(t,M\ )\quad T \times \mathsf{M}} \eta((M, t, M\ )) = 1$ for all $M \qquad \mathsf{M}$. We overload this notation with $\eta : \mathsf{E} \to [0,1]$ where $\eta((M, M\ )) = \sum_{t \quad T} \eta((M, t, M\ ))$. We write $M \underset{p}{\overset{t}{\Rightarrow}} M$ and $M^{i_0} \underset{q}{\Rightarrow} M^{i_n}$ for denoting the firing of t at M with probability $p = \eta((M, t, M\ ))$ and for denoting the existence of some (not necessarily unique) firing sequence $(M^{i_0}, ..., M^{i_n})$ with $q = \prod_{i=0}^{n-1} \eta((M^{i_j}, t_i, M^{i_{j+1}}))$ where $M^{i_j} \overset{t_i}{\Rightarrow} M^{i_{j+1}} \quad j = 0, ..., n - 1$.

Definition 3.2. (Regular Probability Function). A probability function is called *regular*, if for all $(M, M\ ) \quad \mathsf{E}, t \quad T$ with $M \overset{t}{\Rightarrow} M\ $, $\eta((M, t, M\ )) > 0$.

In Fig. 5, probabilities were added to the labels of the running example's state space from Fig. 4. By choosing various values for α, β, we can describe all possible probability functions on that space.
The property that for every marking, the sum of probabilities of outgoing edges equals to 1 formalizes the intuition that the state of the described processes should be well-defined at any time-step. This coheres to the nature of time-discrete Markov processes. Even if at some point in time there is no transition enabled, at least the process stays in its current state, and thus a technical transition might be added that is enabled at that current state but does not change the state of the net. This technical property corresponds to the continuity property of workflows demanded in Def. 2.5, (5). The property of *regularity* is added to prevent the problems in the analysis of correctness criteria for transitions with firing probability 0, more on that

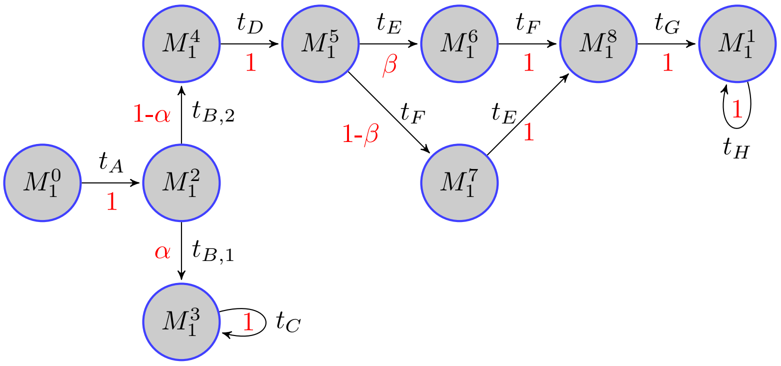


Figure 5: The reachability graph R_1 for the Petri net (N_1, M_1^0) presented in Fig. 3. The probabilities on the edges describe all probability functions η_1.

will be explained in Sec. 3.5. Now, given some initial state, we can conduct such a Markov process and explore the state space in a probabilistic fashion.

Definition 3.3. (Markov Process). Let (N, M^0) be a marked Petri net with reachability graph $R = (M, E)$, $|M| = k \leq \infty$. Let then η be a probability measure on R. The transition matrix $T \quad [0,1]^{k\times k}$ is defined as follows:

$$T_{ij} = \eta((M^i, M^j))$$

The *initial state* s^0 is a probability distribution over all markings:

$$s^0 \quad [0,1]^{1\times k}, \sum_j s_j^0 = 1$$

The *Markov sequence* or *Markov walk* $(s^n)_{n \quad \mathrm{N}}$ with initial member s^0 is defined as follows:

$$s^{n+1} = s^n \cdot T$$

The *Markov limit* s^∞, if it exists, is defined as:

$$s^\infty = \lim_{n\to\infty} s^n$$

An important property of the state probabilities is the so-called Chapman-Kolmogorov-Equation.

Theorem 3.1. (Chapman-Kolmogorov-Equation, see e.g. Meintrup and Schäffler (2006)). For some markings M^i, M^j, n $\geq$ 1, let

$$P_{ij}^n := \{((M^i, ..., M^j), p) \mid M^i \underset{p_1}{\rightarrow} ... \underset{p_n}{\rightarrow} M^j, p = \prod_{r=1}^{n} p_r\} \quad M^n \times [0,1]$$

the set of all firing sequences of length n from M_i to M_j. Then, for all $N \quad \mathsf{N}$,

$$s_j^{N+n} = \sum_{i=0}^{k} s_i^N \cdot \sum_{(M,p) \quad \mathsf{P}_{ij}^n} p$$

In particular, for $i = 0$ and $N = 0$, with $s_0^0 = 1$, we have

$$s_j^n = \sum_{(M,p) \quad \mathsf{P}_{0j}^n} p$$

This theorem states that the total probability to be in a certain state after a certain length in time is determined over all single paths of that length leading to the state. *Example.* For the graph depicted in Fig. 5, we have the following transition matrix with respect to the parameters α, β.

$$\mathsf{T} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & \alpha & 1-\alpha & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \beta & 1-\beta & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Starting with the initial marking M_1^0 governs our initial state s^0.

$$M_1^0 = \begin{array}{ccccccccc} i & p_B & p_C & p_D & p_E & p_F & p_{G,1} & p_{G,2} & o \\ \left(\ 1\right. & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \left.0\ \right) \end{array}$$

$$s_0 = \begin{array}{ccccccccc} M_1^0 & M_1^1 & M_1^2 & M_1^3 & M_1^4 & M_1^5 & M_1^6 & M_1^7 & M_1^8 \\ \left(\ 1\right. & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \left.0\ \right) \end{array}$$

This leads to a Markov sequence as follows.

$$\begin{aligned} s^1 &= (\ 0,\ 0,\ 1,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0\ ) \\ s^2 &= (\ 0,\ 0,\ 0,\ \alpha,\ 1-\alpha,\ 0,\ 0,\ 0,\ 0\ ) \\ &\vdots \\ s^\infty &= (\ 0,\ 1-\alpha,\ 0,\ \alpha,\ 0,\ 0,\ 0,\ 0,\ 0\ ) \end{aligned}$$

The limit of the Markov sequence reflects the expected state of the process in the long term. This limit will be particularly interesting for our purpose, because we would like to assess correctness properties by examining the long-term behavior of workflows. However, the examplary workflow in Fig. 6 shows that this limit does not necessarily exist, in particular, the state of processes alters and thus the Markov sequence does not converge. For the initial state $s^0 := (1,\ 0,\ 0,\ 0)$, we have

$$\begin{aligned} s^n &= (0,\ 1/2,\ 1/2,\ 0) \text{ for } n \text{ uneven} \\ s^n &= (0,\ 1/2,\ 0,\ 1/2) \text{ for } n \text{ even} \end{aligned}$$

and therefore the Markov limit $\lim_{n \to \infty} s^n$ does not exist. However, as an alternative, one might consider the *mean* long-term state of the process.

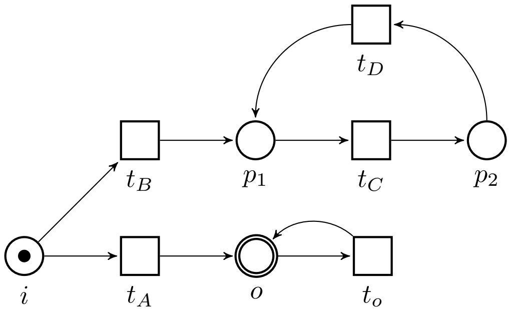
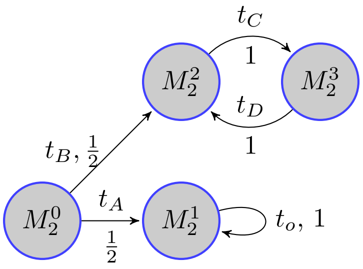


Figure 6: An examplary initially marked Petri net N_2 whose Markov sequence does not converge.



$$T = \begin{pmatrix} 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Figure 7: The state space R_2 of the net N_2 from Fig. 6 with an exemplary probability measure η_2 and the corresponding (stochastic) transition matrix.

Definition 3.4. (Mean state). For a Markov sequence s^n, the *mean state sequence* $\bar{s}^n$ is defined as follows:

$$\bar{s}^n = \frac{1}{n+1} \cdot \sum_{i=0}^{n} s^n$$

The *mean state limit* $\bar{s}^\infty$ is defined as follows:

$$\bar{s}^\infty = \lim_{n \to \infty} \bar{s}^n$$

Theorem 3.2. The mean state limit is well-defined, i.e. the mean state sequence converges.

Proof. See Doob (1991), Th. 2.1. □

Example. For the state space depicted in Fig. 7, we have:

time step n	s^n	$\bar{s}^n$
0	$(1,0,0,0)$	$(1,0,0,0)$
1	$(0,\frac{1}{2},\frac{1}{2},0)$	$(\frac{1}{2},\frac{1}{4},\frac{1}{4},0)$
2	$(0,\frac{1}{2},0,\frac{1}{2})$	$(\frac{1}{3},\frac{1}{3},\frac{1}{6},\frac{1}{6})$
3	$(0,\frac{1}{2},\frac{1}{2},0)$	$(\frac{1}{4},\frac{3}{8},\frac{1}{4},\frac{1}{8})$
$\vdots$	$\vdots$	$\vdots$
∞	- - -	$(0,\frac{1}{2},\frac{1}{4},\frac{1}{4})$

Thus, while the Markov limit does not always exist, the mean state limit does. Also, both concepts closely relate to each other, as the following result shows.

Theorem 3.3. If the Markov limit s^∞ exists, then $s^\infty = \bar{s}^\infty$, i.e. the Markov limit and the mean state limit coincide.

Proof. Let $\epsilon > 0$. Since s^n converges, there is N $\mathbb{N}$ such that for all $n > N$ we have $|s^n - d| < \epsilon$, where d denotes the Markov limit of s^n. For the mean state, we then have for $n > N$:

$$\begin{aligned}
|\bar{s}^n - d| &= |\frac{1}{n+1} \cdot \sum_{i=0}^{n} s^i - \frac{n+1}{n+1} \cdot d| \\
&= |\frac{1}{n+1} \cdot (\sum_{i=0}^{n} s^i - d)| \\
&= |\frac{1}{n+1} \cdot (\sum_{i=0}^{N} s^i - d) + \frac{1}{n+1} \cdot (\sum_{i=N+1}^{n} s^i - d)| \\
&\leq \frac{1}{n+1} \cdot \sum_{i=0}^{N} |s^i - d| + \frac{1}{n+1} \cdot \sum_{i=N+1}^{n} |s^i - d|
\end{aligned}$$

The left term converges to zero for $n \to \infty$, because N is fixed. For the right term,

$$\frac{1}{n+1} \cdot \sum_{i=N+1}^{n} |s^i - d| < \frac{1}{n+1} \cdot \sum_{i=N+1}^{n} \epsilon = \frac{n-(N+1)}{n+1} \cdot \epsilon \xrightarrow{n \to \infty} \epsilon$$

Therefore, $\bar{s}_n$ also converges with the limit d. □

This result substantiates the mean state as a concept suitable for describing the long-term behavior of a workflow. With the tools of probabilistic state space analysis at hand, we now can propose first measures to assess the correctness of workflows. Such a measure, in general, should be well-defined on any initially marked workflow with a probability measure on its state space.

Definition 3.5. (Probabilistic Workflows). A *probabilistic workflow* $\mathsf{W} = (N, M^0, \eta)$ is a workflow (N, M^0) with a state space that is subject to a probability measure η. The set of all probabilistic workflows is denoted as N.

$$\begin{aligned} \mathsf{N} := \{\ (N, M^0, \eta) \mid (N, M^0) &\text{ is a workflow with state space } \mathsf{R}, \\ \eta &\text{ is a probability function on } \mathsf{R}\ \} \end{aligned}$$

We also differ concerning the subset of workflows with a regular probability function.

$$\mathsf{N}^+ := \{\ (N, M^0, \eta) \quad \mathsf{N} \mid \eta \text{ is regular }\}$$

Definition 3.6. (Inconsistency Measures on Probabilistic Workflows). An *inconsistency measure on probabilistic workflows* is a function

$$\mathsf{I} : \mathsf{N} \to \mathsf{R}_{\geq 0}$$

Next, we address the criteria *option to complete, liveness* and *fairness* as introduced in Sec. 2.2 one by one by proposing and discussing measures for each criterion.

3.2. Assessing the Option to Complete

Recall the *option to complete* from Def. 2.6. This property states that whatever course a process takes, there is always the possibility of a firing sequence which leads to the final marking. There might be different scenarios where a workflow hurts this criterion. For example, the workflow might run into a *dead end* without the possibility to reach the final place at all. Both examples discussed so far feature this scenario: In Fig. 5, the marking M_2 constitutes such a dead end and in Fig. 7, the subgraph (M_1, M_2) does. Another potential scenario is that the final place o is reached in some parallel execution branch but unreached in some other branch. In any case, there is some discrepancy between the designated final marking and the actual marking in the long run of the process.

Definition 3.7. (L-distance). Let (P, T, A, W, M^0) be a workflow with final place $o \quad P$ and $M : P \to \mathsf{N}$ be a marking. The *L-distance* of M is defined as

$$L := \sum_{p \quad P \setminus \{o\}} M(p) + |M(o) - 1|$$

One might adapt this definition depending on the application scenario. For example, if multiple tokens should be allowed in the final place, one could either replace the term $|M(o) - 1|$ with some term indicating the general existence of tokens in o, or comply with the original concept of a single final marking by adding a suitable technical transition t with $W(o,t) = 2$ and $W(t,o) = 1$. In the following, we respect the original definition of soundness and define a first inconsistency measure for the *option to complete*, referred to as *dead end inconsistency* for a smoother denotation. Also, for the moment, we consider only *finite* state spaces, i.e. the number of markings is bounded. This discussion will be extended to the infinite case in Sec. 3.2.1.

Definition 3.8. (Token-sensitive Dead End Inconsistency, finite case). Let $W = (N, M^0, \eta) \quad N$ where (N, M^0) has state space $R = (M, E)$, $|M| = k + 1 < \infty$. The probability function η induces a Markov sequence on R with mean state limit $\bar{s}^\infty$. Then, the *token-sensitive dead end inconsistency* on W is defined as:

$$I_D(W) := \sum_{j=0}^{k} \bar{s}_j^\infty \cdot L_j$$

A token-*insensitive* measure might disregard the token discrepancies from the final state L_j and simply add up the probabilities of not reaching the final marking.

Definition 3.9. (Token-insensitive Dead End Inconsistency). Let $W = (N, M^0, \eta) = (P, T, A, W, M^0, \eta) \quad N$ where (N, M^0) has state space $R = (M, E)$, $|M| = k+1 < \infty$ or $|M| = k = \infty$. The probability function η induces a Markov sequence with mean state limit $\bar{s}^\infty$. Then, the *token-insensitive dead end inconsistency* on the marked net w.r.t η is defined as:

$$I_{TID}(W) := 1 - \bar{s}_1^\infty$$

During the course of working on this thesis, the token-sensitive case turned out to require a very careful and extensive discussion, while the insensitive case did not exhibit such a degree of complexity. Therefore, we will stick with the token-sensitive case in the following and neglect the token-insensitive case to a far extend. We argue that there may be sufficiently many application scenarios in which an analysis of the token behavior will be necessary, for example, if a growth of tokens corresponds to a growth in costs for a business process.

Example. In the state space depicted in Fig. 5, for the non-final markings $M^0, M^2..., M^8$, there is one or two tokens in some of the non-final places and no token in the final place. The discrepancies L_j corresponding to the M_j therefore amount as follows.

$$L_j = \begin{cases} 2, & \text{if } j = 0 \text{ or } 2 \leq j \leq 3 \\ 3, & \text{if } 4 \leq j \leq 8 \\ 0, & \text{if } j = 1 \end{cases}$$

Recall that $\bar{s}_3^\infty = \alpha$ and $\bar{s}_1^\infty = 1 - \alpha$ and $\bar{s}_j^\infty = 0$ for all other j, i.e. the process either settles in the dead end state M^3 or the final state M^1. Then, for the inconsistency values we have

$$\mathcal{I}_D((N_1, M_1^0, \eta_1)) = \alpha \cdot 2 \qquad \mathcal{I}_{TID}((N_1, M_1^0, \eta_1)) = \alpha$$

The computation for the second example in Fig. 7 proceeds in a similar fashion and yields values of

$$\mathcal{I}_D((N_2, M_2^0, \eta_2)) = \frac{1}{4} \cdot 2 + \frac{1}{4} \cdot 2 = 1 \qquad \mathcal{I}_{TID}((N_2, M_2^0, \eta_2)) = \frac{1}{4} + \frac{1}{4} = \frac{1}{2}$$

The inconsistency measures were proposed to be assessments for the criterion *option to complete*. Obviously, the two examples do not satisfy that criterion, because there are markings from where the final marking is unreachable. Thus, the non-zero inconsistency values appear to be meaningful. Later, we will examine in how far the measure generally corresponds to the property *option to complete*, that is, whether it is a reliable indicator for this property to be or not to be satisfied. Firstly, we will extend the discussion of the dead-end criterion to infinite state spaces. After that, we will touch the other two correctness criteria and then turn to a more formal analysis in Sec. 3.5.

3.2.1. Infinite State Spaces

Until now, the discussion of the token-sensitive dead end inconsistency measure was limited to workflows with a finite state space. However, in the general case, state spaces need not be finite even for finite Petri nets. In the following, we illustrate the infinite case with an example and then discuss in how far the measure can be adapted for infinite state spaces.

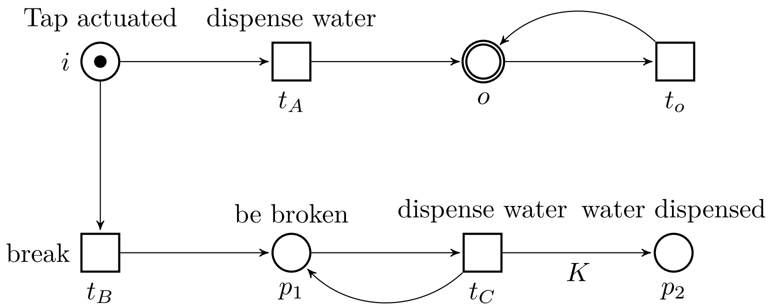


Figure 8: A workflow (N_3, M_3^0) modelling a disfunctional water tap. The weight of each ark is 1, except from (t_C, p_2), where the weight is K.

The workflow shown in Fig. 8 models a water tap which can be actuated to dispense water. After being actuated, either the tap works as expected and *dispenses* a small

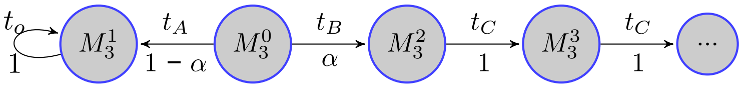


Figure 9: The beginning of the state space R_3 of the net from Fig. 8, parametrized with a general probability function η_3. Together, $\mathsf{W}_3 = (N_3, M_3^0, \eta_3)$

amount of water or runs into a technical error and *breaks*. If no error occurs, the workflow terminates. If the error occurs, the tap keeps on *dispensing* small amounts of water infinitely many times, modelled with the help of the transition t_C which then is always enabled and keeps on producing $K > 0$ tokens that pile up in p_2.
This net has an infinite state space which is sketched in Fig. 9. All probability functions on the exemplary state space differ only concerning the risk to fire t_B from the initial marking, this risk is denoted with α. This constitutes the probabilistic workflow $\mathsf{W}_3 = (N_3, M_3^0, \eta_3)$, where the complete state space $\mathsf{R}_3 = (\mathsf{M}_3, \mathsf{E}_3)$ is described as

$$\begin{aligned}
\mathsf{M}_3 &= \bigcup_{j=0}^{\infty} M_3^j \\
M_3^0 &= (1,0,0,0) \\
M_3^1 &= (0,1,0,0) \\
M_3^i &= (0,0,1,K \cdot (i-1)) \quad \text{for } i \geq 2
\end{aligned}$$

with respect to the place ordering o, i, p_1, p_2, and the firings

$$\begin{aligned}
M_3^0 &\xrightarrow[1-\alpha]{t_A} M_3^1 \\
M_3^0 &\xrightarrow[\alpha]{t_B} M_3^2 \\
M_i &\xrightarrow[1]{t_C} M_3^{i+1} \quad \text{for } i \geq 2 \\
M_3^1 &\xrightarrow[1]{t_o} M_3^1
\end{aligned}$$

describe the edges E_3 and all probability functions η_3^α for $\alpha \quad [0,1]$.
The case that the workflow takes the turn via transition t_B and runs into an error is not desirable. In Sec. 3.2, a token-based measure was introduced which sums up the discrepancy between each marking and the final marking, weighted by the probability to reach these markings. In order to assess the suitability of this approach for the infinite case, let us first consider the behavior of the Markov sequence $s^n \quad [0,1]^\infty$ and its mean state $\bar{s}^n$ in Table 1.
While for all $n \quad \mathsf{N}$, $\bar{s}^n$ is a stochastic vector, $\bar{s}^\infty$ is not a stochastic vector, since the probabilities of the markings sum up to 1 for each n, but the individual probabilities for all $M_3^i, i \geq 0$ vanish. It is therefore not feasible to simply weight the $L-$distances at each marking with their marginal probability. Instead, the $L-$distances and the

time step n	s^n	$\bar{s}^n$
0	$(1, 0, 0, ...)$	$(1, 0, 0, ...)$
1	$(0, 1-\alpha, \alpha, 0, ...)$	$(\frac{1}{2}, \frac{1-\alpha}{2}, \frac{\alpha}{2}, 0, ...)$
2	$(0, 1-\alpha, 0, \alpha, 0, ...)$	$(\frac{1}{3}, \frac{2\cdot(1-\alpha)}{3}, \frac{\alpha}{3}, \frac{\alpha}{3}, 0, ...)$
3	$(0, 1-\alpha, 0, 0, \alpha, 0, ...)$	$(\frac{1}{4}, \frac{3\cdot(1-\alpha)}{4}, \frac{\alpha}{4}, \frac{\alpha}{4}, \frac{\alpha}{4}, 0, ...)$
$\vdots$	$\vdots$	$\vdots$
∞	- - -	$(0, 1-\alpha, 0, ...)$

Table 1: Markov and mean state sequence for the state space in Fig. 9.

marking probabilities should be considered as a whole. An adapted inconsistency measure might look as follows:

$$\lim_{n\to\infty}\sum_{j=0}^{\infty}\bar{s}_j^n\cdot L_j \tag{1}$$

For the running example, we obtain for $\alpha > 0$:

$$\begin{aligned}
&\lim_{n\to\infty}\Big(\frac{1}{n}\cdot 2+\frac{n\cdot(1-\alpha)}{n+1}\cdot 0+\frac{\alpha}{n}\cdot(2\cdot(K+1)+3\cdot(K+1)+...+n\cdot(K+1))\Big)\\
=&\lim_{n\to\infty}\Big(\frac{2}{n}+\frac{\alpha}{n}\cdot(K+1)\cdot(\frac{n\cdot(n+1)}{2}-1)\Big)\\
=&\lim_{n\to\infty}\Big(\frac{2}{n}+(n+1)\cdot\frac{\alpha\cdot(K+1)}{2}-\frac{\alpha\cdot(K+1)}{n})\Big)\\
=&\ \infty
\end{aligned}$$

This value seems appropriate for the application example insofar as the expected amount of water to be dispensed is infinite as long as there is a non-zero probability for the water tap to break. On the other hand, this value is totally insensitive against changes in the breaking risk α. Moreover, the value is insensitive against variations of the ark weight $K = W\big((t_C, p_2)\big)$, corresponding to a weaker or stronger water dispension of the broken tap.

