HANDLING INCONSISTENCY IN BUSINESS RULE BASES

Dissertation

Genehmigte Dissertation zur Erlangung des akademischen Grades
eines Doktors der Naturwissenschaften (Dr. rer. nat.)

Fachbereich 4: Informatik
Universität Koblenz-Landau

Carl Corea

Vorsitzender des Promotionsausschusses: Prof. Dr. Ralf Lämmel
Vorsitzende der Promotionskommission: Prof. Dr. Viorica Sofronie-Stokkermans
Berichterstatter:
Prof. Dr. Patrick Delfmann
PD Dr. Matthias Thimm
Prof. Dr. Jan Mendling

Datum der wissenschaftlichen Aussprache: 16.12.2020

Abstract

EN - Within the field of Business Process Management, *business rules* are commonly used to model company decision logic and govern allowed company behavior. An exemplary business rule in the financial sector could be for example: *"A customer with a mental condition is not creditworthy"*. Business rules are usually created and maintained collaboratively and over time. In this setting, modelling errors can occur frequently. A challenging problem in this context is that of *inconsistency*, i.e., contradictory rules which cannot hold at the same time. For instance, regarding the exemplary rule above, an inconsistency would arise if a (second) modeller entered an additional rule: *"A customer with a mental condition is always creditworthy"*, as the two rules cannot hold at the same time. In this thesis, we investigate how to handle such inconsistencies in business rule bases. In particular, we develop methods and techniques for the *detection, analysis* and *resolution* of inconsistencies in business rule bases.

DE - Im Kontext des Geschäftsprozessmanagements werden häufig sogenannte Business Rules (Geschäftsregeln) als zentrales Artefakt zur Modellierung von unternehmensinterner Entscheidungslogik sowie der Steuerung von Unternehmensaktivitäten eingesetzt. Eine exemplarische Geschäftsregel aus dem Finanzsektor wäre z.B. *"Ein Kunde mit geistiger Behinderung ist nicht geschäftsfähig"*. Business Rules werden hierbei meist von mehreren Mitarbeitern und über einen längeren Zeitraum erstellt und verwaltet. Durch dieses kollaborative Arbeiten kann es jedoch leicht zu Modellierungsfehlern kommen. Ein großes Problem in diesem Kontext sind Inkonsistenzen, d.h. sich widersprechende Regeln. In Bezug auf die oben gezeigte Regel würde beispielsweise eine Inkonsistenz entstehen, wenn ein (zweiter) Modellierer eine zusätzliche Regel *"Kunden mit geistiger Behinderung sind voll geschäftsfähig"* erstellt, da diese beiden Regeln nicht zeitgleich einhaltbar sind. Die vorliegende Arbeit beschäftigt sich mit dem Umgang mit solchen Inkonsistenzen in Business Rule-Repositorien. Hierbei werden im Speziellen Methoden und Techniken zur *Erkennung, Analyse* und *Behebung* von Inkonsistenzen in Regelbasen entwickelt.

Publications and Disclaimer

This thesis contains 11 publications which have already been published:

Conferences

1. Carl Corea, Matthias Thimm. Towards Inconsistency Measurement in Business Rule Bases. In Proceedings of the 24th European Conference on Artificial Intelligence (ECAI). Santiago de Compostela, 2020. (Citation [35])
2. Carl Corea, Sabine Nagel, Patrick Delfmann. Effects of Visualization Techniques on Understanding Inconsistencies in Automated Decision-Making. In Proceedings of the 53rd Hawaii International Conference on System Sciences (HICSS). Maui, 2020. (Citation [33])
3. Carl Corea, Jonas Blatt, Patrick Delfmann. A Tool for Decision-Logic Level Verification in DMN Decision Tables. In Proceedings of the Demonstration Track at BPM 2019 co-located with the 17th International Conference on Business Process Management (BPM). Wien, 2019. (Citation [27])
4. Carl Corea, Patrick Delfmann. Quasi-Inconsistency in Declarative Process Models. In Business Process Management Forum co-located with the 17th International Conference on Business Process Management (BPM). Wien, 2019. (Citation [31])
5. Carl Corea, Matthias Deisen, Patrick Delfmann. Resolving Inconsistencies in Declarative Process Models based on Culpability Measurement. In Proceedings der 14. Internationalen Tagung der Wirtschaftsinformatik (WI). Siegen, 2019. (Citation [28])
   Winner of the best-paper award.
6. Sabine Nagel, Carl Corea, Patrick Delfmann. Effects of Quantitative Measures on Understanding Inconsistencies in Business Rules. In Proceedings of the 52nd Hawaii International Conference on System Sciences (HICSS). Maui, 2019. (Citation [89])
7. Carl Corea, Patrick Delfmann. Supporting Business Rule Management with Inconsistency Analysis. In Proceedings of the Industrial Track at BPM 2018 co-located with the 16th International Conference on Business Process Management (BPM). Sydney, 2018. (Citation [29])
8. Carl Corea, Patrick Delfmann. A Tool to Monitor Consistent Decision-Making in Business Process Execution. In Proceedings of the Demonstration Track at BPM 2018 co-located with the 16th International Conference on Business Process Management (BPM). Sydney, 2018. (Citation [30]).

Journals

9. Carl Corea and Matthias Thimm. On Quasi-Inconsistency and its Complexity. In *Artificial Intelligence* (AIJ). 2020 (Citation [34])
10. Faruk Hasic, Carl Corea, Jonas Blatt, Patrick Delfmann and Estefania Serral. Decision Model Change Patterns for Dynamic System Evolution. In *Knowledge and Information Systems* (KAIS). 2020 (Citation [64])
11. Carl Corea and Patrick Delfmann. A Taxonomy of Business Rule Organizing Approaches in Regard to Business Process Compliance. In *International Journal of Conceptual Modeling* (EMISA). 2020. (Citation [32])

The publication *Towards Inconsistency Measurement in Business Rule Bases* was written by myself under the supervision of Matthias Thimm. Mr. Thimm provided some proofs which can be found in the appendix of that publication.

The publications *Effects of Visualization Techniques on Understanding Inconsistencies in Automated Decision-Making* and *Effects of Quantitative Measures on Understanding Inconsistencies in Business Rules* were co-authored by Sabine Nagel under the supervision of Patrick Delfmann. The publications were written collaboratively and the corresponding experiments were conducted in cooperation with Ms. Nagel. My contributions to these two publications were specifically 1) initializing the ideas of the papers, 2) development of the underlying inconsistency measures used in the experiments, 3) positioning the works in the scope of related work, 4) identification of statistical methods and evaluation measures, 5) evaluation of collected data.

The publication *A Tool for Decision-Logic Level Verification in DMN Decision Tables* was written by myself under the supervision of Patrick Delfmann. The demonstration presented in that paper is based on a prototype developed by Jonas Blatt in his research thesis written under the supervision of myself and Patrick Delfmann.

The publication *Resolving Inconsistencies in Declarative Process Models* was written by myself under the supervision of Patrick Delfmann. The run-time experiments in that publication were conducted using the library developed by Matthias Deisen in his master thesis "(Quasi-)Inconsistency Library for Business Rule Management" [40] written under the supervision of myself and Patrick Delfmann. I am very happy to have received the best-paper award for this publication at the WI 2019.

The publications *Supporting Business Rule Management with Inconsistency Analysis*, *Quasi-Inconsistency in Declarative Process Models*, *A Tool to Monitor Consistent Decision-Making in Business Process Execution* and *A Taxonomy of Business Rule Organizing Approaches in Regard to Business Process Compliance* were written by myself under the supervision of Patrick Delfmann. I want to acknowledge that the usage-example shown in [30] is based on a prototype developed in the research lab "Business Rule Management" (WS 2017/18) supervised by myself and Patrick Delfmann. The students of that research lab were awarded with the first annual Debeka Innovation award 2019 for their results.

The publication *On quasi-inconsistency and its complexity* was co-authored by Matthias Thimm and written collaboratively. My contribution to this publication was specifically 1) initializing the idea of the paper, 2) initial conceptualization and formalization of rationality postulates, 3) adaptation of inconsistency measures, 4) positioning the work in the context of business rules management.

The publication *Decision Model Change Patterns for Dynamic System Evolution* was co-authored by Faruk Hasic, myself, Jonas Blatt, Patrick Delfmann and Estefania Serral. My contribution to this publication was specifically 1) classifying inconsistency types in the DMN standard, 2) identifying corresponding change patterns for restoring consistency, 3) conceptualization of the developed prototype. I want to acknowledge that the demonstration shown in this publication is based on the tool developed by Jonas Blatt in his master thesis "Prototyping a verification tool for Decision Model and Notation" [15] supervised by myself and Patrick Delfmann.

Further details on the exact author percentages can be found in Section 6. Other than already mentioned, many people have contributed to this thesis through discussions and guidance, and this work would not have been possible without them!

As a further disclaimer, the currentness of all presented results refers to the individual publications. For example, the results of the literature review and the discussion of related work presented in Section 3.5 considers only results up to the publication of that respective publication (June 2020), and not necessarily the publication date of this thesis.

Acknowledgements

So many people have supported me in my Ph.D. years and I can't thank you enough!

I want to thank Patrick Delfmann for his supervision and guidance through the years, and especially for giving me the freedom to pursue the research that interested me the most! I came to work motivated every single day, so thank you for creating this amazing work atmosphere!

I also thank Matthias Thimm for his guidance through the years! It was the seminar paper in your Semantic Web seminar that even got me thinking about academia back when I was studying and I don't think I would be here today otherwise, so thank you!

I would especially like to thank all my co-authors and (former) students for their support in all the research. Especially, thanks to the guys from the BRM research lab, Matthias Deisen, for his hard work, Jonas Blatt, for his relentless efforts (you are a legend!), Faruk Hasic, for an amazing last stretch ;), and Sabine Nagel, for her incredible input and help.

I would like to thank the entire BPM conference series for the inspiration throughout the years and an incredible work atmosphere. From my first BPM in Barcelona (which btw, Josep, was one of the best conferences I have ever been to), I felt so welcome and directly knew I had found a scientific home! Thank you to all the amazing and interesting people met there and hopefully see you again in person soon.

Last, I especially want to thank my family for their love and support! Thank you to my dad, sister, grandma and parents in law for always being there for me. I love you all and I couldn't have done this without you.

And of course, thank you to my amazing wife, Kerstin Corea. I cannot thank you enough for your love and support, and I could not have done any of this without you! I am so thankful to have you in my life and can't wait to see what the next chapter holds. I love you!

Dedicated in memory of my mom, Doris Corea, who I know would have loved this. Also dedicated to Ernest Corea, probably the smartest man I ever knew.

Contents

Part I - Exposé

Part II - Individual Contributions

Appendix

List of Rule Bases

$$
\begin{aligned}
\mathcal{B}_1 &= \{\mathit{newCustomer} \to \mathit{creditWorthy}; \mathit{newCustomer} \to \neg \mathit{creditWorthy}\}\\
\mathcal{B}_2 &= \{\mathit{mentalCondition} \to \neg \mathit{contractuallyCapable}; \neg \mathit{contractuallyCapable} \to \neg \mathit{creditWorthy}\}\\
\mathcal{B}_3 &= \{\mathit{customer}; \mathit{mentalCondition}; \mathit{mentalCondition} \to \neg \mathit{contractuallyCapable},\\
&\qquad \neg \mathit{contractuallyCapable} \to \neg \mathit{creditWorthy}\}\\
\mathcal{B}_4 &= \{\mathit{platinumCustomer}; \mathit{platinumCustomer} \to \mathit{contractuallyCapable};\\
&\qquad \mathit{mentalCondition}; \mathit{mentalCondition} \to \neg \mathit{contractuallyCapable}\}\\
\mathcal{B}_5 &= \{a \to b; b \to c; a \to \neg c\}\\
\mathcal{B}_6 &= \{a \to e; c \to \neg e; a \to c; b \to e\}\\
\mathcal{B}_7 &= \{a \to b; b \to c; c \to d\}\\
\mathcal{B}_8 &= \{a; \neg a\}\\
\mathcal{B}_9 &= \{a; a \to b; a \to \neg b\}\\
\mathcal{B}'_9 &= \{a; \neg b; c; \neg e; a \to b; a \to \neg b; a \to d; d \to e\}\\
\mathcal{B}_{10} &= \{a; a \to b; a \to \neg b; c; \neg c\}\\
\mathcal{B}_{11} &= \{a; a \to b; \neg b\}\\
\mathcal{B}_{12} &= \{a \to b; a \to \neg b\}\\
\mathcal{B}_{13} &= \{a \to b; b \to c; b \to d; a \to \neg c; d \to e; e \to c\}\\
\mathcal{B}_{14} &= \{a; a \to b; a \to \neg b; \neg a\}\\
\mathcal{B}_{15} &= \{a \to b; b \to c; b \to d; a, e \to \neg c; a \to \neg b; d \to e; e \to c\}\\
\mathcal{B}_{16} &= \{a \to n; a \to \neg n; b \to d; c \to \neg d; x \to z; y \to \neg z\}
\end{aligned}
$$

For readability, we sometimes use the ”;” sign to separate elements in order to avoid confusion with ”,” signs used inside rule premises.

1

Introduction

The topic of the work at hand is handling inconsistency in business rule bases. Essentially, this relates to *identifying*, *analyzing* and *resolving* inconsistent business rules, i.e., contradictory rules which cannot hold at the same time. In this chapter, we introduce the background and motivation of this thesis. We present the status quo on fields related to business rules and inconsistency handling. Then, based on an identification of research gaps, we develop a main research question and present our specific contributions.

1.1 Background and Motivation

Business rules have gained major attention in the context of business process management [2, 71]. Here, business rules are used to encode policies, norms, regulations or laws as a declarative business logic, aimed to ensure that company activities comply with such regulatory controls. For example, the company should only conduct activities as constrained by the set of business rules. Otherwise, the behavior might be inefficient due to a disregard to internal policies, or might even violate external legal regulations, the latter of which could result in sensitive financial fines, or even criminal prosecution [63]. Consequently, company efforts are being directed to ensure that activities follow internal and external regulations by means of business rules. Following GOVERNATORI ET AL., we define these company efforts as *compliance*, i.e., *"an act or process to ensure that business operations, processes, and practices are in accordance with prescriptive [internal or external] documents"* [63][p.3].

While using business rules to verify the compliance of company activities is promising, it comes with increased demands on the quality of the business rules themselves, as the rules become the de facto artifact used to monitor compliant behavior. However, as company rule bases are usually maintained by multiple modelers, modelling errors can occur frequently [9, 29, 106, 120]. For instance, BATOULIS AND WESKE (2017) report on a case study with a large insurance company, which revealed that 27% of analyzed rules contained modelling errors. In result, modelling errors in business rules are broadly recognized as an important problem for companies, as they impede using business rules for their intended purpose of ensuring compliant processes, see e.g. [29, 92, 106, 120] for an overview and [11, 21, 28, 42] for some recent works.

A potential problem here is that of *inconsistency*, i.e., (multiple) rules that contradict each other. For example, consider the following business rule base $\mathcal{B}_1$ in Figure 1.1 (we will formalize syntax and semantics later) with the intuitive meaning that we have two contradictory business rules, stating that 1.) new customers are generally credit worthy, and 2.) new customers are not credit worthy. Given a process instance where we encounter a new customer, the rule base is inconsistent in the classic-logical sense, as it would entail contradictory conclusions. Subsequently, the rule base cannot be used to draw meaningful conclusions. This jeopardizes previous company efforts that have been invested in modelling, for example, it might be impossible to execute the process model as shown in Figure 1.1 (right).

$$
\begin{aligned}
\mathcal{B}_1 = \{&\mathit{newCustomer} \rightarrow \mathit{creditWorthy},\\
&\mathit{newCustomer} \rightarrow \neg\mathit{creditWorthy}\}
\end{aligned}
$$

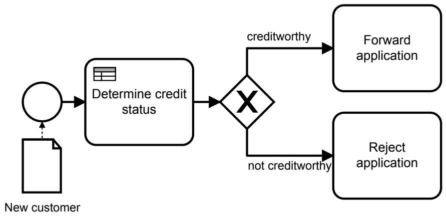


Figure 1.1: Exemplary rule base and corresponding business process.

To counteract such problems, companies need to be supported with means for the detection and analysis of inconsistencies in the scope of *business rules management* (BRM). In case of inconsistencies, methods are needed that analyze the inconsistency and present the users with a careful analysis as a strategic basis for re-modeling.

The field of *inconsistency measurement* [61, 130] is about analyzing inconsistency in logic-based knowledge representation formalisms and therefore represents a good candidate for an application in BRM. In this field, a central object of study are *quantitative measures*, which allow to assign a numerical value to (elements of) a knowledge base, with the informal meaning that a higher value reflects a higher degree of inconsistency. These measures could be useful in the context of BRM to assess inconsistencies in rule bases, i.e. *quantify* the severity of inconsistency, and *pin-point* the causes of inconsistency.

Consequently, the aim of this thesis is to support business rules management in handling inconsistent business rules by developing technical means for *detecting, analyzing* and *resolving* inconsistencies in business rule bases, while extending results from the scientific field of inconsistency measurement.

In particular, we will adapt and develop inconsistency measures for quantifying inconsistency in business rule bases. We will now present the status quo of related fields and identify current research gaps. The research question is then presented in Section 1.4.

1.2 Status Quo

The problem of handling inconsistencies in business rule bases touches a multitude of different scientific fields and streams of research. Specifically, this work studies the intersection of *Business Process Management (BPM), Business Process Compliance (BPC), Business Rule Management (BRM)* and *Inconsistency Measurement* (the latter of which is a sub-field of Knowledge Representation and Reasoning). To clarify the relations of these fields and why it is necessary to investigate this intersection, we present the state of the art and show respective limitations. For readability, we will display a breadcrumb-navigation showing the interconnection of chapters using the "▷" symbol.

1.2.1 Business Process Management

▷ *BPM*

The topicality of business rules is strongly interconnected to the discipline of *Business Process Management* (BPM), as a central aim of business rules is to facilitate compliant models and process execution. Here, BPM is concerned with concepts, methods, and techniques towards the design, configuration, enactment, and evaluation of company processes, i.e., sequences of company activities [43,140]. In this context, *process models* are the de facto object of study to represent such company processes [140]. Process models are usually represented by means of graphical modelling languages – such as the Business Process Model and Notation[1] – and aim to capture the prescribed sequence in which company activities should be conducted in, as well as the socio-technical environment in which the process is embedded in. Following a similar definition as in [41,116], we define a process model as a tuple M, via

$$M = (V, E, T_v, T_e, C, \alpha, \beta, \gamma),$$

where V is a set of nodes in the process model graph and $E = E_D \cup E_U$ is the set of edges between nodes, where $E_D \subseteq V \times V$ is the set of directed edges and $E_U \subseteq \{\{v_1, v_2\} \mid v_1, v_2 \in V\}$ is the set of undirected edges. T_v/T_e are sets of node/edge types. C is a set of captions, i. e., labels used in the process model. α, β, γ are functions, where α assigns a node in V to its type in T_v, β assigns an edge in E to its type in T_e, and γ assigns a node in V or an edge in E to a label in C.

---

[1] `https://www.omg.org/spec/BPMN/2.0/About-BPMN/`

Example 1. Figure 1.2 shows an exemplary process model in the Business Process Model and Notation (BPMN) standard. The model in Figure 1.2 can be formalized via $V = \{v_1, v_2, v_3, v_4, v_5\}$, $E = \{e_1, e_2, e_3, e_4\}$, $e_1 = (v_1, v_2), e_2 = (v_2, v_3), e_3 = (v_3, v_4), e_4 = (v_3, v_5)$ $T_v = \{$StartEvent, BusinessRuleTask, Task, XOR$\}$, $T_e = \{$controlflow$\}$, $C = \{$"Determine Credit Status", "Forward application","Reject application", "creditworthy", "not creditworthy"$\}$, $\alpha(v_1) =$ "Start Event", $\alpha(v_2) =$ "BusinessRuleTask", $\alpha(v_3) =$ "XOR", $\alpha(v_4) = \alpha(v_5) =$"Task", $\beta(e_1) = \beta(e_2) = \beta(e_3) = \beta(e_4) =$"controlflow", $\gamma(v_2) =$"Determine Credit Status", $\gamma(v_4) =$"Forward application", $\gamma(v_5) =$"Reject application", $\gamma(e_3) =$"creditworthy", $\gamma(e_4) =$"not creditworthy".

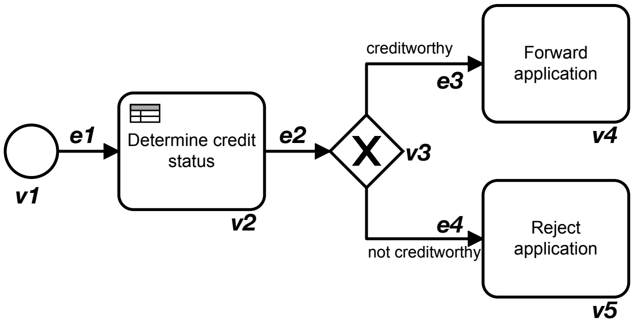


Figure 1.2: Exemplary process model.

Given such process models, the intuition is that systems and employees should perform activities as prescribed by the model. An execution of a process model (i.e., one traversal through a process model) is referred to as a case, or a process instance [1,140]. That is, a company activity is performed at a given point in time, and related to a case. We denote performing a company activity by means of an *event* e. In regard to a process model $M = (V, E, T_v, T_e, C, \alpha, \beta, \gamma)$, an event e is defined via

$$e = (a, c, i)$$

where $a \in C$ is an activity identifier (i.e. a well-defined step in a process, which could be for example a task or a message occurrence in BPMN), $c \in \mathbb{N}$ is an id that refers to the case (i.e., a process instance), and $i \in \mathbb{N}$ is a sequence number that provides the order of e w.r.t. c (i.e., following [1], we assume all company activities can be captured sequentially and thus events can be ordered w.r.t. the respective case)[2]. In the following, we denote the activity identifier of e as $a(e)$, the case of e as $c(e)$ and the order of e as $i(e)$. Also, the universe of all possible events is denoted as $\mathcal{E}$.