A remedy to obtain more discriminative values, at least for the tap example, is to weight state probabilities quadratically in order to balance out the quadratical term induced by the repeated K-token addition. The computed value changes as follows:

$$\begin{aligned}
&\lim_{n\to\infty}(2\cdot(\frac{1}{n})^2+(\frac{\alpha}{n})^2\cdot(K+1)\cdot(\frac{n\cdot(n+1)}{2}-1)\Big)\\
=&\lim_{n\to\infty}\frac{2}{n^2}+\frac{\alpha^2}{n^2}\cdot(K+1)\cdot n^2\cdot(\frac{1+\frac{1}{n}}{2}-\frac{1}{n^2})\\
=&\frac{\alpha^2\cdot(K+1)}{2}
\end{aligned}$$

In fact, this value incorporates both the breaking risk α and the number of produced tokens K. Now, the question arises whether this measurement approach delivers discriminative, finite inconsistency values for all kinds of nets with infinite state spaces. Also, the question arises how this measure behaves for finite nets. First, let us formally define the adapted inconsistency measure.

Definition 3.10. (Token-Sensitive Dead End Inconsistency, infinite case). Let $\mathsf{W} = (N, M^0, \eta) \quad \mathsf{N}$ with $N = (P, T, A, W, M^0)$ a workflow with state space $\mathsf{R} = (\mathsf{M}, \mathsf{E}), |\mathsf{M}| = k + 1 < \infty$ or $|\mathsf{M}| = \infty$. On R, η induces a Markov walk with mean state sequence $(\bar{s}^n)_{n \quad \mathsf{N}}$. Then, the *token-sensitive dead end inconsistency* on W is defined as:

$$\mathsf{I}_D^{\infty}(\mathsf{W}) = \lim_{n \to \infty} \sum_{j=0}^{k} (\bar{s}_j^n)^2 \cdot L_j$$

We propose that this measure delivers finite values on all kinds of workflows, even with infinite state spaces.

Proposition 3.4. Let $\mathsf{W} \quad \mathsf{N}$. Then, for some $R \quad \mathsf{R}$,

$$\mathsf{I}_D^{\infty}(\mathsf{W}) = R < \infty.$$

Proof (Idea). Consider the sequence of inconsistency terms $(\iota^n)_{n \quad \mathsf{N}}$ up to time step n

$$\iota^n = \sum_{j=0}^{\infty} (\bar{s}_j^n)^2 \cdot L_j$$

and show that this sequence is bounded. For this, choose in each time step n a probability function which maximizes the inconsistency terms up to that time step. Intuitively, there might be such a maximizing function that deterministically chooses a path through the state space (i.e. assigns a probability of 0 or 1 to each edge) and works as a supremum for all other probability functions concerning their inconsistency value. Since the token growth in the net is linearly bounded, we argue that the inconsistency values ι^n do converge.

Example. We investigate the behaviour of the adapted inconsistency measure on finite state spaces. Consider Fig. 10, where the water tap is modelled as before with the slight difference that after breaking, it does not dispense water infinitely many times, but only one time and then remains inert. The corresponding state space is

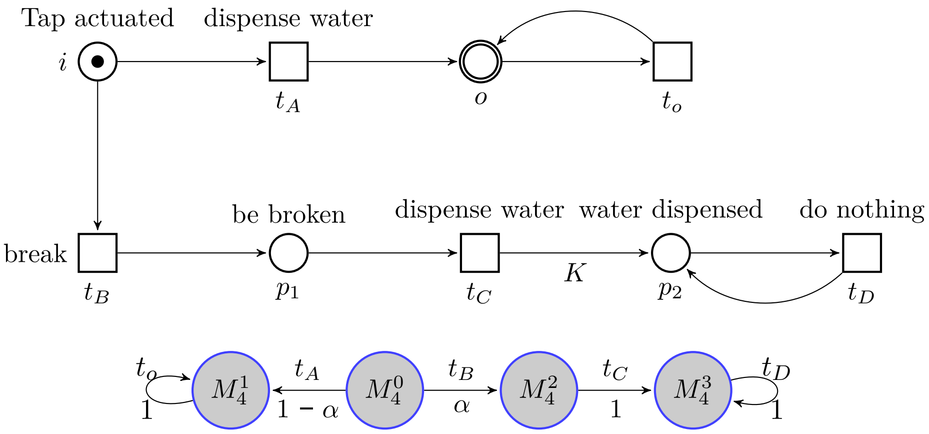


Figure 10: A workflow $\mathsf{W}_4 = (N_4, M_4^0, \eta_4^\alpha)$ with its state space R_4 modelling a different version of the disfunctional water tap.

finite. The markings are defined as follows with respect to the places (i, o, p_1, p_2):

$$\begin{aligned}
\mathsf{M}_4 &= \{M_4^0, M_4^1, M_4^2, M_4^3\}\\
M_4^0 &= (1, 0, 0, 0)\\
M_4^1 &= (0, 1, 0, 0)\\
M_4^2 &= (0, 0, 1, 0)\\
M_4^3 &= (0, 0, 0, K)
\end{aligned}$$

The $L-$distances read as follows:

$$\begin{aligned}
L_0 &= 2\\
L_1 &= 0\\
L_2 &= 2\\
L_3 &= K + 1
\end{aligned}$$

Together with $\bar{s}^\infty = (0, 1 - \alpha, 0, \alpha)$, we obtain

$$\mathsf{I}_D^\infty(\mathsf{W}_4) = \alpha^2 \cdot (K + 1)$$

This is twice the inconsistency value from the workflow in Fig. 8, which appears to be contraintuitive: the first example exhibits an infinite state space which corresponds to an infinite growth of dispensed water and should be reflected in a higher inconsistency value, while for the finite example, the amount of dispensed water remains constant, which should be reflected in a lower inconsistency value. Obviously, this is a shortcoming due to the technical adaption of squaring individual probabilities: In the infinite case, the convergence of the individual marking probabilities to zero is reinforced by this squaring.

All in all, the mismatch between finite and infinite state spaces could not be reconciled. For application purposes, one should consider what each of the measures mean, and what kinds of workflows are to be assessed. I_D^{∞} might serve well in assessing workflows with infinite state space. I_D, again, reflects the expected token discrepancy from the final marking, and can be applied in cases where this discrepancy is of interest. Also, the first proposition for an adapted measure as shown in (1) can be utilized, even though the value might diverge, for example, when comparing workflows with infinite state spaces concerning their speed of divergence under that term.

3.3. Assessing Liveness

Recall the criterion *liveness* from Def. 2.6 and the criterion for *freeness from dead transitions.* These criteria are closely related, both make qualitative statements about the firing of transitions. If a workflow is free from dead transitions, all transitions may fire in some feasible firing sequence. However, the mathematical means introduced so far neglect transitions: The states in the Markov sequence only describe markings and respectively places. However, the concept of the Markov walk easily extends to the effect of also tracking transition firings. Since we do not only seek to make qualitative, but also quantitative statements about transition firings in the course of assessing fairness, we introduce the concept of *control vectors* as counters of the transition firings.

Definition 3.11. (Control matrix). Let $W = (N, M^0, \eta) = (P, T, A, W, M^0, \eta) \quad N$, where (N, M^0) has the state space $R = (M, E)$ and $|M| = k < \infty$ or $|M| = \infty, |T| = m$. The *control matrix* $C \quad [0,1]^{k \times m}$ is defined as

$$C_{ij} := \begin{cases} p, & \text{if } M^i \xrightarrow[p]{t_j} M \text{ for some } M \quad M \\ 0, & \text{otherwise} \end{cases}$$

Thus C_{ij} is the probability of t_j to fire at marking M_i. By means of this definition, the Markov walk extends by a sequence of *probabilistic* control vectors stating the *expected* times of each transition to fire during the Markov walk.

Definition 3.12. (Control vector sequence). The control vector sequence $(c^n)_{n \quad N}$ of the Markov sequence from Def. 3.3 is defined as follows.

$$\begin{aligned} c^0 &= (0, ..., 0) \quad R^{1 \times m} \\ c^{n+1} &= c^n + s^n \cdot C \end{aligned}$$

Furthermore, we normalize each control vector by the sum of its elements and define the limit of the normalized control vector accordingly.

$$\bar{c}^n := \frac{c^n}{c^n \quad {}_1} \text{ for n } \geq 1$$
$$\bar{c}^\infty := \lim_{n \to \infty} \bar{c}^n$$

Lemma 3.5. For the normalized control vector, we have $\bar{c}^n = \frac{c^n}{n}$ for $n \geq 1$.

Proof. Each row in the control matrix sums up to 1, because of our assumption that the sum of the probabilities of outgoing edges at each markings sums up to 1. The control vector delta $c^{n+1} - c^n = s^n \cdot C$ in every step sums up to 1 because s^n are stochastic. Therefore, the norm of the control vectors increases by 1 in each step. □

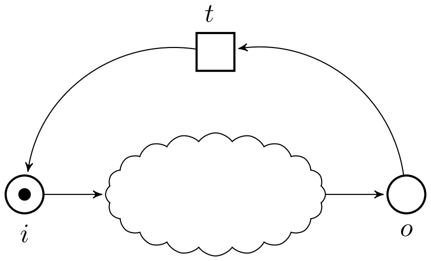


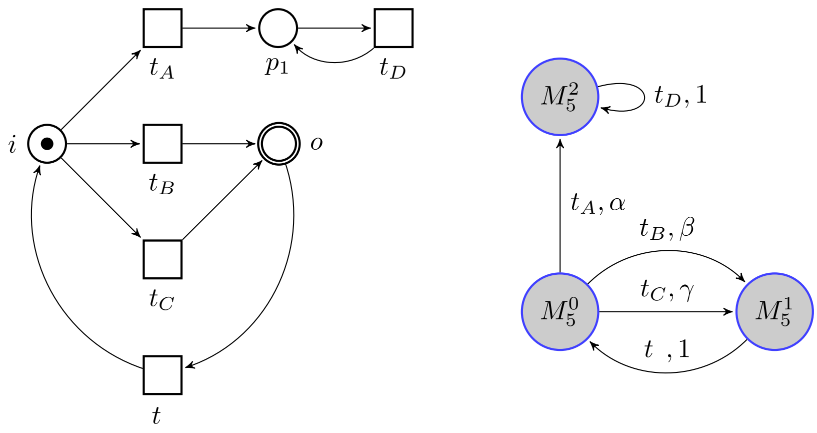
Figure 11: Short-circuit of a workflow

In order to obtain proper results for the control sequence, the definition of a workflow needs to be adapted slightly, by wiring the final place to the initial place as drawn in Fig. 11 to obtain the *short-circuited net.* The reason to extend the net in this way is that in the original net, the final marking merely enables the final transition t_o, which would then fire infinitely often. The result would be a less meaningful normalized control vector $\bar{c}^\infty$.

Definition 3.13. (Short-circuit). Let $\mathsf{W} = (N, M^0, \eta) = (P, T, A, W, M^0, \eta)$ N. The *short-circuit* of W (van der Aalst et al., 2011)) is a tuple $(N\ , M^0, \eta\ ) = (P, T\ , A\ , W\ , M^0, \eta\ )$ with

- $T\ = (T \quad \{t\ \}) \setminus \{t_o\}$
- $A\ = (A \quad \{(o, t\ ), (t\ , i)\}) \setminus \{(o, t_o), (t_o, o)\}$
- $W\ ((o, t\ )) = W\ ((t\ , i)) = 1$ and $W\ (e) = W(e)$ for all other $e \quad A\ $.
- $\eta\ ((M_1, M^0)) = 1$ and $\eta\ ((M, M\ )) = \eta\ ((M, M\ ))$ for all $(M, M\ ) \quad \mathsf{E} \setminus \{(M_1, M_1)\}$.

The short-circuit of a workflow can be utilized together with the liveness property to verify soundness.



(a) The short-circuit (left). Its state space, subject to a probability function where $\alpha + \beta + \gamma = 1$ (right).

$$\mathsf{T} = \begin{pmatrix} 0 & \beta + \gamma & \alpha \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad C = \begin{pmatrix} \alpha & \beta & \gamma & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

(b) The transition matrix (left). The control matrix w.r.t the transition ordering $t_A, t_B, t_C, t_D, t\ \ $ (right).

Figure 12: The short-circuit $(N_5\ , M_5^0, \eta_5)$ of some exemplary workflow (N_5, M_5^0, η_5) with its state space, transition matrix and control matrix.

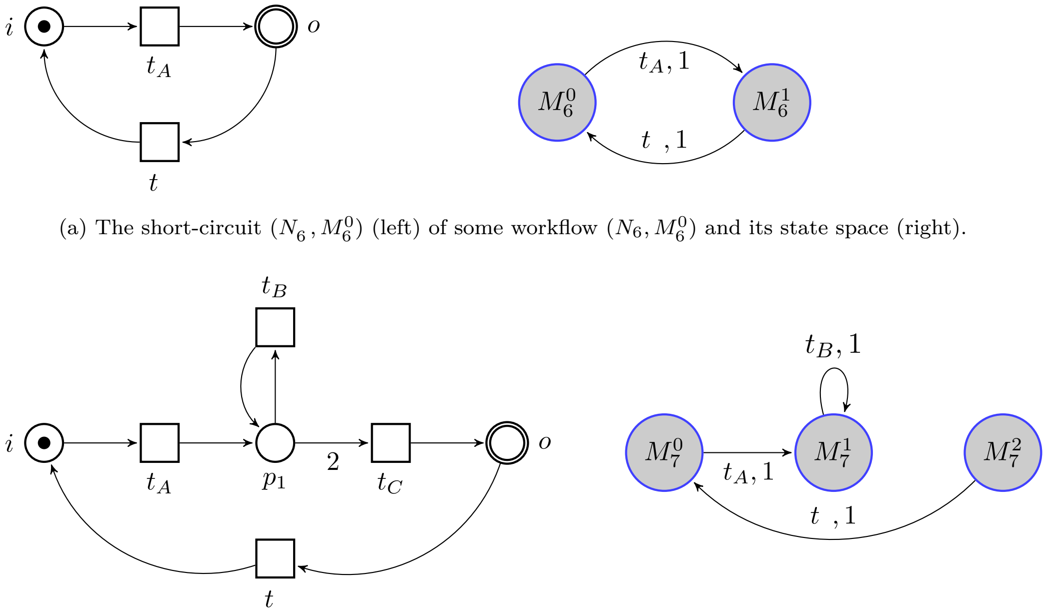
Theorem 3.6. A workflow (N, M^0) is sound if and only if $(N\ \ , M^0)$ is live and bounded.

Proof. See van der Aalst (1997). □

Example. To illustrate the newly introduced concepts we consider a new exemplary workflow in Fig. 12a. The net is comparable to the previously discussed examples in that it contains a dead end, in M_1. However, it differs from the other examples in that it reaches the final marking by firing either of the two transitions t_B, t_C. We consider the resulting control vectors and their normalizations with the help of Fig. 12b.

time step n	s^n	c^n	$\bar{c}^n$
0	$(1, 0, 0)$	$(0, 0, 0, 0, 0)$	$(0, 0, 0, 0, 0)$
1	$(0, \alpha, \beta + \gamma)$	$(\alpha, \beta, \gamma, 0, 0)$	$(\alpha, \beta, \gamma, 0, 0)$
2	$(\beta + \gamma, \alpha, 0)$	$(\alpha, \beta, \gamma, \alpha, \beta + \gamma)$	$1/2 \cdot (\alpha, \beta, \gamma, \alpha, \beta + \gamma)$
3	$(0, \alpha \cdot (1 + \beta + \gamma), (\beta + \gamma)^2)$	$c_2 + s_2 \cdot C$	$1/3 \cdot c_3$
$\vdots$	$\vdots$	$\vdots$	$\vdots$

The limit $\bar{c}^\infty$ depends on the choice of the risk α to turn into the dead end at M^1. If $\alpha > 0$, there is at least a small chance that the process takes the turn into that dead



(a) The short-circuit (N_{6}, M_6^0) (left) of some workflow (N_6, M_6^0) and its state space (right).

(b) The short-circuit (N_{7}, M_7^0) (left) of some workflow (N_7, M_7^0) and its state space (right). Note that $W\big((p_1, t_C)\big) = 2$.

Figure 13: Two more examples for workflows. These are their short-circuits.

end after re-entering the initial state via t . Therefore, in the long run, t_D will fire infinitely more times than all other transitions and we obtain $\bar{c}^{\infty} = (0, 0, 0, 1, 0)$. If $\alpha = 0$, t_A, t_D never fire and we obtain $\bar{c}^{\infty} = (0, \beta/2, \gamma/2, 0, 1/2)$. However, recall that we demand $\eta\big((M, t, M\ )\big) > 0$ for all possible firings $M \xrightarrow{t} M$, because otherwise, the respective transitions might as well be omitted from the workflow.

The short-circuited net (and its original net where o is linked to itself by the usual transition t_o) is free from dead transitions, but not live, and therefore unsound by Th. 3.6. For concrete inconsistency measures, one can differ between these two criteria. An approach to quantify *unfreeness from dead transitions* could be to determine the ratio of dead transitions in the net. An approach to quantify *unliveness* could be to measure the firing ratio between transitions, such that transitions that are relatively unlive become apparent, that is, transitions that fire infinitely less than others. Then, the inconsistency value reads as the ratio of transitions that are relatively unlive.

Definition 3.14. (Dead Transition Inconsistency and Unliveness Inconsistency). Let $W = (N, M^0, \eta) \quad N$ with short-circuit $N \ = (P, T \ , A \ , W \ , M^0, \eta \ )$. Let R be the state space of the short-circuit endowed with η . For the markov sequence on R , let $(c^n)_{n \quad \mathsf{N}}$ be the corresponding control vector sequence with normalized limit $\bar{c}^{\infty}$.

Then, the *dead transition inconsistency* on W is defined as follows:

$$I_{DT}(W) = \lim_{n \to \infty} \frac{|\{ j \quad \{1,...,m\} \mid c_j^n = 0 \}|}{|T \ |}$$

The *unliveness inconsistency* on W is defined as follows:

$$I_L(W) = \frac{|\{ j \quad \{1,...,m\} \mid \bar{c}_j^\infty = 0 \}|}{|T \ |}$$

Note that while both measures are defined via the Markov sequence on the short-circuit net, for the dead end inconsistency measure, the primal net would work as well, since the transitions that never fire in absolute terms is the same and also $|T| = |T \ |$.
Example. Consider again the examples W_1, W_2 from the previous sections. Consider also the examples from Fig. 13. We obtain the inconsistency values as depicted in Tab. 13. The first three workflows W_1, W_2, W_5 behave in a similar fashion; there is a small ratio of transitions that participate in a dead end and a big ratio of transitions not participating in a dead end. However, all transitions can fire in some firing sequence. The fourth workflow W_6 is sound. The fifth workflow W_7 is comparable to the first three workflows, however, it features two also transitions t_C, t which never can fire.

Workflow W	$I_{DT}(N, M^0, \eta)$	$I_L(N, M^0, \eta)$
W_1 with $\alpha, \beta > 0$ (cf. Fig. 5)	0	8/9
W_2 (cf. Fig. 7)	0	3/5
W_5 with $\alpha > 0$ (cf. Fig. 12a)	0	4/5
W_6 (cf. Fig. 13a)	0	0
W_7 (cf. Fig. 13b)	2/4	3/4

Table 2: Dead Transition and Unliveness Inconsistency Values for some workflows.

3.4. Assessing Fairness

Lastly, we consider the criterion of fairness (Murata, 1989). A Petri Net is said to be *fair* if for any (possibly infinite) firing sequence and every pair of transition, the number of times one transition fires while the other transition does not fire is bounded. In other words, in an unfair net, there is a firing sequence and some transitions t_1, t_2 such that $f(t_1)/f(t_2) = 0$, where f is the number of times each transition fires.
While fairness is not a part of the classical soundness notion, it is still a potentially interesting concept for characterising workflows. For example, consider an industrial production process where work is distributed between machines $A_1, ..., A_n$, described

by n tasks *Machine* A_1 *produces, ... , Machine* A_n *produces.* For the sake of profitability and maintainability, there might be an interest in a good load balancing. Thus, there might be an interest in *fairness* of a Petri net describing this business process. Speaking in terms of *measuring inconsistency* of workflows, we would like to assess varying degrees of fairness to workflows depending on the ratio of transition firings in a firing sequence. That is, if $f(t_i)/f(t_j) = 1$ for all transitions t_i, t_j, we demand an inconsistency value of 0, and the higher the maximal firing ratio between two transitions gets, the higher the inconsistency value should be.
As a proposition to assess the fairness of transition firings, we define the following *unfairness inconsistency* over the entropy in the control vector limit, where a high entropy corresponds to a fair distribution of transition firings. We assume $|T| \geq 2$ such that the concept of fairness between multiple transitions may be meaningful.