An ordered sequence of events that refer to the same case is referred to as a trace. In other words, a trace t is a non-empty sequence of events $\epsilon = \langle e_1, ..., e_{|\epsilon|} \rangle$

---

[2] We acknowledge that events may also comprise other properties, however in this work we confine the discussion to activity identifiers, case ids and the order.

(where $\mid \epsilon \mid$ is the length of t), s.t. for every $k <\mid \epsilon \mid$: $c(e_k) = c(e_{k+1})$ and $i(e_k) < i(e_{k+1})$. We denote the overall case id of a trace t as $c(t)$ by a slight abuse of notation. Also, we denote the universe of all possible traces as $\mathcal{T}$. We assume all traces are complete, i.e., given a trace $t \in \mathcal{T}$, there exists no event $e' \in \mathcal{E}$, s.t. $e' \notin t$ and $c(e') = c(t)$.

Then, the actually performed company activities/traces can be captured in *event logs* [1]. We define an event log E via

$$E \subseteq \mathcal{T},$$

i.e., a set of (completed) traces.

Example 2. Table 1.1 shows an exemplary event log corresponding to the process model in Figure 1.2. The event log captures all company activities. As can be seen, there have been three cases, where the loan application in the first two cases is forwarded and rejected in case 3.

Activity	Case ID	Order
Determine Credit Status	1	1
Forward Application	1	2
Determine Credit Status	2	1
Forward Application	2	2
Determine Credit Status	3	1
Reject Application	3	2

Table 1.1: Exemplary event log.

1.2.2 Business Process Compliance

▷ *BPM* ▷ *BPC*

While business process models are created by the company, the actual allowed sequences and behavior are often impacted by external factors, such as laws and regulations [63,107]. Consequently, it becomes imperative to verify that the designed process models and their execution are *compliant* with such regulations. Otherwise, violating regulations can lead to sensitive financial fines, or even criminal prosecution [63,84,85,105].

To this aim, the discipline of *Business Process Compliance* (BPC) has emerged, which comprises methods and techniques concerned with ensuring the regulatory compliance of business processes [105]. To ensure compliance, there are basically three different perspectives, namely a *design-time* perspective, a *run-time* perspective, and a *post-execution* perspective [63][3].

---

[3] We acknowledge that there are hybrid approaches that combine different perspectives, but continue to discuss these three basic perspectives for simplicity.

- **Design-time Compliance.** Design-time compliance can be defined as a preventative strategy, with the goal of facilitating compliant-by-design process models [63, 106]. Here, process models are analyzed in order to detect compliance violations.

- **Run-time Compliance.** In run-time compliance management, the compliant execution of process models is verified during the execution of the process [63]. In case of detected compliance violations, means to react to such violations are needed, such that no compliance breaches are accidentally committed.

- **Post-execution Compliance.** Post-execution compliance focuses on the analysis of event logs by domain experts after process execution [63]. Here, the actually observed behavior, e.g., the company activities recorded in event logs during process execution, and the relations of different process instances (or traces) can be verified for compliance.

Regarding which strategy is best, there are advantages to all approaches. Following HASHMI ET AL. (2018), *"the increased pressure and threat of possible criminal prosecutions [...] make the auditing method a less attractive compliance reporting strategy"* [63][p. 83]. Intuitively, compliant-by-design processes are desirable for companies. In this context, works such as [84, 85] however point out that some compliance violations might not be detectable a priori, as they might be dependent on case-specific contexts. Thus, run-time compliance management might be necessary, despite design-time compliance management efforts. Last, companies can also benefit from post-execution compliance by means of a retrospective compliance analysis from a global perspective, for example, further comparing the interrelations between different cases to identify commonalities or interdependencies [18,63]. Ideally, companies should thus implement all three compliance management strategies [63, 105].

Following HASHMI ET AL. (2018), BPC can be defined as *"an act or process to ensure that business operations, processes, and practices are in accordance with prescriptive (often legal) documents"*. From this definition, we can entail two domains of BPC. First, there is a *business process domain*, which comprises all aspects related to the analysis of process models and their execution. Second, there is a *(legal) requirements domain*, which comprises all aspects related to the actual internal or external regulations, which are imposed on the business process domain. It becomes necessary to capture and represent these (legal) requirements in such a way that the business process domain can be evaluated against the regulatory requirements. Here, *business rules* – which are essentially declarative constraints – have evolved as the central object of study used for such a formalization of requirements and to consequently verify process models and company activities against these requirements [58, 102]. Note that this is sometimes also referred to as the business rules approach [102], explained as follows.

1.2.3 Business Rules and their Relation to BPC

▷ *BPM* ▷ *BPC* ▷ *Business Rules*

Following Graham (2007)[p. 7], a business rule is a *"declarative statement about an aspect of a business"*, which specifies obligations, permissions and restrictions that constrain how company activities should be performed. Business rules are usually divided into *structural business rules* (which describe constraints in data/structures), and *behavioral business rules* (which describe how company activities should be conducted) [58, 80, 138][4]. In this work, we focus on behavioral business rules, as they directly affect company behavior and are the main object of study within BPC.

Following [58, 106] behavioral rules are of the general form

$$if\ \ A_1, ..., A_n\ \ then\ \ B \tag{1.1}$$

where $A_1, ..., A_n$ represents the premise of the rule (condition), and the conclusion B can be entailed, if the premise holds. This representation of conditions and behavioral conclusions allows to model business rules as a basis for BPC.

Regarding the different compliance strategies of design-time-, run-time-, and post-execution compliance, business rules play different roles in regard to BPC.

For design-time compliance management, the compliance of process models can be verified against a set of business rules. Identifying erroneously modelled parts in a process model is important for companies, as otherwise it can be expected that non-compliant (sequences of) activities will be performed later at run-time following the erroneous model. Figure 1.3 shows an example for design-time compliance management. For the sake of the example, we assume that the shown business rule can be understood s.t. every task A must be directly followed by a task B. Accordingly, model checking or model query can be used to verify that the modelled sequences of activities violates the (external) regulation, cf. [96] for an overview.

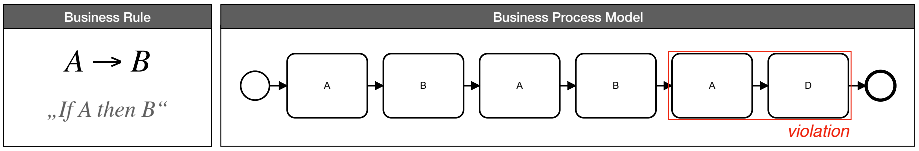


Figure 1.3: Example for design-time BPC using business rules.

In run-time compliance management, business rules can be used to govern compliant process execution. That is, case-dependent information is validated against the set of business rules for decision-making, i.e., to determine a route through the process model for individual process instances. As an example,

---

[4] We acknowledge that Weiden (2002) also identifies the class of managerial business rules, however these are not related to business process compliance but rather to strategic company aspects beyond the scope of BPC (e.g. expected added-values of processes).

consider the exemplary rule base shown in Figure 1.4, with the intuitive meaning that 1.) entities who have a mental condition are not contractually capable, and 2.) entities who are not contractually capable are not creditworthy. For a running process instance, if one assumes a customer referenced in the current case has a mental condition, the rules can be used to entail that this entity is also not creditworthy.

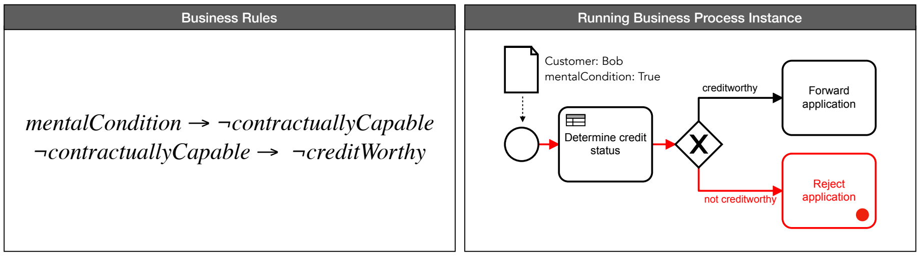
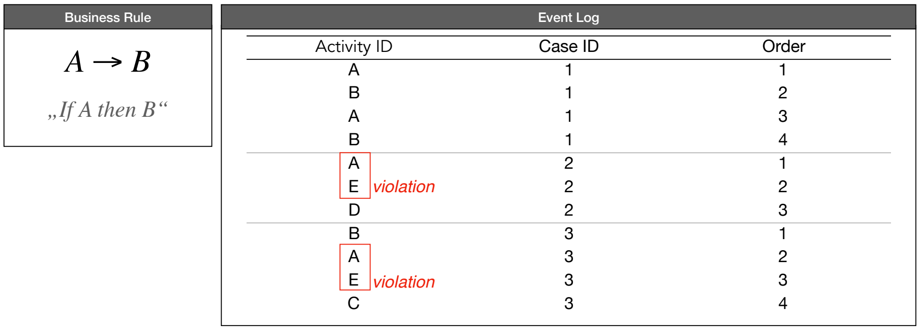


Figure 1.4: Example for ensuring compliant process execution during run-time.

Last, in post-execution compliance, the behavior recorded in event logs can be verified for compliance by checking the conformance of observed behavior to business rules. Figure 1.5 shows an example of post-execution auditing. As before, the depicted business rule states that every task A be should be directly followed by task B. As can be seen, there were two cases where this rule was violated. Here, the relations of cases can also be considered, e.g., in the example it appears the violations occured in cases where task E was executed.



Business Rule

$A \rightarrow B$

„If A then B“

Event Log

Activity ID	Case ID	Order
A	1	1
B	1	2
A	1	3
B	1	4
A	2	1
E *violation*	2	2
D	2	3
B	3	1
A	3	2
E *violation*	3	3
C	3	4

Figure 1.5: Example for post-execution auditing using business rules.

Regardless of the specific BPC strategy, we can observe that business rules are needed as a respective basis. Therefore, it is necessary to capture and maintain a set of business rules as a prerequisite for BPC. This is not a trivial task, as companies need to identify relevant laws and regulations, derive business rules from these regulations and store these rules such that company systems can access them in the scope of BPC. To conquer these important challenges, the discipline of Business Rules Management has emerged.

1.2.4 Business Rules Management

▷ *BPM* ▷ *BPC* ▷ *Business Rules* ▷ *Business Rule Management*

Business rules management (BRM) is geared towards the creation and maintenance of business rule repositories [58]. In essence, business rule management can be defined as *"a systematic and controlled approach [...] to support the elicitation, specification, verification, validation, deployment, execution, governance and monitoring of both business decisions and business logic"* [80][p.2]. Adapted from Nelson (2014), typical business rule management lifecycles usually contain the steps of strategic alignment, rule management and implementation, as shown in Figure 1.6. In this thesis, we focus on the rule management aspect, which contains the phases of *capturing*, *authoring* and *organizing*.

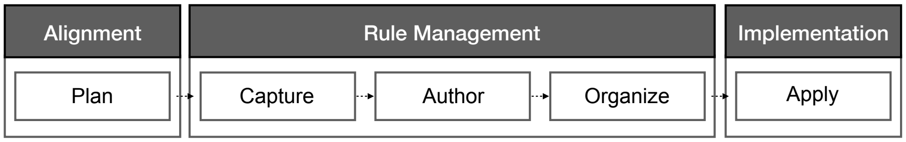


Figure 1.6: Business Rule Management lifecycle, adapted from [92].

In rule capturing, relevant business rules need to be identified. This is often performed by legal experts or using results from rule mining [91, 110]. Here, companies need to ensure they identify all relevant regulations that affect their domain. Note that this phase is also often referred to as rule elicitation in literature [92].

Then, identified rules need to be formalized in the authoring phase, i.e., representing the identified rules in standards and rule languages [110]. Authoring is usually a manual, collaborative and incremental process [91, 106]. In such a setting, modellers might make mistakes, or model redundant rules due to a lack of oversight [11,29]. Worse, modellers with different understandings on the same domain of interest might model business rules in a contradictory manner [120]. Thus, the authored rules must be assessed, in order to ensure error-freeness within the set of business rules. This is performed in the organizing phase.

Definition 3 (Business Rule Organizing)**.** In this work, we define business rule organizing (BRO) as understanding, grouping and selecting authored rules, with the goal to warrant error-freeness within the set of business rules.

Note that rule organizing as defined in this work is also sometimes referred to as verification in academia [111, 119, 120].

From the perspective of BPC, business rule organizing is a mandatory prerequisite. Otherwise, given errors within the modelled business rules, it may not be possible to conduct BPC efforts correctly, or even not at all. Yet, in many traditional BRM lifecycle models, rule organizing is described simply as a task that requires *"extensive manual intervention and analysis"* [91][p.5]. In recent years, academia has however recognized the difficulties in manual business rule organizing. In an agenda-setting contribution by SADIQ in 2011, that author states the need for approaches to provide *"organizational capacity to manage [the] compliance knowledge base"* [108][p.3]. This has also been affirmed by following research, for example the empirical evidence in [9] or the interviews with key practitioners conducted in [118,120]. For the latter, the practitioners in those reports unanimously stated that they are experiencing problems in actually implementing the rule organization phase, as a manual analysis was perceived as unfeasible.

As a main point of emphasis and motivation for this thesis, there is a broad consensus that rule organizing is a challenging task that is often too complex for manual analysis, and thus approaches are needed to support companies in ensuring error-freeness within business rules [9,11,21,42,81,106,108]. To provide a better understanding of challenges in business rule organizing, we subsequently discuss current challenges.

1.2.5 Challenges in Business Rule Organizing

▷ *BPM* ▷ *BPC* ▷ *Business Rules* ▷ *Business Rule Management* ▷ *Business Rule Organizing*

Following the taxonomy on business rule organizing approaches in [32], we can identify the three general challenges of a *detection* of errors, an *assessment* of errors and the *resolution* of errors.

A central challenge in BRO is the detection of errors within a set of business rules. This requires knowledge of the error types that can occur in rule bases. Regarding "which" errors can occur, different classifications exist that try to define error types in business rules, e.g. [64, 120, 135]. In our literature review in [32], we therefore proposed to generally group error types in business rules and the corresponding detection capabilities into two main groups, namely *simplification-related* error types and *inconsistency-related* error types.

Simplification-related errors refer to multiple business rules that should be merged, or reduced. For example, two identical rules can usually be merged. Hence, for simplification approaches, resolution of the error is trivial, or at least undisputed (e.g. in the example it is clear that one can simply delete one of two duplicate rules).

Inconsistency-related errors refer to multiple business rules with contradictory conclusions, such as the conclusions *creditWorthy* and *not creditWorthy* for the same customer. Here, handling such errors is not trivial. It is not clear how to resolve such an issue, as two contradictory pieces of information exist, which requires careful analysis by experts.

Moreover, inconsistency in business rules is not only more difficult to resolve, but also has more severe effects on business process compliance. For example, considering again a case with *identical rules*, two identical rules do not necessarily impose a problem with regard to business process compliance (e.g., the worst case is that a compliance check would be conducted twice). On the other hand, in case of contradictory rules, the inconsistency makes it *impossible* to use the business rules for their intended purpose of governing compliant business process execution. While simplification should not be neglected, handling inconsistencies in business rules is an important challenge to address for companies.

Continuing with business rule organizing, a further important challenge is the (quantitative) analysis of the detected errors [4, 81, 106]. Such an analysis can be presented to modellers in order to a) assess the *severity* of the detected errors, and b) provide a *prioritization* in which order rules should be attended to in the scope of re-modelling. Recent studies conducted for this thesis show that quantitative insights are associated with better understanding accuracy, better understanding efficiency and less mental effort needed for understanding problems in the scope of business rule organizing [33, 89].

Last, rule organizing also requires means for the actual resolution of the detected (and analyzed) problems. This can range from semi-automated resolution by means of recommendation systems to fully automated resolution algorithms.

To summarize, handling inconsistency would ideally require means to detect, analyze and resolve inconsistencies in business rules. Also, this should ideally be possible from multiple BPC perspectives. However, from a literature review conducted in the scope of this thesis [32], we see that there is currently no approach satisfying these requirements, illustrated in the following.

1.2.6 Inconsistency Handling in BRM

▷ BPM ▷ BPC ▷ Business Rules ▷ Business Rule Management ▷ Business Rule Organizing ▷ Inconsistency Handling in BRM

Table 1.2 shows an excerpt of our literature review on business rule organizing approaches in [32], namely a classification of those approaches identified that are geared towards *inconsistency-related errors in business rules*. We refer the reader to [32] for an in-depth discussion of the research method for the mentioned literature review.

A recent approach here is that by Di Ciccio et al. (2017). Those authors utilize automata representations of business rules. In [42], inconsistency is defined as a set of rules that cannot be satisfied, e.g., there cannot exist a (future) sequence of activities that satisfies the set of constraints – hence the automata product is empty. The approach by those authors allows to *ensure* the consistency of business rules. To this aim, a new business rule base is created by incrementally adding rules from the original rule base. Before ”transferring” a rule, it is tested whether adding that rule would render the rule base inconsistent – in which case the rule is dropped and not added. Intuitively, the rule order has an impact on which rules will be accepted ultimately. While those authors propose various metrics to select a smart order, that approach is not able to compute or consider all existing contradictions between business rules. This would however be needed to allow domain experts to analyze the contradictions and influence which rules should be kept.

	BPC Strategy			**Capabilities**			**Applicability**	
Literature	Design-Time Compliance	Run-time Compliance	Post-Execution Compliance	Detection	Quantitative Analysis	Resolution	Rule Formalism	Tool
Kardasis et al. (2004)	X						Text	
Hicks (2007)	X			X			Text	X
Cheng et al. (2009)	X			X			n/a	
Maggi et al. (2011)		X		X	X		LTL	
Maggi et al. (2011)		X		X	X		LTL	
Decker et al. (2013)	X			X	X		FOL	
Berstell-Silva et al. (2014)	X			X		A	FOL	
Cuzzocrea et al. (2014)	X			X	X		FOL	
Santos et al. (2014)	X			X		A	SBVR	X
Zhang al. (2014)	X			X		A	n/a	X
Cemus et al. (2015)	X						n/a	
Gomez et al. (2015)		X		X			n/a	
Agli et al. (2016)	X			X		A	ILOG	X
De Smedt et al. (2016)	X			X			LTL	
Di Ciccio et al. (2016)	X			X		A	LTL	X
Gomez et al. (2016)		X		X			n/a	X
Houari et al. (2016)	X			X			n/a	X
Di Ciccio et al. (2017)	X			X		A	LTL	X
Anand et al. (2018)	X			X			SBVR	X
De Smedt et al. (2018)	X			X			LTL	

Resolution: A (Automated).
Rule Formalisms: Text (textual description/natural language), n/a (Non-standard or individual rule formalisms, if-then-structures), FOL (First-order logic), FCL (Formal Contract Language), LTL (Linear Temporal Logic, resp. Declare), SBVR (Semantics of Business Vocabulary and Business Rules), Ilog (IBM Ilog).

Table 1.2: Overview of inconsistency-related approaches for Business Rule Organizing, classified into the dimensions of BPC Strategy, Capabilities and Applicability.

Next to this design-time approach, there are works geared towards detecting contradictions during run-time, e.g. inconsistencies in fact values during run-time [53], or contradictions in business rules relative to observed (unexpected) behavior [84, 85]. Here, a common baseline is to monitor running process instances and throw an error, should contradictory constraints be activated. While this is positive and can help to mitigate compliance breaches, this however does not help companies to understand the errors and resolve them, such that such an issue does not occur again. Again, methods are needed to detect all contradictions at design-time, such that an inconsistent rule set is not deployed in the first place. On that note, we would however like to emphasize that almost all works on run-time approaches provide some examples where run-time errors occurred that could not be detected at design-time, thus design-time compliance does not subsume run-time compliance, and both strategies should be implemented ideally.

Last, we can observe that there exist no approaches supporting business rule organizing during post-execution compliance. Due to the increased availability of event logs [1, 2, 18], methods for a posteriori business rule organization would be desirable, as considering the relations of different process instances could reveal even new insights that cannot be understood from a local run-time perspective. Also, approaches that cover multiple/all compliance management phases would be highly desirable following [63, 105].

We acknowledge that the approach by DE SMEDT ET AL. [114, 115] also analyzes interrelations between all business rules at design-time, based on their notion of *hidden dependencies.* Those authors study hidden relations of activities that can block each other in case of certain sequences of activities. However, this does not relate to inconsistencies between business rules, but rather on specific orderings in which rules may be activated. We also acknowledge that the approach by CALVANESE ET AL. [20,21] is able to detect deviations between rule conclusions (i.e., *different*, but not necessarily inconsistent conclusions). To clarify, in the use-case of those authors, it can be correct that multiple rules with *different* (but not contradictory) conclusions are activated at the same time, i.e., this is a different type of problem. Also, it is noteworthy that those authors analyze the rule base not as a whole but in partitions. However, as there can exist redundancies in rule bases, the rule base should be analyzed as a whole unit, and not partitioned, which cannot be performed with the approach in [21], cf. [11, 27, 29] for a related discussion.

We acknowledge that there are also inconsistency handling approaches beyond the scope of the organizing phase. First, there is a line of approaches that aim to facilitate reasoning in the presence of contradictory business rules, e.g., by using results from inconsistency-tolerant reasoning or reasoning under uncertainty [3, 24, 37, 39]. However, following [11], storing contradictory business rules impedes business rule management: Contradictory rules are confusing to modellers and have no added-value for the company. Unsurprisingly, qualitative research such as [117, 120] reveals that companies share this viewpoint and explicitly state the need for support in detecting and resolving errors in business rules (as opposed to means for helping to live with the errors). Second, we acknowledge that there are hybrid approaches that aim to connect rule authoring and rule organizing [67, 119]. The idea here is that procedure models on *how to model business rules* are proposed, such that inconsistencies are proactively counteracted. However, while this could be a good addition, based on the evidence in [9, 117, 118, 120, 121], it seems that companies are having trouble to implement such procedure models. Again, this advocates the need for rule organizing after the rule authoring phase.

Interim Result 1. From our literature review, we can observe that there currently exists no approach that can be used to compute all (sub)sets of contradictory business rules at design-time. There are also no approaches that support all three compliance management perspectives.

Continuing, research on *analyzing* inconsistencies in business rules is sparse and still at an early stage, despite clear suggestions in related works that quantitative error assessments are an important driver for re-modelling [81, 106, 139]. Here, quantitative assessments can contribute towards helping companies understand their errors, such that process- and decision-logic can be improved. Experiments conducted in the scope of this thesis show that providing quantitative assessments creates positive cognitive effects in understanding errors, i.e., better understanding efficiency, better understanding accuracy, and less mental effort needed to understand inconsistencies [89]. Further research on analyzing inconsistencies could therefore promote design-time compliance management and compliant-by-design business processes, as well as speed-up understanding and prioritizing errors that occurred during run-time.

Interim Result 2. Despite strong advocation in the literature, we can observe that there are currently no sufficient means for quantitatively assessing inconsistencies in business rules.

To conclude, there is a need for approaches to detect and analyze inconsistencies in business rules, especially for design-time compliance management. Especially the latter aspect is in line with the results in [4], who present a meta-analysis of the artifacts addressed in scientific works on BPC and explicitly point out the need for more research on metrics. A scientific field which is concerned with such an analysis and resolution of inconsistencies in knowledge representation formalisms is that of *inconsistency measurement*. Due to this conceptual alignment, we propose to develop new means for handling inconsistency in the scope of business rule organizing, while reusing and extending results from the scientific field of inconsistency measurement.

1.2.7 Inconsistency Measurement

▷ *BPM* ▷ *BPC* ▷ *Business Rules* ▷ *Business Rule Management* ▷ *Business Rule Organizing* ▷ *Inconsistency Handling in BRM* ▷ *Inconsistency Measurement*

The scientific field of *inconsistency measurement* – cf. [61, 130] for an overview – studies means to quantify the severity of inconsistency in knowledge bases. The main objects of study are *inconsistency measures*, which are functions that assign a numerical value to a knowledge base – i. e. a set of logical formulas or rules and facts that describe a certain domain of interest – with the intuitive meaning that a larger value indicates a larger "degree" of inconsistency.

The traditional setting for inconsistency measurement is that of classical propositional logic knowledge bases, i.e., a set of formulas $\mathcal{K}$ built on top a a set of propositional atoms $\mathcal{A}$ and the connectives $\neg$ (negation), $\vee$ (disjunction), $\wedge$ (conjunction) and $\rightarrow$ (implication).

For example, given the set of atoms $\mathcal{A}_0 = \{\mathit{platinumCustomer}, \mathit{mentalCondition}, \mathit{creditWorthy}, \mathit{contractuallyCapable}\}$, consider the following exemplary knowledge base $\mathcal{K}_1 = \{\phi_1, \phi_2, \phi_3\}$, where

$$\begin{aligned}
\phi_1 &= \mathit{platinumCustomer} \rightarrow \mathit{creditWorthy}\\
\phi_2 &= \mathit{mentalCondition} \rightarrow \neg \mathit{contractuallyCapable}\\
\phi_3 &= \neg \mathit{contractuallyCapable} \rightarrow \neg \mathit{creditWorthy},
\end{aligned}$$

with the intuitive meaning that 1.) platinum customers are generally creditworthy, 2.) customers who have a mental condition are not contractually capable, and 3.) customers who are not contractually capable are not creditworthy.

Informally, a knowledge base is *inconsistent* if one can derive contradictory formulas. For example, consider the knowledge base $\mathcal{K}_1' = \{\phi_1, \phi_2, \phi_3, \phi_4, \phi_5\}$, with

$$\phi_4 = \mathit{platinumCustomer} \qquad \phi_5 = \mathit{mentalCondition}$$

which contains the same formulas as $\mathcal{K}_1$, but also two additional formulas, with the intuitive meaning that we have a platinum customer who also has a mental condition. The knowledge base $\mathcal{K}_1'$ is consequently inconsistent as it is possible to derive both *creditWorthy* (via ϕ_4, ϕ_1) and $\neg$*creditWorthy* (via ϕ_5, ϕ_2, ϕ_3).

Instead of viewing inconsistency as a binary concept, the field of inconsistency measurement studies quantitative measures to assess the severity of inconsistency for a knowledge base $\mathcal{K}$.

In the following, let $\mathbb{R}^{\infty}_{\geq 0}$ be the set of non-negative real values including ∞, and $\mathbb{K}$ be the universe of all knowledge bases.

Definition 4. An inconsistency measure $\mathcal{I}$ is a function $\mathcal{I} : \mathbb{K} \rightarrow \mathbb{R}^{\infty}_{\geq 0}$ that assigns a non-negative real value to a knowledge base [130].

The intuition is that a larger value indicates a larger inconsistency in $\mathcal{K}$. For the remainder of the thesis, we denote $\mathcal{I}(\mathcal{K})$ as the degree of inconsistency of $\mathcal{K}$ (wrt. $\mathcal{I}$). As the concept of a "severity" of inconsistency is not easily characterizable, there have been different proposals as to which aspects should be considered for the quantification of inconsistency, see e. g. [124, 130] for an overview. An exemplary approach to quantify inconsistency is the MI-inconsistency measure $\mathcal{I}_{\mathsf{MI}}(K)$, which considers minimal inconsistent subsets in a knowledge base K. $M \subseteq K$ is a minimal inconsistent subset, if it is inconsistent and minimal in terms of set inclusion, i.e., removing any element resolves the inconsistency. For example, in K_1', there exists only one minimal inconsistent subset $M_1 = \{\phi_1, \phi_2, \phi_3, \phi_4, \phi_5\}$, as M_1 is inconsistent and there exists no $M' \subset M_1$ s.t. M' is inconsistent as well. Thus, considering again the MI-inconsistency measure, which counts the number of minimal inconsistent subsets, we have that $\mathcal{I}_{\mathsf{MI}}(K) = 1$. As mentioned, there have been a lot of different proposals as to how inconsistency can be quantified, and we will revisit central approaches in the course of this work.

Next to measures that assess the severity of inconsistency for the *entire* knowledge base, the field of inconsistency measurement also studies so-called *inconsistency values* (or *culpability values*), which are, in substance, element-based measures. Let $\mathbb{F}$ denote the universe of all possible formulas.

Definition 5. A culpability measure $\mathcal{C}$ is defined as a function $\mathcal{C} : \mathbb{K} \times \mathbb{F} \rightarrow [0, \infty)$ that assigns a non-negative real value to individual formulas of a knowledge base.

The intuition is that a higher value represents a higher blame that the formula carries in the overall context of inconsistency. Culpability measures thus provide quantitative insight that can help companies to understand and resolve problems in their rule bases [81, 106]. An exemplary culpability measure is the $\mathcal{C}_{\#}$ measure [70], that counts the number of minimal inconsistent subsets that an individual formula is contained in. Hence, this is essentially a scoring function that quantifies how many minimal inconsistent subsets can be resolved, if a formula is deleted. As with inconsistency measures, there have been numerous proposals for culpability measures, see e. g. [87] for an overview. As with inconsistency measures, we revisit central approaches in the course of this work.

To summarize, the field of inconsistency measurement studies means that can be used to quantitatively assess the degree of inconsistency for (individual elements of) knowledge bases. Such means could be very useful for inconsistency handling in business rule organizing, e.g., for assessing the severity of inconsistency, pin-pointing individual business rules which are responsible for the overall inconsistency, and providing a basis for an informed decision on determining resolution strategies. However, an integration of these two fields has not yet been investigated, despite its described potential and demand in both academia and practice. We therefore envisage to develop new means for handling inconsistency in business rules by adapting and extending results from inconsistency measurement. In the following, we confine the research scope and show the corresponding research gap.

1.3 Scope and Research Gap

Based on the previous discussion of the status quo, the scope of this thesis, shown in Figure 1.7[5], is defined as follows:

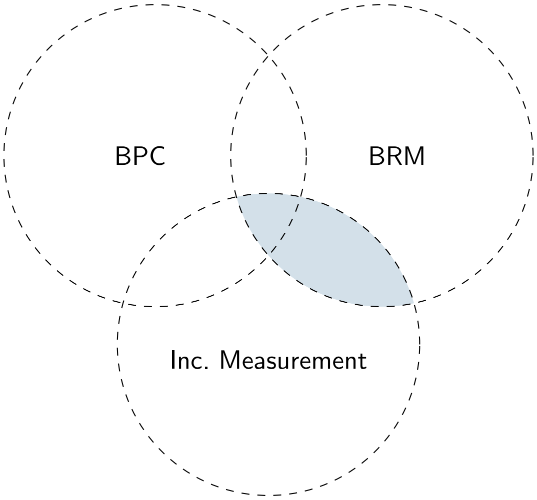


Figure 1.7: Research Scope (marked in color).

---

[5] The overlapping areas in this Figure are just for illustrative purposes and in no way mean to depict the actual percentage of overlaps of the shown scientific fields.

Within BPC, we see that business rules are needed as a basis for BPC. This raises the need to create and maintain business rules, which is handled in the scope of BRM. Within BRM, rule organization, i.e., ensuring error-freeness within a set of business rules, is an important challenge. Then, within rule organizing, handling inconsistent business rules is a current challenge for companies. In this scope, our research aim is to develop new means to handle inconsistency in business rule bases. Here, we can identify the following research gaps.

First, while there is a broad consensus that modelling errors often occur in practice and companies need to be supported in detecting these errors, there are currently no means to detect all subsets of contradicting business rules. Second, despite a strong advocation for quantitative measures, there currently exist no means to quantify inconsistencies in business rules. Such a quantification could however provide valuable insights for companies, e.g. could serve as a basis for an informed decision in the scope of inconsistency resolution.

Given the similarity of propositional rules and the general business rule form shown in (1.1), adapting results from inconsistency measurement might seem intuitive. However, due to some conceptual mismatches, a straightforward application of inconsistency measurement is not plausible, explained as follows.

In the classical setting of inconsistency measurement, the formulas in a knowledge base do not have a distinguishable level of granularity. However, business rule bases are divided into *facts* and *rules*. A separation between facts and rules however has strong implications, the simplest being that it may not be plausible to delete facts from a rule base, e.g. in the scope of inconsistency resolution. For example, considering again the example in $\mathcal{K}_1'$, one cannot change the mental condition of a customer, just to make the set of business rules consistent. As inconsistency measures cannot differentiate between facts and rules, current measures cannot be plausibly applied to quantify inconsistencies in business rule bases, which raises the need for an adaptation and extension of measures.

Furthermore, in the context of BRM, the question arises whether facts in business rule bases can be assumed to be known at all. For instance, companies need to determine at design-time, whether their set of business rules is consistent. No instance facts are available here, i.e. it is not known which rules might eventually be activated. For example, recall rule base $\mathcal{B}_1$, defined via

$$\begin{aligned}\mathcal{B}_1 = \{&newCustomer \rightarrow creditWorthy,\\ &newCustomer \rightarrow \neg creditWorthy\}.\end{aligned}$$

This rule base is, in a sense, "contradictory", and it makes no sense for companies to store both rules at the same time. Yet, it is not possible to infer information about actual inconsistencies for $\mathcal{B}_1$ as no conclusions can be made without information about activation, i.e. facts. Indeed, virtually all inconsistency measures, such as the introduced $\mathcal{I}_{\mathsf{MI}}$, thus return a value of 0, indicating that the rule base is logically consistent. In the scope of BRM, inconsistency measures should however ideally be able to assess which rules are modelled in a contradictory way. This motivates the need to develop new measures, which allow to analyze inconsistency on a rule-level, i.e. independently of facts.

Finally, the question has to be asked how the existing results could actually help companies and experts. That is, the background and skill-set of experts in BPC might be far different from that of experts in the field of inconsistency measurement, thus the plausibility of utilizing results from inconsistency measurement "in practice" needs to be addressed.

The research gaps can thus be summarized as follows:

- From a business rule management perspective, there is a need for methods to detect, analyze and resolve inconsistencies in business rules.
- Applying results from inconsistency measurement for this use-case seems promising, however a straightforward application is not plausible due to conceptual mismatches.
- New means have to be developed to fit certain requirements in design-time-, run-time-, and post-execution compliance management.
- The plausibility of applying concepts from the field of inconsistency measurement to BRM in general must be carefully investigated.

On a last note, a further problem impeding a straightforward application of inconsistency measurement results is that it is highly doubtful that business rules will be formalized with propositional logic (which is as mentioned the traditional setting for inconsistency measurement). Rather, business rule formalisms have evolved. To promote the adaptation of our results in enterprise contexts, it is therefore important to align our results with important rule standards. To this aim, we propose to refer to recent literature reviews in the area of business rules and to consider the rule standards supported by the approaches identified in those reviews. Based on our own literature review [32], we can identify three seminal works, namely [63, 71, 105]. These are to the best of our knowledge the most recent literature reviews on business rules in general [71], and business process compliance approaches in general [63,105], and are therefore considered to select commonly addressed rule standards[6].

Based on the identified reviews [32,63,71,105], rule standards that are commonly addressed are the *Formal Contract Language (FCL)* (cf. e.g. [56]), *Temporal Logic (Declare)* (cf. e.g. [42]) and *Decision Model and Notation (DMN)* (cf. e.g. [21]). As a design decision, we will therefore consider these three rule standards in this work, such that our results are available for these commonly addressed rule standards[7].

---

[6] We acknowledge that these literature reviews do not provide empirical evidence for any distribution of supported standards in practice, however, many of the works identified in the reviews include case-studies or analyze standards and tools, thus we argue these reviews are a reasonable basis to identify important business rule standards.

[7] Please note that our results are in no way limited to these three rule standards. All our results are discussed based on the general form shown in 1.1. Thus, it is possible to map our results to any rule standard based on this general form of behavioral business rules. An exemplary mapping from FCL to a different rule standard has for example already be shown in a Master thesis [99], supervised by myself and Patrick Delfmann.

1.4 Research Questions and Contributions

Based on the identified research gaps, we consequently raise the following main research question.

RQ: How can inconsistencies in business rule bases be detected, analyzed and resolved in the scope of Business Rule Management?

This research question is aligned with our main research aim to support business rules management in handling inconsistent business rules. To this aim, we also raise the following subsidiary research questions:

SRQ1: How must current results from inconsistency measurement be extended to make them applicable in BRM?

In the previous section, we motivated that current results from inconsistency measurement cannot be plausibly applied within BRM. We consequently investigate how the notion of inconsistency needs to be extended in a BRM setting. This however has implications on how the "severity" of inconsistency can be defined. Subsequently, we raise the following research question:

SRQ2: How can inconsistency be measured in business rules?

We investigate how to adapt current means to fit the intended use-case and analyze formal properties of our proposed results. Here, following calls from literature [81,106,108,120], we also investigate how such quantitative insights can be exploited, e.g. for re-modelling business rules and improving operations:

SRQ3: How can quantitative measures be used to serve as a basis for inconsistency resolution?

We investigate element-based inconsistency measures (also referred to as culpability measures) and how these measures can be exploited for inconsistency resolution, e.g. in the form of recommendations. In this thesis, a strong focus is also set on the evaluation of the developed means:

SRQ4: How feasible is the application of inconsistency analysis in business rule bases, and what is the added-value for BRM?

We evaluate the feasibility of implementing our proposed means and investigate whether the developed means can actually support human modellers in BRO via experiments with human participants. Last, the question arises how our results, concerning different BPC phases, can be combined into a unified framework:

SRQ5: How can a BRO approach supporting multiple BPC strategies by the proposed means be created?

As recent scientific literature strongly implies advantages of implementing multiple BPC strategies, we discuss how our results can be combined in a multi-phase framework for handling inconsistencies in the context of BRO.

Given our research question(s), the contributions of this thesis are consequently as follows.

- **C1. Novel means to detect inconsistency in business rules at design-time.** We show that the current definition of inconsistency is not applicable for all use-cases in business rule organization. To detect inconsistent business rules at design-time, we introduce the novel notion of quasi-inconsistency, which describes rules that will always be activated together, but have logically contradictory outcomes.
- **C2. Measuring Inconsistency in Business Rule Bases.** We propose new rationality postulates for inconsistency measures applied to a business rule context, i.e. properties that need to be met by measures for plausible application in a business rule context. We then survey a selection of existing inconsistency measures and show that these measures cannot be plausibly applied in business rule organizing. Then, we adapt these measures for design-time and run-time compliance. Allthough it is beyond the scope of this thesis, we also discuss initial ideas regarding novel measures for post-execution compliance.
- **C3. Resolving Inconsistency based on culpability measurement.** We investigate element-based inconsistency measures and propose an initial approach to resolve inconsistency based on culpability measurement.
- **C4. Algorithms and Library.** To support companies in rule organizing and allow for a seamless adaptation of our results, the proposed framework will not only be discussed formally, but will also be implemented for currently used rule standards. We have identified FCL, Declare and DMN as current important rule standards and consequently present algorithms and tools for handling inconsistencies in real-life business rule bases.
- **C5. Effects of quantitative measures in BRM.** To evaluate the plausibility of our proposed means, we conduct experiments with human participants. We show that our results help human modellers to better understand and handle inconsistencies, in a faster time, and with less mental effort needed.
- **C6. A framework for handling inconsistencies in business rule bases.** As there is currently no rule organizing approach supporting multiple BPC phases, we will not only investigate inconsistency handling for these phases individually, but will also discuss how our results can be combined to support multiple BPC phases.

2

Outline and Research Method

In general, this thesis is divided into two parts - *part I*, which summarizes the research belonging to it, and *part II*, which consists of the individual published works. On a high-level, part I is divided into the following chapters:

1. **Introduction.** Chapter 1 gives an overview of this thesis by providing a problem statement, the motivation of the research, the scope of the research as well as the research question.
2. **Research Design.** Chapter 2 presents the research design and the outline of the conducted research.
3. **Novel Concepts.** Chapter 3 discusses the research results and also provides a summarizing discussion.
4. **Evaluation.** Chapter 4 provides an evaluation of the research results.
5. **Summary and Conclusion.** Chapter 5 summarizes the research results and points out further points of interest for future work.

Then, on a high-level, part II is divided into the following chapters:

6. **Overview of Publications.** Chapter 6 provides an overview of the publications corresponding to this thesis and contextualizes the individual contributions in the scope of the dissertation regulations of the Faculty of Computer Science at the University of Koblenz-Landau.
7. **Published Contributions.** In chapter 7, all published works are presented in full.

Table 2.1 shows the published works corresponding to this thesis in descending chronological order and distinguished into conference and journal publications. We refer the reader back to the Disclaimer for details on the respective contributions to these works by the author.

ID	Title	Outlet	Ranking	Section(s)
[35]	Towards Inconsistency Measurement in Business Rule Bases	ECAI 2020	A	3.2.2, 3.3.1, 4.1
[33]	Effects of Visualization Techniques on Understanding Inconsistencies in Automated Decision-Making	HICSS 2020	A	4.4
[27]	A Tool for Decision-Logic Level Verification in DMN Decision Tables	BPM 2019 (Demo-Track)	A	3.5, 4.3.1
[31]	Quasi-Inconsistency in Declarative Process Models	BPM 2019 (Forum)	A	3.2.1, 3.4, 4.3.1
[28]	Resolving Inconsistencies in Declarative Process Models based on Culpability Measurement (**Winner of the Best-Paper award**)	WI 2019	C	3.3, 3.4.1, 4.3
[89]	Effects of Quantitative Measures on Understanding Inconsistencies in Business Rules	HICSS 2019	A	4.4
[29]	Supporting Business Rule Management with Inconsistency Analysis	BPM 2018 (Industry-Track)	A	1.2.5, 3.5
[30]	A Tool to Monitor Consistent Decision-Making in Business Process Execution	BPM 2018 (Demo-Track)	A	3.5, 4.3.1
[34]	On Quasi-Inconsistency and its Complexity	AIJ (**Journal**)	A*	3.1, 3.2.1, 4.1.1, 4.2
[32]	A Taxonomy of Business Rule Organizing Approaches	EMISA (**Journal**)	C	1, 3.5
[64]	Decision Model Change Patterns for Dynamic System Evolution	KAIS (**Journal**)	B	3.5, 4.3.1

Table 2.1: Overview of publications corresponding to this thesis.

Table 2.1 also shows the publication outlet, as well as the conference or journal ranking according to the CORE-ranking[1]. Due to the focus on computer science and information technology in this thesis, the CORE-ranking was selected as a design choice in correspondence with the supervisors of this thesis. A further discussion of these works in the context of this ranking, especially in the scope of the recommendations for cumulative dissertations of the Faculty of Computer Science at the University of Koblenz-Landau, is presented in Chapter 6.

In the following, we present methodological considerations and the logic of investigation of this thesis.

2.1 Methodological Considerations

The aim of this thesis is to develop new means for inconsistency handling in order to support companies in business rule management, and is therefore within the scope of business informatics [76]. Based on the classification of goals of the field of business informatics by Becker et al. (2003), we therefore identify the following four goals:

The focus of this thesis is a goal of design ("Gestaltungsziel"), which is man-

---

[1] `http://www.core.edu.au/`

ifested by the aim of developing methods and techniques for handling inconsistencies in business rule bases (Goal of design $D1$). Here, these techniques are also distilled into a holistic framework and implemented as tools (Goal of design $D2$). As subordinate goals to $D1$ and $D2$, this thesis also pursues goals of understanding (”Erkenntnisziel”). First, the status quo on inconsistency handling in business rules has to be identified to guide this research (Goal of understanding $U1$). Second, a major goal of understanding is an evaluation of the developed methods and techniques (Goal of understanding $U2$). Here, the evaluation is performed by means of formal analysis, feasibility studies and plausibility studies. The goals of design follow a methodic mission (cf. [12]), as they are not oriented towards an individual enterprise domain. While the evaluation also has a primary methodic mission, certain data-sets analyzed originate from specific domains, thus this goal in part also follows a functional mission. The evaluation and design are iteratively performed with the goal of incrementally improving the primary goals of design.