Definition 3.15. (Unfairness Inconsistency Measures). Let $\mathsf{W} = (N, M^0, \eta)$ N with short-circuit $N = (P, T, A, W, M^0, \eta), |T| = m \geq 2$. Let R be the state space of the short-circuit endowed with η. For the markov sequence on R, let $(c^n)_{n\ \mathsf{N}}$ be the corresponding control vector sequence with normalized limit $\bar{c}^\infty$. Then, the *unfairness inconsistency* on W is defined as:

$$\mathsf{I}_F(\mathsf{W}) = \sum_{j=1}^{m} (\bar{c}_j^\infty - \frac{1}{|T|})^2$$

The *entropy-based unfairness inconsistency* on W is defined as:

$$\mathsf{I}_{EF}(\mathsf{W}) = 1 + \sum_{j=1}^{m} \bar{c}_j^\infty \cdot \log_m \bar{c}_j^\infty$$

In the first measure, the squaring is conducted to punish higher deviances from the firing mean $1/|T|$, or, in other words, to flatten slighter deviances. The measure takes its minimum value of 0 if and only if all transitions are in that firing mean $1/|T|$. The entropy-based measure also takes the minimal value of 0 under that uniform firing distribution (see e.g. Cover and Thomas (2001), Th. 2.6.4). When setting $0 \cdot \log_m 0 := 0$, it is easy to see that the entropy-based measure takes its maximum value of 1 if and only if one of two cases occur: Either there is exactly one transition t_i with $\bar{c}_i^\infty = 1$ and $\bar{c}_j^\infty = 0$ for all other transitions t_j, i.e. one transition fires infinitely more often than all others. Then, with $1 \cdot \log_m 1 = 0$, we have $\mathsf{I}_{EF}(\mathsf{W}) = 0$. Secondly, there might be no transition enabled at all and then also the maximal value of 1 is reached, but such a workflow violates Def. 2.5.
Example. $\mathsf{W}_1, \mathsf{W}_2, \mathsf{W}_5, \mathsf{W}_6$ and W_7 show control vector limits and inconsistency values as depicted in Tab. 3
W_5 is alluded in two different parametrizations to illustrate the case when in fact the load distributions on the two concurrent transitions t_B, t_C comes into play for the inconsistency value. Comparable to W_5 in terms of firing fairness is W_2, but that net still has higher inconsistency values because only two transitions fire, while for W_5,

Workflow W	|T|	$\bar{c}^{\infty}$	$I_F(N, M^0, \eta)$	$I_{EF}(N, M^0, \eta)$
W_1 with $\alpha, \beta > 0$ (cf. Fig. 5)	9	(0,0,1,0...,0)	≈ 0.889	1
W_2 (cf. Fig. 7)	5	$(0, 0, \frac{1}{2}, \frac{1}{2}, 0)$	0.3	≈ 0.569
W_5 with $\alpha > 0$ (cf. Fig. 12a)	5	(0,0,0,1,0)	0.8	1
W_5 with $\alpha = 0, \beta = \gamma = \frac{1}{2}$	5	$(0, \frac{1}{4}, \frac{1}{4}, 0, \frac{1}{2})$	0.175	≈ 0.354
W_6 (cf. Fig. 13a)	2	$(\frac{1}{2}, \frac{1}{2})$	0	0
W_7 (cf. Fig. 13b)	4	(0,1,0,0)	0.75	1

Table 3: Unfairness and Entropy-based Unfairness Inconsistency Values for some workflows.

three transitions fire. The nets W_1, W_7 both have one transition that constitutes a dead end and fires infinitely many more times than all other transitions. W_6 is a very simple workflow where both transitions fire in turn and equally often.

Comparing the two inconsistency measures, three observations are striking. Firstly, the measures seem to be *order-compatible* (Grant and Hunter, 2011) in the sense that for all workflows W_1, W_2, we have $I_F(W_1) < I_F(W_2)$ if and only if $I_{EF}(W_1) < I_{EF}(W_2)$. Secondly, also $I_F(W) \leq I_{EF}(W)$ for all exemplary workflows, that is, the values of the entropy-based measure are at least as high as values of the non-entropy measure. Of course, the validity of these statements for the general case remains to be shown. These two observations suggest that the two measures are similar in the way they capture the concept of unfairness. However, a third observation suggests a difference: While the entropy-based measure takes its maximal value of 1 for several workflows (W_1, W_5 with $\alpha > 0, W_7$), when exactly one transition fires infinitely more often than all others, the measure not based on entropy is more discriminative. We argue that both evaluations are justified: One might wish to assign the same maximal value for that extreme case of one transition firing extremely often; but one also might wish to assign a higher unfairness value if more other transitions remain relatively unlive.

It is also possible to restrict the fairness analysis to a subset of the workflows. A scenario of concurrent machines where a load should be well-balanced, as described in the introduction to this section, might be described with the help of a parallel construct like t_B, t_C in the workflow N_5. Then, the normalized control vector could be restricted to t_B, t_C and the inconsistency value can be computed based on that restricted transition subset.

3.5. Rationality Postulates

In Sec. 2.3, we learned to know the framework of inconsistency measurement on knowledge basis in general. The question how to quantify inconsistencies and what properties measures should satisfy is a very controversial one (Thimm, 2019). In the course of the previous chapters it has become apparent that even defining the term inconsistency on courses of business processes is difficult. In the end, this depends on what properties are considered to be desirable, for example soundness and fairness. Here, we could draw a simple analogy to the rationality postulate of *consistency* as introduced in Sec. 2.3. Consistency on classical inconsistency measures means that a knowledge base is mapped to the inconsistency value of zero if and only if it is consistent. This postulate can be applied to workflow criteria, by considering the adherence to these criteria to be consistency.

Definition 3.16. (Rationality Postulates for Inconsistency Measures on Workflows). Let $I : N \to R_{\geq 0}$ be an inconsistency measure on workflows. The measure satisfies

- **Consistency with the Option to Complete**, if for all (N, M^0, η) N^+, $I((N, M^0, \eta)) = 0$ iff (N, M^0) holds the *Option to Complete.*
- **Consistency with Dead Transition Freeness**, if for all (N, M^0, η) N^+, $I((N, M^0, \eta)) = 0$ iff $(N\ , M^0)$ holds *Freeness from Dead Transitions.*
- **Consistency with Liveness**, if for all (N, M^0, η) N^+, $I(N, M^0, \eta) = 0$ iff $((N\ , M^0))$ holds *Liveness.*

Note that the second and the third postulates incorporate the short-circuit N of the assessed net N. The consistency postulates are defined via workflows W N^+ with a regular probability function η for the reason that, if a transition is assigned a zero probability under a non-regular function, the Markov sequence ignores that transition and misleading conclusions about correctness criteria could be drawn, because the transition never fires. Such a transition would behave like a dead transition while it is actually not dead.

Now, the question arises whether the measures defined for the different correctness criteria in the previous sections satisfy the respective postulates above. In the following, this question will be answered in detail. After that, we will briefly discuss ideas for further meaningful rationality postulates.

3.5.1. Consistency with the Option to Complete

Firstly, we examine the measure concerned with the option to complete, the token-sensitive dead end inconsistency measure (c.f. Def. 3.8). Here, we stick with the measure for the finite case. In preparation for the verification of the consistency postulate, we need to prove a few auxiliary properties on the Markov sequence.

Lemma 3.7. Let $R = (M, E)$ a state space with probability measure η, transition matrix T and mean state limit $\bar{s}^\infty$. Then $\bar{s}^\infty \cdot T = \bar{s}^\infty$.

Proof.

$$\begin{aligned}
\bar{s}^{\infty} \cdot T &= \lim_{n \to \infty} (\frac{1}{n+1} \cdot \sum_{i=0}^{n} s^i) \cdot T \\
&= \lim_{n \to \infty} (\frac{1}{n+1} \cdot \sum_{i=0}^{n} s^0 \cdot T^i) \cdot T \\
&= \lim_{n \to \infty} (\frac{1}{n+1} \cdot \sum_{i=0}^{n} s^0 \cdot T^{i+1}) \\
&= \lim_{n \to \infty} (\frac{1}{n+1} \cdot (\sum_{i=0}^{n} s^0 \cdot T^i + s^0 \cdot T^{n+1} - s^0)) \\
&= \lim_{n \to \infty} \frac{1}{n+1} \cdot \sum_{i=0}^{n} s^0 \cdot T^i + \lim_{n \to \infty} \frac{1}{n+1} s^0 \cdot T^{n+1} - \lim_{n \to \infty} \frac{1}{n+1} s^0 \\
&= \bar{s}^{\infty} + 0 - 0 \\
&= \bar{s}^{\infty} .
\end{aligned}$$

□

In the following, we assume that the entries of the state vectors are ordered with respect to the ordering of markings, that is, for a state vector s, the entry s_j refers to the marking M^j. Also, recall that the marking M^1 denotes the final marking.

Lemma 3.8. Let $R = (M, E)$ a state space with final marking M^1, regular probability measure η, transition matrix T and mean state limit $\bar{s}^{\infty} = (a_0, ..., a_k)$. Then for all $j = 0, 2, ..., k$: If $M^j \overset{t}{\Rightarrow} M^1$, then $a_j = 0$.

Proof. In the transition matrix T, we have $T_{11} = 1$ by the definition of the final place in workflows, see Def. 2.5 and the definition of probability functions, cf. 3.1. Then, by Lemma 3.7,

$$\bar{s}^{\infty} = \bar{s}^{\infty} \cdot T$$

$$(a_0, ..., a_k) = (a_0, ..., a_k) \cdot \begin{pmatrix} . & . & \cdots & \cdots \\ . & 1 & \cdots & \cdots \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & T_{j1} & \cdots & \cdots \\ \vdots & \vdots & \ddots & \vdots \end{pmatrix}$$

$$\begin{aligned}
= \quad a_1 &= \sum_{i=0}^{k} a_i \cdot T_{i1} \\
&\geq a_1 \cdot 1 + a_j \cdot T_{j1} \\
= \quad a_j &\leq 0
\end{aligned}$$

In the last step, we used that $T_{j1} > 0$ by Def. 3.2. Together with $a_j \geq 0$, this implies $a_j = 0$. □

Lemma 3.9. Let $R = (M, E)$ a state space with regular probability measure η, transition matrix T and mean state limit $\bar{s}^\infty = (a_0, ..., a_k)$. Then for all $i, j = 0, ..., k$: If $M^i \overset{t}{\Rightarrow} M^j$ and $a_j = 0$, then $a_i = 0$.

Proof. Analogously to the proof of Lemma 3.8,

$$\begin{aligned} \bar{s}^\infty &= \bar{s}^\infty \cdot T \\ (a_0, ..., a_k) &= (a_0, ..., a_k) \cdot T \\ = \quad a_j &= \sum_{l=0}^{k} a_l \cdot T_{lj} \\ &\geq a_i \cdot T_{ij} \\ = \quad a_i &\leq 0 \end{aligned}$$

In the last step, we used that $T_{ij} > 0$ and the premise $a_j = 0$. Together with $a_i \geq 0$, this implies $a_i = 0$. □

Theorem 3.10. The token-sensitive dead end inconsistency measure satisfies *consistency for the option to complete.*

Proof. Given some $(N, M^0, \eta) \quad N^+$, we have to show: $I_D((N, M^0, \eta)) = 0$ iff (N, M^0) holds the property *option to complete.*
” = ”. Let $(N, M^0, \eta) \quad N^+$ a probabilistic workflow with final marking M^1 where there is the option to complete for (N, M_0). Then, for all reachable markings $M \quad M \setminus \{M^1\}$, there is a firing sequence to the final marking. Application of Lemma 3.8 for the direct predecessor markings of M^1 and repeated application of Lemma 3.9 for all other non-final markings shows that $\bar{s}_j^\infty = 0$ for all $j = 1$. Therefore, $I_D((N, M^0, \eta)) = 0$.
“ = ”. Infer

$$\begin{aligned} & I_D((N, M^0, \eta)) = 0 \\ = \quad & \sum_{j=0}^{k} \bar{s}_j^\infty \cdot \Big(\sum_{p \quad P \setminus \{o\}} M_j(p) + |M(o) - 1| \Big) = 0 \\ = \quad & \quad j = 0, ..., k : \bar{s}_j^\infty = 0 \text{ or } M^j \text{ is final, i.e. } j = 1 \end{aligned}$$

Therefore, $\bar{s}_j^\infty = 0$ for all non-final indices $j = 0, 2, ..., k$. Next, show that from every marking there is a firing sequence leading to the final state M^1. Let M^i a non-final marking. We assume that there is no such sequence, i.e. the final marking is not reachable. Let

$$\begin{aligned} \mathsf{M}^{i} &:= \{M \quad \mathsf{M} \mid M^{i} \Rightarrow M\} \\ I &:= \{j \quad \{0, 2, ..., k\} \mid M^{i} \Rightarrow M^{j}\} \end{aligned}$$

the nodes of the reachability graph starting from M^i and their index set. Note that M^i is not empty because of the continuability of workflows, cf. Def. 2.5, (5). Note also that M^i is closed under firing, i.e. if $M \quad \mathsf{M}^i$ and $M \rightarrow M$ then also $M \quad \mathsf{M}^i$. For the transition matrix, aka the transition probabilities, this means

$$\sum_{j \quad I} \mathsf{T}_{rj} = 1 \quad r \quad I$$

Since M^i is reachable, there is $N \quad \mathsf{N}$ with $s_i^N > 0$. The idea of the proof now is to show that this non-zero probability remains in the subgraph, which does not contain a final marking, and to derive a contradiction with the premise that $\bar{s}_j^{\infty} = 0$ for all non-final j. For this, define the sequence $(b^n)_{n \quad \mathsf{N}}$ of the total probability to be in M^i at time step n:

$$b^n := \sum_{j \quad I} s_j^n$$

We have $b^N \geq s_i^N > 0$. The sequence is monotonic increasing:

$$\begin{aligned} b^{n+1} &= \sum_{j \quad I} s_j^{n+1} \\ &= \sum_{j \quad I} (s^n \cdot \mathsf{T})_j \\ &= \sum_{j \quad I} \sum_{r=0}^{k} s_r^n \cdot \mathsf{T}_{rj} \\ &= \sum_{r=0}^{k} s_r^n \sum_{j \quad I} \mathsf{T}_{rj} \\ &= \sum_{r \quad I} s_r^n \sum_{j \quad I} \mathsf{T}_{rj} + \sum_{r \quad I} s_r^n \sum_{j \quad I} \mathsf{T}_{rj} \\ &\geq \sum_{r \quad I}^{k} s_r^n \underbrace{\sum_{j \quad I} \mathsf{T}_{rj}}_{1} \\ &= \sum_{r \quad I} s_r^n \\ &= b^n \end{aligned}$$

Now also restrict the mean state $\bar{s}^n$ to the subgraph to obtain the subsequence $(\bar{b}^n)_{n \quad \mathsf{N}}$.

$$\bar{b}^n := \frac{1}{n+1}\sum_{j=0}^{n} b^j$$

For $n \geq N$, we get

$$\begin{aligned}\bar{b}^n &= \frac{1}{n+1}\sum_{j=0}^{N-1} b^j + \frac{1}{n+1}\sum_{j=N}^{n} b^j\\ &\geq \frac{1}{n+1}\sum_{j=0}^{N-1} b^j + \frac{1}{n+1}\sum_{j=N}^{n} b^N\\ &= \underbrace{\frac{1}{n+1}\sum_{j=0}^{N-1} b^j}_{\xrightarrow{n \to \infty} 0} + \frac{1}{n+1} \cdot (n-N) \cdot b^N \xrightarrow{n \to \infty} b^N\end{aligned}$$

Since $\bar{b}^\infty = \sum_{j \quad I} \bar{s}_j^\infty$, the fact $\bar{b}^\infty \geq b^N$ implies that for some $j \quad I$, at least $\bar{s}_j^\infty \geq b^N/|I| > 0$. By contradiction to the premise that $\bar{s}_j^\infty = 0$ for all $j = i$, the proposition follows that the workflow holds the option to complete. □

3.5.2. Consistency with Dead Transition Freeness

Secondly, we examine the measure concerned with dead transition freeness, the dead transition inconsistency measure (c.f. Def. 3.14). Here, also the proposition is raised that the measure satisfies the corresponding rationality postulate, the consistency with dead transition freeness.

Theorem 3.11. The dead transition inconsistency measure satisfies *consistency with dead transition freeness.*

Proof. Given some $(N, M^0, \eta) \quad \mathsf{N}^+$, we have to show: $\mathsf{I}_{DT}((N, M^0, \eta)) = 0$ iff $(N \ , M^0)$ holds the property *dead transition freeness.*
“ = ”. Mind that for all transitions t_j, the control vector entries c_j^n monotonically increase.

$$\begin{aligned}& \mathsf{I}_{DT}((N, M^0, \eta)) = 0\\ = \quad & \lim_{n \to \infty} \frac{|\{ j \quad \{1, ..., m\} \mid c_j^n = 0 \}|}{|T \ |} = 0\\ = \quad & \lim_{n \to \infty} |\{ j \quad \{1, ..., m\} \mid c_j^n = 0 \}| = 0\\ = \quad & N \quad \mathsf{N} : c_j^N > 0 \quad j = 1, ..., m\\ = \quad & j = 1, ..., m : \quad N_j \quad \mathsf{N} : c_j^{N_j} > 0 \text{ and } c_j^{N_j - 1} = 0\end{aligned}$$

Now, with $c^{N_j} = c^{N_j-1} + s^{N_j-1} \cdot C$,

$$\begin{aligned} 0 < c_j^{N_j-1} &= \left(s^{N_j-1} \cdot C\right)_j \\ &= \sum_{i=0}^{k} s_i^{N_j-1} \cdot C_{ij} \end{aligned}$$

So there is $i \quad \{0,...,k\}$ with $s_i^{N_j-1} > 0$ and $C_{ij} > 0$. The former means that M^i is reachable (from M^0) and the latter means that t_j is enabled at M^i. Thus, every transition t_j can be activated in some firing sequence from M^0. Thus, the workflow and its short-circuit satisfy freeness from dead transitions.
“ $=$ ”. Assume that N satisfies freeness from dead transitions. Then, for every transition t_j, there is a firing sequence $M^0 \underset{q}{\Rightarrow} M^i \underset{p}{\xrightarrow{t_j}} M^l$ of length n+1 for some $n \quad \mathsf{N}, i, l \quad \{0,...,k\}$. This means $s_i^n \geq q > 0$ and, as t_j is enabled at M_i, $C_{ij} > 0$. Together,

$$\begin{aligned} c_j^{n+1} &= c_j^n + (s^n \cdot C)_j \\ &\geq (s^n \cdot C)_j \\ &= \sum_{r=0}^{k} s_r^n \cdot C_{rj} \\ &\geq s_i^n \cdot C_{ij} \\ &> 0 \end{aligned}$$

This holds for all transitions t_j. With the monotonicity of the control vector entries,

$$\lim_{n \to \infty} |\{\, j \quad \{1,...,m\} \mid c_j^n = 0 \,\}| = 0$$

Then, also

$$\mathsf{I}_{DT}((N, M^0, \eta)) = \lim_{n \to \infty} \frac{|\{\, j \quad \{1,...,m\} \mid c_j^n = 0 \,\}|}{|T\ |} = 0$$

□

3.5.3. Consistency with Liveness

Thirdly, we examine the measure concerned with liveness, the unliveness inconsistency measure (c.f. Def. 3.14). Here, also the proposition is raised that the measure satisfies the corresponding rationality postulate, the consistency with liveness.

Theorem 3.12. The unliveness inconsistency measure satisfies *consistency with liveness.*

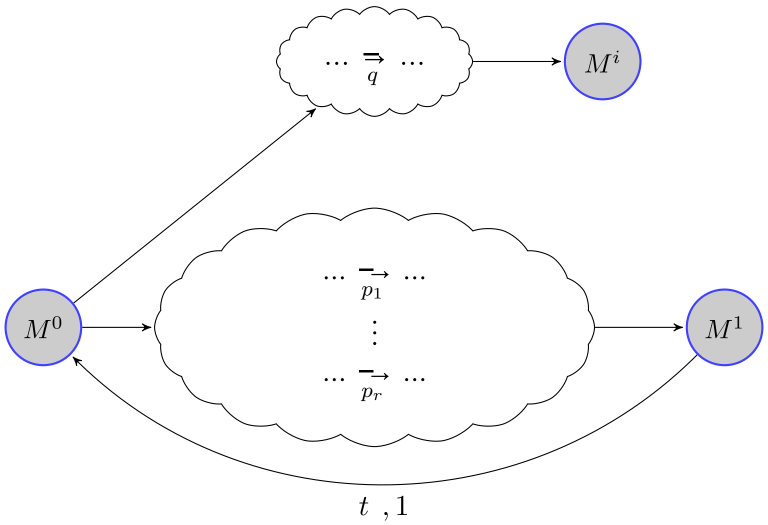

$\dots \overset{}{\underset{q}{\Rightarrow}} \dots$
M^i
$\dots \underset{p_1}{\twoheadrightarrow} \dots$
$\vdots$
$\dots \underset{p_r}{\twoheadrightarrow} \dots$
M^0
M^1
t , 1

Figure 14: When a reachable marking M^i cannot reach the final marking, at least the transition t will eventually "die", i.e. stop firing, because $p_1 + \dots + p_r \leq 1 - q < 1$.

Proof. Given some (N, M^0, η) N^+, we have to show: $\mathcal{I}_L((N, M^0, \eta)) = 0$ iff $(N\ , M^0)$ holds the property *liveness*.
" =   ". By contraposition: Assume that $(N\ , M^0)$ is not live. We have to find a transition t_j with $\bar{c}_j^\infty = 0$, so that $\mathcal{I}_L((N, M^0, \eta)) > 0$.
If $(N\ , M^0)$ is not live, there is a reachable marking M^i and a transition t_j such that t_j can not be fired in any firing sequence starting from M^i. Now, we distinguish two cases.
If the initial marking is reachable from M^i (we are talking about the short-circuit), then t_j also cannot be fired in any firing sequence from the initial marking, then t_j is dead. This means $c_j^n = 0$ for all n N, so $\bar{c}_j^\infty = 0$, so $\mathcal{I}_L((N, M^0, \eta)) > 0$.
If the initial marking is not reachable from M^i, then the final marking is not reachable from M^i, because the initial marking is always reachable from the final marking in the short-circuit. Note that then M^i is neither the initial marking nor the final marking. But since M^i is reachable from the initial marking, for some probability q $(0, 1]$, we have $M^0 \underset{q}{\Rightarrow} M^i$. The idea of the proof now is the following: Since the final marking is not reachable from M^i, the total probability to reach the final marking in the first run through the workflow is smaller or equal to $1 - q < 1$. When restarting the workflow by firing t , the final marking will be reached with an even smaller probability, because again a path to M^i can be taken (cf. Fig. 14). Re-iterating the workflow exhaustively up to the point where the firing of the circuit transition t stops delivers an control vector limit entry equal to zero and thus a non-zero inconsistency value.

Formally, define for all $n \in \mathsf{N}, n \geq 0$

$$\mathsf{P}_n := \{((M^0, ..., M^1), p) \mid M^0 \xrightarrow[p_1]{} ... \xrightarrow[p_r]{} M^1, \, p = \prod_{j=1}^{r} p_j \text{ and } t \text{ fires exactly } n \text{ times } \}$$

the set of all firing sequences from the initial to the final marking on which the short-circuit is iterated n times via transition t , together with their probabilities p. Now, recall the marking M^i which is reachable with a probability q and from which the final marking is not reachable. By this, we have

$$\sum_{(M,p) \in \mathsf{P}_0} p \leq 1 - q$$

After every re-iteration of the workflow, the path to M^i demands at least this probability q. Therefore, for all $n \geq 0$,

$$\sum_{(M,p) \in \mathsf{P}_n} p \leq (1 - q)^{n+1}$$

Consider also that the union over all P_n delivers all possible paths to the final marking.

$$\bigcup_{n \in \mathsf{N}} \mathsf{P}_n = \{((M^0, ...M^1), p) \mid M^0 \xrightarrow[p_1]{} ... \xrightarrow[p_r]{} M^1, p = \prod_{j=1}^{r} p_j\}$$

By Th. 3.1, the probabilities s_1^n to be at the final marking M^1 after n steps is found by summing up the probabilities p of all paths of length n to that marking. Now, we do not know exactly how long the paths in the P_n are, but we know that all those paths together yield all paths of all lengths to the final marking.

$$\sum_{n \in \mathsf{N}} s_1^n = \sum_{n \in \mathsf{N}} \sum_{(M,p) \in \mathsf{P}_n} p$$

We can simply sum up all p since the P_n are disjoint. Then,

$$\begin{aligned} \sum_{n \in \mathsf{N}} s_1^n &= \sum_{n \in \mathsf{N}} \sum_{(M,p) \in \mathsf{P}_n} p \\ &\leq \sum_{n \in \mathsf{N}} (1 - q)^{n+1} \\ &= \frac{1}{q} - 1 \end{aligned}$$

because with $0 < 1 - q < 1$ we have a geometric series. Note that the case q = 1 might also occur, but is trivial because then the final marking can never be reached and the short-circuit transition t is dead.

Next, we consider the control vector entry for t . For indexing, we assume $t \quad = t_m$. With

$$c^{n+1} = c^n + s^n \cdot C$$

we get

$$\begin{aligned} c_m^{n+1} &= c_m^n + (s^n \cdot C)_m \\ &= c_m^n + \sum_{j=0}^{k} s_j^n \cdot C_{jm} \\ &= c_m^n + s_1^n \cdot C_{1m} \\ &= c_m^n + s_1^n \end{aligned}$$

Here, note that $C_{1m} = 1$ and $C_{jm} = 0$ for all other j because t_m is only enabled at marking M^1. Together with $c_m^0 = (0, ..., 0)$, this implies

$$c_m^n = \sum_{j=0}^{n-1} s_1^j$$

and thus

$$\lim_{n \to \infty} c_m^n = \sum_{n \quad \mathsf{N}} s_1^n \leq \frac{1}{q} - 1$$

Then, with Lemma 3.5

$$\begin{aligned} \bar{c}_m^\infty &= \lim_{n \to \infty} \bar{c}_m^n \\ &= \lim_{n \to \infty} \frac{1}{n} c_m^n \\ &\leq \lim_{n \to \infty} \frac{1}{n} \cdot (\frac{1}{q} - 1) \\ &= 0 \end{aligned}$$

So with the short-circuit transition being unlive, there is at least one unlive transition in the workflow. In conclusion,

$$\mathit{I}_L((N, M^0, \eta)) > 0.$$

“ $\quad =$ ”. Assume that $(N \;, M^0)$ is live. Given any transition $t_j \quad T$, we have to show that $\bar{c}_j^\infty > 0$. For every $M^i \quad \mathsf{M}$, we directly apply the liveness property and derive some M^i , M^i M and p_i, q_i $(0, 1]$ with

$$M^i \xrightarrow[p_i]{} M^i \xrightarrow[q_i]{t_j} M^i$$

For every $i = 0, ..., k$, let l_i be the length of that firing sequence and $N := \max_{i=0,...,k} l_i$. Each of the firing sequences $M^i \Rightarrow M^i \xrightarrow{t_j} M^i$ is executed completely at least once in N time steps. Because of this, the *firing delta* $c_j^{n+N} - c_j^n$ of transition t_j, that is, the increase of transition firings after N steps, grows constantly

$$c_j^{n+N} - c_j^n \geq \sum_{i=0}^{k} s_i^n \cdot p_i \cdot q_i$$

for each n N. Setting $m := \min_{i=0,...,k} p_i \cdot q_i$,

$$\begin{aligned} c_j^{n+N} - c_j^n &\geq \sum_{i=0}^{k} s_i^n \cdot m \\ &= 1 \cdot m, \end{aligned}$$

because s^n is a stochastic vector. The transition t_j therefore fires constantly at least m expected times after at most N steps. In conclusion,

$$\bar{c}_j^\infty \geq \frac{m}{N} > 0.$$

□

We have now shown the consistency of three inconsistency measures, for measuring dead ends, dead transitions and liveness, with their corresponding workflow correctness criterion. The measure for fairness was omitted. In the scope of this thesis, fairness is interpreted as a highly data-dependent criterion to particularly assess load balancing. It is not claimed that this interpretation is exclusively valid. However, we argue that the dead transition and the unliveness inconsistency measures are means that sufficiently describe other transition (aka task) dependent workflow correctness. We have proven that the classical notion of soundness is well-described by these two measures. Together with Th. 3.6, especially the unliveness inconsistency measure might be a very powerful tool to verify workflow soundness.

3.5.4. Other Rationality Postulates

We just argued that by fulfilling the postulates of consistency, the measures have proven expressive to the extend that at least the hard validity of workflow criteria can be checked. In inconsistency measurement for logics, other postulates are common which may reflect other meaningful requirements, apart from proving hard validity of consistency criteria. We learned to know some of them in Sec. 2.3, namely *norma-*

lization, monotony, free-formula independence and *dominance*. The question arises in how far these postulates can be transfered from classical logics to a procedural language. In the following, we argue that one postulate, normalization, can be transferred easily, but for the other postulates, this is not the case. To this end, we will firstly define normalization and apply it to the recently defined measures. Then, we will discuss the remaining postulates.

Definition 3.17. (Normalization). Let $I : N \rightarrow \mathsf{R}_{\geq 0}$ be an inconsistency measure on workflows. The measure satisfies *normalization*, if $0 \leq I(W) \leq 1$ for all $W \quad N$.

Theorem 3.13. (Normalization in Inconsistency Measures for Workflows). The measures I_{TID} (Def. 3.8), I_{DT}, I_L (Def. 3.14) and I_{EF} (Def. 3.15) satisfy normalization.

Proof. For the first three measures, this becomes clear when looking at the definitions of each measure.

$$I_{TID}(W) := 1 - \bar{s}_1^{\infty}$$

Here, $0 \leq I_{TID}(W) \leq 1$ because $\bar{s}^{\infty}$ is a stochastic vector.

$$I_{DT}(W) = \lim_{n \rightarrow \infty} \frac{|\{ j \quad \{1, ..., m\} \mid c_j^n = 0 \}|}{|T \;|}$$
$$I_L(W) = \frac{|\{ j \quad \{1, ..., m\} \mid \bar{c}_j^{\infty} = 0 \}|}{|T \;|}$$

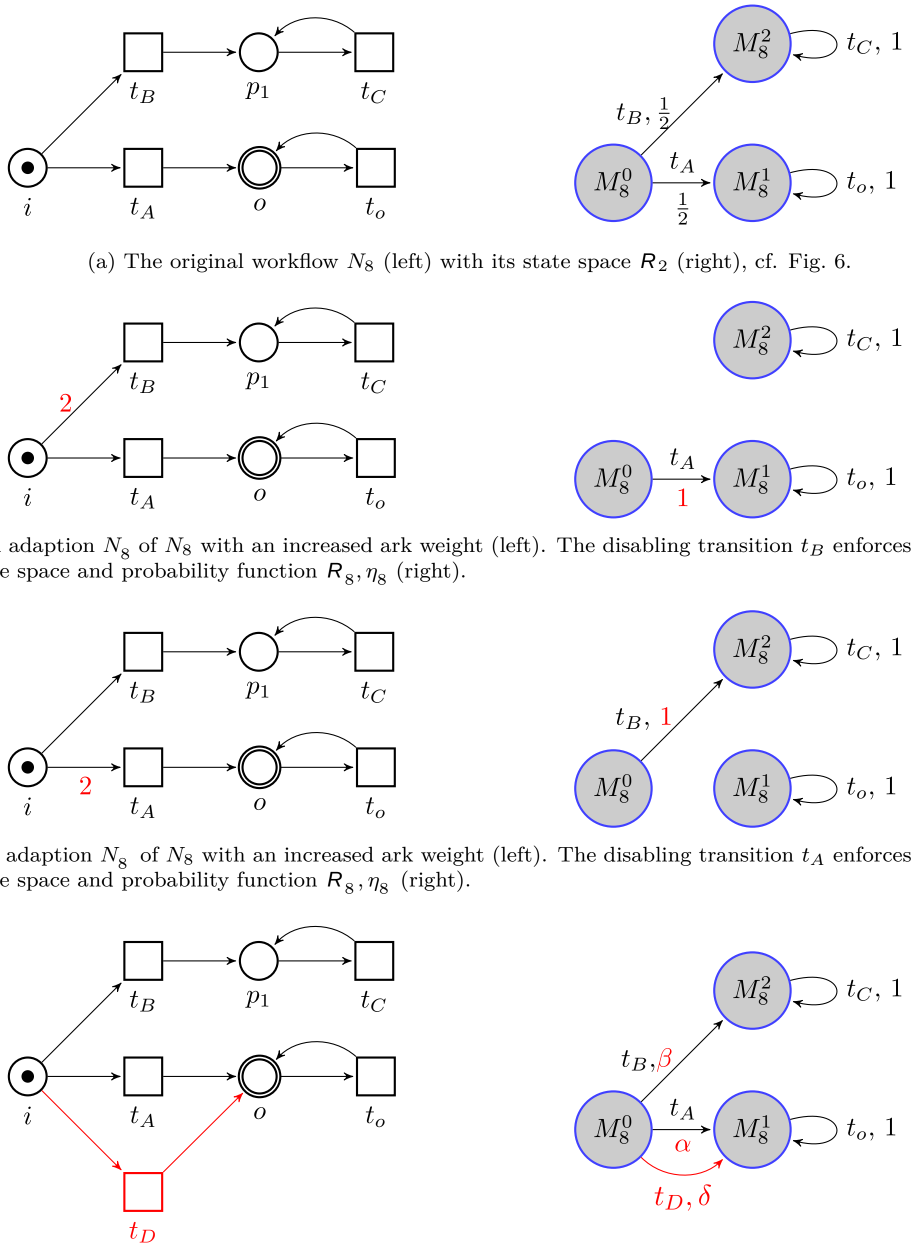
The dead transition freeness and unliveness measures are defined each as some ratio of elements in some set, which naturally yields a value between 0 and 1. For the entropy-based unfairness inconsistency,

$$I_{EF}(W) = 1 + \sum_{j=1}^{m} \bar{c}_j^{\infty} \cdot \log_m \bar{c}_j^{\infty}$$

we again point at literate about entropy properties (Cover and Thomas, 2001). □

These measures can therefore compare workflows of different sizes. However, the token-sensitive dead end measure (Def. 3.8) does not satisfy normalization, as the examples in Sec. 3.2 and Sec. 3.2.1 have shown. Anyway, the normalization postulate does not seem to be meaningful here; in fact, the aim of the token-sensitive measure is to count tokens in absolute terms which does not comply with a relative property such as normalization.

Concerning the other rationality postulates, monotony, free-formula independence and dominance, firstly note that in classical logics, these are properties that make statements about the behaviour of the inconsistency values when the measured knowledge base undergoes changes, that is, if formulae are added to or removed from the knowledge base. To examine whether a transfer of the concepts of those postulates may be possible, we firstly have to examine in what way adding elements to a workflow



(a) The original workflow N_8 (left) with its state space R_2 (right), cf. Fig. 6.

(b) An adaption N_8 of N_8 with an increased ark weight (left). The disabling transition t_B enforces a change in state space and probability function R_8, η_8 (right).

(c) An adaption N_8 of N_8 with an increased ark weight (left). The disabling transition t_A enforces a change in state space and probability function R_8, η_8 (right).

(d) An adaption N_8 of N_8 with a new transition t_D and arks linking t_D to the existing workflow (left). The definition of probability functions requires that this newly enabled transition is granted a positive firing probability, resulting in a change in state space and probability function R_8 , η_8 , with $\alpha + \beta + \delta = 1$ (right).

Figure 15: A probabilistic workflow W_8 with two different adaptions W_8, W_8 where the ark weights are changed, and an adaption W_8 where a transition is added and wired. By adding or removing elements, markings become reachable or loose reachability.

Workflow W	$\bar{s}^\infty$	(L_0, L_1, L_2)	$\bar{c}^\infty$	T_{dead}
W_8 (Fig. 15a)	$(0, \frac{1}{2}, \frac{1}{2})$	$(2, 0, 2)$	$(0, 0, 1, 0)$	
W_8 (Fig. 15b)	$(0, 1, 0)$	$(2, 0, 2)$	$(\frac{1}{2}, 0, 0, \frac{1}{2})$	$\{t_B, t_C\}$
W_8 (Fig. 15c)	$(0, 0, 1)$	$(2, 0, 2)$	$(0, 0, 1, 0)$	$\{t_A, t\ \}$
W_8 (Fig. 15d)	$(0, \alpha + \delta, \beta)$	$(2, 0, 2)$	$(0, 0, 1, 0, 0)$	

Table 4: Characteristics for the workflow variations from Fig. 15 that are necessary for determining inconsistency values.

Workflow W	I_D	I_{DT}	I_L	I_F
W_8	1	0	0.75	0.75
W_8	0	0.5	0.5	0.25
W_8	2	0.5	0.75	0.75
W_8	2β	0	0.8	0.8

Table 5: Inconsistency values for the workflow variations from Fig. 15.

and removing elements from a workflow may affect the inconsistency values. To this end, consider the workflows in Fig. 15. The original workflow in Fig. 15a fires from the initial state each of the transitions t_B, t_A with probability 1/2. This workflow is slightly changed in Fig. 15b by increasing the weight of the ark (i, t_B), in Fig. 15c, the weight of the ark (i, t_A) is increased. The result is that markings become unreachable, because the increased ark weights impede the transitions t_B and t_A, respectively, from firing. Thus, the state space changes as depicted. Then also, necessarily, the probability function can not remain the same as in the original workflow, because a well-defined probability function has the probabilities of all enabled transitions to fire at each marking summing up to 1. Also, in Fig. 15d, the adding of a transition and wiring it via arks to the rest of the net enforces a change in the state space and probability function. In total, the Markov sequence and the related metrical values change as depicted in Tab. 4. Inconsistency values, according to some of our measures, amount as displayed in Tab. 5. Next, let us discuss the induced changing of inconsistency values in respect of the remaining rationality postulates.

Monotony. A measure should be considered monotonous if adding elements (arks, places or transitions) to a workflow either always increases or always decreases the measured values. We argue that manipulating the ark weights, as in the example above, can be considered as an illustration for adding arks. For example, adding a new ark (a, b) between some nodes a, b can be considered as increasing the ark weight $W((a, b))$ from 0 to 1.

The token-sensitive dead end inconsistency I_D may increase or decrease, depending on whether the reachability of the final marking M^1 or the dead end M^2 in Fig. 15a is removed. Hence, this measure behaves not monotonously with respect to adding elements. Also, note that adding a new transition, namely t_D, enforces a change in the state space that brings up ambiguity concerning the probability function, because it is not clear what probability the newly enabled transition should be granted for firing. The dead transition inconsistency I_{DT} increases in the adaptions W_8, W_8 , because in each case one transition turns dead. However, this does not mean that adding elements always increases this kind of inconsistency. Also, adding more transitions and wiring them via arks such that these new transitions are not dead can decrease the dead transition inconsistency, because the ratio of dead transitions may decrease. W_8 introduces such a transition, but the ratio of dead transition already was 0 beforehand. The unliveness and unfairness inconsistency I_L, I_F decrease in W_8, because in (W_8) , two transitions $(t_A, t\ )$ fire infinitely many times, as opposed to one transition (t_C) in W_8. However, adding elements may also increase these kinds of inconsistencies, as the adaption W_8 shows. Here, the new transition t_D is indeed not dead, but unlive as it cannot be brought to firing in any firing sequence starting from the dead end M^2. All in all, neither of the measures exhibits a monotonic nature. Presumably, since adding elements may affect the state space in any kind, inconsistency measurement on probabilistic workspaces in the way we have reasoned about it may be amonotonous per se.

Free Formula Independence. In inconsistency measurement for classical logic, free formula independence on a measure means that in any knowledge base, removing a formula not entangled in inconsistencies should leave the inconsistency value unaltered.

For workflows, we have proposed varying concepts of consistency, namely *option to complete, dead transition freeness, liveness* and *fairness*, leaned on the respective criteria from literature. For the option to complete, we might define *freeness* over markings. A marking $M \quad M$ may be considered to be *free* if $M \Rightarrow M_1$, that is, if it has the option to complete, or if $M \ \Rightarrow M_1$ for all M with $M \Rightarrow M$, that is, if it completes in every case. However, a problem arises here: A marking is not an atomic element in a Petri net, but an element derived from the interplay of the atomic places, transitions and arks. Therefore, markings cannot be added or removed from workflows in an atomic operation the same way that formulae can be added to or removed from knowledge bases in classical logics. The other three consistency concepts however, dead transition freeness, liveness and fairness, rely on atomic elements, namely transitions. Here, a transition may be called *free* if it is not dead, live, or fires in a fair ratio, respectively. However, the example from Fig. 15d has shown that adding a transition which is free in that sense potentially impacts the whole state space. All in all, we refrain from defining a correspondent of the free formula independence postulate for workflow measures.

Dominance. In inconsistency measurement for classical logic, dominance on a measure means that in any knowledge base, adding some formula increases the inconsistency

value at least as much as adding a formula logically implied by the first formula.
For workflows, as procedural models, there is no concept directly corresponding to that of logical implication. We may consider between markings $M, M \quad \mathsf{M}$ an implication relation $M \quad M$ if $M \Rightarrow M$, or even between atomic nodes $a, b \quad P \quad T$, and set $a \quad b$ if there is a directed path of arks in A leading from b to a. The idea here is that the existence of a succeeding node logically implies the existence of the preceding node. Anyway, a is only a successor of b if we regard the workflow as a whole, with all nodes and arks leading from b to a. b may only be predecessor of a after adding b to the workflow, and b is no longer predecessor of a as an atomic node after removing it from the context of the workflow. All in all, we cannot propose a meaningful correspondent of the dominance postulate for workflow measures.
In conclusion, meaningful rationality postulates could be defined especially for consistency, but also for normalization, and verified, if applicable, on our inconsistency measures. Other rationality postulates from classical logics which describe the behavior of inconsistency measures under manipulation of the knowledge base could not be transfered to the measurement on workflows. This was due to the procedural nature of the knowledge bases and thus to a high degree of interdependence between elements in these knowledge bases which we aim to assess. Also, workflows as a subset of Petri nets tailored for the need of modelling business processes impose very specific structural constraints that impede the definition of meaningful rationality postulates for general knowledge base manipulations.

4. Culpability Measures on Workflows

Until now, workflows were assessed in a global manner. Based on various correctness criteria, namely soundness (van der Aalst et al., 2011; Weske, 2012), liveness (van der Aalst, 1997) and fairness (Murata, 1989), measures were proposed (Sec. 3.2- 3.4) which, given a workflow, quantify to what degree that workflow fulfills the respective correctness criteria. In fact, we confidently claim that the measures do capture the correctness criteria since the consistency with these criteria was formally verified (Sec. 3.5).

When applying such measures in business process modelling, a positive inconsistency value might be helpful insofar as it alerts a modeller and by the magnitude of the value gives an estimate concerning the severity of inconsistency. For remodelling, it would be helpful to identify within the workflow elements which cause the inconsistency, such that corrections can be made locally on that elements and possibly a remodelling of the complete workflow can be avoided. The field of *culpability measurement* (Hunter and Konieczny, 2006; Thimm, 2012) is engaged with such a localization of culprits. For applications in business process modelling, efforts were made in line with inconsistency measurement on business rule bases to pinpoint culprits for compliance violations (Corea and Delfmann, 2018; Corea et al., 2019).

For this thesis, we aim to pinpoint elements in workflows that could be considered as culpable for the presence of global inconsistency in the light of the inconsistency measures. The correctness criteria and their measures make statements about workflows, as Petri nets, and more exactly, about the different components of these nets, that is, places, transitions and states. Based on which criterion is to be assessed, we will therefore make the following conceptual distinction in the respective culpability measures.

Definition 4.1. (Culpability Measures on Workflows). Recall the set of all probabilistic workflows, defined as

$$\mathsf{N} := \{\ (N, M^0, \eta) \mid (N, M^0) \text{ is a workflow with state space } \mathsf{R}, \\ \eta \text{ is a probability function on } \mathsf{R}\ \}$$

A *place culpability measure* C^P is for each $\mathsf{W} \quad \mathsf{N}$ with places P a function

$$\mathsf{C}^P_{\mathsf{W}} : P \rightarrow \mathsf{R}_{\geq 0}$$

A *transition culpability measure* C^T is for each $\mathsf{W} \quad \mathsf{N}$ with transitions T a function

$$\mathsf{C}^T_{\mathsf{W}} : T \rightarrow \mathsf{R}_{\geq 0}$$

A *state culpability measure* C^{M} is for each $\mathsf{W} \quad \mathsf{N}$ with markings M a function

$$\mathsf{C}^{\mathsf{M}}_{\mathsf{W}} : \mathsf{M} \rightarrow \mathsf{R}_{\geq 0}$$

The notation C^Y, for Y denoting P, T or M, will be used if we talk about a culpability measure in general, the notation C^Y_{W} *binds* that measure to a particular probabilistic workflow. In the following, we will address the various workflow correctness criteria and discuss how culpability might be determined in case such a criterion is hurt. For this, we revisit the previously used exemplary workflows and refer to the previously defined inconsistency measures.

4.1. Place and State Culpability: Option to Complete

Consider the version of the water tap workflow with *finite* state space W_4, in Fig. 10. This workflow is inconsistent with respect to the token-sensitive dead end inconsistency I_D, because $\mathsf{I}_D(\mathsf{W}_4) = \alpha \cdot (K+1) > 0$ (note that this is not the value under I^∞_D). Arguably, this inconsistency is due to the dead end at M^3, because there is no path leading from M^3 to the final marking. One might argue that also M^2 contributes to this inconsistency because there is also no path from M^2 to the final marking. In the following, we content with a point of view that spots the culpability at M^3, because the means we use point only at this marking: The Markov sequence on this workflow's state space has $\bar{s}^\infty_3 > 0$ but $\bar{s}^\infty_2 = 0$.
A culpability measure in the sense of the token-sensitive dead end inconsistency could assess either places or states. The former might be tailored to a modeler's needs, because places are more fundamental building blocks of workflows. The latter might be tailored to an operator's needs, because states reflect the behaviour of the workflow in production.