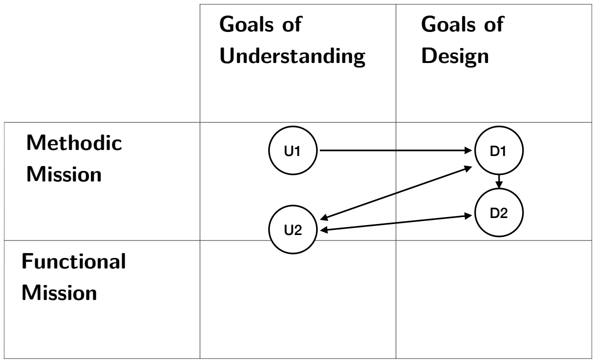


Figure 2.1: Goals of this work in the scope of the research goals in business informatics as defined in [12].

Due to the primary focus on goals of design, this thesis follows a design-science approach, which can be described as a construction-oriented research paradigm, with the primary focus on design and development of *artifacts* capable of solving specific and relevant problems[2] [8, 52, 65, 134].

Following [86], artifacts can be subdivided into *constructs, models, methods* and *instantiations*. Constructs describe the language and terminology used to formalize information, while models use this terminology to represent problems [141]. Methods describe a problem-solving process [141]. Finally, instantiations are aggregates of the above artifact types, e.g. an implemented tool using constructs and allowing to apply a method [141].

In this thesis, the focus lies on the development of constructs, methods and instantiations. Constructs, e.g. formal definitions of certain subset types in rule bases, will be specified as notation for inconsistency in rule bases. Then, these

---

[2] Note that the goals of understanding $U1$ and $U2$ are seamlessly integrated within this approach.

constructs will be used to develop new methods for handling inconsistency in rule bases, which are then implemented, i.e. instantiated. As a result of goal $U1$, a model of the status quo on rule organizing will also be developed in the form of a taxonomy, however this is seen as a basis for the primary contributions in the form of constructs, methods and instantiations.

Based on the suggestion by Hevner et al. (2004), the development of these artifacts should be conducted by applying suitable scientific research methods. Following the classification of business informatics research methods by Laudon et al. (2010), the methods applied in this work are *formal-conceputal analysis*, *prototyping* and *experiments*.

- **Formal-Conceputal Analysis.** Logic-deductive methods are applied on a mathematic-formal level (cf. Sections 3.2, 3.3.1) and on an argumentative level (cf. Sections 3.3.2, 3.4, 3.5) to develop constructs and methods. This is also applied for analytical evaluation of the developed artifacts (cf. Sections 4.1, 4.2).
- **Prototyping.** The developed methods are instantiated through prototyping (cf. Sections 3.3.2, 3.4.2, 3.5, Appendix A). Note that this method as defined in [76] also explicitly includes feasibility evaluation (cf. Section 4.3).
- **Experiments**. Controlled experiments with human participants are conducted to test the causal effects of the developed artifacts (cf. Section 4.4).

For a more fine-granular view, Table 2.2 provides an overview of the goals of the individual publications, the applied methods, as well as the created artifact types.

ID	**Research Method**	**Goals**				**Artifacts**
		D1	**D2**	**U1**	**U2**	
[35]	Formal-Conceputal Analysis	X		o	X	Constructs, Methods
[33]	Literature Analysis, Experiments			o	X	Constructs, Instantiation
[27]	Literature Analysis, Prototyping	X	X	o	o	Methods, Instantiation, Constructs
[31]	Literature Analysis, Formal-Conceputal Analysis, Prototyping	X	X		o	Methods, Instantiation, Constructs
[28]	Literature Analysis, Formal-Conceputal Analysis, Prototyping	X	o	o	X	Methods, Instantiation, Constructs
[89]	Literature Analysis, Experiments				X	Constructs, Instantiation
[29]	Literature Analysis, Formal-Conceputal Analysis	o		X		Constructs, Methods
[30]	Literature Analysis, Prototyping	X	X		o	Methods, Instantiation
[32]	Literature Analysis			X		Model, Constructs
[34]	Formal-Conceputal Analysis	X			X	Constructs, Methods
[64]	Literature Analysis, Prototyping	o	X	o		Methods, Instantiation, Constructs

X = Primary goal, o = Secondary goal.

Table 2.2: Research Methods, Research Goals and Artifacts addressed in the individual publications.

As a design choice, during creating these artifact types and while applying the design science research method we will follow the guidelines by Hevner et al. (2004) in order to warrant a high degree of scientific rigor based on these widely acknowledged principles. In particular, these guidelines concern the topicalities of *artifact creation, relevance, evaluation, contribution, rigor, design as a search process* and *communication of results*:

- **Artifact Creation.** The central outcome of design science research should be artifacts, e.g., as classified in [86]. As mentioned, the central outcomes of this thesis are artifacts in the form of constructs, methods and instantiations.
- **Relevance.** The research should address a relevant problem, where this relevance should be shown by the researcher. In Chapter 1 of this thesis, the current state of the art in business rule organizing was introduced to identify current research gaps through literature analysis (cf. also the literature review in [32]). Subsequently, the identified research gaps were used to derive research questions, aimed at solving these gaps. Based on our findings, the need for means to handle inconsistencies in business rule bases can be shown by literature analysis [32], based on recent case-studies from the field [9, 120], empirical evidence from evaluating real-life data sets (cf. Section 4.3), and based on the arguments presented in Section 1.3.
- **Evaluation.** The created artifacts should be evaluated after creation. In this work, feasibility is evaluated through proof-of-concept implementations, complexity analysis and run-time tests (cf. Sections 4.2, 4.3). Plausibility is evaluated through experiments with human participants (cf. Section 4.4). Also, as this work is an interdisciplinary work with roots in theoretical computer science, evaluation methods from that field such as formal analysis are a core focus (cf. Sections 4.1, 4.2). Please see Section 4.5 for a further meta-evaluation of the applied evaluation techniques.
- **Contribution.** The (scientific) contribution is to be shown by the researcher. Based on the literature review conducted for this thesis, there currently exists no means to address the research questions raised in this work, cf. also Section 1.3 (research gap) and Section 1.4 (Contributions). Also, virtually all publications corresponding to this thesis contain a related work analysis, where the contribution of the individual work in regard to existing results is described. In a way, the individual contributions and relevance of the publications (and therefore this work) were also ensured by the peer-review processes of the individual publication outlets.
- **Rigor.** The development and evaluation of artifacts should be conducted by applying suitable scientific research methods. Please see the above discussion for details on the applied research methods. Also, please see Section 4.5 for a meta-evaluation on the selected evaluation methods.
- **Design as a Search Process.** The creation process should be conducted incrementally. As mentioned, development and evaluation were also iteratively

cycled to ensure incremental improvement, following the suggestsions of [65, 134] (See also the following Section 2.2).

- **Communication.** The results should be (continuously) communicated. A focus here should be set on emphasizing both the contributions of results for science, as well as practice. This thesis is based on several works published at relevant outlets in the fields of business informatics (e.g. *The international conference on Business Process Management*), and artificial intelligence (e.g. *Artificial Intelligence* or *the European Conference on Artificial Intelligence*), cf. Table 2.1. This peer-reviewed dissemination during the writing of this thesis allowed to present intermediate results and get constructive feedback from experts in the field. Also, different paper types (e.g. research papers, works in industry-tracks, or works in demonstration-tracks) were communicated to emphasize different relevant aspects of this thesis.

2.2 Logic of Investigation

Regarding procedural approaches in Design Science research (DSR), there have been various proposals, cf. e.g. [65, 75, 95, 134]. A shared consensus however is that design science research entails new scientific insight through incremental design and construction in phases. Following Vaishnavi and Kuechler (2004), a typical DSR process typically includes the phases of *problem awareness, suggestion, development, evaluation* and *conclusion*, shown in Figure 2.2.

Figure 2.2: Procedural DSR approach by [134].

In this thesis, we will follow these DSR phases as proposed in [134], explained as follows. The first step is the identification and initial analysis of a problem in a domain of interest. In this thesis, this relates to the identification of problems in business rule organizing presented in Section 1.2.5, resp. 1.3. Vaishnavi and Kuechler (2004) then propose to suggest a tentative design, or tentative approach of how to solve the problem. In this thesis, this suggestion relates to the research aim of developing new means to conquer the identified problem. Next, the development and evaluation phase are of central importance. The focus in these phases is the creation of artifacts, cf. Section 3 (results). An important part of DSR is that the development and evaluation phases are performed iteratively and in turn [134]. On every iteration, the performance measures of the evaluation contribute towards knowledge and can be integrated to improve the developed artifacts. In this work, such an iterative cycle is warranted through the evaluation of the developed artifacts in the individual publications. Here, limitations could be identified to guide future development. Also, through presenting the intermediate works at scientific venues, feedback from experts could be attained through insightful discussions. The iterative cycles can be concluded if the designed artifacts meet the intended aim, cf. Section 5. [65,134]. The design science knowledge attained in this project is subsequently summarized in this thesis to conclude the applied research approach.

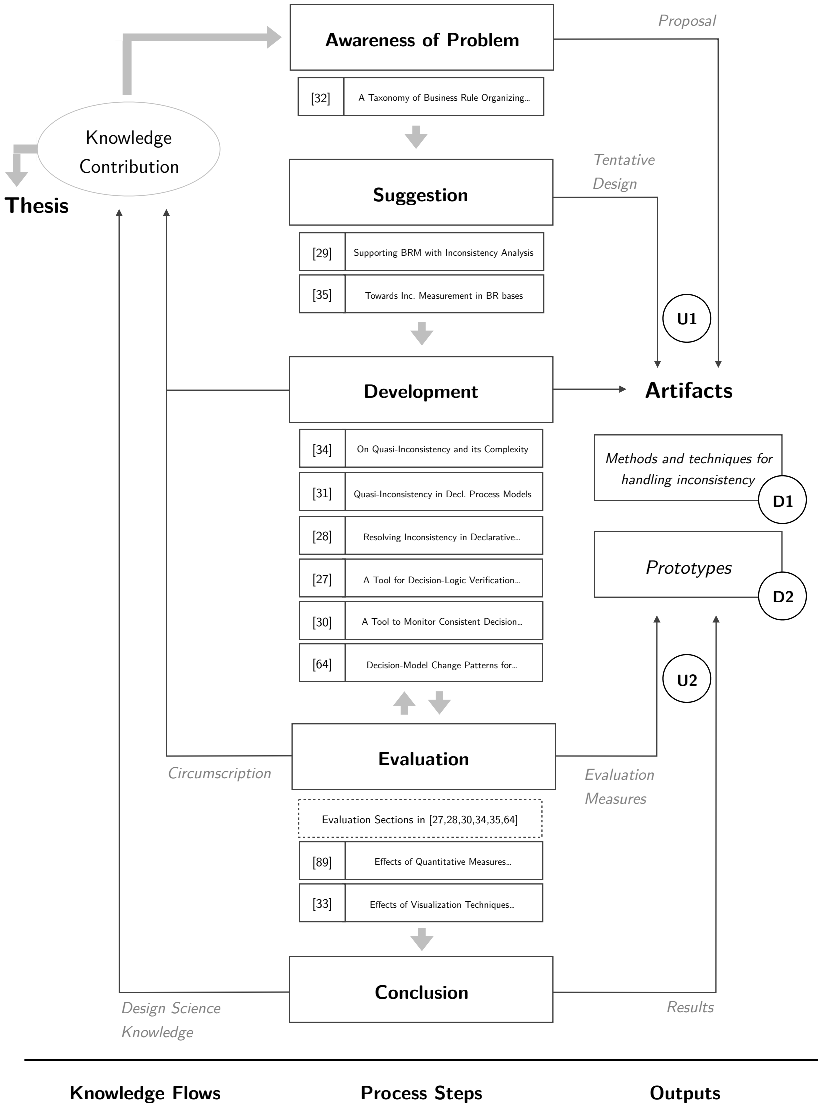


Figure 2.3: Logic of investigation, following the suggestion in [134].

Figure 2.3 shows the logic of investigation of this thesis, following the procedural approach as proposed in [134]. Figure 2.3 also provides an overview of the placement of the published works within the logic of investigation[3]. Due to space reasons, some titles are abbreviated – please see the respective reference number. We also refer the reader to Section 6 for further details on the interrelations of the published works.

---

[3] The presented order of works within the depicted logic of investigation does not fully correspond to the publication order. Following authors such as [95], this is however in no way a requirement, and can have many contextual reasons, such as long review times for journal articles.

3

Novel Concepts

This chapter presents the central research results, which relate to *methods and techniques for quantifying inconsistency in business rule bases, pin-pointing elements responsible for the overall inconsistency*, and *resolving inconsistency.* The results and their interrelations are also summarized in a concluding discussion. As stated in the disclaimer, the following results have already been published in various works. Subsequently, the following section includes many materials of these works, which are cited accordingly. All original material was written by myself, except if explicitly stated otherwise.

3.1 On the Notion of Inconsistency

> *"We know dragons do not exist, but we can still talk about them as much as we want. Just if you actually see one, then your knowledge base becomes inconsistent – but then you also have other problems."*
>
> – Matthias Thimm, *Semantic Web Lecture 2014*

▷ *This section corresponds to the works [31, 34, 35] by the author.*
▷ *The general definition of business rule bases used as a basis for this work is taken from [35] by the author.*

The notion of inconsistency is often quite ambiguous in academia. In this work, we consider classic-logical inconsistency, which usually refers to a knowledge base containing multiple pieces of information that cannot hold at the same time. In this section, we describe the formal framework used to represent business rules in this work, and define inconsistency from a viewpoint of BRM.

In order to formalise business rules bases, we rely on a simple (monotonic) logic programming language, cf. [50]. For that, we consider a finite set $\mathcal{A}$ of atoms. Let $\mathcal{L}$ be the corresponding set of literals, i. e., atoms and negations of atoms. We abbreviate $\overline{\neg a} = a$ and $\overline{a} = \neg a$ for an atom a.

Rules are of the form

$$r : \quad l_1, \ldots, l_m \to l_0. \tag{3.1}$$

with $l_0, \ldots, l_m \in \mathcal{L}$. We abbreviate $head(r) = l_0$ and $body(r) = \{l_1, \ldots, l_k\}$. If $body(r) = \emptyset$ we call r a *fact* and simply write l_0 instead of $\to l_0$. A *rule base* $\mathcal{B}$ is then a pair

$$\mathcal{B} = (\mathcal{F}, \mathcal{R}) \tag{3.2}$$

where $\mathcal{F}$ is a set of facts and $\mathcal{R}$ is a multiset of rules of the form in (3.1)[1]. Let $\mathbb{B}$ be the set of all possible rule bases. Note that the ";" sign is used sometimes in this thesis to separate rule base elements in order to avoid confusion with "," signs used inside rule bodies. For a rule base $\mathcal{B}$ let $\mathcal{F}(\mathcal{B})$ denote the facts in $\mathcal{B}$, resp. let $\mathcal{R}(\mathcal{B})$ denote the rules in $\mathcal{B}$.

Example 6. Consider the rule base $\mathcal{B}_3$, defined via

$$\begin{aligned}\mathcal{B}_3 = \{&customer, mentalCondition,\\ &mentalCondition \to \neg contractuallyCapable,\\ &\neg contractuallyCapable \to \neg creditWorthy\}.\end{aligned}$$

Then we have

$$\begin{aligned}\mathcal{F}(\mathcal{B}_3) &= \{mentalCondition, customer\}\\ \mathcal{R}(\mathcal{B}_3) &= \{mentalCondition \to \neg contractuallyCapable,\\ &\quad \neg contractuallyCapable \to \neg creditWorthy\}.\end{aligned}$$

For X being a rule base, or a set of literals, let $\mathcal{A}(X)$ denote the set of atoms appearing in X.

A set M of literals is *closed* wrt. $\mathcal{B}$ if for every rule of the form 3.1, if $l_1, \ldots, l_m \in M$ then $l_0 \in M$. The *minimal model* M of a rule base $\mathcal{B}$, denoted by $\min(\mathcal{B})$ is the smallest (wrt. set inclusion) closed set of literals. A set M of literals is called consistent if it does not contain both a and $\neg a$ for an atom a. A rule base $\mathcal{B}$ is called consistent if its minimal model is consistent. If the minimal model of $\mathcal{B}$ is not consistent, we say that $\mathcal{B}$ is inconsistent, denoted by $\mathcal{B} \models \perp$.

Example 7. Consider the rule base $\mathcal{B}_4$, defined via

$$\begin{aligned}\mathcal{B}_4 = \{&platinumCustomer,\\ &platinumCustomer \to contractuallyCapable,\\ &mentalCondition,\\ &mentalCondition \to \neg contractuallyCapable\}\end{aligned}$$

$\mathcal{B}_4$ is inconsistent in the classic-logical sense, as it entails the contradictory conclusions $contractuallyCapable$ and $\neg contractuallyCapable$.

---

[1] Note that we model rule bases as multi-sets in order to allow for duplicates of rules, which can also be the case in BRM (cf. also the discussion in [14]).

An important notion in this work is that of design-time compliance management. During this phase, business rules are initially modelled and stored in rule bases. Here, this modelling is conducted independently of facts, as (case-dependent) facts will only occur later during run-time. Consequently, during design-time, we have that $\mathcal{F}(\mathcal{B}) = \emptyset$ for a rule base $\mathcal{B}$.

To distinguish such rule bases, we say that a rule base $\mathcal{B}$ with an empty set of facts is the design-time view of $\mathcal{B}$, denoted with $\mathcal{B}^{\setminus\mathcal{F}}$. For readability, we will also refer to such rule bases as "design-time rule bases".

The *F-minimal model M* of a design-time rule base $\mathcal{B}^{\setminus\mathcal{F}}$, denoted by $\min_F(\mathcal{B}^{\setminus\mathcal{F}})$ is the smallest (in terms of set inclusion) closed set of facts (=literals) wrt. $\mathcal{B} \cup F$. A design-time rule base $\mathcal{B}^{\setminus\mathcal{F}}$ is called *F-consistent* if its F-minimal model is consistent. If not, we call $\mathcal{B}^{\setminus\mathcal{F}}$ *F-inconsistent*. In other words, a design-time rule base is *F-inconsistent*, if adding the set of facts F makes the resulting rule base inconsistent. We say that a rule base $\mathcal{B}^{\setminus\mathcal{F}}$ is *minimally F-inconsistent* if F is consistent, $\mathcal{B}^{\setminus\mathcal{F}}$ is *F-inconsistent* and for every $F' \subsetneq F$, $\mathcal{B}^{\setminus\mathcal{F}}$ is *F'-consistent*.

Example 8. Consider the design-time rule base $\mathcal{B}_5^{\setminus\mathcal{F}}$, defined via

$$\mathcal{B}_5^{\setminus\mathcal{F}} = \{a \to b; b \to c; a \to \neg c\}$$

Then we have that $\{a\}$ is a consistent set of facts, $\mathcal{B}_5^{\setminus\mathcal{F}}$ is $\{a\}$-inconsistent, and $\mathcal{B}_5^{\setminus\mathcal{F}}$ is $\{\}$-consistent , thus, $\mathcal{B}_5^{\setminus\mathcal{F}}$ is minimally $\{a\}$-inconsistent.

When considering design-time rule bases, we see that every design-time rule base is *consistent* in the classic-logical sense, i.e., $\emptyset$-consistent [34]. In the scope of design-time modelling, we are however interested if there exists a set of facts F, s.t. a rule base is F-inconsistent. That is, we are interested in cases where a rule base will become inconsistent, should certain facts be introduced. Addressing this problem is of high interest in industrial settings, as it cannot always be foreseen which combination of (case-dependent) facts will occur *"later"* during run-time. Here, it becomes important for companies to analyze whether there exist *potential* inconsistencies in the rule base, so experts can improve operations and counteract potential compliance breaches during process execution.

Example 9. Consider the design-time rule base $\mathcal{B}_6^{\setminus\mathcal{F}}$, defined via

$$\mathcal{B}_6^{\setminus\mathcal{F}} = \{a \to e; c \to \neg e; a \to c; b \to e\}$$

Observe that $\mathcal{B}_6^{\setminus\mathcal{F}}$ is minimally $\{a\}$-inconsistent and minimally $\{b, c\}$-inconsistent. However, the $\{a\}$-, resp. $\{b, c\}$-inconsistency, entail different types of problems:

The $\{b, c\}$-inconsistency says that whenever b and c are added to the rule base, we activate both $c \to \neg e$ and $b \to e$, which yields an inconsistent conclusion. However, there can of course be cases where only b, or only c occurs, thus it is possible to activate the rules individually and draw meaningful conclusions from the rule base. We therefore denote the $\{b, c\}$-inconsistency as a *potential issue*, as it is still possible that $c \to \neg e$ or $b \to e$ are activated independently.

On the other hand, the $\{a\}$-inconsistency says that whenever the rule $a \rightarrow e$ is activated, we automatically derive both e and $\neg e$. To clarify, the rules $a \rightarrow e$, and $a \rightarrow c$, $c \rightarrow \neg e$ can either not be activated at all, or, they will always be activated together but in this case yield inconsistent conclusions. Thus, these two rules cannot be used for any meaningful reasoning. We therefore denote the $\{a\}$-inconsistency as an *(actual) issue*.

Intuitively, it is important to resolve *(actual) issues* in the scope of design-time compliance management, as they clearly indicate a modelling error in the set of business rules. For that purpose, we will now introduce the notion of *quasi-inconsistency*. Informally, we say that a rule base is *quasi-inconsistent* if it contains rules that will always be activated together but yield inconsistent conclusions. For that we need some further notation.

Definition 10 (Rule Set activation, [34])**.** A set of facts X *activates* a set of rules R iff there is a sequence $< r_1, \ldots, r_n >$ with $\{r_1, \ldots, r_n\} = R$ such that

1. $body(r_1) \subseteq X$
2. for all $i = 2, \ldots, n$ we have $body(r_i) \subseteq \{head(r_1), \ldots, head(r_{i-1})\} \cup X$

A set of facts X *minimally activates* a set of rules R iff X activates R and there is no proper subset of X that activates R.

If X activates R we also say that X is an *activation set* of R.

Example 11. Consider the design-time rule base $\mathcal{B}_7^{\setminus\mathcal{F}}$, defined via

$$\mathcal{B}_7^{\setminus\mathcal{F}} = \{a \rightarrow b; b \rightarrow c; c \rightarrow d\}.$$

For each rule, its activation set consists of the body of the rule, e.g., $\{a\}$ is an activation set of $\{a \rightarrow b\}$. Also, the set $\{a\}$ activates the entire set $\mathcal{B}_7^{\setminus\mathcal{F}}$.

In the following, we assume a rule base to be acyclic, i.e., the dependency graph[2] of $\mathcal{B}_7^{\setminus\mathcal{F}}$ is acyclic. Some further observations for cyclic rule bases are provided in [34]. We are now ready to define *quasi-inconsistency* as follows.

Definition 12 (Quasi-Inconsistency, [34])**.** Let $R_1, R_2 \subseteq \mathcal{R}_{\mathcal{L}}$ be rule bases and X_1, X_2 be consistent sets of literals. A tuple (R_1, X_1, R_2, X_2) is called an *issue* iff

1. $X_1 \subseteq X_2$.
2. X_1 minimally activates R_1.
3. X_2 minimally activates R_2.
4. R_1 is X_1-consistent and R_2 is X_2-consistent.
5. $R_1 \cup R_2$ is X_2-inconsistent.

---

[2] The *dependency graph* $G_{\mathcal{B}}$ of a rule base $\mathcal{B}$ is a directed graph $G_{\mathcal{B}} = (\mathcal{B}, E_{\mathcal{B}})$ where $(r_1, r_2) \in E_{\mathcal{B}}$ for $r_1, r_2 \in \mathcal{B}$ iff $head(r_1) \in body(r_2)$.

A tuple (R_1, X_1, R_2, X_2) is called a *minimal issue* iff there are no R_1', R_2', X_1', X_2' with $R_1' \subseteq R_1$ and $R_2' \subseteq R_2$ (one of these set inclusions being proper) such that (R_1', X_1', R_2', X_2') is an issue. Note that two activation sets X_1 and X_2 are considered for the case that one activation set subsumes the other, cf. Example 13.

A rule base $\mathcal{B}$ is *quasi-inconsistent* iff there is an issue (R_1, X_1, R_2, X_2) with $R_1, R_2 \subseteq \mathcal{B}$. Then we also say that (R_1, X_1, R_2, X_2) is an issue of $\mathcal{B}$. Let Issues($\mathcal{B}$), MinIssues($\mathcal{B}$) be the set of all (minimal) issues of $\mathcal{B}$, respectively.

In other words, an issue (R_1, X_1, R_2, X_2) describes a case where the activation of one set of rules R_1 implies the activation of a second set of rules R_2 and both sets together derive an inconsistency (while being consistent on their own).

Example 13. We recall the (design-time) rule base $\mathcal{B}_1$ from the introduction, with

$$\begin{aligned}\mathcal{B}_1 = \{&newCustomer \rightarrow creditWorthy,\\ &newCustomer \rightarrow \neg creditWorthy\}.\end{aligned}$$

Then we have a minimal issue m_1, with

$$\begin{aligned}m_1 = (&\{newCustomer \rightarrow creditWorthy\}, \{newCustomer\},\\ &\{newCustomer \rightarrow \neg creditWorthy\}, \{newCustomer\}).\end{aligned}$$

The shown m_1 is a minimal issue of $\mathcal{B}_1$, i.e., $\mathcal{B}_1$ is quasi-inconsistent (despite being classically consistent). Experts need to resolve this modelling errors, as these two rules cannot be used for any meaningful conclusions in the scope of compliance reasoning. Note that in the minimal issue m_1, the two activation sets X_1 and X_2 are identical ($newCustomer$). Yet, using two activation sets as proposed in Definition 12 is needed for cases where one rule condition subsumes the other. For instance, consider the following modified version of $\mathcal{B}_1$:

$$\begin{aligned}\mathcal{B}_1* = \{&newCustomer \rightarrow creditWorthy,\\ &newCustomer, male \rightarrow \neg creditWorthy\}.\end{aligned}$$

In this case, we have a minimal issue m_2, with

$$\begin{aligned}m_2 = (&\{newCustomer \rightarrow creditWorthy\}, \{newCustomer\},\\ &\{newCustomer, male \rightarrow \neg creditWorthy\}, \{newCustomer, male\}).\end{aligned}$$

The shown $\mathcal{B}_1*$ is also quasi-inconsistent via 1.) in Definition 12, as the fact combination $newCustomer, male$ yields contradictory conclusions.

To summarize, we have introduced two main cases of "inconsistency", namely classic-logical inconsistency and potential inconsistency, the latter of which only refers to design-time rule bases. For potential inconsistency, we further distinguish between potential issues, i.e., any F-inconsistency, and actual issues, i.e., an issue which satisfies the conditions in Definition 12. If there exists an actual issue, we say that a rule base is quasi-inconsistent. We conclude with an example distinguishing these cases.

Example 14. Consider the following exemplary rule bases (a), (b) and (c).

a		
$a \rightarrow b$	$a \rightarrow b$	$b \rightarrow d$
$a \rightarrow \neg b$	$a \rightarrow \neg b$	$c \rightarrow \neg d$
(a) (Classic) Inconsistency	(b) Potential Inconsistency (Actual Issue)	(c) Potential Inconsistency (Potential Issue)

In case (a), we see that the rule base consists of the fact a, and two contradictory rules. The fact a can be used to instantiate the rule and infer the contradictory conclusions $b, \neg b$. Thus, the rule base (a) is inconsistent in the classic-logical sense.

For cases (b) and (c), we see that no facts are present, i.e., these are design-time rule bases. Yet, we see that there are potential inconsistencies for these two design-time rule bases.

In case (b), the rule base is $\{a\}$-inconsistent. The two rules will always be activated together, i.e. we have an (actual) issue. Following Definition 12, we therefore say that the rule base (b) is quasi-inconsistent (while being classically consistent),

In case (c), we see that the rule base is $\{b, c\}$-inconsistent. Here, it can of course be the case that we only encounter either b, or either c, individually during run-time. Therefore, the rule base in (c), while being classically consistent, has a potential issue as is therefore potentially inconsistent (but no actual issue, i.e., rule base (c) is not quasi-inconsistent).

On this matter, we argue that actual issues represent modelling errors in the business rules which should be resolved at design-time. To clarify, yes, we do not know whether a fact a will actually occur, but the rules which are part of actual issues cannot be used for any form of meaningful reasoning. On the contrary, potential issues can be correct in many cases (simply two different conditions with "opposing" conclusions). Likewise, an expert could also determine that it is highly unlikely, that a problematic fact combination will actually occur in practice and keep rules such as in (c) as-is. Here, the potential issues should however be monitored during run-time to mitigate compliance breaches due to unexpected behavior.

Summary: On the Notion of Inconsistency

>The term inconsistency refers to (classic-logical) inconsistency of a rule base, i.e., $\mathcal{B} \models \bot$

>It follows that design-time rule bases cannot be inconsistent in the classic-logical sense, but can be potentially inconsistent

>If a design-time rule base has at least one actual issue, we say that the rule base is quasi-inconsistent

3.2 Measuring Inconsistency in Business Rule Bases

Next to *detecting* (quasi-)inconsistency, we are now interested in methods that can *measure* (quasi-)inconsistency. Such a quantitative assessment can be useful for companies in order to a) assess the *severity* of the detected errors, and b) provide a *prioritization* in which order rules should be attended to in the scope of re-modeling [29]. As mentioned in the Introduction, the field of inconsistency measurement studies *inconsistency measures* to this aim, which are functions that assign a non-negative numerical value to a rule base, where a larger value indicates a larger "severity" of inconsistency. In this section, we develop new means for measuring inconsistency in business rule bases, while reusing and extending results from the field of inconsistency measurement.

As the notion of a "severity" of inconsistency is not well-defined in academia, a quantification of inconsistency is not easily characterizable, and there have been numerous proposals how to measure inconsistency, see e. g. [129] for an overview and [16, 17, 62] for some recent works.

Following [69], there are essentially two approaches to assess atomic inconsistency, namely a) on a formula-level, and b) on the level of atoms. Also, there are c) hybrid approaches that aim to combine a) and b). For each of these three forms a)-c), we have selected a representative inconsistency measure based on the suggestions in THIMM (2019), shown in Figure 3.2. Specifically, we consider the MI−inconsistency measure $\mathcal{I}_{\mathsf{MI}}$ [70], the contension measure $\mathcal{I}_c$ [69] and the mv-inconsistency measure $\mathcal{I}_{mv}$ [142]. Note that we also consider the the problematic inconsistency measure $\mathcal{I}_p$ [69], as it is a variation if the MI inconsistency measure, as well as the drastic inconsistency measure $\mathcal{I}_d$ [70], which is a baseline measure that does not fall into the abovementioned categories. We acknowledge there have been numerous other proposals, yet virtually all other measures can be also grouped according to Figure 3.2, and thus our results can be extended to arbitrary inconsistency measures.

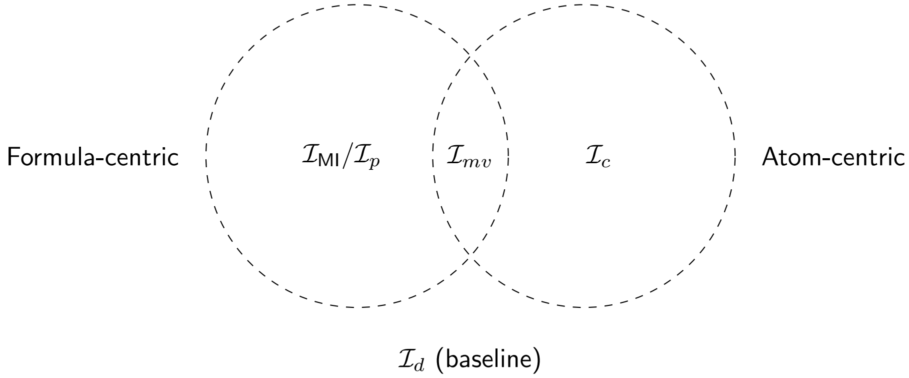


Figure 3.2: Classification of considered measures following [69].

In the following, we introduce the 5 selected inconsistency measures, shown in Figure 3.3.

$$\mathcal{I}_d(\mathcal{B}) = \begin{cases} 1 & \text{if } \mathcal{B} \models \perp \\ 0 & \text{otherwise} \end{cases}$$

$$\mathcal{I}_{\mathsf{MI}}(\mathcal{B}) = |\mathsf{MI}(\mathcal{B})|$$

$$\mathcal{I}_p(\mathcal{B}) = |\bigcup_{M \in \mathsf{MI}(\mathcal{B})} M|$$

$$\mathcal{I}_c(\mathcal{B}) = \min\{|i^{-1}(\mathsf{b})| \mid i \models^3 \mathcal{B}\}$$

$$\mathcal{I}_{mv}(\mathcal{B}) = \frac{|\bigcup_{M \in \mathsf{MI}(\mathcal{B})} \mathcal{A}(M)|}{|\mathcal{A}(\mathcal{B})|}$$

Figure 3.3: Definitions of the considered inconsistency measures, taken from [124].

The traditional setting for inconsistency measurement is that of propositional logic knowledge bases. The considered measures, namely the drastic inconsistency measure $\mathcal{I}_d$, the MI-inconsistency measure $\mathcal{I}_{\mathsf{MI}}$, the problematic inconsistency measure $\mathcal{I}_p$, the contension measures $\mathcal{I}_c$ and the mv-inconsistency measure $\mathcal{I}_{mv}$, are thus originally defined for propositional logic. For readability, we define these measures over business rule bases in this thesis. A definition for propositional logic can be found in [130].

A baseline is the drastic inconsistency measure $\mathcal{I}_d$ [70], which only differentiates between inconsistent and consistent knowledge bases.

Formula-level measures, cf. Figure 3.2, consider *elements* of the rule base, i.e. facts and rules. Here, the measures $\mathcal{I}_{\mathsf{MI}}$ and $\mathcal{I}_p$ (cf. e.g. [69,70]) consider different aspects of minimal inconsistent subsets of a rule base. A minimal inconsistent subset of $\mathcal{B}$ is defined as a set $M \subseteq \mathcal{B}$, s.t. $M \models \perp$ and there is no $M' \subset M$ s.t. M' is consistent. We denote $\mathsf{MI}(\mathcal{B})$ as the set of all minimal inconsistent subsets of $\mathcal{B}$. Here, the $\mathcal{I}_{\mathsf{MI}}$ measure counts the number of minimal inconsistent subsets, whereas the $\mathcal{I}_p$ measure counts the number of distinct elements, that appear in any minimal inconsistent subset.

Measures on the level of atoms, cf. Figure 3.2, consider the atoms of the rule base, resp. their interpretations. An interpretation is usually a function $\omega : \mathcal{A} \rightarrow \{true, false\}$, which assigns a truth value to an atom [49]. An interpretation ω satisfies $a \in \mathcal{A}$, iff $\omega(a) = true$. The $\mathcal{I}_c$ measure (cf. e.g. [69]) is defined using paraconsistent semantics based on three-valued interpretations [98]. A three-valued interpretation is a function $i : \mathcal{A} \rightarrow \{\mathsf{t}, \mathsf{f}, \mathsf{b}\}$, which assigns one of three truth value to an atom. The values t and f represent the classic *true* and *false*, and b stands for *both*. The intuition of the latter truth value *both* is that there exist conflicting truth values for an atom. For example, consider a rule base $b = \{a; a \to b; a \to \neg b\}$. In such a case, no conclusion about the atom b can be made without a further investigation by experts, thus, the atom b should be marked with this conflicting truth value to indicate this problem. The interpretation function i can be extended to arbitrary elements of a rule base as shown in Table 3.1 (Note that $i(\alpha, \beta)$ is the logical conjunction of α and β, and $i(\alpha \to \beta)$ denotes the logical $\neg\alpha \vee \beta$). We say an interpretation i satisfies a rule or fact $\alpha \in \mathcal{B}$ (denoted by $i \models^3 \alpha$) if either $i(\alpha) = \mathsf{t}$ or $i(\alpha) = \mathsf{b}$. The $\mathcal{I}_c$ measure then measures inconsistency by seeking an interpretation i that assigns b to a minimal number of atoms, and then counting the number of atoms that were assigned the truth value b.

α	β	$i(\alpha, \beta)$	$i(\alpha \to \beta)$
t	t	t	t
t	b	b	b
t	f	f	f
b	t	b	t
b	b	b	b
b	f	f	b
f	t	f	t
f	b	f	t
f	f	f	t

α	$i(\neg\alpha)$
t	f
b	b
f	t

Table 3.1: Truth tables for three-valued logic, adapted from [98].

There are also hybrid approaches, which aim to combine the aspects of formula-centric and atom-centric inconsistency. The $\mathcal{I}_{mv}$ measure [142] combines measurement through multi-valued semantics and MIs, by counting the fraction of atoms in a rule base that appear in any minimal inconsistent subset.

Example 15. Consider the exemplary rule base $\mathcal{B}'_9$, defined via

$$\begin{aligned}\mathcal{B}'_9 = \{&a, \neg b, c, \neg e,\\ &a \to b,\\ &a \to \neg b,\\ &a \to d,\\ &d \to e\}\end{aligned}$$

First, it can directly be seen that $\mathcal{B}'_9$ is inconsistent, for example, the contradictory conclusions $b, \neg b$ can be derived, e.g. via $\{a; a \to b; a \to \neg b\}$. Therefore, $\mathcal{I}_d(\mathcal{B}'_9) = 1$ per definition.

Next, regarding minimal inconsistent subsets, we have that

$$\mathsf{MI}(\mathcal{B}'_9) = \{\{a; a \to b; a \to \neg b\}, \{a; a \to b; \neg b\}, \{a; a \to d; d \to e; \neg e\}\}.$$

Consequently, we have that $\mathcal{I}_{\mathsf{MI}}(\mathcal{B}'_9) = |\mathsf{MI}(\mathcal{B}'_9)| = 3$ (as there are three minimal inconsistent subsets), and $\mathcal{I}_p(\mathcal{B}'_9) =$

$$|\bigcup_{M \in \mathsf{MI}(\mathcal{B}'_9)} M| = |\{a; a \to b; a \to \neg b; \neg b; a \to d; d; \neg e\}| = 7.$$

Regarding $\mathcal{I}_c$, consider the exemplary interpretation $i_1 : \{a, b, c, d, e\} \to \{\mathsf{t}, \mathsf{f}, \mathsf{b}\}$, defined via

$$i_1(a) = \mathsf{b} \qquad i_1(b) = \mathsf{f} \qquad i_1(c) = \mathsf{t} \qquad i_1(d) = \mathsf{f} \qquad i_1(e) = \mathsf{f}$$

The interpretation i_1 satisfies all elements $\alpha \in \mathcal{B}'_9$, explained as follows. Recall that an interpretation i satisfies any $\alpha \in \mathcal{B}$, iff $i(\alpha) = \mathsf{t}$ or $i(\alpha) = \mathsf{b}$. Given i_1 and the extension mapping in Table 3.1, it follows that:

$i_1(a)$ = b (per assumption)
$i_1(\neg b)$ = t (as $i_1(b) = \mathsf{f}$, cf. Table 3.1)
$i_1(c)$ = t (per assumption)
$i_1(\neg e)$ = t (as $i_1(e) = \mathsf{f}$, cf. Table 3.1)
$i_1(a \to b)$ = b (as $i_1(a) = \mathsf{b}$ and $i_1(b) = \mathsf{f}$, cf. Table 3.1)
$i_1(a \to \neg b)$ = t (as $i_1(a) = \mathsf{b}$ and $i_1(b) = \mathsf{f}$, cf. Table 3.1)
$i_1(a \to d)$ = b (as $i_1(a) = \mathsf{b}$ and $i_1(d) = \mathsf{f}$, cf. Table 3.1)
$i_1(d \to e)$ = t (as $i_1(d) = \mathsf{f}$ and $i_1(e) = \mathsf{f}$, cf. Table 3.1)

We see that i_1 satisfies all $\alpha \in \mathcal{B}'_9$ and there is no interpretation which assigns the truth value b to fewer atoms. The truth value b was assigned to 1 atom in i_1, and thus $\mathcal{I}_c(\mathcal{B}'_9) = 1$.

Last, we recall the minimal inconsistent subsets of $\mathcal{B}'_9$

$$\mathsf{MI}(\mathcal{B}'_9) = \{\{a; a \to b; a \to \neg b\}, \{a; a \to b; \neg b\}, \{a; a \to d; d \to e; \neg e\}\}.$$

Then we have that $\bigcup_{M \in \mathsf{MI}(\mathcal{B}'_9)} \mathcal{A}(M) = \{a, b, d, e\}$. We recall that $\mathcal{A}(\mathcal{B}'_9) = \{a, b, c, d, e\}$, and thus $\mathcal{I}_{mv}(\mathcal{B}'_9) =$

$$\frac{\mid \{a, b, d, e\} \mid}{\mid \{a, b, c, d, e\} \mid} = 0.8$$

To conclude, we have that

$$\mathcal{I}_d(\mathcal{B}'_9) = 1 \qquad \mathcal{I}_{\mathsf{MI}}(\mathcal{B}'_9) = 3$$
$$\mathcal{I}_p(\mathcal{B}'_9) = 7 \qquad \mathcal{I}_c(\mathcal{B}'_9) = 1$$
$$\mathcal{I}_{mv}(\mathcal{B}'_9) = 0.8$$

As can be seen in Example 15, different measures may return different values for the same rule base, and the notion of a "severity" of inconsistency is still under investigation. In classical inconsistency measurement, research is therefore driven by the development of rationality postulates, i. e., desirable properties that should be satisfied by concrete approaches – cf. [127] for an overview. In this work, we also follow this approach and will investigate postulates that specify the expected behavior in a BRM use case, as well as the relation of inconsistency measures to such postulates.

3.2.1 Design-Time Inconsistency Measurement

▷ *This section corresponds to the work [34] by the author.*

In this section, we are interested in measures of inconsistency that can assess quasi-inconsistency. In the following, we denote this as quasi-inconsistency measures $\mathcal{I}^Q$.

Formally speaking, we are interested in functions $\mathcal{I}^Q : \mathbb{B}^{\setminus \mathcal{F}} \to \mathbb{R}^{\infty}_{\geq 0}$ that assign to a design-time rule base $\mathcal{B}^{\setminus \mathcal{F}}$ a non-negative real number $\mathcal{I}^Q(\mathcal{B}^{\setminus \mathcal{F}})$, which represents the degree of quasi-inconsistency in $\mathcal{B}^{\setminus \mathcal{F}}$; with 0 representing "not quasi-inconsistent" and larger values indicating more severe quasi-inconsistency.

To specify this behavior, we propose the postulate of *quasi-inconsistency* [34].