Definition 4.2. (Dead End Place Culpability, finite case, and Dead End State Culpability). Let $\mathsf{W} = (N, M^0, \eta) = (P, T, A, W, M^0, \eta) \quad \mathsf{N}$ where (N, M^0) has the state space $\mathsf{R} = (\mathsf{M}, \mathsf{E})$, $|\mathsf{M}| = k+1 < \infty$. Let $\bar{s}^\infty$ be the mean state limit of the Markov walk on R w.r.t η. Then, the *dead end place culpability* CD^P is defined on W for each place $p \quad P \setminus \{o\}$ as:

$$\mathsf{CD}^P_{\mathsf{W}}(p) = \sum_{j=0}^{k} \bar{s}^\infty_j \cdot M^j(p)$$

If $|\mathsf{M}| \leq \infty$, the *dead end state culpability* CD^{M} is defined on W for each state $M^j \quad \mathsf{M} \setminus \{M^1\}$ as:

$$\mathsf{CD}^{\mathsf{M}}_{\mathsf{W}}(M_j) = \bar{s}^\infty_j$$

The place culpability measure describes, for each place, the expected number of tokens at that place in the long run of the workflow. The state culpability measure describes, for each marking, the probability to land at that marking in the long run of the workflow.
Example. For the probabilistic workflow $\mathsf{W}_4 = (N_4, M^0_4, \eta_4)$ depicted in Fig. 10, we have the following relevant terms for the culpability values:

Marking M^j	$\bar{s}_j^{\infty}$	$M^j(i)$	$M^j(p_1)$	$M^j(p_2)$
M^0	0	1	0	0
M^1	1-α	0	0	0
M^2	0	0	1	0
M^3	α	0	0	K

Therefore,

$$\begin{aligned} \mathsf{CD}^P_{\mathsf{W}_7}(i) &= 0 & \qquad \mathsf{CD}^M_{\mathsf{W}_7}(M^0) &= 0 \\ \mathsf{CD}^P_{\mathsf{W}_7}(p_1) &= 0 & \qquad \mathsf{CD}^M_{\mathsf{W}_7}(M^2) &= 0 \\ \mathsf{CD}^P_{\mathsf{W}_7}(p_2) &= \alpha \cdot K & \qquad \mathsf{CD}^M_{\mathsf{W}_7}(M^3) &= \alpha \end{aligned}$$

This means that the expected of amount of tokens found at p_2 after exhaustively executing the workflow amounts to $\alpha \cdot K$, which naturally corresponds to the breaking risk α and the amount of water K lost in the case of an error. The evaluation of the state culpability measure points out the dead end M^3 with the severity α, that is, the mere error risk, insensitive to token loads.
The dead end place culpability measure, for the infinite case, faces problems similar to the token-sensitive dead end inconsistency measure (cf. Sec. 3.2.1): As the number of tokens at a place might be unbounded, the resulting culpability value may be unbounded and thus two places that both bear infinite amounts of tokens may become indistinguishable with respect to culpability. However, a remedy here may be to not count tokens in absolute terms, but rather consider the portion of tokens at each place in relation to the total number of tokens in the net.

Definition 4.3. (Relative Dead End Place Culpability).
Let $\mathsf{W} = (N, M^0, \eta) = (P, T, A, W, M^0, \eta) \quad \mathsf{N}$ where (N, M^0) has the state space $\mathsf{R} = (\mathsf{M}, \mathsf{E})$, $|\mathsf{M}| = k + 1 < \infty$ or $|\mathsf{M}| = k = \infty$. Let $(\bar{s}^n)_{n \quad \mathsf{N}}$ be the mean state sequence of the Markov walk on R w.r.t η. Then, the *relative dead end place culpability* CRD^P is defined on W for each place $p \quad P \setminus \{o\}$ as:

$$\mathsf{CRD}^P_{\mathsf{W}}(p) = \lim_{n \to \infty} \frac{\sum\limits_{j=0}^{k} \bar{s}_j^n \cdot M^j(p)}{\sum\limits_{j=0}^{k} \sum\limits_{p \quad P \setminus \{o\}} \bar{s}_j^n \cdot M^j(p\ )}$$

If the denominator (and thus, for each place also the enumerator) equals to 0, we set $\mathsf{CRD}^P_{\mathsf{W}}(p) = 0$. This corresponds to the case when the net converges to the final state. Note that we exclude the final place from the token counting. One might also include the final place when aiming for some weighting of non-terminating against terminating tokens.

Example. For the infinite case of the water tap example (Fig. 8, Fig. 9), if the process takes the course of breaking (via t_B), there is always a token remaining in p_1, but the tokens accumulating in p_2 outweigh the token in p_1. For the finite case (Fig. 10), place p_2 is the only non-final place where tokens accumulate in the long term. Thus,

$$\begin{array}{lll} \mathit{CRD}^P_{W_3}(i) = 0 & \mathit{CRD}^P_{W_3}(p_1) = 0 & \mathit{CRD}^P_{W_3}(p_2) = 1 \\ \mathit{CRD}^P_{W_4}(i) = 0 & \mathit{CRD}^P_{W_4}(p_1) = 0 & \mathit{CRD}^P_{W_4}(p_2) = 1 \end{array}$$

The places of the finite and infinite case of the tap workflow are equivalent under the relative dead end measure. In conclusion, this measure might come in handy if not the absolute number of tokens is of interest, but the distribution across places in the net.

4.2. Transition Culpability: Liveness and Fairness

From a conceptual point of view, places in Petri nets together with the tokens they carry may serve the purpose of modelling resources, such as water in the tap example, or merely indicate the state of the process. In Sec. 2.1, we have seen this on the basis of an exemplary mapping from a BPMN process to a workflow, and in that course also grasped the purpose of transitions. Namely, tasks or activities in a business process are conducted and may determine the process course based on their result. In terms of a Petri net, transitions effectuate a change of the net's state. From an application point of view, individual tasks and activities may carry the blame for some inconsistent state just like places do. Therefore, in the following, we will aim at defining culpability measures for transitions.

Consider again the first examplary workflow presented in the course of this thesis, in Fig. 3, and its probabilistic state space in Fig. 5. We have briefly touched the workflow in terms of unfairness in Sec. 3.4 and noted that with $\bar{c}^\infty = (0, 0, 1, 0, ..., 0)$, the workflow is highly unfair with respect to the unfairness inconsistency measure, for example $\mathit{I}_{EF} = 1$. Concerning liveness, it is easy to see that under the corresponding inconsistency measures the workflow turns out to be highly unlive, because there is exactly one transition which fires infinitely many times (t_C) while all ten other transitions will fire only finitely many times. However, the workflow is free from dead transitions, because all transitions may fire at least once.

$$\mathit{I}_{DT}((N_1, M_1^0, \eta_1)) = 0 \qquad \mathit{I}_L((N_1, M_1^0, \eta_1)) = \frac{8}{9}$$

These global measures rely on the firing rates of single transitions. So, a positive inconsistency value is connected with individual transitions t_j firing too seldomly, relatively to each other, so that $\bar{c}_j^\infty = 0$. Putting the blame for the inconsistency to these transition seems reasonable, but also contraintuitive or at least arguable, as the example W_1 shows. Here, if the culpability for the inconsistency is put on those with $\bar{c}_j^\infty = 0$, then eight out of nine transitions are culpable. The converse point of view

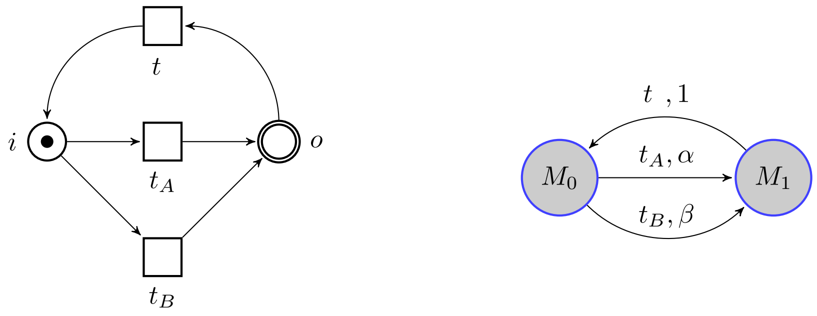


Figure 16: The short-circuit N_8 of some net (N_8, M_0) (left). Its state space, subject to a probability function where $\alpha + \beta = 1$ (right).

may seem more reasonable, that is, not eight out of nine transitions firing too little, but one out of eleven transitions, namely t_C, fires too much. A dilemma now is that also examples are imaginable where a high ratio of transitions fires a lot and a small ratio fires too little.

As a solution, we propose to actually define the unliveness and dead transition culpability via the criterion of zero-valued control vector entries. This conforms to a literal understanding of the concepts of liveness and dead transition freeness: A transition is unlive, if it can not be brought to fire in any sequence from some marking, and this is the case, after all, for eight out of nine transitions in N_1. However, multiple unlive or dead transitions should portion out the culpability.

Definition 4.4. (Dead Transition Culpability and Unliveness Culpability). Let $\mathsf{W} = (N, M^0, \eta) \quad \mathsf{N}$ with short-circuit $N \ \ = (P, T\ \ , A\ \ , W\ \ , M^0, \eta\ \ )$. Let $\mathsf{R}\ \ $ be the state space of the short-circuit endowed with $\eta\ $. Let $(c^n)_{n\ \ \mathsf{N}}$ be the control vector sequence corresponding to the Markov sequence on $\mathsf{R}\ \ $ with normalized limit $\bar{c}^{\infty}$. Define the set of dead transitions T_{dead} and the set of unlive transitions T_{unlive} as follows:

$$
\begin{aligned}
T_{dead} &:= \{t_j \quad T \mid c_j^n = 0 \quad n \quad \mathsf{N}\,\} \\
T_{unlive} &:= \{t_j \quad T \mid \bar{c}_j^{\infty} \ = 0\ \}
\end{aligned}
$$

Then, the *dead transition culpability* CD^T is defined on W for each transition $t \quad T$ as follows:

$$
\mathsf{CD}^T_{\mathsf{W}}(t) = \begin{cases} \frac{1}{|T_{dead}|}, & \text{if } t \quad T_{dead} \\ 0, & \text{else} \end{cases}
$$

The *unliveness culpability* CL^T is defined on W for each transition $t \quad T$ as follows:

$$\mathsf{CL}^T_{\mathsf{W}}(t) = \begin{cases} \frac{1}{|T_{unlive}|}, & \text{if } t \quad T_{unlive} \\ 0, & \text{else} \end{cases}$$

The concerns raised above that in W_1, not the subset of eight out of nine transitions, but one out of nine transitions should be regarded as culpable, may be considered in an appropriate definition of *unfairness* culpability.

Definition 4.5. (Unfairness Culpability). Let $\mathsf{W} = (N, M_0, \eta) \quad \mathsf{N}$ with short-circuit $N \quad = (P, T \ , A \ , W \ , \eta \ )$. Let R be the state space of the short-circuit endowed with the probability function η . Let $(c^n)_{n \quad \mathsf{N}}$ be the control vector sequence on R corresponding to the Markov sequence with normalized limit $\bar{c}^{\infty}$. Then, for each $t_j \quad T$, the unfairness culpability is defined as follows:

$$\mathsf{CF}^T_{\mathsf{W}}(t_j) = (\bar{c}^{\infty}_j - \frac{1}{|T|})^2$$

Example. The transitions of the workflows $\mathsf{W}_1, \mathsf{W}_6, \mathsf{W}_8$ show the culpabiltiy values listed in Tab. 6.

W	T_{dead}	T_{unlive}	Transition t_j	$\bar{c}^{\infty}_j$	$\mathsf{CD}^T_{\mathsf{W}}(t)$	$\mathsf{CL}^T_{\mathsf{W}}(t)$	$\mathsf{CF}^T_{\mathsf{W}}(t)$
W_1		$T \setminus \{t_C\}$	t_C	1	0	0	$\frac{72}{81}$
			$t_j = t_C$	0	0	$\frac{1}{8}$	$\frac{1}{81}$
W_6	$\{t_C, t \ \}$	$\{t_A, t_C, t \ \}$	t_A	0	0	$\frac{1}{3}$	$\frac{1}{16}$
			t_B	1	0	0	$\frac{9}{16}$
			t_C	0	$\frac{1}{2}$	$\frac{1}{3}$	$\frac{1}{16}$
			t	0	$\frac{1}{2}$	$\frac{1}{3}$	$\frac{1}{16}$
W_8			t_A	$\frac{\alpha}{2}$	0	0	$(\frac{\alpha}{2} - \frac{1}{3})^2$
			t_B	$\frac{\beta}{2}$	0	0	$(\frac{\beta}{2} - \frac{1}{3})^2$
			t	$\frac{1}{2}$	0	0	$\frac{1}{36}$

Table 6: Dead Transition-, Unliveness- and Unfairness Culpabilities in various probabilistic workflows. W_1 cf. Fig. 3 and 5, W_6 cf. Fig. 13b, W_8 cf. Fig. 16.

In the first net, W_1, as discussed above, the transition in the dead end t_C is considered to be not culpable for the unliveness, but the other transitions share the culpability for unliveness. However, t_C is regarded as highly culpable for unfairness. The net W_6 is alluded to illustrate the difference between dead and unlive transitions: Here, t_A, t_C, t are unlive, but only t_C, t are also dead. The net W_8 has no dead or unlive transitions. However, it exemplifies a workflow with load balancing between two transitions t_A, t_B. Here, the unfairness culpability measure depends on this load, that is, the choice of α and β.

4.3. Rationality Postulates

Just as for inconsistency measurement, in culpability measurement the question has to be raised what properties a measures should satisfy, such as to make different measures comparable from a more abstract point of view. Just as for inconsistency measurement, rationality postulates for culpability measures were proposed parallel to their counterparts in inconsistency measures for classical logic (Corea and Thimm, 2020; Hunter and Konieczny, 2006). These very postulates also transfer to our graph-based framework only to a small degree. For instance, postulates were proposed for the behaviour of free formulae and logically implied formulae in alignment with the respective free-formula independence and dominance postulates on inconsistency measures (Hunter and Konieczny, 2006). We do not transfer those and refer to the discussion in Sec. 3.5.4. However, we do transfer a very intuitive concept, namely that of *distributivity.* Distributivity directly relates inconsistency measures to culpability measures by stating that a global inconsistency should be regarded as the aggregate of all local inconsistencies. Beside distributivity, we define *normalization* in accordance with the normalization on inconsistency measures (Def. 3.17).

Definition 4.6. (Rationality Postulates for Culpability Measures on Workflows). Let I be an inconsistency measure and C^Y a place, transition or state culpability measure, with the symbol Y denoting P, T or M. We define the following properties:

- **Distributivity** (Hunter and Konieczny, 2006). I is said to *distribute over* C^Y, if for every $W \quad N$, $I(W) = \sum_{y \quad Y} C^Y_W(y)$.
- **Normalization**. C^Y is said to be *normalized*, if for every $W \quad N$, either $\sum_{y \quad Y} C^Y_W = 0$ or $\sum_{y \quad Y} C^Y_W = 1$.

The second postulate is proposed to express that in each workflow, either there is no inconsistency or there is an inconsistency and the culpability measure describes the *relative* culpability of each element for that inconsistency. Of course, each culpability measure can be normalized. However, this may require to firstly calculate all culpability values.

Theorem 4.1. (Validity of Rationality Postulates on Culpability Measures). The following statements hold:

- $\mathcal{I}_{TID}$ (Def. 3.9) distributes over $\mathcal{CD}^{M}$ (Def. 4.2).
- $\mathcal{I}_{F}$ (Def. 3.15) distributes over $\mathcal{CF}^{T}$ (Def. 4.5).
- $\mathcal{CRD}^{P}$ (Def. 4.3) is normalized.
- $\mathcal{CD}^{T}$ (Def. 4.4) is normalized.
- $\mathcal{CL}^{T}$ (Def. 4.4) is normalized.

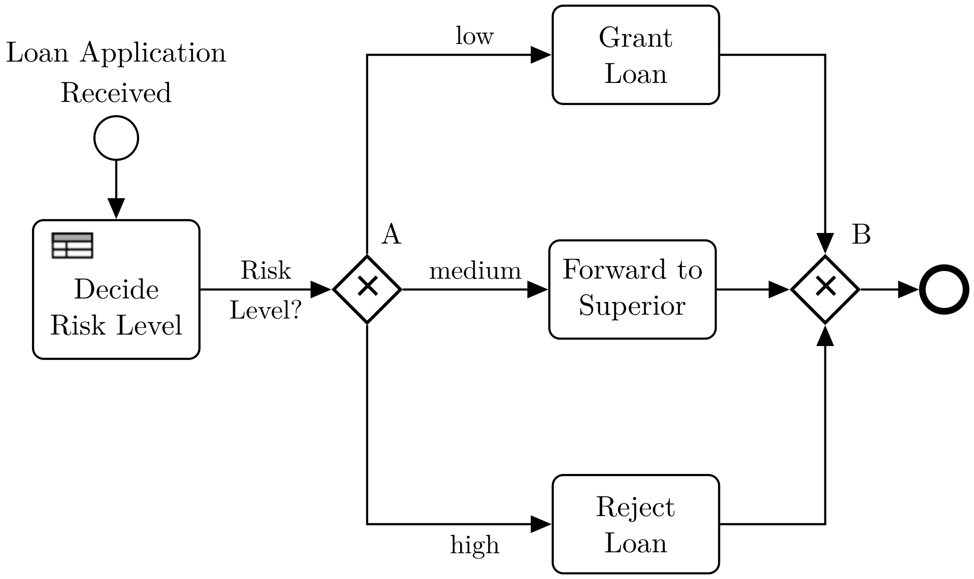
Proof. Clear. □

We have motivated the definitions of the culpability measures in this section by the same ideas that substantiate the inconsistency measures. Therefore, the distributivity postulate and its verification on various measures simply formalizes this natural correspondence.

4.4. Expressivity and Complexity

So far, inconsistency and culpability measures that quantify various correctness criteria on workflows were introduced. We have formally verified the correspondence between the measures and the criteria as well as a correspondence between certain inconsistency and culpability measures. To complete the theoretical discussion, we will examine in how far our tools remain applicable on complex contemporary process models. Process modelling languages such as BPMN are expressive by incorporating, for example, data flow and by deploying process variables for explicit process routing, and manipulating these variables. An extension to BPMN which makes that possible is the *Decision Model and Notation* (DMN). Until now, our examples were of a rather technical nature. To gather more practical insights, we will investigate a sample BPMN process with DMN decision logic by mapping its semantics into our Petri net framework and applying the inconsistency measures on that representation. However, we refrain from defining a complete formal mapping of DMN, since this would be out of scope.

With the help of DMN, so-called *decision tasks* can be defined which manipulate global process variables. This happens on the basis of *decision rules* over the current assignment of these global variables. The actual routing in the control flow is conducted via conditional statements over these variables. Fig. 17a depicts a process model describing the handling of a loan application in some commerical context. This workflow incorporates such a decision logic in the decision task *Decide Risk Level.* This task is not a manual one, but it automatically assigns a value to the variable *Risk Level* by accessing the DMN Table in Fig. 17b and finding that rule (row) which matches the instance assignment of the input variables *Loan size, Criminal Record.* The head of the first column *U* indicates that this table is subject to a unique hit policy, which demands that there is at most one rule matching each possible combination of input variable assignments, that is, rules do not overlap. After deciding the value of *Risk Level*, the further process course can be determined



(a) A DBPMN model. The task *Decide Risk Level* is a decision task with rules as specified below.

Decide Risk Level			
U	Loan size	Criminal Record?	Risk Level
1	<= 1000€	false	low
2	1000 − 5000€	false	medium
3	> 5000€	false	medium

(b) The DMN Table corresponding to the Decision Task.

Figure 17: A DBPMN model, describing the handling of a loan application by some credit grantor (a). Rules for the decision task are specified in DMN (b).

explicitly by reading that variable at the exclusive gateway A and routing to either of the three tasks *Grant Loan, Forward to Superior* and *Reject Loan.*
The examples discussed before featured exclusive gateways with an implicit routing. For example, the gateway B in Fig. 1 does not specify criteria that determine whether the process is continued at node C or D. The respective formal representation in Fig. 3 thus features two transitions being activated at the marking with $M(p_B) = 1$. Non-deterministically, either of them fires subject to some probability α (cf. Fig 4). This probability might depend on some statistical basis, but merely a basis of process traces, since we regarded only the probabilities on the superficial routing directions and disregarded the underlying routing causes. With the help of DMN, the statistics can be detached from this procedural trace data and be bound to instance-specific case data. This makes it also possible to re-use data at multiple points. In the following, we will illustrate that our framework is capable of expressing this semantics to a certain extent.