Quasi-Inconsistency (QI) $\mathcal{I}^Q(\mathcal{B}) > 0$ iff $\mathcal{B}$ is quasi-inconsistent.

This postulate states that a quasi-inconsistency measure should be able to distinguish between rule bases that are quasi-inconsistent and those that are not.

We recall the introduced inconsistency measures, and the (design-time) rule base $\mathcal{B}_1$ from the introduction, defined via

$$\mathcal{B}_1 = \{newCustomer \to creditWorthy, newCustomer \to \neg creditWorthy\}.$$

This rule base is quasi-inconsistent, thus we would expect an assessment to yield a value larger than zero. However, for all considered inconsistency measures, we have

$$\begin{aligned} \mathcal{I}_d(\mathcal{B}_1) &= 0 & \mathcal{I}_{\mathsf{MI}}(\mathcal{B}_1) &= 0 \\ \mathcal{I}_p(\mathcal{B}_1) &= 0 & \mathcal{I}_c(\mathcal{B}_1) &= 0 \\ \mathcal{I}_{mv}(\mathcal{B}_1) &= 0 \end{aligned}$$

This result is not surprising, as these measures satisfy the postulate of *consistency* (CO) [68], which demands that the returned value should be zero iff the rule base is consistent. This however impedes the expected behavior as dictated by QI.

Proposition 16 ([34]). *CO is incompatible with QI.*

Following Proposition 16, observe that virtually all classical inconsistency-measures cannot be used as quasi-inconsistency measures, as the uniformly satisfy CO [127]. This calls for new approaches to measure quasi-inconsistency.

As a main contribution, we adapted the above classical inconsistency measures such that they satisfy the desired behavior [34]. This allows to utilize the amenities of these well-studied measures for the use-case of measuring quasi-inconsistency. Figure 3.4 shows an overview of the developed measures. We refer the reader to [34] for a deeper discussion and examples. Below, we briefly describe the developed measures, using $\mathcal{B}_1$ as an example.

$$\mathcal{I}_d^Q(\mathcal{B}^{\setminus\mathcal{F}}) = \begin{cases} 1 & \text{if } \mathcal{B}^{\setminus\mathcal{F}} \text{ is quasi-inconsistent} \\ 0 & \text{otherwise} \end{cases}$$

$$\mathcal{I}_{\mathsf{MI}}^Q(\mathcal{B}^{\setminus\mathcal{F}}) = |\mathsf{MinIssues}(\mathcal{B}^{\setminus\mathcal{F}})|$$

$$\mathcal{I}_p^Q(\mathcal{B}^{\setminus\mathcal{F}}) = |\bigcup_{(R_1,X_1\ R_2,X_2)\in\mathsf{MinIssues}(\mathcal{B}^{\setminus\mathcal{F}})} R_1 \cup R_2|$$

$$\mathcal{I}_c^Q(\mathcal{B}^{\setminus\mathcal{F}}) = min\{i^{-1}(\mathsf{b}) \mid i \models^3 (\bigcup_{(R_1,X_1\ R_2,X_2)\in\mathsf{MinIssues}(\mathcal{B}^{\setminus\mathcal{F}})} R_1 \cup R_2 \cup \mathcal{L}(\mathcal{B}^{\setminus\mathcal{F}})\}$$

$$\mathcal{I}_{mv}^Q(\mathcal{B}^{\setminus\mathcal{F}}) = \frac{|\bigcup_{(R_1,X_1\ R_2,X_2)\in\mathsf{MinIssues}(\mathcal{B}^{\setminus\mathcal{F}})} \mathcal{A}((R_1,X_1\ R_2,X_2))|}{|\mathcal{A}(\mathcal{B}^{\setminus\mathcal{F}})|}$$

Figure 3.4: Definitions of the proposed quasi-inconsistency measures [34].

The drastic quasi-inconsistency measure is a baseline to distinguish quasi-inconsistent design-time rule bases. For the exemplary $\mathcal{B}_1$, $\mathcal{I}_d^Q$ is therefore 1 per definition.

The MI-quasi-inconsistency measure counts the number of minimal issues (i.e., actual issues). Likewise, the problematic quasi-inconsistency measure counts the number of *rules* that participate in at least one minimal issue. For the exemplary $\mathcal{B}_1$, we recall from Example 13 that there is one minimal issue m_1, with

$$\begin{aligned} m_1 =&(\{newCustomer \to creditWorthy\}, \{newCustomer\}, \\ &\{newCustomer \to \neg creditWorthy\}, \{newCustomer\}). \end{aligned}$$

Consequently, $\mathcal{I}_{\mathsf{MI}}^Q = 1$ and $\mathcal{I}_p^Q = 2$.

The quasi contension measure seeks an interpretation that satisfies all rules that participate in any minimal issue (as opposed to the traditional $\mathcal{I}_c$ which seeks an interpretation that satisfies the entire rule base), while assigning the truth value b to a minimal number of atoms, and then counts the number of atoms that were assigned the truth value b. Note that next to the rules (cf. $R_1 \cup R_2$), also the rule literals (cf. $\mathcal{L}(\mathcal{B}^{\setminus\mathcal{F}})$) are needed for inference (as no conclusions could be made without activations). We recall the minimal issue m_1 in $\mathcal{B}_1$. Then we consider $b_1 \subseteq \mathcal{B}_1$ via the definition of $\mathcal{I}_c^Q$, with

$$\begin{aligned} b_1 =& \bigcup_{(R_1,X_1\ R_2,X_2)\in\mathsf{MinIssues}(\mathcal{B}_1)} R_1 \cup R_2 \cup \mathcal{L}(\mathcal{B}_1) \\ =& \{newCustomer \to creditWorthy; newCustomer \to \neg creditWorthy; \\ & newCustomer; creditWorthy; \neg creditWorthy\} \end{aligned}$$

For the interpretation $i_2 : \{newCustomer, creditWorthy\} \to \{\mathsf{t}, \mathsf{f}, \mathsf{b}\}$, defined via

$$i_2(newCustomer) = \mathsf{t} \qquad\qquad i_2(creditWorthy) = \mathsf{b}$$

we see that i_2 satisfies all elements $\alpha \in b_1$ (i.e., $i_2 \models^3 b_1$), and there is no interpretation that assign b to fewer atoms. As b was assigned to 1 atom in i_2, $\mathcal{I}^Q_c(\mathcal{B}_1) = 1$.

Last, the mv-quasi-inconsistency measure considers the fraction of those atoms that are part of problematic rules, relative to the atoms of all rules. We recall the minimal issue m_1, and see that all atoms of the rule base are part of at least least one minimal issue, thus $\mathcal{I}^Q_{mv} = 2/2 = 1$.

Regarding the proposed postulate of QI, we recall the introduced quasi-inconsistency measures, and that for the above $\mathcal{B}_1^{\setminus\mathcal{F}}$, we expect a quasi-inconsistency assessment to yield a value > 0. Also, we expect the measures to yield a value of zero for a rule base which is not quasi-inconsistent, e.g. the above $\mathcal{B}_7^{\setminus\mathcal{F}}$. Then, we have

$$\begin{aligned}
\mathcal{I}^Q_d(\mathcal{B}_1^{\setminus\mathcal{F}}) &= 1 & \qquad \mathcal{I}^Q_d(\mathcal{B}_7^{\setminus\mathcal{F}}) &= 0\\
\mathcal{I}^Q_{\mathsf{MI}}(\mathcal{B}_1^{\setminus\mathcal{F}}) &= 1 & \mathcal{I}^Q_{\mathsf{MI}}(\mathcal{B}_7^{\setminus\mathcal{F}}) &= 0\\
\mathcal{I}^Q_p(\mathcal{B}_1^{\setminus\mathcal{F}}) &= 2 & \mathcal{I}^Q_p(\mathcal{B}_7^{\setminus\mathcal{F}}) &= 0\\
\mathcal{I}^Q_c(\mathcal{B}_1^{\setminus\mathcal{F}}) &= 1 & \mathcal{I}^Q_c(\mathcal{B}_7^{\setminus\mathcal{F}}) &= 0\\
\mathcal{I}^Q_{mv}(\mathcal{B}_1^{\setminus\mathcal{F}}) &= 1 & \mathcal{I}^Q_{mv}(\mathcal{B}_7^{\setminus\mathcal{F}}) &= 0
\end{aligned}$$

In result, all presented measures capture the desired behavior.

Proposition 17 ([34]). *$\mathcal{I}^Q_d$, $\mathcal{I}^Q_{\mathsf{MI}}$, $\mathcal{I}^Q_p$, $\mathcal{I}^Q_c$ and $\mathcal{I}^Q_{mv}$ satisfy QI.*

The proofs are presented in Section 4.1.1. Our main result is that while classical inconsistency measures do not suffice to measure quasi-inconsistency (cf. Proposition 16), we see that the proposed measures satisfy the intended behavior and can thus be used to measure quasi-inconsistency. Such an assessment can provide useful insights for companies in the scope of resolving human modeling errors during design-time, which was not possible with existing means. This result is summarized in Table 3.2.

$\mathcal{I}$	$\mathcal{I}_d$	$\mathcal{I}_{\mathsf{MI}}$	$\mathcal{I}_p$	$\mathcal{I}_c$	$\mathcal{I}_{mv}$	$\mathcal{I}^Q_d$	$\mathcal{I}^Q_{\mathsf{MI}}$	$\mathcal{I}^Q_p$	$\mathcal{I}^Q_c$	$\mathcal{I}^Q_{mv}$
QI	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓

Table 3.2: Compliance with the postulate QI of the proposed quasi-inconsistency measures [34].

3.2.2 Run-Time Inconsistency Measurement

▷ *This section corresponds to the work [35] by the author.*

In this section, we are interested in measures of inconsistency that can assess inconsistency in business rule bases at run-time.

During run-time, we actually have facts, and classic-logical inconsistency is possible. Thus, it would seem intuitive to simply apply the existing inconsistency measures. However, as described in the Introduction, a straightforward application is not plausible: In the classical setting of inconsistency measurement, knowledge bases are constituted of propositional formulas, where these formulas do not have a distinguishable level of granularity. On the other hand, business rule bases distinguish between *facts* and *rules*. That is, facts have a different conceptual quality as their veracity is unconditionally assumed [58][3]. Thus, a rule base consists of a set of (*indisputable*) facts and rules. This impedes using existing measures in this setting, explained as follows.

Example 18. We recall the rule base $\mathcal{B}_4$, with

$$\begin{aligned}\mathcal{B}_4 = \{&platinumCustomer, mentalCondition,\\ &platinumCustomer \rightarrow contractuallyCapable,\\ &mentalCondition \rightarrow \neg contractuallyCapable\}.\end{aligned}$$

As can be seen, the rule base is inconsistent, but we can see that by removing the fact $mentalCondition$ it becomes consistent. However, the facts $mentalCondition$, and $platinumCustomer$ are provided by a given case input and have to be kept as-is, even in the scope of inconsistency handling. For instance, one cannot change the mental condition of a customer just to make the set of business rules consistent.

Consequently, methods are needed to analyze inconsistency based on a distinction between (indisputable) facts and rules. To further frame the problem, consider the following exemplary rule bases $\mathcal{B}_8$, $\mathcal{B}_9$, $\mathcal{B}_{10}$, defined via

$$\begin{aligned}\mathcal{B}_8 &= \{a, \neg a\}\\ \mathcal{B}_9 &= \{a; a \rightarrow b; a \rightarrow \neg b\}\\ \mathcal{B}_{10} &= \{a; a \rightarrow b; a \rightarrow \neg b; c; \neg c\}.\end{aligned}$$

In our setting, we are interested only in inconsistencies comprising at least one business rule, as this indicates a human modelling error in the set of business rules. In turn, we will not consider inconsistencies such as in $\mathcal{B}_8$ (this can be handled by existing results from inconsistency measurement), but want to develop new "rule-based" inconsistency measures, in the following denoted as $\mathcal{I}^{RB} : \mathbb{B} \rightarrow \mathbb{R}^{\infty}_{\geq 0}$, which can specifically assess actual modeling errors, i. e., inconsistencies including at least one rule.

To this aim, we first propose some novel postulates which should be satisfied by rule-based inconsistency measures [35].

Rule Consistency (RC) $\mathcal{I}^{RB}(\mathcal{B}) = 0$ if and only if for all consistent sets $F' \subseteq \mathcal{F}(\mathcal{B})$, $\mathcal{R}(\mathcal{B}) \cup F'$ is consistent.

---

[3] We acknowledge there could also be fact contradictions, however these are clear data issues and beyond the scope of this thesis. In this thesis, we are interested in inconsistencies resulting from modelling errors in business rules.

The above rationality postulate is a weakening of the classical postulate *consistency* [127], which requires $\mathcal{I}(\mathcal{B}) = 0$ if and only if $\mathcal{B}$ is consistent. Following RC, measures should assess the degree of inconsistency for $\mathcal{B}_8$ as 0. Vice versa, in case there is at least one inconsistency which touches at least one rule, the returned inconsistency value should not be 0.

In any case however, we want to ensure that only conflicts including at least one rule are valued towards the quantification of inconsistency. Atoms appearing only as facts should not alter the degree of inconsistency for rule-based measures, if they do not contradict any rules. We subsequently define the property of fact elision [35].

Fact Elision (FE) If $\forall r \in \mathcal{B} : \mathit{head}(r) \neq \alpha$ then $\mathcal{I}^{RB}(\mathcal{B}) = \mathcal{I}^{RB}(\mathcal{B} \cup \{\overline{\alpha}\})$.

This postulate is closely related to the postulate safe-formula independence (SI) [127], which states that formulas which do not share the signature with the existing propositions of a knowledge base, i. e., safe formulas, should not alter the degree of inconsistency. The proposed postulate is a weakening of SI, and states that a formula only needs to be safe w.r.t. the business rules. That is, even if an added formula is *not* safe w.r.t. facts in the knowledge base, this should not alter the score. Consequently, for any $\mathcal{I}^{RB}$, $\mathcal{I}^{RB}(\mathcal{B}_9)$ should be equal to $\mathcal{I}^{RB}(\mathcal{B}_9 \cup c)$, and $\mathcal{I}^{RB}(\mathcal{B}_{10})$ should be equal to $\mathcal{I}^{RB}(\mathcal{B}_{10} \cup c)$.

Last, we consider a further aspect of rule-based inconsistency measures. Consider again the rule base $\mathcal{B}_9$ and the rule base $\mathcal{B}_{11}$, defined as

$$\mathcal{B}_9 = \{a; a \rightarrow b; a \rightarrow \neg b\} \qquad\qquad \mathcal{B}_{11} = \{a; a \rightarrow b; \neg b\}.$$

In the traditional setting of inconsistency measurement, one could argue that both rule bases are equally inconsistent. However, we see that the inconsistency in $\mathcal{B}_{11}$ can only be resolved in one way in our setting – namely by modifying or deleting the rule $a \rightarrow b$ – as the given facts a and $\neg b$ are indisputable. On the contrary, the inconsistency in $\mathcal{B}_9$ is caused by contradicting rules, thus, this inconsistency is more complex to handle and requires attention by domain experts. To identify such cases, we introduce a third, optional property of rule emphasis [35]. For that, a formula $a \in \mathcal{B}$ is called a *free formula*, if $a \notin M, \forall M \in \mathsf{MI}(\mathcal{B})$. We denote the free formulas of $\mathcal{B}$ as $\mathsf{Free}(\mathcal{B})$.

Rule Emphasis (RE) If $\alpha \rightarrow \beta \notin \mathcal{B}$ and $\alpha \rightarrow \beta \notin \mathsf{Free}(\mathcal{B} \cup \{\alpha \rightarrow \beta\})$ then $\mathcal{I}(\mathcal{B} \cup \{\alpha \rightarrow \beta\}) > \mathcal{I}(\mathcal{B} \cup \{\beta\})$.

This postulate states that adding a rule $\alpha \rightarrow \beta$ to a rule base, where this rule is not a free formula, should increase the inconsistency more than adding only the head of that rule, i. e. as a fact β. This postulate ensures that measures valuate the conflicts involving contradictory rules as more significant than a conflict resulting from a rule and a non-negotiable fact (as the former type of inconsistency might be more complex to resolve than the latter).

Example 19. For the rule bases $\mathcal{B}_8$, $\mathcal{B}_9$, $\mathcal{B}_{10}$, and $\mathcal{B}_{11}$ from before, we expect a rule-based inconsistency assessment $\mathcal{I}^{RB}$ satisfying the postulates RC, FE, and RE to give

$$\begin{aligned} 0 = \mathcal{I}^{RB}(\mathcal{B}_8) < \mathcal{I}^{RB}(\mathcal{B}_9) = \mathcal{I}^{RB}(\mathcal{B}_{10}) \qquad \text{and} \\ \mathcal{I}^{RB}(\mathcal{B}_{11}) < \mathcal{I}^{RB}(\mathcal{B}_9) \end{aligned}$$

However, for the considered "traditional" inconsistency measures we get

$$\begin{array}{rrrr} \mathcal{I}_d(\mathcal{B}_8) = 1 & \mathcal{I}_d(\mathcal{B}_9) = 1 & \mathcal{I}_d(\mathcal{B}_{10}) = 1 & \mathcal{I}_d(\mathcal{B}_{11}) = 1 \\ \mathcal{I}_{\mathsf{MI}}(\mathcal{B}_8) = 1 & \mathcal{I}_{\mathsf{MI}}(\mathcal{B}_9) = 1 & \mathcal{I}_{\mathsf{MI}}(\mathcal{B}_{10}) = 2 & \mathcal{I}_{\mathsf{MI}}(\mathcal{B}_{11}) = 1 \\ \mathcal{I}_p(\mathcal{B}_8) = 2 & \mathcal{I}_p(\mathcal{B}_9) = 3 & \mathcal{I}_p(\mathcal{B}_{10}) = 5 & \mathcal{I}_p(\mathcal{B}_{11}) = 3 \\ \mathcal{I}_c(\mathcal{B}_8) = 1 & \mathcal{I}_c(\mathcal{B}_9) = 1 & \mathcal{I}_c(\mathcal{B}_{10}) = 2 & \mathcal{I}_c(\mathcal{B}_{11}) = 1 \\ \mathcal{I}_{mv}(\mathcal{B}_8) = 1 & \mathcal{I}_{mv}(\mathcal{B}_9) = 1 & \mathcal{I}_{mv}(\mathcal{B}_{10}) = 1 & \mathcal{I}_{mv}(\mathcal{B}_{11}) = 1 \end{array}$$

We see that none of the considered measures is capable of capturing the desired outcome. Specifically, we see that for the above measures (in the following abbreviated as $\mathcal{I}$ by a slight misuse of notation):

- $\mathcal{I}(\mathcal{B}_8) > 0$ for all measures, thus violating RC.
- $\mathcal{I}(\mathcal{B}_9) \neq \mathcal{I}(\mathcal{B}_{10})$ for all measures except $\mathcal{I}_d$ and $\mathcal{I}_{mv}$, thus broadly violating FE.
- $\mathcal{I}(\mathcal{B}_{11}) \not< \mathcal{I}(\mathcal{B}_9)$ for all measures, thus violating RE.

Regarding $\mathcal{I}(\mathcal{B}_8) > 0$, this is intuitive, as all considered measures satisfy the postulate of *consistency* (CO)[70], which demands that the returned value should only be zero iff the rule base is consistent. As a result, we have the following:

Proposition 20 ([35]). *CO is incompatible with RC.*

Following from Proposition 20, virtually all existing inconsistency measures cannot be used as rule-based inconsistency measures, as they uniformly satisfy CO (cf. [127]) and thus broadly violate the proposed rationality postulates as motivated from the business use-case. This impedes using existing results in a company context and calls for an adaptation of measures to fit this use-case.

As a main contribution, we adapted the above classical inconsistency measures such that they satisfy the desired behavior [35]. This allows to utilize the amenities of these well-studied measures for the use-case of assessing modelling errors during run-time, i.e., measure inconsistencies comprising at least one business rule. Figure 3.5 shows an overview of the developed measures. Below, we briefly describe the developed measures. Regarding notation, recall that a rule base is not rule-consistent if it contains at least one minimal inconsistent subset, that itself contains at least one rule. To verify this condition, we consider only those minimal inconsistent subsets that do not contain two complementary facts

$a, \neg a$. Formally, we define $\mathsf{MI}^{\setminus\mathcal{F}}(\mathcal{B}) = \{M \in \mathsf{MI}(\mathcal{B}) \mid \nexists a \in \mathcal{A} : a, \neg a \in M\}$. If $a, \neg a \in M$ we also call M a pure fact set (note that indeed $a, \neg a \in M$ implies $M = \{a, \neg a\}$).

$$\mathcal{I}_d^{RB}(\mathcal{B}) = \begin{cases} 1 & \text{iff } \mathsf{MI}^{\setminus\mathcal{F}}(\mathcal{B}) \neq \emptyset \\ 0 & \text{otherwise} \end{cases}$$

$$\mathcal{I}_{\mathsf{MI}}^{RB}(\mathcal{B}) = |\mathsf{MI}^{\setminus\mathcal{F}}(\mathcal{B})|$$

$$\mathcal{I}_p^{RB}(\mathcal{B}) = |\bigcup_{M \in \mathsf{MI}^{\setminus\mathcal{F}}(\mathcal{B})} M \setminus \mathcal{F}(M)|$$

$$\mathcal{I}_c^{RB}(\mathcal{B}) = min\{|v^{-1}(\mathsf{b})| \mid v \models^3 \bigcup_{M \in \mathsf{MI}^{\setminus\mathcal{F}}(\mathcal{B})} M\}$$

$$\mathcal{I}_{mv}^{RB}(\mathcal{B}) = \frac{|\bigcup_{M \in \mathsf{MI}^{\setminus\mathcal{F}}(\mathcal{B})} \mathcal{A}(M)|}{|\mathcal{A}(\mathcal{B})|}$$

Figure 3.5: Definitions of the proposed rule-based inconsistency measures [35].