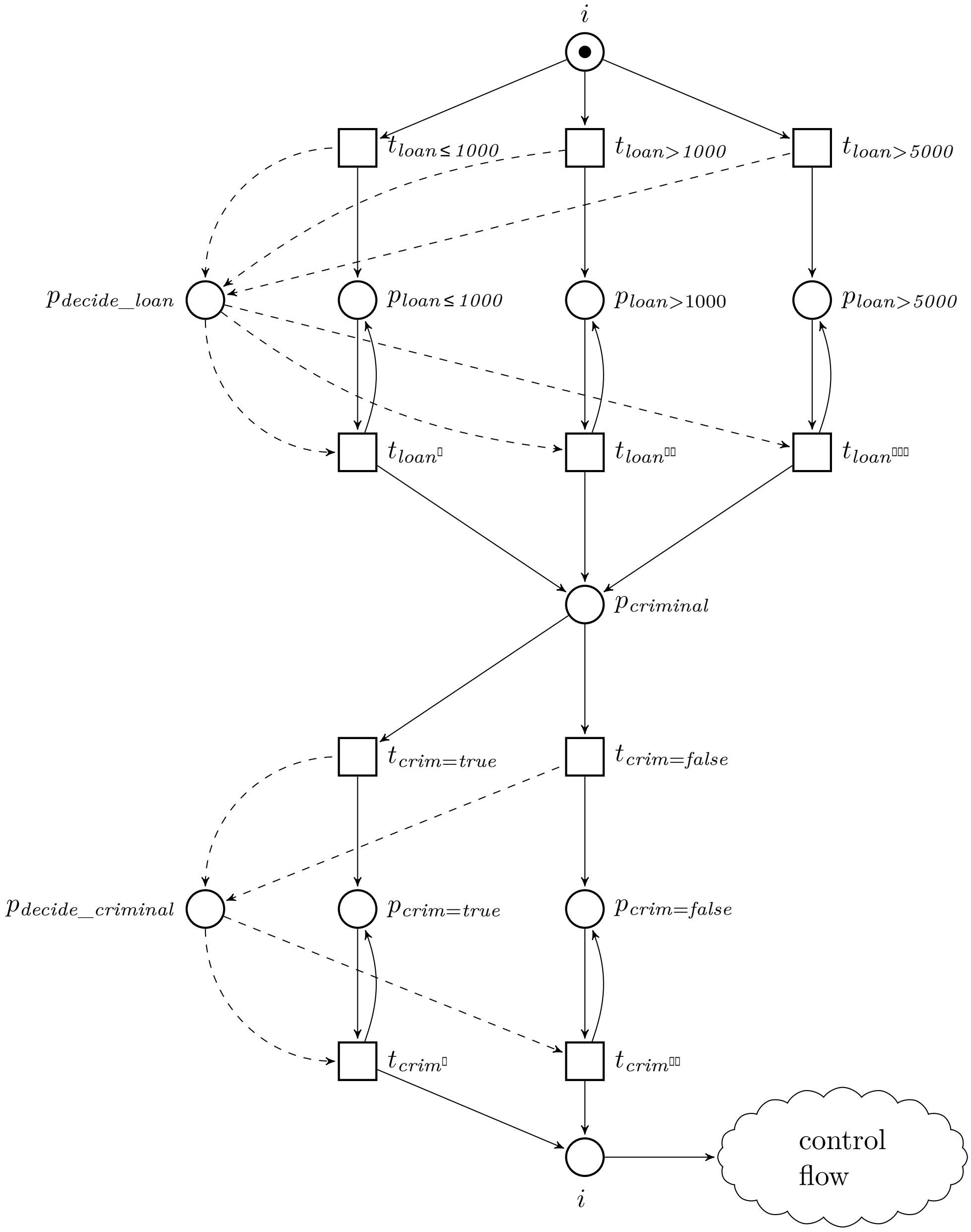
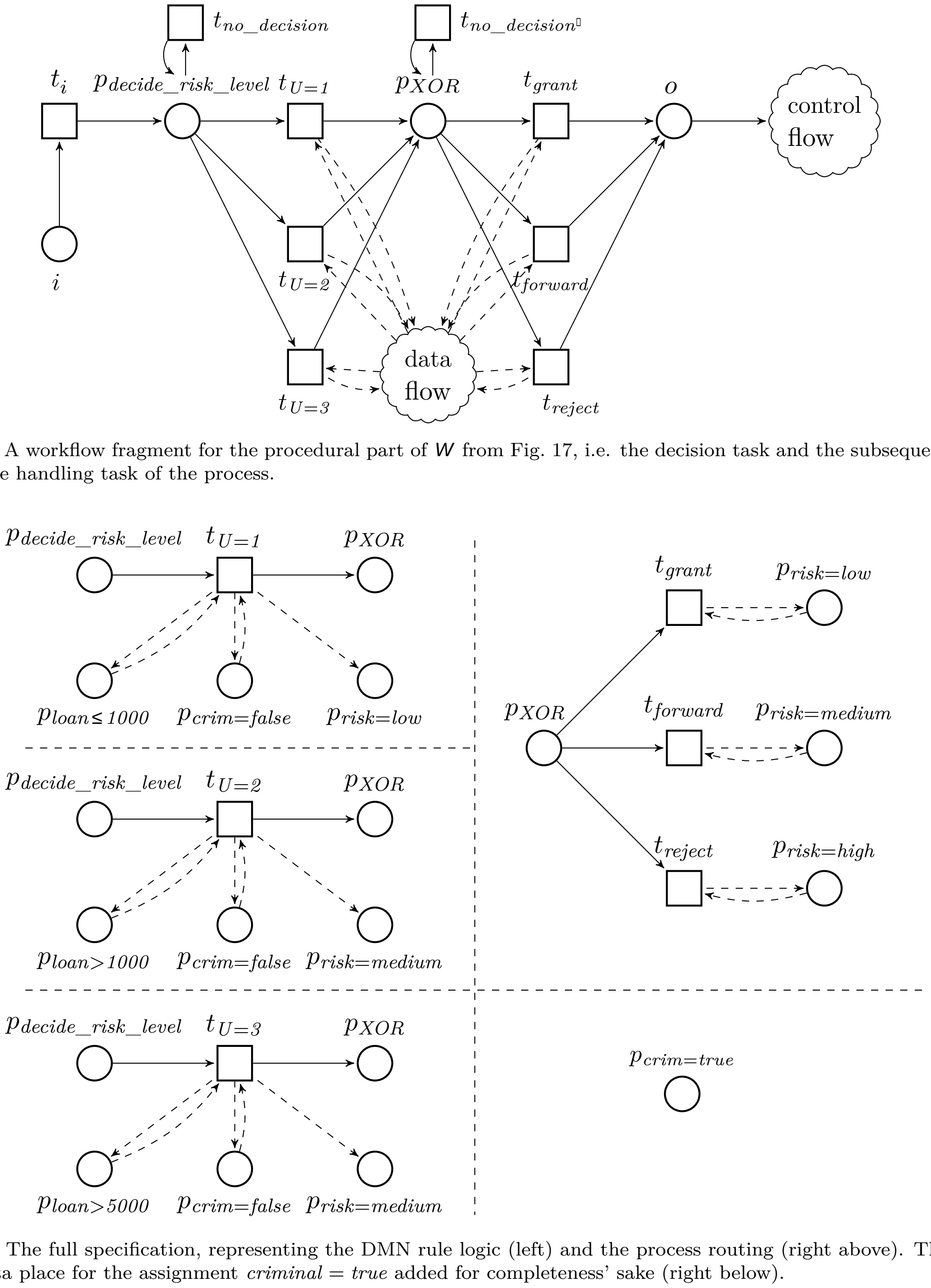


Figure 18: The start of a possible workflow representation of model from Fig. 17. Here, the input variables are set. Then, the actual procedural model (control flow) is iterated as specified in Fig. 19. Ark dashing merely provides visual support.



(a) A workflow fragment for the procedural part of W from Fig. 17, i.e. the decision task and the subsequent case handling task of the process.

(b) The full specification, representing the DMN rule logic (left) and the process routing (right above). The data place for the assignment *criminal* = *true* added for completeness' sake (right below).

Figure 19: A workflow fragment (above) describing the business process from Fig. 17. Ark dashing provides visual support to differ between control- and data flow. Fig. 20 specifies the control flow continuation. Completion of the specification of the fragment (below).

The mapping of the process and its decision logic to a workflow in Petri net terms will proceed in three steps. The first part of the workflow formalizes the setting of the input variables' assignment with the help of designated data places as a preliminary step before the actual control flow. The second part will cover the process of decision making which is based on reading these variables and writing the relevant output variables, which corresponds again to a manipulation of data places. Here, also the actual routing by reading the information about *Risk Level* is conducted. The third part is dedicated to the completion of the process. Here, we will formalize what corresponds to a clearing of the variable assignments. After the workflow has been defined, we will apply the inconsistency and culpability measures on it and discuss the results.

Setting Case Variables. For the preliminary step, consider Fig. 18. As usual, the workflow starts with a token in the designated initial place. After the initial place, two blocks of transitions and places are cascaded where each block corresponds to one of the input variables *Loan Size* and *Criminal Record.* Since for the three possible values of *Loan Size*, exactly one value must be assigned (if not so, we can add a designated place representing *undefined*), exactly one of the corresponding data places $p_{loan \leq 1000}, p_{loan > 1000}, p_{loan > 5000}$ must receive exactly one token. Also, the technical place p_{decide_loan} receives a token. After these two tokens have been placed, the process takes its course to $p_{criminal}$ by firing either of the transitions $t_{loan}, t_{loan}, t_{loan}$, which removes the token from p_{decide_loan} but the token in the data place remains. Without the technical place safeguarding the latter transitions, they would remain enabled and keep on producing tokens in the filled data place. The assignment of the second input variable *Criminal Record* is conducted in the same way as the assignment of the first variable. In the end, there resides exactly one token in i and one token in a data place for *Loan Size* and one token in a data place for *Criminal Record.*

Decision Task and Routing. The naming i ajar to the initial place suggests that the procedural part of the workflow is still waiting in the starting blocks. This procedural part only consists of two major steps: Firstly, the variable *Risk Level* is decided based on the input variables' assignment. Then, depending on the outcome, one of the three tasks *Grant Loan, Forward to Superior* and *Reject Loan* is conducted and the business process terminates. In Fig. 19a, the workflow representation is sketched. After the place $p_{decide\_risk\_level}$ receives a token, at most one of the three transitions $t_{U=1}, t_{U=2}, t_{U=3}$ activates, depending on the assignment of input variables, that is, depending on which of the data places contain tokens. After one of the transitions fires, the control flow moves to the place p_{XOR}, the data places for input variables remain untouched and one of the data places for the ouput variable *Risk Level* contains a token, cf. 19b. However, note that in case there is no rule matching the input variable assignment (for example, if *Criminal Record = true*), the process gets stuck at the decision task. Also, if a rule matches but at the exclusive gateway, no continuation for the output is specified (which does not apply to this example, but might happen), the process gets stuck. For that reason, the technical

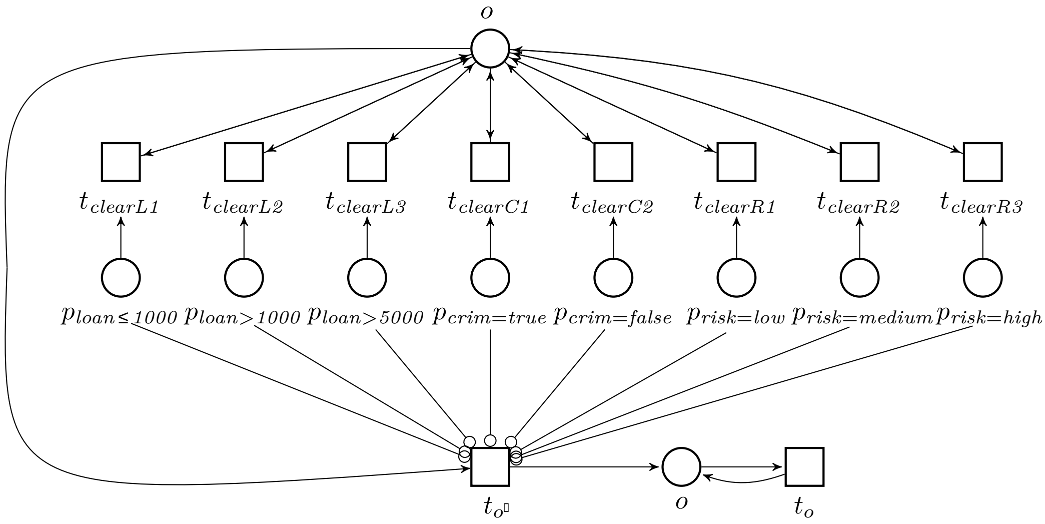


Figure 20: The final part of the workflow representation of the DBPMN model from Fig. 17. The transition t_o is inhibited by the data places, such that the final state cannot be reached before all variable assignments are cleared.

transitions $t_{no\_decision}$,$t_{no\_decision}$ were added. If the business process terminates without issue, a token appears at the pseudo-final place o .

Resetting and Termination. We call o a pseudo-final place because a token in that place only indicates the conclusion of the procedural part of the workflow. However, the data places representing the variables' assignment are still filled with tokens, such that the state of the workflow does not conform to the workflow specification from Def. 2.5 which demands that all places but the final place are empty. We would like to adhere to that definition, such that, for example, if a short-circuit net needs to be constructed by wiring the final place to the initial place, the initial state can be entered with a clear variable assignment. Fig. 20 depicts the desired clearing of the data places via the transitions $t_{clearL1}, ..., t_{clearR3}$. Note that the pre-sets of all these transitions contain the pseudo-final place such that the data places cannot be emptied before the procedural part, which may need to access them, has been completed. The pseudo-final place is also wired to the true final place o by t_o such that the workflow can enter the true final state. However, a problem arises here: Technically, the true final place might be entered any time after t_o is activated by a token entering o . It is desirable that this transition remains deactivated until all data places have been emptied. To overcome the problem, we make the following slight extension to Petri nets.

Definition 4.7. An *Extended Petri Net* is a tuple $\mathring{N} = (P, T, A, \mathring{A}, W)$ where $N = (P, T, A, W)$ is a Petri net and $\mathring{A} \quad (P \times T)$ is called the *inhibitor arks* of the net. In $\mathring{N}$, a transition $t \quad T$ is called *enabled* at a marking M if t is enabled in N at M and for all $(p, t) \quad \mathring{A}, M(p) = 0$.

		Loan Size			
		≤ 1000€	1000-5000€	> 5000€	∑
Criminal	true	4%	1%	5%	10%
Record?	false	36%	19%	35%	90%
	∑	40%	20%	40%	100%

Table 7: An exemplary statistical basis for cases of the DMN-BPMN process in Fig. 17.

The building blocks of our probabilistic framework subsequent to the underlying Petri net definition, for example the concepts of state spaces and Markov sequences, remain untouched by this adaption. The question whether the validity of the postulates from Sec. 3.5 remains true is to be revised. For the running DMN-BPMN-example, Fig. 20 depicts inhibitor arks between the data places and the transition t_o indicated by a small circle at the inhibited transition.

Applying Inconsistency & Culpability Measurement. The workflow representation W of the example being completed, we are ready to investigate the behaviour of the workflow concerning inconsistencies as defined in the previous course of this thesis. To this end, we regard an exemplary statistical basis of the input variables as listed in Tab. 7. This statistics determines the probability function on the reachability graph $\mathsf{R} = (\mathsf{M}, \mathsf{E})$ of the workflow. The graph with specifications of all markings can be found in the appendix in Fig. 22 . Note that some of the transition probabilities in Fig. 22 are determined as conditional probabilities via Tab. 7, for example, $\eta((M^0_{l1}, M_{l1c1})) = P(\mathit{criminal} = \mathit{true} \mid \mathit{loansize} \leq \mathit{1000}€) = 4/40 = 0.1$. Note secondly a significant simplification in the part from Fig. 23 : After a token enters o , the firing order of the clearing transitions $t_{ClearL1}, ..., t_{ClearR3}$ is fixed in our non-regular (w.r.t Def. 3.2) choice of the probability function, while a regular probability function would demand a non-deterministic firing order. However, it is easy to see that this does not affect the final outcome of inconsistency values and culpability values.

At the decision task, we set for any marking M with $M \xrightarrow{t_{no\_decision}} M$:

$$\eta((M, t_{no\_decision}, M)) = \begin{cases} 0 & \text{if} \quad M \quad \mathsf{M}, j \quad \{1, 2, 3\} : M \xrightarrow{t_{U=j}} M \\ 1 & \text{else} \end{cases}$$

i.e. the transition $t_{no\_decision}$ should fire if and only if there is no applicable rule in the DMN table matching the variable assignment. The members of the resulting Markov sequence are expanded in the appendix in Tab. 14, together with information about the control vector sequence on the short-circuit in Tab. 15. Note that since the Markov limit s^∞ exists, we can use $s^\infty = \bar{s}^\infty$ by Th. 3.3 for the determination of the inconsistency values. The behavior of the control vector deltas yields

$$\bar{c}^{\infty}(t_{no\_decision}) = 1$$

i.e. the entry of the control vector limit for the transition $t_{no\_decision}$ is 1 and all other entries are 0. Then,

$$\begin{aligned} T_{dead} &= \{t_{no\_decision}\ , t_{reject}, t_{ClearC1}, t_{ClearR3}\} \\ T_{unlive} &= T \setminus \{t_{no\_decision}\} \end{aligned}$$

Note here the difference between $t_{no\_decision}$ and $t_{no\_decision}$. T_{dead} can be retraced by observing which of the transitions from Fig. 18-20 do not occur in the state space in Fig. 22-23.

$I_D(W)$	$I_{TID}(W)$	$I_D^{\infty}(W)$	$I_{DT}(W)$	$I_L(W)$	$I_F(W)$	$I_{EF}(W)$
0.4	0.1	0.168	$\frac{4}{29}$	$\frac{28}{29}$	$\frac{28}{29}$	1

Table 8: Inconsistency Values for the DMN-BPMN workflow W.

Tab. 8 depicts the values of W under all inconsistency measures defined in the course of this thesis. The token-insensitive dead end value $I_{TID}(W)$ is due to the probability to land at a non-final marking amounting to 0.1. The token-sensitive dead end value $I_D(W)$ is due to the $L-$distances of all these non-final markings amounting to 4, because at these markings there is a token in two data places (for the variables *Loan size* and *Criminal Record*), one at the place $p_{decide\_risk\_level}$ and no token in the final place o. The I_D^{∞} yields a rather odd value that is due to the squaring of mean state limit entries, which illustrates again that the infinite state measure might not meaningfully apply to finite state spaces, as discussed in Sec. 3.2.1. The dead transition measure I_{DT}, the unlivess measure I_L and the unfairness measures I_F, I_{EF} yield values that can easily be retraced with the help of the control vector limit $\bar{c}^{\infty}$ and the set of dead transitions T_{dead}.

Place p	$CD_W^P(p)$	$CRD_W^P(p)$
$p_{loan \leq 1000}$	0.04	$\frac{4}{30}$
$p_{loan > 1000}$	0.01	$\frac{1}{30}$
$p_{loan > 5000}$	0.05	$\frac{5}{30}$
$p_{crim=true}$	0.1	$\frac{10}{30}$
$p_{decide\_risk\_level}$	0.1	$\frac{10}{30}$

Table 9: Place Culpability Values for the DMN-BPMN workflow W.

Marking M	$CD^M_W(M)$
M^2_{l1c1}	0.04
M^2_{l2c1}	0.01
M^2_{l3c1}	0.05

Table 10: State Culpability Values for the DMN-BPMN workflow W.

Transition t	$CD^T_W(t)$	$CL^T_W(t)$	$CF^T_W(t)$
$t_{no\_decision}$	0	0	$(\frac{28}{29})^2$
$t \quad T_{dead}$	$\frac{1}{4}$	$\frac{1}{28}$	$(\frac{1}{29})^2$
$t \quad T_{unlive} \setminus T_{dead}$	0	$\frac{1}{28}$	$(\frac{1}{29})^2$

Table 11: Transition Culpability Values for the DMN-BPMN workflow W.

Tab. 9,10 and 11 gather all culpability values that can be calculated for any places, transitions or states of the workflow W. For the places, as listed in Tab. 9, the token-sensitive measure CD^P yields a value of, for example, 0.04 for $p_{loan\leq 1000}$. This means that the expected amounts of tokens in the net in the long run of executing the Markov sequence is 0.04. In more practical terms, it means that there is a 0.04 probability that a token gets stuck in that place. The values of the relative dead end place measure CRD^P, in this example with a finite state space, simply correspond to a normalization of the values under the CD^P-measure such that all culpabilities sum up to 1. For the states, as listed in Tab. 10, we find the dead end state culpability CD^M_W distributing the overall token-insensitive dead end inconsistency $I_{TID}(W)$. This culpability measure indicates the distinct probabilities of each of the listed states to get stuck at it. For the transitions, the dead transition measure CD^T_W and the unlive measure CL^T_W uniformly distribute a value of 1 over all dead and unlive transitions, respectively. The unfairness measure CF^T_W, due to the control vector limit which assigns a value of 1 to $t_{no\_decision}$ and 0 to all other 28 transitions, punishes this relatively high deviation from the mean 1/29 while the unfairness value for the other transitions are relatively low.

Discussion. The global non-termination risk of $I_{TID} = 0.1$ corresponds directly to the exemplary a priori probability for the variable assignment *Criminal Record = true*. In fact, any case with that variable assignment will get stuck at the decision task because no rule is specified for variable combinations with *Criminal Record = true*. The value 0.4 of the token-sensitive measure is not so meaningful in the application context, because the token load in the net is partially due to tokens in data places, and thus a business process with more different input variables that is mapped in a similar way would be assigned to a higher degree of inconsistency, because more data places are filled with tokens. In a refined definition of the inconsistency measures and the workflows in the underlying framework, we could differ more formally between data places and control flow places. The value of the unliveness measure $\mathcal{I}_L$, 28/29, is also not so meaningful here: This value is due to the transition $t_{no\_decision}$ that fires if no DMN rule is applicable at the decision task and which is the only live transition out of 29. Imagine that at the business process runs into another error, namely that a decision is made at the decision task, but there is no matching routing at the exclusive gateway A. Then, a second transition $t_{no\_decision}$ would become live, resulting in a lower inconsistency value, while the business process is in fact more problematic. The unfairness measures $\mathcal{I}_F$, $\mathcal{I}_{EF}$ also yield values that trace back to the ratio between live and unlive transition. In this case, a more meaningful result would be even more urgently desirable. There is, in fact, a ratio of transition firings based on the exemplary statistics, for example, the transitions for filling the variable places $t_{crim=true}, t_{crim=false}$ fire in a 1/9 ratio, an information which is lost in the control vector limit used for calculating the unfairness inconsistencies. Therefore, the way of determining unfairness inconsistencies should be refined. Lastly, the dead transition measure $\mathcal{I}_{DT}$ can be considered helpful indeed. The ratio of dead transitions, in this case, four out of 29, gives a rough estimate about how large the part of the workflow is that can actually be reached. A more distinct assessment, however, is possible when discussing the culpability values.

The state culpability measures allow a conclusion about problematic variable assignments, because for a state with non-zero dead end culpability, one can directly read out from the definition of the marking which data places are filled in that marking and thus what the problematic assignment is. For example, the marking M^2_{l3c1} holds the highest culpability of 0.05, so we might infer that the missing of a rule for the assignment *Loan size > 5000€, Criminial Record = true* is the most urgent deficiency. The place culpabilities, on the one hand, allow more fine-grained assessments for single variables. For example, the high culpability of the data place $p_{crim=true}$ suggests an issue with that particular variable. On the other hand, the place measures allow to pinpoint problematic elements in the *cFontrol* flow more directly: The place $p_{decide\_risk\_level}$ as the only control flow place with a positive culpability directly points at the decision task as the problematic control flow element. As for the transition culpabilities, especially spotting the dead transitions via $\mathcal{CD}^T_{\mathcal{W}}$ might come in handy. Positive values for $t_{reject}, t_{ClearC1}$ and $t_{ClearR3}$ indicate a business task never being executed, namely *Reject Loan*, as well as variable assignments under which the workflow cannot terminate or which are never taken, namely *Criminal Record = true* and *Risk Level = high.* On the other hand, the values under the unliveness and unfairness culpability measures seem less meaningful, as discussed above in the scope of the inconsistency measurement.