The rule-based drastic measure is a baseline to assess rule bases which are not rule-consistent, i.e., $\mathcal{I}_d^{RB}$ is 1 for any rule base which is not rule-consistent. The rule-based MI-measure counts the number of minimal inconsistent subsets without pure fact sets. Likewise, the rule-based problematic counts the number of rules that appear in at least one minimal inconsistent subset. The rule based contension measure seeks an interpretation that satisfies all $M \in \mathsf{MI}^{\setminus\mathcal{F}}(\mathcal{B})$ for a rule base $\mathcal{B}$, where this interpretation assigns the truth value b to a minimal number of atoms, and then counts the number of atoms that were assigned the truth value b. Last, the rule-based mv-measure is a relative measure that measures the fraction of those atoms that are part of problematic elements, relative to the atoms of the rule base, while ignoring pure fact sets. We refer the reader to [35] for a more detailed discussion of the introduced measures.

Regarding the proposed postulates, we recall the introduced rule-based measures and the rule bases $\mathcal{B}_8 - \mathcal{B}_{11}$, and that we expect a rule-based inconsistency assessment $\mathcal{I}^{RB}$ satisfying the postulates RC, FE, and RE to give

$$0 = \mathcal{I}^{RB}(\mathcal{B}_8) < \mathcal{I}^{RB}(\mathcal{B}_9) = \mathcal{I}^{RB}(\mathcal{B}_{10}) \qquad \text{and}$$
$$\mathcal{I}^{RB}(\mathcal{B}_{11}) < \mathcal{I}^{RB}(\mathcal{B}_9)$$

For our adapted measures, we have that

$$\begin{array}{llll}
\mathcal{I}_{d}^{RB}(\mathcal{B}_8) = 0 & \mathcal{I}_{d}^{RB}(\mathcal{B}_9) = 1 & \mathcal{I}_{d}^{RB}(\mathcal{B}_{10}) = 1 & \mathcal{I}_{d}^{RB}(\mathcal{B}_{11}) = 1 \\
\mathcal{I}_{\mathsf{MI}}^{RB}(\mathcal{B}_8) = 0 & \mathcal{I}_{\mathsf{MI}}^{RB}(\mathcal{B}_9) = 1 & \mathcal{I}_{\mathsf{MI}}^{RB}(\mathcal{B}_{10}) = 1 & \mathcal{I}_{\mathsf{MI}}^{RB}(\mathcal{B}_{11}) = 1 \\
\mathcal{I}_{p}^{RB}(\mathcal{B}_8) = 0 & \mathcal{I}_{p}^{RB}(\mathcal{B}_9) = 2 & \mathcal{I}_{p}^{RB}(\mathcal{B}_{10}) = 2 & \mathcal{I}_{p}^{RB}(\mathcal{B}_{11}) = 1 \\
\mathcal{I}_{c}^{RB}(\mathcal{B}_8) = 0 & \mathcal{I}_{c}^{RB}(\mathcal{B}_9) = 1 & \mathcal{I}_{c}^{RB}(\mathcal{B}_{10}) = 1 & \mathcal{I}_{c}^{RB}(\mathcal{B}_{11}) = 1 \\
\mathcal{I}_{mv}^{RB}(\mathcal{B}_8) = 0 & \mathcal{I}_{mv}^{RB}(\mathcal{B}_9) = 1 & \mathcal{I}_{mv}^{RB}(\mathcal{B}_{10}) = 2/3 & \mathcal{I}_{mv}^{RB}(\mathcal{B}_{11}) = 1
\end{array}$$

In result, we see that our alterations have improved all measures, s.t. they satisfy RC. Also, all measures except $\mathcal{I}_{mv}^{RB}$ could be improved s.t. they satisfy FE. Here, note however that the original $\mathcal{I}_{mv}$ is intentionally designed to consider all atoms (including those of simple facts) and is therefore inherently incompatible with FE by design. Additionally, the optional postulate RE is satisfied by our alteration of the original $\mathcal{I}_p$ measure.

Proposition 21 ([35]). *$\mathcal{I}_{d}^{RB}$, $\mathcal{I}_{\mathsf{MI}}^{RB}$, $\mathcal{I}_{c}^{RB}$ satisfy RC and FE, and do not satisfy RE.*

Proposition 22 ([35]). *$\mathcal{I}_{p}^{RB}$ satisfies RC, FE and RE.*

Proposition 23. *$\mathcal{I}_{mv}^{RB}$ satisfies RC, and does not satisfy FE and RE.*

Proofs for $\mathcal{I}_{d}^{RB}$, $\mathcal{I}_{\mathsf{MI}}^{RB}$ and $\mathcal{I}_{p}^{RB}$ can be found in [35]. The proofs for $\mathcal{I}_{mv}^{RB}$ and $\mathcal{I}_{c}^{RB}$ are analogous, cf. Section 4.1.1. Some other properties of the proposed measures are also discussed in [35]. However, a main result is that while classical inconsistency measures cannot be used plausibly as rule-based measures (cf. Proposition 20), we see that the proposed rule-based measures satisfy the intended behavior (Note that RE was introduced as an optional property, and thus all proposed measures satisfy the mandatory postulates of RC and FE. For the exception in $\mathcal{I}_{mv}^{RB}$, cf. the above discussion). This result is summarized in Tables 3.3 and 3.4. Such an assessment can provide useful insights for companies in the scope of resolving inconsistencies during run-time under a distinction between (indisputable) facts and rules, which was not possible with existing means.

$\mathcal{I}$	RC	FE	RE
$\mathcal{I}_d$	✗	✗	✗
$\mathcal{I}_{\mathsf{MI}}$	✗	✗	✗
$\mathcal{I}_p$	✗	✗	✗
$\mathcal{I}_c$	✗	✗	✗
$\mathcal{I}_{mv}$	✗	✗	✗

Table 3.3: Compliance with rationality postulates of the traditional inconsistency measures [35].

$\mathcal{I}^{RB}$	RC	FE	RE
$\mathcal{I}_d^{RB}$	✓	✓	✗
$\mathcal{I}_{\mathsf{MI}}^{RB}$	✓	✓	✗
$\mathcal{I}_p^{RB}$	✓	✓	✓
$\mathcal{I}_c^{RB}$	✓	✓	✗
$\mathcal{I}_{mv}^{RB}$	✓	✗	✗

Table 3.4: Compliance with rationality postulates of the proposed rule-based inconsistency measures [35].

3.2.3 Post-Execution Analysis

Post-Execution auditing is a retrospective analysis. Consequently, measuring inconsistency in this phase is essentially a combination of the preceding two compliance management strategies. For example, all process instances and the corresponding case-dependent facts can be inspected relative to the business rule base. This allows for a posteriori analysis as a basis for handling inconsistencies.

In future work, it may however also be beneficial to investigate the relations of different process instances. This could be useful for gaining a better overall understanding on the quality of business rules relative to the observed processes. Although this is beyond the scope of this thesis, we present some exemplary measures to this aim for clarification.

For further post-execution analysis of the run-time inconsistencies (and their relations) one could consider a "snapshot" of a rule base together with all facts that occurred during a process instance. In the following, we denote such a snapshot (i.e. union) of business rules and the facts referring to one process instance as a *case*. A baseline measure could then be a form of "drastic" measure, i.e., it would return 1 if there was at least one case, in which there was an inconsistency. A further possibility would be to count the number of cases, in which there was at least one inconsistency, or to count the number of overall inconsistencies across all cases. An interesting measure could also be to consider the (number of) distinct minimal inconsistent subsets across all cases.

While these were only some exemplary measures, it shows that considering not only a single run-time perspective, but also considering the interrelations of different cases can yield novel insights for experts in regard to understanding and resolving inconsistencies. Here, such a multi knowledge base inconsistency measurement (where the knowledge bases have a shared set of atoms) is a novel research stream for the field of inconsistency measurement, which should be investigated in future work.

It could also be beneficial for companies to consider temporal aspects in inconsistency measurement during the post-execution phase. Instead of viewing the union of case-dependent facts and business rules as a static snapshot, it might be beneficial to further consider the temporal order in which facts or events occurred. While this is beyond the scope of this thesis, it is noteworthy that results from drift detection could be used in this context to further understand errors in business rules. For instance, one could iterate through event logs using sliding window techniques as in [143]. Then, for every window, which contains certain case-dependent facts, one could count the number of inconsistencies arising from that set of facts in regard to the prevailing business rules. This would allow to plot the number of inconsistencies over time, e.g. a histogram where the x-axis corresponds to the individual windows, and the y-axis shows the number of inconsistencies for that window. Following [143], this could be beneficial to categorize the inconsistencies (arising over time), by presenting experts with such visualizations of the "development history" of inconsistencies, as in Figure 3.6.

Visualizations of the number of inconsistencies in cases *over time* allow to categorize the inconsistencies as ”drifts”, where a drift is defined as a deviation from the allowed behavior, i.e., fact occurrences that cause inconsistencies in regard to the prevailing business rules. Such a drift categorization could be useful to assess the severity of drifts, e.g., a case where the drift can be described as an outlier might be a negligible error which might have been caused by accident, while cases where there is an increase of inconsistencies in recent time could indicate that the observed facts cause drifts, and thus maybe the business rules should be carefully reconsidered. We would like to point out that this proposed method of visualizing inconsistencies over time would satisfy all requirements for process drift analysis proposed in [143], namely (R1) *drift identification*, (R2) *drift categorization*, (R3) *drill down and roll-up analysis*, (R4) *quantitative analysis*, (R5) *qualitative analysis*. The proposed approach could be used to identify and categorize drifts via the drift categorization in [143] or [48] (R1-R2). By means of post-execution (drift) analysis, inconsistencies can also be assessed by different levels of granularity, e.g. inconsistencies relative to an entire case (see exemplary measures above), or inconsistencies in regard to individual business rules (see Section 3.3.1 on culpability measurement) (R3). Here, the severity of the drift can be quantified (R4), and can be used to convey which business rules should be changed (R5, cf. Section 3.3.1 on culpability measurement).

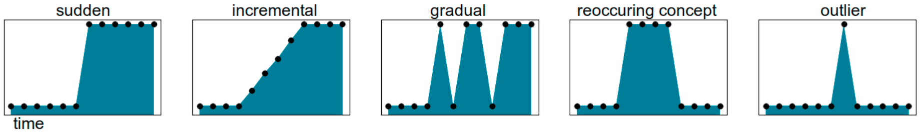


Figure 3.6: Exemplary plots for inconsistencies over time and corresponding drift types, taken from [143].

Many other inconsistency measures could be investigated, e.g., it might be interesting to consider the fraction of potential issues (identified at design-time) that became actual issues (later, during run-time). It seems this is an interesting area for future work with great potential benefits for companies in regard to further understanding inconsistencies in business rules. In general, it seems that considering not only different compliance management strategies individually, but also considering their interrelations can yield valuable insights for companies, which will be discussed in Section 3.5 on the proposed framework for handling inconsistency in business rule bases.

Summary: Measuring Inconsistency in Business Rule Bases

Due to several intricacies of BRM, existing inconsistency measures cannot be plausibly applied. We therefore developed new means for:

>Design-time inconsistency measurement (incl. postulate QI)

>Run-time inconsistency measurement (incl. postulates RC, FE and RE)

3.3 Resolving Inconsistency based on Culpability Measurement

So far, we considered measures that assess the *entire* rule base. In the context of inconsistency handling, this is, however, often not sufficient. Companies need to pin-point those rules in their rule bases that contribute towards the overall inconsistency, e. g. as a basis for inconsistency resolution. The field of inconsistency measurement also studies so-called *culpability measures.* These are essentially functions which assign a numerical value to individual formulas of a rule base, with the intuition that a higher value indicates a higher blame which a formula carries in the context of the overall inconsistency.

In the following, we discuss whether this element-based approach is plausible for an application in the domain of business rules management, and how it can be used as a basis for inconsistency resolution.

3.3.1 Culpability Measurement in Business Rule Bases

▷ *This section corresponds to the works [29, 35] by the author.*

As with inconsistency measures, there have been numerous proposals for specific culpability measures, see e. g. [87] for an overview. In this work, we consider the Shapley inconsistency value as proposed by HUNTER & KONIECZNY (2010), explained as follows: The Shapley inconsistency value is a generalized measure which is parametrized with arbitrary inconsistency measures. The intuition is that the *overall* degree of inconsistency, as quantified by an inconsistency measure $\mathcal{I}$, is distributed among all elements of a rule base, based on the "blame" that an individual element carries in the context of the overall inconsistency. Here, the Shapley inconsistency value uses notions from game theory to determine the blame – also referred to as payoff – that each formula carries w.r.t. the assessment of an arbitrary inconsistency measure. We consequently discuss this approach, as it allows to use arbitrary inconsistency measures, e.g., based on company needs (cf. the classification in Figure 3.2).

Definition 24 ([69])**.** Let $\mathcal{I}$ be an inconsistency measure, $\mathcal{B}$ be a rule base and $\alpha \in \mathcal{B}$. Then, the Shapley inconsistency value of α w.r.t. $\mathcal{I}$, denoted $S^{\mathcal{I}}_{\alpha}$ is defined via

$$S^{\mathcal{I}}_{\alpha}(\mathcal{B}) = \sum_{B \subseteq \mathcal{B}} \frac{(b-1)!(n-b)!}{n!} (\mathcal{I}(B) - \mathcal{I}(B \setminus \alpha))$$

where b is the cardinality of $B \subseteq \mathcal{B}$, and n is the cardinality of the initial $\mathcal{B}$.

The Shapley inconsistency value can be computed for any $\alpha \in \mathcal{B}$, in regard to an inconsistency measure $\mathcal{I}$. To this aim, all $B \subseteq \mathcal{B}$ are considered, also referred to as *coalitions*. For any B, the "change" of the overall inconsistency by removing α from B is calculated (cf. $\mathcal{I}(B) - \mathcal{I}(B \setminus \alpha)$). If α was not responsible for the overall inconsistency in B (w.r.t. $\mathcal{I}$), the right multiplicand will equate to zero, thus the blame of α w.r.t. B is 0. Otherwise, the change of the overall inconsistency in B after removing α (w.r.t. $\mathcal{I}$) is weighted by the cardinality of B relative to the cardinalities of $\mathcal{B}$, following results on game-theory in [112]. In this way, the blame for any $\alpha \in \mathcal{B}$ (w.r.t. $\mathcal{I}$) is the sum of blame over all coalitions, i.e., $S^{\mathcal{I}}_{\alpha}(\mathcal{B})$ is the amount of blame, that an element α carries in regard to $\mathcal{I}(\mathcal{B})$. In the following, we consider all elements α of a rule base as a vector $(\alpha_1, \alpha_2, ..., \alpha_n)$, and denote $S^{\mathcal{I}}(\mathcal{B})$ as the vector of corresponding Shapley inconsistency values, i. e., $S^{\mathcal{I}}(\mathcal{B}) = (S^{\mathcal{I}}_{\alpha_1}(\mathcal{B}), S^{\mathcal{I}}_{\alpha_2}(\mathcal{B}), ..., S^{\mathcal{I}}_{\alpha_n}(\mathcal{B}))$.

Example 25. We recall the rule base $\mathcal{B}_9 = \{a, a \to b, a \to \neg b\}$. We denote the blame that an element α carries in regard to a coalition, cf. Definition 24, as $blame(\alpha)$. Then, for the Shapley inconsistency value of a w.r.t. $\mathcal{I}_d$, we have

$B \subseteq \mathcal{B}$	b	n	$\mathcal{I}_d(B)$	$\mathcal{I}_d(B \setminus a)$	$blame(a)$
a	1	3	0	0	0
$a \to b$	1	3	0	0	0
$a \to \neg b$	1	3	0	0	0
$a, a \to b$	2	3	0	0	0
$a, a \to \neg b$	2	3	0	0	0
$a \to b, a \to \neg b$	2	3	0	0	0
$a, a \to b, a \to \neg b$	3	3	1	0	$\frac{(2!)(0!)}{3!} * (1-0) = \frac{1}{3}$
					$S^{\mathcal{I}}_{a}(\mathcal{B}_9) = \frac{1}{3}$ (sum)

The Shapley value of the element a (w.r.t. $\mathcal{I}_d$) for the rule base $\mathcal{B}_9$ is consequently the sum of all blame values over all coalitions. The calculations for the other elements in $\mathcal{B}_9$ are analogous, thus $S^{\mathcal{I}_d}(\mathcal{B}_9) = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$. The results can be interpreted s.t. the overall "blame mass" (derived from $\mathcal{I}_d$) in $\mathcal{B}_9$ is evenly distributed over all elements.

As we have discussed, existing inconsistency measures cannot be used for measuring (quasi-)inconsistency in an *entire* business rule base, due to conceptual mismatches. Here, it is also not feasible to "plug-in" existing inconsistency measures into the Shapley inconsistency value to determine element-based culpability in this setting, explained as follows:

During design-time, we see via Proposition 16 that virtually all classical inconsistency measures cannot be used here, e.g. given the design-time rule base $\mathcal{B}_{12} = \{a \to b; a \to \neg b\}$, for the Shapley inconsistency values w.r.t. $\mathcal{I}_d$, we have $S^{\mathcal{I}_d}(\mathcal{B}_{12}) = (0, 0)$, allthough $\mathcal{B}_{12}$ is quasi-inconsistent. For this setting, we are interested in culpability measures that can rank rules of a design-time rule base by the amount of blame they carry in the context of the overall quasi-inconsistency.

During run-time, we recall that this setting underlies the assumption of given, i.e. indisputable, facts. This has a major impact on culpability measurement, namely whether facts should even be considered as blameable. Indeed, deleting facts is not feasible in many cases here, thus methods are needed that can rank only the rules of a (run-time) rule base by their degree of culpability, as these are the actual modelling errors that the company needs to identify.

This calls for new means to assess element-culpability in business rule bases. Subsequently, we discuss how culpability can be measured during different compliance phases.

For the following discussion, we assume all used quasi-inconsistency measures satisfy QI, and all used rule-based inconsistency measures satisfy RC. Also, we assume the used measures satisfy the two basic properties of monotony and free-formula independence [70], which are usual properties satisfied by most measures [127, 130].

Monotony (MO) If $\mathcal{B} \subseteq \mathcal{B}'$ then $\mathcal{I}(\mathcal{B}) \leq \mathcal{I}(\mathcal{B}')$

Free-formula independence (IN) If $\alpha \in \mathsf{Free}(\mathcal{B})$ then
$\mathcal{I}(\mathcal{B}) = \mathcal{I}(\mathcal{B} \setminus \{\alpha\})$

MO demands that the addition of information cannot decrease the degree of inconsistency. IN states that free formulas should not affect the degree of inconsistency.

Design-time Culpability Measurement

We have presented adapted versions of classical inconsistency measures for the use-case of analyzing inconsistency in a set of rules only. As rule bases will only contain rules at design-time, the quasi-inconsistency measures can directly be applied to determine element-based culpability of business rules.

Definition 26 (Shapley quasi-inconsistency value)**.** Let $\mathcal{I}^Q$ be an quasi-inconsistency measure, $\mathcal{B}^{\setminus\mathcal{F}}$ be a rule base and $\alpha \in \mathcal{B}^{\setminus\mathcal{F}}$. Then, the Shapley quasi-inconsistency value of α w.r.t. $\mathcal{I}^Q$, denoted $S_\alpha^{\mathcal{I}^Q}$ is defined via

$$S_\alpha^{\mathcal{I}^Q}(\mathcal{B}^{\setminus\mathcal{F}}) = \sum_{B \subseteq \mathcal{B}^{\setminus\mathcal{F}}} \frac{(b-1)!(n-b)!}{n!}(\mathcal{I}^Q(B^{\setminus\mathcal{F}}) - \mathcal{I}^Q(B^{\setminus\mathcal{F}} \setminus \alpha))$$

where b is the cardinality of B, and n is the cardinality of $\mathcal{B}$.

For culpability measurement, we consider the Shapley inconsistency value as it allows to utilize arbitrary inconsistency measures. We would however like to mention that we proposed an adaptation of one other culpability measure, namely the $MIS_{\#}$ measure as proposed in [70]. This is an important measure as it can be considered as a scoring function which indicates how many minimal issues of a rule base will can be eliminated by deleting a rule α. As this measure will be used in the course of this work, we briefly discuss the specific adaptation and conclude with an example showing the behavior of the proposed culpability measures.

Definition 27 (Cardinality-Based Culpability Measure, [28])**.** Define the *cardinality based culpability measure* $\mathcal{C}_{\#}$ via

$$\mathcal{C}_{\#}(\mathcal{B}, \alpha) = |\{M \in \mathsf{MinIssues}(\mathcal{B})\}|\alpha \in M|$$

This measure counts the number of minimal issues that a rule α appears in.