We did not cover the whole of BPMN or DMN in examples or even in a complete formalization, which would allow us to draw general conclusions about the expressivity and complexity of our framework. But by discussing the running examples, especially the DMN-BPMN model, we could develop an understanding that allows for some assumptions. To completely specify the DMN-BPMN model, we have identified the need for inhibitor arks, which make the Petri net formalism Turing-complete (Zaitsev, 2014). Since the probabilistic methodology works even for such nets with inhibitor arks, we are confident that our framework technically could capture DMN and BPMN in their whole expressivity. However, it might be convenient to extend to Colored Petri nets Jensen (1986), alone for the sake of readability in more complex data and decision driven process models. The DMN-BPMN example also illustrated the problem of exponential growth in state spaces. Locally at decision tasks, up to 2^n rules may be required for n different discretely valued input variables. However, note that in the state spaces of our formalism, this problem turns from a local one into a global one, since the variable assignments require consideration in the continuation of markings. In other words, a simple state space may reproduce to a wide extend for every possible variable assignment, as seen in Fig. 22 and 23.

5. Related Work

In its methodology and application domain, this thesis closely relates to the work of Haarmann et al. (2018). The authors perform a state space analysis of BPMN models formalized as Coloured Petri Nets (CPN) in order to verify the conformance of processes against compliance rules. Furthermore, Batoulis et al. (2017) extend the classical notion of soundness for business processes with decision logic formalized in DMN by two decision-aware requirements. Firstly, there should be no decision task at which some execution instance might find no matching decision rule for its variable assignment, and any output that may be produced for some case is covered by a matching process continuation (*state-based decision deadlock freedom*). Secondly, any control flow branch that is specified should be able to be executed by some process instance (*state-based dead branch absence*). The example from Fig. 17 is not state-based decision deadlock free: A case with a variable assignment where *Criminal Record* is true can arrive at the decision task but does not find a matching decision rule. The example also does not satisfy state-based dead branch absence: The execution directive for the output assignment *Risk Level = high* can never be realized, because that value is never assigned at the decision task. Arguably, the violations of the criteria might be detected with the means developed in this thesis. In the example, the problematic variable assignment which causes the violation of the state-based dead branch absence criterion is indicated by the dead end state culpability values under CD^{M}. Also, the state-based dead branch where *Risk Level* is assigned to *high* is indicated by the dead transition culpability values under CD^{T}. However, a more thorough and general discussion is needed to establish and formally verify a link between the decision-aware assessment framework by Batoulis et al. and the quantitative means developed in this thesis.

The authors also extend with their decision-aware criteria several weaker notions of structural soundness. The decision-aware extensions aside, we will list some of these structural notions here and assess whether our probabilistic approach might conform to them. Martens (2003) proposed in the *weak soundness* for workflows to drop the criterion of dead transition freeness. The author argues that his application context of web service construction requires a compositional workflow modelling that yields systems where not necessarily all components are used. One should therefore allow for dead transitions. Since we have built our inconsistency measures on a fine-granular level discerning between soundness subproperties, one might simply focus on the dead end measures and neglect the measures for dead transition freeness to evaluate workflows with respect to weak soundness. Dehnert and Rittgen (2001) proposed *relaxed soundness* where it is no longer required that from every state, there is the option to complete, but every transition should fire at least in one firing sequence with the option to complete. This notion was introduced to accommodate modelers who might not be aware of every single possible process execution in complex models, but at least one path with a sound behavior should be guaranteed for all workflow elements. In our probabilistic framework, we do not find an obvious conformance to this type of soundness notion, because markings and places on the one hand and transitions on the other hand were dealt with separately in Markov sequence and control vector sequence, but the relaxed soundness makes a composite statement about markings and transitions.

In a fashion similar to the analysis of decision aspects by Batoulis et al., de Leoni et al. (2018) incorporate data flow in CPN representations of business process models in order to verify soundness criteria. Furthermore, de Leoni et al. (2021) incorporate the results of Batoulis et al. (2017) concerning decision-aware notions of soundness. Incorporating data aspects, arguably, may yield a more abstract modelling framework than incorporating only decision aspects, if decisions are data-based and so the decision models are a subset of data models in general. In this thesis, we have realized execution semantics of processes depending on rule-based decisions by mapping variable values to places and coupling transitions that correspond to a matching rule to these places. This idea makes rule matching under a unique hit policy possible, it is not claimed that the full expressivity of DMN is covered (Hasic et al., 2020). But also, process routing could be be made explicit and deterministic by safeguarding exclusive splitting gateways with data places if the splitting condition can be expressed as an intersection of input variable assignments, in the exemplary case a unary intersection for the variable *Risk Level* (cf. Fig. 17a). In conclusion, our framework might apply to working with data in general, but possible only to a limited extent because we have worked under the assumptions of finite Petri nets while random variables might be more than finitely valued.

We observe that in its methodology of state space analysis this thesis can be related to other work, but that other work, arguably, is not engaged with inconsistency measurement or quantitative assessments in general. Links to this research area, however, can be found in work on the application domain of business process management (Ciccio et al., 2017; Corea and Delfmann, 2019; Corea and Thimm, 2020; Lu et al., 2008) as well as in work on defining inconsistency for procedural formalisms (Unruh et al., 2021). Lu et al. (2008) focus on assessing compliance as a high-level correctness as opposed to the low-level correctness we engage with, but also use trace data in order to relate these compliance rules to the reality of process executions. The result is a measurement of the extent to which compliance directives are transacted in business practise. This is not to be confused with the inconsistency measurement on compliance rules conducted by Corea and Delfmann (2019) and Corea and Thimm (2020). Here, the measurement captures the compatibility of compliance rules among each other and does not affect process models, let alone execution instances. Unruh et al. (2021) aims at endowing such process models, namely Petri nets, with metrics on inconsistencies. Petri nets are considered there in a general fashion as opposed to the rather specific workflows we discussed which are structurally constrained to conform to business work practise. The discussed idea of inconsistency is comparable to our concept of *dead ends* in so far that their assessment of *traps* gleans subsets of places which, once filled with tokens, remain filled. The concept of *deadlocks*, on the other hand, describes subsets of places which cannot get marked if not marked from the beginning. These deadlocks point out a direction into which the means of this thesis might be extended, because we have discussed the concept of *dead transitions*, but disregarded a conceptualization of *dead places*.

6. Conclusions

This chapter gives a review of the work conducted for the thesis at hand. Results are assessed in the light of the research aim that was set initially. Beside that, insights gained during the course of working on this thesis inspire future work, which will be presented in the end.

6.1. Summary

This thesis is engaged with the research areas of business process management, more exactly with the modelling of business processes, and inconsistency measurement. We aimed at tying these fields together by identifying desirable characteristics of business process models in general and thus conceptualizing a notion of consistency on these models which may be quantified. We deployed the Petri net formalism to describe execution semantics of process models as an interplay of tokens stored in places driven by firing of transitions. We investigated a subclass of these nets, the so-called workflows, for which we found correctness traits that are potentially desirable in business practise, namely *soundness* with its subproperties *option to complete* as a place-centered and *dead transition freeness* as a transition-centered property, and besides soundness, also *liveness* and *fairness* as transition-centered properties.
As a fundamental framework to make quantitative assessments firstly about procedural behavior in general, and then also about the correctness characteristics, we interpreted the state space of workflows as time-discrete Markov chains. We argued that the probabilities determining the stochastic process may be found in real-life process logs of the analyzed workflows. We used the term *probabilistic workflows* to denote such workflows which in their execution behaviour are subject to a probability function, and delved into defining and discussing means to analyze these workflows concerning the correctness criteria. By monitoring the Markov process on the state space of probabilistic workflows with the help of vectors indicating the state probability distribution and vectors controlling the transition firings in each time step, we were able to define inconsistency measures on probabilistic workflows. We proposed a formal correspondence between each measure and one correctness criterion, respectively, as indicated in Tab. 12.

Criterion	Option to Complete	Dead Transition Freeness	Liveness	Fairness
Measures	I_D, I_{TID}	I_{DT}	I_L	I_{EF}, I_F
	Def. 3.8, 3.9	Def. 3.14	Def. 3.14	Def. 3.15

Table 12: The inconsistency measures of this thesis.

We were able to prove formal correspondences for the measures on the option to complete, dead transition freeness and liveness. That is, we showed that a workflow holds a certain criterion if and only if the workflow is mapped to zero under

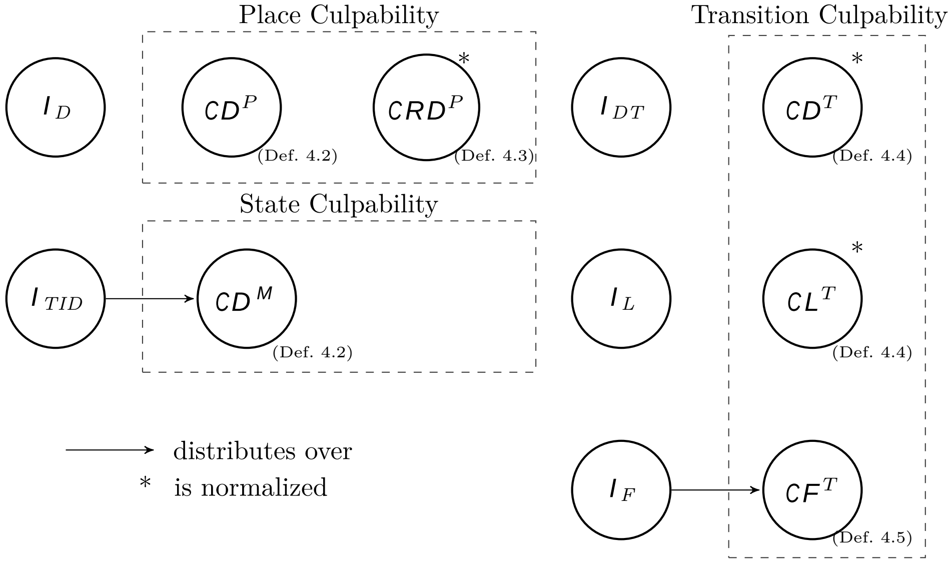


Figure 21: The culpability measures of this thesis and relations to the inconsistency measures.

the respective inconsistency measure. For the unfairness inconsistency measures, we refrained from defining or verifying a formal correspondence. To deepen the link to inconsistency measurement in classical logic, we attempted to transfer into our procedural framework rationality postulates that are commonly assigned to measures from logic. Here, we faced difficulties because the behavior of classical measures is oftentimes characterized by the nature of logical implications, a concept for which we failed to identify a direct correspondent in procedural models. However, we had anchored the procedural framework in inconsistency measurement by means of the basic consistency postulates.

After finishing the discussion of global measures for inconsistencies, we proposed measures that apply to particular workflow elements and aim at assessing the culpability of these elements for the global inconsistency. Again, we differed concerning the various correctness criteria, but also concerning the type of workflow elements to be assessed, namely the atomic places and transitions and the higher-level states. In particular, we either tried to *distribute* global inconsistency values among the workflow elements, or assess for each element a relative portion of culpability for global inconsistency, yielding *normalized* measures. Fig. 21 depicts all culpability measures and possible relations to inconsistency measures. Note that I_D, I_{DT} and I_L do not formally correspond to a culpability measure, but were depicted next to the culpability measures describing the same workflow correctness aspect.

Finally, we put our framework on trial by exploring its capability to capture and assess an exemplary BPMN model which uses a contemporary extension, namely DMN, in order to incorporate decision logic. We successfully captured the execution semantics of the process model and thus claimed that our framework is able to describe at least a subset of DMN-enhanced BPMN models, namely those with a unique hit policy. However, we required the Petri net formalism to extend to a Turing-complete version with inhitor arks in order to correctly map the execution semantics. We also noted a general problem of state explosion, because we might need to reproduce simple state spaces in an exponential order when respecting an exponentially growing number of variable combinations.

6.2. Research Aim Revisal

In the following, we will work up the initially set research aim by firstly revisiting each practical research objective, then answering the presupposed research questions and concluding with a discussion of the overall research aim.

RO1: Identify desirable low-level characteristics of workflows.
RO2: Identify desirable characteristics for inconsistency measures.
RO3: Develop methods to quantify workflows concerning these characteristics.
RO4: Assess the measures in the light of the postulates.
RO5: Evaluate the implemented measures in the light of (RQ3).

For the workflow characteristics to be assessed (RO1), we have relied on classical notions of soundness, liveness and fairness. As opposed to the other concepts, we interpreted fairness as a highly data-dependent characteristics to particularly assess load balancing, while the other characteristics describe workflows in a more structural, data-independent way. By a literature review, some customizations of classical soundness were found, i.e. relaxed and weak soundness, but these were disregarded. Concerning the inconsistency measures (RO2), we availed ourselves with rationality postulates from the well-researched domain of propositional logic, but most characteristics were found to be not applicable to our domain of procedural models. However, a basic link to classical inconsistency measurement was established by meaningfully adapting the postulate of consistency. Actual measures for each of the workflow characteristics (RO3) were then defined based on a probabilistic state space analysis, and a correspondence to the consistency postulates was formally verified (RO4).

RQ1: How should inconsistency on workflow models be defined and how to measure it?
RQ2: What are desirable properties that inconsistency measures on workflow models should have?
RQ3: To what extend can the newly defined measures capture the expressivity of business process modelling languages?

The question how inconsistency on workflows should be defined highly depends on the requirements imposed by the modeller. We have suggested inconsistency notions based on some established notions of workflow correctness and provided a framework expressive enough to capture all suggested notions (RQ1). However, recall that we identified a need for revision in particular for the notion of unfairness inconsistency. The claim that the other notions are captured indeed is based on our results concerning (RQ2), because the most fundamental desirable property identified is that of the consistency between inconsistency measure and assessed correctness criterion. More generic properties for measures could not be identified. As to (RQ3), we are confident that our framework may capture complex contemporary process models and also yield meaningful results concerning inconsistency and culpability assessment, as exemplified in a process with decision logic.

RA: Develop measures that suitably describe inconsistencies
in workflow models and pinpoint culpable elements.

In conclusion, we claim that the measures developed in this thesis may come in handy as a modelling support, because modellers may be alerted by global inconsistency values and also pointed to local culprits. Moreover, the measures are built on a solid framework that was proven flexible as it carries the different correctness notions, such that we are confident that also other concepts of workflow correctness can be incorporated.

6.3. Future Work

Related work by Lu et al. (2008) engaged with business compliance rules and applied these rules to process models in order to assess the degree of compliance in those process models. We see a chance to develop such a notion of compliance degrees based on our probabilistic framework, for example by identifying states or sets of states which hurt a certain compliance directive and then defining the compliance degree via the probability to meet the defective states in the course of execution. In order to complete our framework to the full expressivity, it will be necessary to either verify the completeness of workflows with inhibitor arks concerning its ability to capture data- and decision-aware processes, or tie up our work to the work of Haarmann et al. (2018) and de Leoni et al. (2021) who conduct state space analysis based on Colored Petri nets. The latter approach may be preferred to leverage a compact, well-readable notion as opposed to the very technical notion that we used to describe the semantics of decision-aware process models.

References

Kimon Batoulis and Mathias Weske. Soundness of decision-aware business processes. In Josep Carmona, Gregor Engels, and Akhil Kumar, editors, *Business Process Management Forum - BPM Forum 2017, Barcelona, Spain, September 10-15, 2017, Proceedings*, volume 297 of *Lecture Notes in Business Information Processing*, pages 106–124. Springer, 2017. doi: 10.1007/978-3-319-65015-9_7. URL `https://doi.org/10.1007/978-3-319-65015-9_7`.

Kimon Batoulis, Stephan Haarmann, and Mathias Weske. Various notions of soundness for decision-aware business processes. In Heinrich C. Mayr, Giancarlo Guizzardi, Hui Ma, and Oscar Pastor, editors, *Conceptual Modeling - 36th International Conference, ER 2017, Valencia, Spain, November 6-9, 2017, Proceedings*, volume 10650 of *Lecture Notes in Computer Science*, pages 403–418. Springer, 2017. doi: 10.1007/978-3-319-69904-2_31. URL `https://doi.org/10.1007/978-3-319-69904-2_31`.

Leopoldo E. Bertossi, Anthony Hunter, and Torsten Schaub. Introduction to inconsistency tolerance. In Leopoldo E. Bertossi, Anthony Hunter, and Torsten Schaub, editors, *Inconsistency Tolerance [result from a Dagstuhl seminar]*, volume 3300 of *Lecture Notes in Computer Science*, pages 1–14. Springer, 2005. doi: 10.1007/978-3-540-30597-2_1. URL `https://doi.org/10.1007/978-3-540-30597-2_1`.

Claudio Di Ciccio, Fabrizio Maria Maggi, Marco Montali, and Jan Mendling. Resolving inconsistencies and redundancies in declarative process models. *Inf. Syst.*, 64: 425–446, 2017. doi: 10.1016/j.is.2016.09.005. URL `https://doi.org/10.1016/j.is.2016.09.005`.

Carl Corea and Patrick Delfmann. Supporting business rule management with inconsistency analysis. In Wil M. P. van der Aalst, Fabio Casati, Raffaele Conforti, Massimiliano de Leoni, Marlon Dumas, Akhil Kumar, Jan Mendling, Surya Nepal, Brian T. Pentland, and Barbara Weber, editors, *Proceedings of the Dissertation Award, Demonstration, and Industrial Track at BPM 2018 co-located with 16th International Conference on Business Process Management (BPM 2018), Sydney, Australia, September 9-14, 2018*, volume 2196 of *CEUR Workshop Proceedings*, pages 141–147. CEUR-WS.org, 2018. URL `http://ceur-ws.org/Vol-2196/BPM_2018_paper_30.pdf`.

Carl Corea and Patrick Delfmann. Quasi-inconsistency in declarative process models. In Thomas T. Hildebrandt, Boudewijn F. van Dongen, Maximilian Röglinger, and Jan Mendling, editors, *Business Process Management Forum - BPM Forum 2019, Vienna, Austria, September 1-6, 2019, Proceedings*, volume 360 of *Lecture Notes in Business Information Processing*, pages 20–35. Springer, 2019. doi: 10.1007/978-3-030-26643-1_2. URL `https://doi.org/10.1007/978-3-030-26643-1_2`.

Carl Corea and Matthias Thimm. Towards inconsistency measurement in business rule bases. In Giuseppe De Giacomo, Alejandro Catalá, Bistra Dilkina, Michela Milano, Senén Barro, Alberto Bugarín, and Jérôme Lang, editors, *ECAI 2020 - 24th European Conference on Artificial Intelligence, 29 August-8 September 2020, Santiago de Compostela, Spain, August 29 - September 8, 2020 - Including 10th Conference on Prestigious Applications of Artificial Intelligence (PAIS 2020)*, volume 325 of *Frontiers in Artificial Intelligence and Applications*, pages 704–711. IOS Press, 2020. doi: 10.3233/FAIA200157. URL `https://doi.org/10.3233/FAIA200157`.

Carl Corea, Matthias Deisen, and Patrick Delfmann. Resolving inconsistencies in declarative process models based on culpability measurement. In Thomas Ludwig and Volkmar Pipek, editors, *Human Practice. Digital Ecologies. Our Future. 14. Internationale Tagung Wirtschaftsinformatik (WI 2019), February 24-27, 2019, Siegen, Germany*, pages 139–153. University of Siegen, Germany / AISeL, 2019. URL `https://aisel.aisnet.org/wi2019/track03/papers/2`.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory.* Wiley, 2001. ISBN 9780471062592. doi: 10.1002/0471200611. URL `https://doi.org/10.1002/0471200611`.

Massimiliano de Leoni, Paolo Felli, and Marco Montali. A holistic approach for soundness verification of decision-aware process models (extended version). *CoRR*, abs/1804.02316, 2018. URL `http://arxiv.org/abs/1804.02316`.

Massimiliano de Leoni, Paolo Felli, and Marco Montali. Integrating BPMN and DMN: modeling and analysis. *J. Data Semant.*, 10(1):165–188, 2021. doi: 10.1007/s13740-021-00132-z. URL `https://doi.org/10.1007/s13740-021-00132-z`.

Juliane Dehnert and Peter Rittgen. Relaxed soundness of business processes. In Klaus R. Dittrich, Andreas Geppert, and Moira C. Norrie, editors, *Advanced Information Systems Engineering, 13th International Conference, CAiSE 2001, Interlaken, Switzerland, June 4-8, 2001, Proceedings*, volume 2068 of *Lecture Notes in Computer Science*, pages 157–170. Springer, 2001. doi: 10.1007/3-540-45341-5_11. URL `https://doi.org/10.1007/3-540-45341-5_11`.

Remco M. Dijkman, Marlon Dumas, and Chun Ouyang. Semantics and analysis of business process models in BPMN. *Inf. Softw. Technol.*, 50(12):1281–1294, 2008. doi: 10.1016/j.infsof.2008.02.006. URL `https://doi.org/10.1016/j.infsof.2008.02.006`.

Jospeh Leo Doob. *Stochastic Processes.* A Wiley-interscience publication. Wiley, 1991. ISBN 9780471523697. URL `https://books.google.de/books?id=zbsrAAAAYAAJ`.

John Grant. Classifications for inconsistent theories. *Notre Dame J. Formal Log.*, 19(3):435–444, 1978. doi: 10.1305/ndjfl/1093888404. URL `https://doi.org/10.1305/ndjfl/1093888404`.

John Grant and Anthony Hunter. Measuring the good and the bad in inconsistent information. In Toby Walsh, editor, *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011*, pages 2632–2637. IJCAI/AAAI, 2011. doi: 10.5591/978-1-57735-516-8/IJCAI11-438. URL https://doi.org/10.5591/978-1-57735-516-8/IJCAI11-438.

Stephan Haarmann, Kimon Batoulis, and Mathias Weske. Compliance checking for decision-aware process models. In Florian Daniel, Quan Z. Sheng, and Hamid Motahari, editors, *Business Process Management Workshops - BPM 2018 International Workshops, Sydney, NSW, Australia, September 9-14, 2018, Revised Papers*, volume 342 of *Lecture Notes in Business Information Processing*, pages 494–506. Springer, 2018. doi: 10.1007/978-3-030-11641-5_39. URL https://doi.org/10.1007/978-3-030-11641-5_39.

Faruk Hasic, Carl Corea, Jonas Blatt, Patrick Delfmann, and Estefanía Serral. A tool for the verification of decision model and notation (DMN) models. In Fabiano Dalpiaz, Jelena Zdravkovic, and Pericles Loucopoulos, editors, *Research Challenges in Information Science - 14th International Conference, RCIS 2020, Limassol, Cyprus, September 23-25, 2020, Proceedings*, volume 385 of *Lecture Notes in Business Information Processing*, pages 536–542. Springer, 2020. doi: 10.1007/978-3-030-50316-1_35. URL https://doi.org/10.1007/978-3-030-50316-1_35.

Anthony Hunter and Sébastien Konieczny. Shapley inconsistency values. In Patrick Doherty, John Mylopoulos, and Christopher A. Welty, editors, *Proceedings, Tenth International Conference on Principles of Knowledge Representation and Reasoning, Lake District of the United Kingdom, June 2-5, 2006*, pages 249–259. AAAI Press, 2006. URL http://www.aaai.org/Library/KR/2006/kr06-027.php.

Kurt Jensen. Coloured petri nets. In Wilfried Brauer, Wolfgang Reisig, and Grzegorz Rozenberg, editors, *Petri Nets: Central Models and Their Properties, Advances in Petri Nets 1986, Part I, Proceedings of an Advanced Course, Bad Honnef, Germany, 8-19 September 1986*, volume 254 of *Lecture Notes in Computer Science*, pages 248–299. Springer, 1986. doi: 10.1007/BFb0046842. URL https://doi.org/10.1007/BFb0046842.

Ruopeng Lu, Shazia Wasim Sadiq, and Guido Governatori. Measurement of compliance distance in business processes. *Inf. Syst. Manag.*, 25(4):344–355, 2008. doi: 10.1080/10580530802384613. URL https://doi.org/10.1080/10580530802384613.

Axel Martens. Usability of web services. In Giuseppe Santucci, Wolfgang Klas, Michela Bertolotto, Coral Calero, and Luciano Baresi, editors, *4th International Conference on Web Information Systems Engineering Workshops, WISE 2003 Workshops, Rome, Italy, December 13, 2003*, pages 182–190. IEEE Computer Society, 2003. doi: 10.1109/WISEW.2003.1286801. URL https://doi.org/10.1109/WISEW.2003.1286801.

D. Meintrup and S. Schäffler. *Stochastik: Theorie und Anwendungen.* Statistik und ihre Anwendungen. Springer Berlin Heidelberg, 2006. ISBN 9783540267072. URL https://books.google.de/books?id=Lg0kBAAAQBAJ.

Tadao Murata. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541–580, 1989. doi: 10.1109/5.24143. URL https://doi.org/10.1109/5.24143.

C.A. Petri. *Kommunikation mit Automaten.* Schriften des Rheinisch-Westfälischen Institutes für Instrumentelle Mathematik an der Universität Bonn. Rheinisch-Westfälisches Institut f. instrumentelle Mathematik an d. Univ., 1962. URL https://books.google.de/books?id=NCZMvAEACAAJ.

August-Wilhelm Scheer, Oliver Thomas, Christian Seel, Gunnar Martin, and Bettina Kaffai. Geschäftsprozessorientierte software-architekturen: Revolution auf dem software-markt? In Peter Dadam and Manfred Reichert, editors, *34. Jahrestagung der Gesellschaft für Informatik, Informatik verbindet, INFORMATIK 2004, Ulm, Germany, September 20-24, 2004, Band 1*, volume P-50 of *LNI*, pages 2–13. GI, 2004. URL https://dl.gi.de/20.500.12116/28828.

Matthias Thimm. *Probabilistic Reasoning with Incomplete and Inconsistent Beliefs*, volume 336 of *DISKI.* AKA, 2012. ISBN 978-3-89838-336-3. URL https://d-nb.info/1017777195.

Matthias Thimm. On the compliance of rationality postulates for inconsistency measures: A more or less complete picture. *Künstliche Intell.*, 31(1):31–39, 2017. doi: 10.1007/s13218-016-0451-y. URL https://doi.org/10.1007/s13218-016-0451-y.

Matthias Thimm. Inconsistency measurement. In Nahla Ben Amor, Benjamin Quost, and Martin Theobald, editors, *Scalable Uncertainty Management - 13th International Conference, SUM 2019, Compiègne, France, December 16-18, 2019, Proceedings*, volume 11940 of *Lecture Notes in Computer Science*, pages 9–23. Springer, 2019. doi: 10.1007/978-3-030-35514-2_2. URL https://doi.org/10.1007/978-3-030-35514-2_2.

Elina Unruh, Patrick Delfmann, and Matthias Thimm. Quantitative deadlock analysis in petri nets using inconsistency measures. In *Proceedings of the 23rd IEEE International Conference on Business Informatics (IEEE CBI 2021)*, September 2021.

Wil M. P. van der Aalst. Verification of workflow nets. In Pierre Azéma and Gianfranco Balbo, editors, *Application and Theory of Petri Nets 1997, 18th International Conference, ICATPN ’97, Toulouse, France, June 23-27, 1997, Proceedings*, volume 1248 of *Lecture Notes in Computer Science*, pages 407–426. Springer, 1997. doi: 10.1007/3-540-63139-9_48. URL https://doi.org/10.1007/3-540-63139-9_48.

Wil M. P. van der Aalst. Business process management as the "killer app" for petri nets. *Softw. Syst. Model.*, 14(2):685–691, 2015. doi: 10.1007/s10270-014-0424-2. URL `https://doi.org/10.1007/s10270-014-0424-2`.

Wil M. P. van der Aalst, Kees M. van Hee, Arthur H. M. ter Hofstede, Natalia Sidorova, H. M. W. Verbeek, Marc Voorhoeve, and Moe Thandar Wynn. Soundness of workflow nets: classification, decidability, and analysis. *Formal Aspects Comput.*, 23(3):333–363, 2011. doi: 10.1007/s00165-010-0161-4. URL `https://doi.org/10.1007/s00165-010-0161-4`.

Mathias Weske. *Business Process Management - Concepts, Languages, Architectures, 2nd Edition.* Springer, 2012. ISBN 978-3-642-28615-5. doi: 10.1007/978-3-642-28616-2. URL `https://doi.org/10.1007/978-3-642-28616-2`.

Dmitry A. Zaitsev. Toward the minimal universal petri net. *IEEE Trans. Syst. Man Cybern. Syst.*, 44(1):47–58, 2014. doi: 10.1109/TSMC.2012.2237549. URL `https://doi.org/10.1109/TSMC.2012.2237549`.

A. Figures

This appendix lists supplementary figures for the main part of this thesis Sec. 1-6.

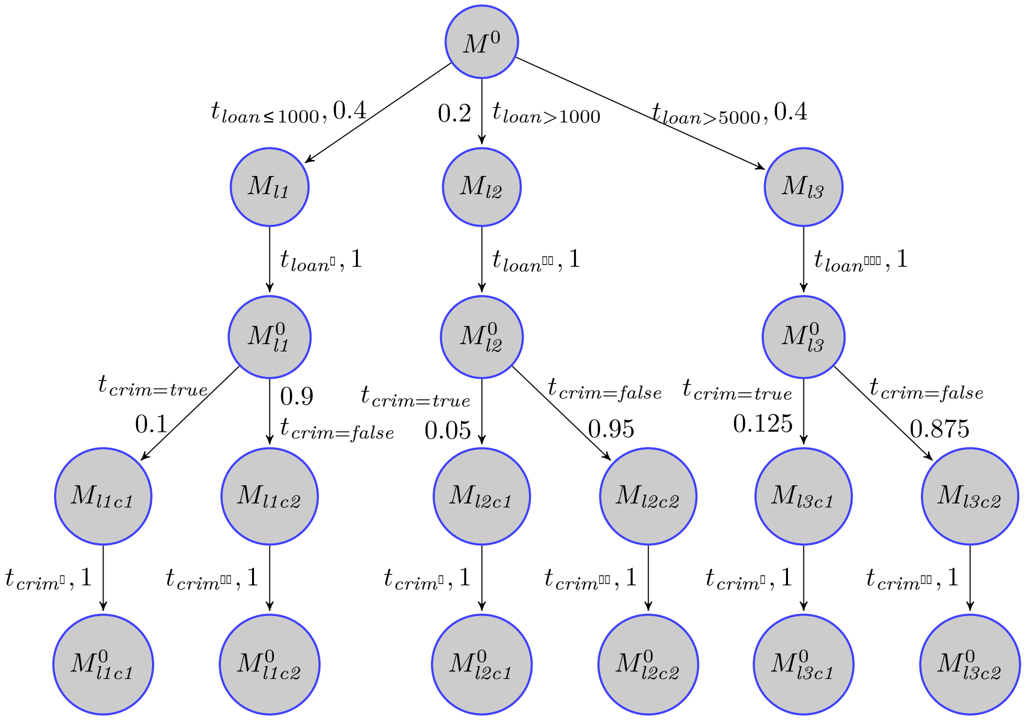


Figure 22: The first part of $R = (M, E)$, the reachability graph from the workflow W depicted in Fig. 18-20.

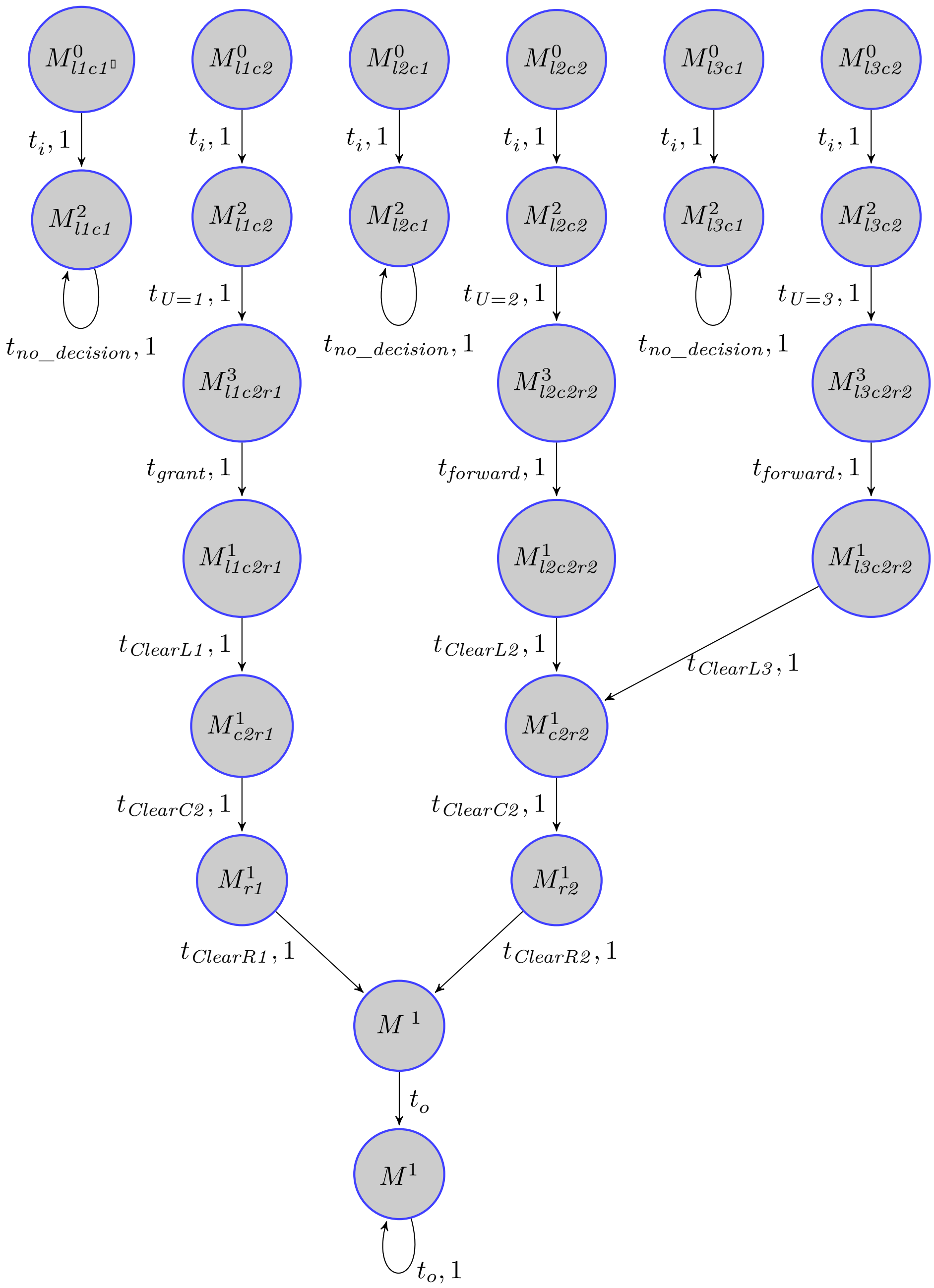


Figure 23: The second part of $R = (M, E)$, the reachability graph from the workflow W depicted in Fig. 18-20.

B. Tables

This appendix lists supplementary tables for the main part of this thesis Sec. 1-6.

Index j	Marking M_j	Places p with $M(p) = 1$
0	M^0	$\{i\}$
2	M_{l1}	$\{p_{decide\_loan}, p_{loan \leq 1000}\}$
3	M_{l2}	$\{p_{decide\_loan}, p_{loan > 1000}\}$
4	M_{l3}	$\{p_{decide\_loan}, p_{loan > 5000}\}$
5	M^0_{l1}	$\{p_{loan \leq 1000}, p_{criminal}\}$
6	M^0_{l2}	$\{p_{loan > 1000}, p_{criminal}\}$
7	M^0_{l3}	$\{p_{loan > 5000}, p_{criminal}\}$
8	M_{l1c1}	$\{p_{loan \leq 1000}, p_{crim=true}, p_{decide\_criminal}\}$
9	M_{l1c2}	$\{p_{loan \leq 1000}, p_{crim=false}, p_{decide\_criminal}\}$
10	M_{l2c1}	$\{p_{loan > 1000}, p_{crim=true}, p_{decide\_criminal}\}$
11	M_{l2c2}	$\{p_{loan > 1000}, p_{crim=false}, p_{decide\_criminal}\}$
12	M_{l3c1}	$\{p_{loan > 5000}, p_{crim=true}, p_{decide\_criminal}\}$
13	M_{l3c2}	$\{p_{loan > 5000}, p_{crim=false}, p_{decide\_criminal}\}$
14	M^0_{l1c1}	$\{p_{loan \leq 1000}, p_{crim=true}, i\ \}$
15	M^0_{l1c2}	$\{p_{loan \leq 1000}, p_{crim=false}, i\ \}$
16	M^0_{l2c1}	$\{p_{loan > 1000}, p_{crim=true}, i\ \}$
17	M^0_{l2c2}	$\{p_{loan > 1000}, p_{crim=false}, i\ \}$
18	M^0_{l3c1}	$\{p_{loan > 5000}, p_{crim=true}, i\ \}$
19	M^0_{l3c2}	$\{p_{loan > 5000}, p_{crim=false}, i\ \}$
20	M^2_{l1c1}	$\{p_{loan \leq 1000}, p_{crim=true}, p_{decide\_risk\_level}\}$
21	M^2_{l1c2}	$\{p_{loan \leq 1000}, p_{crim=false}, p_{decide\_risk\_level}\}$
22	M^2_{l2c1}	$\{p_{loan > 1000}, p_{crim=true}, p_{decide\_risk\_level}\}$
23	M^2_{l2c2}	$\{p_{loan > 1000}, p_{crim=false}, p_{decide\_risk\_level}\}$
24	M^2_{l3c1}	$\{p_{loan > 5000}, p_{crim=true}, p_{decide\_risk\_level}\}$
25	M^2_{l3c2}	$\{p_{loan > 5000}, p_{crim=false}, p_{decide\_risk\_level}\}$
26	M^3_{l1c2r1}	$\{p_{loan \leq 1000}, p_{crim=false}, p_{risk=low}, p_{XOR}\}$
27	M^3_{l2c2r2}	$\{p_{loan > 1000}, p_{crim=false}, p_{risk=medium}, p_{XOR}\}$
28	M^3_{l3c2r2}	$\{p_{loan > 5000}, p_{crim=false}, p_{risk=medium}, p_{XOR}\}$
29	M^1_{l1c2r1}	$\{p_{loan \leq 1000}, p_{crim=false}, p_{risk=low}, o\ \}$
30	M^1_{l2c2r2}	$\{p_{loan > 1000}, p_{crim=false}, p_{risk=medium}, o\ \}$
31	M^1_{l3c2r2}	$\{p_{loan > 5000}, p_{crim=false}, p_{risk=medium}, o\ \}$
32	M^1_{c2r1}	$\{p_{crim=false}, p_{risk=low}, o\ \}$
33	M^1_{c2r2}	$\{p_{crim=false}, p_{risk=medium}, o\ \}$
34	M^1_{r1}	$\{p_{risk=low}, o\ \}$
35	M^1_{r2}	$\{p_{risk=medium}, o\ \}$
36	$M\ ^1$	$\{o\ \}$
1	M^1	$\{o\}$

Table 13: Specifications of the markings of the reachability graph from Fig. 22-23.

Index i	Markov sequence entries $s^i_j > 0$
0	$s^0_0 = 1$
1	$s^1_2 = 0.4, s^1_3 = 0.2, s^1_4 = 0.4$
2	$s^2_5 = 0.4, s^2_6 = 0.2, s^2_7 = 0.4$
3	$s^3_8 = 0.04, s^3_9 = 0.36, s^3_{10} = 0.01, s^3_{11} = 0.19, s^3_{12} = 0.05, s^3_{13} = 0.35$
4	$s^4_{14} = 0.04, s^4_{15} = 0.36, s^4_{16} = 0.01, s^4_{17} = 0.19, s^4_{18} = 0.05, s^4_{19} = 0.35$
5	$s^5_{20} = 0.04, s^5_{21} = 0.36, s^5_{22} = 0.01, s^5_{23} = 0.19, s^5_{24} = 0.05, s^5_{25} = 0.35$
6	$s^6_{20} = 0.04, s^6_{22} = 0.01, s^6_{24} = 0.05, s^6_{26} = 0.36, s^6_{27} = 0.19, s^6_{28} = 0.35$
7	$s^7_{20} = 0.04, s^7_{22} = 0.01, s^7_{24} = 0.05, s^7_{29} = 0.36, s^7_{30} = 0.19, s^7_{31} = 0.35$
8	$s^8_{20} = 0.04, s^8_{22} = 0.01, s^8_{24} = 0.05, s^8_{32} = 0.36, s^8_{33} = 0.54$
9	$s^9_{20} = 0.04, s^9_{22} = 0.01, s^9_{24} = 0.05, s^9_{34} = 0.36, s^9_{35} = 0.54$
10	$s^{10}_{20} = 0.04, s^{10}_{22} = 0.01, s^{10}_{24} = 0.05, s^{10}_{36} = 0.9$
11	$s^{11}_1 = 0.9, s^{11}_{20} = 0.04, s^{11}_{22} = 0.01, s^{11}_{24} = 0.05$
$\vdots$	$\vdots$
$n \to \infty$	$s^n_1 = 0.9, s^n_{20} = 0.04, s^n_{22} = 0.01, s^n_{24} = 0.05$

Table 14: Specifications of the members of the Markov sequence s for R from Fig. 22-23.

Index i	Firing deltas $\delta^i(t_j) = c_j^i - c_j^{i-1} > 0$ for transitions t_j
1	$\delta^1(t_{loan \leq 1000}) = 0.4,\ \ \delta^1(t_{loan > 1000}) = 0.2,\ \ \delta^1(t_{loan > 5000}) = 0.4$
2	$\delta^2(t_{loan\ }) = 0.4,\ \ \delta^2(t_{loan\ \ }) = 0.2,\ \ \delta^2(t_{loan\ \ \ }) = 0.4$
3	$\delta^3(t_{crim=true}) = 0.1,\ \ \delta^3(t_{crim=false}) = 0.9$
4	$\delta^4(t_{crim\ }) = 0.1,\ \ \delta^4(t_{crim\ \ }) = 0.9$
5	$\delta^5(t_i) = 1$
6	$\delta^6(t_{U=1}) = 0.36,\ \ \delta^6(t_{U=2}) = 0.19,\ \ \delta^6(t_{U=3}) = 0.35,$
	$\delta^6(t_{no\_decision}) = 0.1$
7	$\delta^7(t_{grant}) = 0.36,\ \ \delta^7(t_{forward}) = 0.54,\ \ \delta^7(t_{no\_decision}) = 0.1$
8	$\delta^8(t_{ClearL1}) = 0.36,\ \ \delta^8(t_{ClearL2}) = 0.19,\ \ \delta^8(t_{ClearL3}) = 0.35,$
	$\delta^8(t_{no\_decision}) = 0.1$
9	$\delta^9(t_{ClearC2}) = 0.9,\ \ \delta^9(t_{no\_decision}) = 0.1$
10	$\delta^{10}(t_{ClearR1}) = 0.36,\ \ \delta^{10}(t_{ClearR2}) = 0.54,\ \ \delta^{10}(t_{no\_decision}) = 0.1$
11	$\delta^{11}(t_o) = 0.9,\ \ \delta^{11}(t_{no\_decision}) = 0.1$
12	$\delta^{12}(t\ \ ) = 0.9,\ \ \delta^{12}(t_{no\_decision}) = 0.1$
13	$\delta^{13}(loan \leq 1000) = 0.4 \cdot 0.9,\ \ \delta^{13}(loan > 1000) = 0.4 \cdot 0.9,$
	$\delta^{13}(loan > 5000) = 0.4 \cdot 0.9,\ \ \delta^{13}(t_{no\_decision}) = 0.1$
$\vdots$	$\vdots$
18	$\delta^{18}(t_{U=1}) = 0.9 \cdot 0.36,\ \ \delta^{18}(t_{U=2}) = 0.9 \cdot 0.19,\ \ \delta^{18}(t_{U=3}) = 0.9 \cdot 0.35,$
	$\delta^{18}(t_{no\_decision}) = 0.1 + 0.9 \cdot 0.1$
$\vdots$	$\vdots$
$12z + 1$	$\delta^{12z+1}(loan \leq 1000) = 0.4 \cdot 0.9^z,\ \ \delta^{12z+1}(loan > 1000) = 0.4 \cdot 0.9^z,$
	$\delta^{12z+1}(loan > 5000) = 0.4 \cdot 0.9^z,\ \ \delta^{12z+1}(t_{no\_decision}) = \underbrace{0.1 \cdot \sum_{k=0}^{z} 0.9^k}_{\xrightarrow{z \to \infty} 1}$
	$\vdots$
$n \to \infty$	$\delta^n(t_{no\_decision}) = 1$

Table 15: Specifications for the members of the Control vector sequence c for the Markov sequence s from Tab. 14 concerning the reachability graph R from Fig. 22-23. For each time step, the increases in the control vector are depicted.