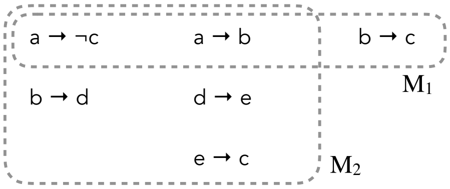
Example 28. Consider the rule base $\mathcal{B}_{13}$, with

$$\mathcal{B}_{13} = \{a \to b; b \to c; b \to d; a \to \neg c; d \to e; e \to c\}$$

with

$$\begin{aligned}
\mathcal{F}(\mathcal{B}_{13}) &= \emptyset \\
\mathsf{MI}(\mathcal{B}_{13}) &= \emptyset \\
\mathsf{MinIssues}(\mathcal{B}_{13}) &= \{M_1, M_2\} \\
M_1 &= (\{a \to b; b \to c\}, \{a\}, \{a \to \neg c\}, \{a\}) \\
M_2 &= (\{a \to b; b \to d; d \to e; e \to c\}, \{a\}, \{a \to \neg c\}, \{a\})
\end{aligned}$$

(visually)



Then we have

$r_1 : a \to b$	$\mathcal{C}_{\#}(\mathcal{B}_{13}, r_1) = 2$	$S_{r_1}^{\mathcal{I}_d^Q}(\mathcal{B}_{13}) = 0.3\overline{6}$	$S_{r_1}^{\mathcal{I}_{\mathsf{MI}}^Q}(\mathcal{B}_{13}) = 0.5\overline{3}$	$S_{r_1}^{\mathcal{I}_p^Q}(\mathcal{B}_{13}) = 1.\overline{6}$
$r_2 : b \to c$	$\mathcal{C}_{\#}(\mathcal{B}_{13}, r_2) = 1$	$S_{r_2}^{\mathcal{I}_d^Q}(\mathcal{B}_{13}) = 0.1\overline{6}$	$S_{r_2}^{\mathcal{I}_{\mathsf{MI}}^Q}(\mathcal{B}_{13}) = 0.\overline{3}$	$S_{r_2}^{\mathcal{I}_p^Q}(\mathcal{B}_{13}) = 0.\overline{6}$
$r_3 : a \to \neg c$	$\mathcal{C}_{\#}(\mathcal{B}_{13}, r_3) = 2$	$S_{r_3}^{\mathcal{I}_d^Q}(\mathcal{B}_{13}) = 0.3\overline{6}$	$S_{r_3}^{\mathcal{I}_{\mathsf{MI}}^Q}(\mathcal{B}_{13}) = 0.5\overline{3}$	$S_{r_3}^{\mathcal{I}_p^Q}(\mathcal{B}_{13}) = 1.\overline{6}$
$r_4 : b \to d$	$\mathcal{C}_{\#}(\mathcal{B}_{13}, r_4) = 1$	$S_{r_4}^{\mathcal{I}_d^Q}(\mathcal{B}_{13}) = 0.0\overline{3}$	$S_{r_4}^{\mathcal{I}_{\mathsf{MI}}^Q}(\mathcal{B}_{13}) = 0.2$	$S_{r_4}^{\mathcal{I}_p^Q}(\mathcal{B}_{13}) = 0.\overline{6}$
$r_5 : d \to e$	$\mathcal{C}_{\#}(\mathcal{B}_{13}, r_5) = 1$	$S_{r_5}^{\mathcal{I}_d^Q}(\mathcal{B}_{13}) = 0.0\overline{3}$	$S_{r_5}^{\mathcal{I}_{\mathsf{MI}}^Q}(\mathcal{B}_{13}) = 0.2$	$S_{r_5}^{\mathcal{I}_p^Q}(\mathcal{B}_{13}) = 0.\overline{6}$
$r_6 : e \to c$	$\mathcal{C}_{\#}(\mathcal{B}_{13}, r_6) = 1$	$S_{r_6}^{\mathcal{I}_d^Q}(\mathcal{B}_{13}) = 0.0\overline{3}$	$S_{r_6}^{\mathcal{I}_{\mathsf{MI}}^Q}(\mathcal{B}_{13}) = 0.2$	$S_{r_6}^{\mathcal{I}_p^Q}(\mathcal{B}_{13}) = 0.\overline{6}$
	Sum:	$\mathcal{I}_d^Q(\mathcal{B}_{13}) = 1$	$\mathcal{I}_{\mathsf{MI}}^Q(\mathcal{B}_{13}) = 2$	$\mathcal{I}_p^Q(\mathcal{B}_{13}) = 6$

This example shows that the Shapley quasi-inconsistency value, as well as the adapted $\mathcal{C}_{\#}$ can be used to quantify element-based culpability of business rules at design-time.

Run-time Culpability Measurement

As with quasi-inconsistency measures, it is theoretically also possible to "plug-in" the proposed rule-based inconsistency measures into the Shapley inconsistency value.

Definition 29 (Rule-based Shapley inconsistency value, [35])**.** Let $\mathcal{I}^{RB}$ be a rule-based inconsistency measure, $\mathcal{B}$ be a rule base and $\alpha \in \mathcal{B}$. Then, the rule-based Shapley inconsistency value of α w.r.t. $\mathcal{I}^{RB}$, denoted $S_{\alpha}^{\mathcal{I}^{RB}}$ is defined via

$$S_{\alpha}^{\mathcal{I}^{RB}}(\mathcal{B}) = \sum_{B \subseteq \mathcal{B}} \frac{(b-1)!(n-b)!}{n!}(\mathcal{I}^{RB}(B) - \mathcal{I}^{RB}(B \setminus \alpha))$$

where b is the cardinality of B, and n is the cardinality of $\mathcal{B}$.

Example 30. Consider the rule base $\mathcal{B}_{14} = \{a; a \to b; a \to \neg b; \neg a\}$. Then, for the Shapley inconsistency values w.r.t. $\mathcal{I}_d^{RB}$ and $\mathcal{I}_{\mathsf{MI}}^{RB}$, we have that $S_a^{\mathcal{I}_d^{RB}}(\mathcal{B}) = S_a^{\mathcal{I}_{\mathsf{MI}}^{RB}}(\mathcal{B}_{14}) = \frac{1}{12} + \frac{1}{4} = \frac{1}{3}$. Also, we have that $S_{\neg a}^{\mathcal{I}_d^{RB}}(\mathcal{B}_{14}) = S_{\neg a}^{\mathcal{I}_{\mathsf{MI}}^{RB}}(\mathcal{B}_{14}) = 0$. Thus, we have $S^{\mathcal{I}_d^{RB}}(\mathcal{B}_{14}) = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3}, 0)$ and $S^{\mathcal{I}_{\mathsf{MI}}^{RB}}(\mathcal{B}_{14}) = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3}, 0)$.

Example 30 shows that the gist of the RC property transfers to the element-based perspective: Only formulas that are part of at least one MI (which itself contains at least one rule) are assigned blame. However, we see that the blame is equally distributed over $\{a, a \to b, a \to \neg b\}$ in both assessments. As facts are viewed as indisputable, this behavior is undesirable. Facts should not be assigned with any blame value, as they have to be kept as-is. To capture this requirement, we therefore propose a new property of fact-minimality [35].

- **Fact-Minimality** $S_f^{\mathcal{I}^{RB}}(\mathcal{B}) = 0$ for any fact f in $\mathcal{B}$.

This extends the minimality property originally proposed in [69], which is necessary for the intended use-case of viewing facts as indisputable. As evidenced by Example 30, the Shapley inconsistency value does not satisfy fact-minimality. Therefore, it is not plausible to apply the Shapley inconsistency value in our use case. We therefore propose an *adjusted* Shapley inconsistency value. The intuition of our approach is as follows [35]:

The original Shapley inconsistency value assigns responsibilities to a number of formulas (or players) in a coalition. Currently, there is no distinction between facts and rules. Following our use case, the idea is to shift the blame from facts to all rules which are part of the inconsistency for that coalition. For that, some notation is necessary.

Definition 31 ([35])**.** Let $\mathcal{I}^{RB}$ be a rule-based inconsistency measure, $\mathcal{B}$ be a rule base and $\alpha \in \mathcal{B}$. Then, the individual Shapley inconsistency coalition value of α w.r.t. $\mathcal{I}^{RB}$ (in a coalition $B \subseteq \mathcal{B}$), is defined via

$$\mathit{CoalPayoff}_{\alpha,\mathcal{B}}^{\mathcal{I}^{RB}}(B) = \frac{(b-1)!(n-b)!}{n!}(\mathcal{I}^{RB}(B) - \mathcal{I}^{RB}(B \setminus \alpha))$$

where b is the cardinality of B, and n is the cardinality of $\mathcal{B}$.

Blame should only be assigned to rules. In turn, the amount of blame that "falls upon" facts from any coalition should be shifted onto the *blamable* rules of that coalition, i. e., the rules which contribute towards the inconsistency for that coalition. We denote the blame that is shifted from facts to the individual blamable rules, as an *additional payoff*.

Definition 32 ([35])**.** Let $\mathcal{I}^{RB}$ be a rule-based inconsistency measure and $\mathcal{B}$ be a rule base, denote the additional blame for a *rule* $r \in \mathcal{R}(\mathcal{B})$ in any coalition $B \subseteq \mathcal{B}$ as

$$
\begin{aligned}
&\mathit{AddPayoff}^{\mathcal{I}^{RB}}_{r,\mathcal{B}}(B) \\
&\quad = \begin{cases} 0 & \text{if } r \in \mathsf{Free}(B) \\ \frac{\sum_{f \in \mathcal{F}(B)} \mathit{CoalPayoff}^{\mathcal{I}^{RB}}_{f,\mathcal{B}}(B)}{|r' \in \mathcal{R}(B) \text{ s.t. } r' \notin \mathsf{Free}(B)|} & \text{otherwise} \end{cases}
\end{aligned}
$$

Example 33. Consider again $\mathcal{B}_9 = \{a, a \to b, a \to \neg b\}$ and $\mathcal{B}_{11} = \{a, a \to b, \neg b\}$. According to the original Shapley value, the blame is evenly distributed for both rule bases. In $\mathcal{B}_9$, given the premise of indisputable facts, a is not to blame for the overall inconsistency. Rather, the blame value of a should be evenly distributed among $a \to b$ and $a \to \neg b$, as both these formulas evenly contribute towards the inconsistency. Next, in $\mathcal{B}_{11}$, both a and $\neg b$ are not to blame in our use case. Here, the blame values of a and $\neg b$ should be transferred to $a \to b$. This is directly in line with the intuition of RE, as for $\mathcal{B}_{11}$, the only way to resolve the inconsistency would be to remove $a \to b$ (thus all blame is relocated to that rule), but for $\mathcal{B}_9$, one can delete either of the two rules (thus the blame is distributed among both rules). In result, for each coalition, the blame is shifted from facts to the blameable rules via the additional payoff.

We are now ready to define the adjusted Shapley inconsistency value.

Definition 34 ([35])**.** Let $\mathcal{I}^{RB}$ be a rule-based inconsistency measure, $\mathcal{B}$ be a rule base and $\alpha \in \mathcal{B}$. Then, the adjusted Shapley inconsistency value of α w.r.t. $\mathcal{I}^{RB}$, denoted $S{*}^{\mathcal{I}^{RB}}_{\alpha}$ is defined via

$$
\begin{aligned}
&S{*}^{\mathcal{I}^{RB}}_{\alpha}(\mathcal{B}) \\
&\quad = \begin{cases} 0 & \text{if } \alpha \in \mathcal{F}(\mathcal{B}) \\ \sum\limits_{B \subseteq \mathcal{B}} (\mathit{CoalPayoff}^{\mathcal{I}^{RB}}_{\alpha,\mathcal{B}}(B) + \mathit{AddPayoff}^{\mathcal{I}^{RB}}_{\alpha,\mathcal{B}}(B)) & \text{otherwise} \end{cases}
\end{aligned}
$$

The adjusted Shapley value assigns the value of 0 to all facts, and computes the blame value of all rules taking into consideration the additional payoff.

Example 35. Consider the rule bases $\mathcal{B}_9 = \{a; a \to b; a \to \neg b\}$ and $\mathcal{B}_{11} = \{a; a \to b; \neg b\}$. Then for the *adjusted* Shapley inconsistency values w.r.t. $\mathcal{I}^{RB}_d$, we have that $S^{\mathcal{I}^{RB}_d}_a(\mathcal{B}_9) = 0, S^{\mathcal{I}^{RB}_d}_{a \to b}(\mathcal{B}_9) = \frac{1}{3}(+\frac{1}{3}/2) = \frac{1}{2}$, and $S^{\mathcal{I}^{RB}_d}_{a \to \neg b}(\mathcal{B}_9) = \frac{1}{3}(+\frac{1}{3}/2) = \frac{1}{2}$. Also, we have that $S^{\mathcal{I}^{RB}_d}_a(\mathcal{B}_{11}) = 0, S^{\mathcal{I}^{RB}_d}_{\neg b}(\mathcal{B}_{11}) = 0$, and $S^{\mathcal{I}^{RB}_d}_{a \to \neg b}(\mathcal{B}_{11}) = \frac{1}{3}(+\frac{2}{3}) = 1$. Thus, we have that $S^{\mathcal{I}^{RB}_d}(\mathcal{B}_9) = (0, \frac{1}{2}, \frac{1}{2})$ and $S^{\mathcal{I}^{RB}_d}(\mathcal{B}_{11}) = (0, 1, 0)$, which is directly in line with RE.

Proposition 36 ([35]). *The adjusted Shapley value satisfies Fact-Minimality.*

Proposition 36 (cf. also Proposition 55) shows that our approach follows the same gist as the original Shapley inconsistency value, but shifts the blame from facts to blamable rules. This is necessary for culpability measurement in run-time compliance management, as companies need to identify erroneously modelled rules relative to a given fact input. Some other properties are also discussed in [35].

Post-Execution Culpability Measurement

As mentioned in Section 3.2.3, post-execution analysis is beyond the scope of this thesis. However, it is worth noting that post-execution culpability measurement may yield further insights for companies in the scope of understanding the interrelations of inconsistencies.

For instance, considering again "cases" as defined in Section 3.2.3, then, for every rule one could measure the number of cases in which the respective rule is part of at least one inconsistency. Likewise, one could count the number of inconsistencies that a business rule is associated with across all cases. As mentioned, such a form of multi knowledge base culpability measurement could be interesting for future work and would contribute towards means for qualitative analysis (i.e., conveying to the experts which business rules to change) in the scope of drift analysis (cf. Section 3.2.3 and 3.5 for further discussion).

3.3.2 Inconsistency Resolution by means of Culpability Measurement

▷ *This section corresponds to the works [28, 29] by the author.*

To resolve inconsistency in a rule base, there are generally two possibilities [42]: One approach is to move elements to a new rule base iteratively, warranted that their introductions do not cause any inconsistencies. A second approach to resolve inconsistencies is to iteratively delete elements from the original rule base until it is consistent. Grant and Hunter (2011) denote this latter approach as *stepwise inconsistency resolution*. In [42], those authors use the first approach to resolve inconsistencies in a (Declare) rule base. In this work, we present a solution for using the second approach, which allows to eliminate inconsistencies while considering all conflicts between rules, and with the goal of mitigating information loss (in terms of deleted elements from the rule base).

Let a rule base $\mathcal{B}$, individual rules $r \in \mathcal{B}$ can be deleted, defined via:

$$deletion(r) = \mathcal{B} \setminus r$$

In the following, we denote deletion as an operation o, for readability. Stepwise inconsistency resolution by deletion is thus a sequence $S =< o_1, ..., o_n >$. We denote $o_i(\mathcal{B})$ as the rule base obtained by applying operations $o_1, ..., o_n$ to $\mathcal{B}$.

We denote $r^S =< r_1, ..., r_n >$ as the individual rules deleted in the sequence S via the operations $o_1, ..., o_n$.

The challenge is to find suitable elements which to attend to, such that the sequence S results in a low amount of information loss (e.g. deleted elements). Here, we propose to use culpability measures as a driver for inconsistency resolution. For example, the $\mathcal{C}_{\#}$ is essentially a scoring function which quantifies how many quasi-inconsistent subsets can be resolved, if a constraint is deleted. We therefore propose a culpability-based ranking to determine which elements to attend to.

Definition 37 (Culpability Ranking, [29]). Let $\mathcal{C}$ be a culpability measure and $\mathcal{B}$ be a rule base, then a culpability ranking w.r.t. $\mathcal{C}$ over all rules $r \in \mathcal{B}$ is any ranking $\langle r_1, ..., r_n \rangle$ that satisfies $\mathcal{C}(r_1) \geq ... \geq \mathcal{C}(r_n)$.

This ranking sorts all rules in $\mathcal{B}$ based on their inconsistency value w.r.t. a culpability measure $\mathcal{C}$ and presents to the modeler a prioritized list of which elements to attend to. As we will show in Chapter 4 (Evaluation), such a ranking can actually help modellers to understand and resolve inconsistencies in rule bases in a faster time, with a better understanding accuracy and with less mental effort needed, and can therefore be a driver for handling inconsistencies in rule bases.

Regarding (semi-) automated inconsistency resolution during modelling, we also propose an approximation algorithm for the problem of restoring consistency by removing a minimal number of rules. This algorithm iteratively performs the steps of analysis, ranking and the operation "deletion" until the rule base is consistent again, shown in Algorithm 1 [28].

Algorithm 1: Stepwise Inconsistency Resolution via deletion

```
   Input  : Quasi-Inconsistent Rule Base $\mathcal{B}$
   Output: Non Quasi-Inconsistent Rule Base $\mathcal{B}$
1  $minIssues \leftarrow$ MinIssues($\mathcal{B}$)
2  $tempCandidate \leftarrow \varnothing$
3  $tempValue \leftarrow 0$
4  while $|minIssues| > 0$ do
5      foreach $r : \mathcal{R}(\mathcal{B})$ do
6          $culpability \leftarrow C_{\#}(r)$
7          if $culpability > tempValue$ then
8              $tempCandidate \leftarrow r$
9              $tempValue \leftarrow culpability$
10     $\mathcal{B} \leftarrow \mathcal{B} \setminus tempCandidate$
11     $minIssues \leftarrow$ MinIssues($\mathcal{B}$)
12 return $\mathcal{B}$
```

The algorithm takes as input parameter a design-time rule base. First, minimal issues are computed in line 1 (we will discuss this further in Section 3.4).

The algorithm then identifies the rules with the highest culpability (lines 6-9), and iteratively deletes these rules (line 10). To clarify, the order in which rules are deleted is to be viewed with caution, e.g. as a recommendation. This recommendation is based on the $\mathcal{C}_{\#}$ measure. This allows to propose an order of deletions such that the highest possible amount of minimal issues is deleted in each step, while deleting only one constraint.

We continue to discuss the *correctness* of Algorithm 1 [104]. Here, we consider an algorithm to be correct if it satisfies the properties of *soundness* and *completeness*, following [104]. We define these two properties as follows:

- Given a problem P and an algorithm that tries to solve this problem, an algorithm is sound if the solution returned by the algorithm is correct.
- Given a problem P and an algorithm that tries to solve this problem, an algorithm is complete if it returns a solution if one exists, or reports failure, if no solution exists.

Theorem 38. *Algorithm 1 is sound for the problem of removing all minimal issues.*

Proof. Assume Algorithm 1 is not sound. Then one of the following has to be true:

1. The returned rule base is not derived from the input $\mathcal{B}$ by deletion of elements.
2. The returned rule base is quasi-inconsistent.

Regarding case 1, observe that the only modification to the input $\mathcal{B}$ is performed in line 10, where an element is deleted. This is performed as many times as the while loop from line 4 is performed. Then, $\mathcal{B}$ is returned in line 12. Thus, Algorithm 1 returns a rule base $\mathcal{B}'$ which is derived by applying a sequence of deletions $o_1, ..., o_n$ to the input $\mathcal{B}$. This contradicts case 1. Regarding case 2, we recall from Definition 12 that a rule base $\mathcal{B}$ is quasi-inconsistent iff there exists a minimal issue in $\mathcal{B}$. It follows that in a non quasi-inconsistent rule base $\mathcal{B}'$ we have that $\text{MinIssues}(\mathcal{B}') = \emptyset$. The algorithm deletes elements from the input rule base while the number of minimal issues is > 0 (line 4). Thus, it terminates if the number of minimal issues is 0 (which is always reached, in the extreme case deleting all but one formula). This contradicts case 2. We reach a contradiction in all cases, hence, Algorithm 1 is sound. □

Corollary 39. *Let a rule base $\mathcal{B}$, $\text{MinIssues}(\mathcal{B}) = \emptyset$ iff $\mathcal{B}$ is not quasi-inconsistent.*

Theorem 40. *Algorithm 1 is complete for the problem of removing all min. issues.*

Proof. Observe that via Corollary 39, any rule base $\mathcal{B}$ can be transformed into a non quasi-inconsistent rule base by deleting all elements. Thus, there is always a solution for Algorithm 1 and there cannot exist a case where failure should be reported.

Then, Algorithm 1 always returns a rule base $\mathcal{B}$, explained as follows. Let an input rule base $\mathcal{B}$ (cf. algorithm signature). In lines 1-3, no alterations are made to the input of $\mathcal{B}$. Then, in line 4, a loop is entered. In this loop, a formula is selected (line 9). Up to this line, no alteration is made to $\mathcal{B}$. Then, in line 10, an element is deleted from $\mathcal{B}$. The while loop is terminated if there are no more minimal issues in $\mathcal{B}$, i.e., $\mathcal{B}$ is not quasi-inconsistent. No alterations to $\mathcal{B}$ are made in line 11 and $\mathcal{B}$ is always returned in line 12. Thus, Algorithm 1 is complete by invariance. □

We would like to point out that our algorithm can easily be extended to accept arbitrary culpability measures, based on company needs (line 6). Also, while we envisage to apply this algorithm during design-time (in order to eliminate quasi inconsistencies), our algorithm can also be used in classical inconsistency measurement (using minimal inconsistent sets instead of minimal issues).

Interestingly, the approximation quality of Algorithm 1 can be assured by a constant factor of 2.

Theorem 41. *Algorithm 1 is a 2-factor approximation algorithm for the problem of removing all minimal issues while removing a minimal number of formulas.*

Proof. Let a rule base $\mathcal{B}$, Algorithm 1 utilizes the $\mathcal{C}_{\#}$ value to identify formulas that will resolve a maximal number of minimal issues in $\mathcal{B}$, if deleted. The maximal number of formulas that can be deleted in any circumstance via Algorithm 1 is $|\mathcal{B}| - 1$. Regarding optimal solutions, the most number of formulas would need to be deleted in a case where we have multiple sets of completely isolated minimal issues each containing two complementary rules. In such a case, an optimal solution would require at most to delete $\frac{|\mathcal{B}|}{2}$ formulas in order to resolve all minimal issues. If an optimal solution deletes $\frac{|\mathcal{B}|}{2} - i$ formulas, this means there are either $2i$ free formulas (or formulas part of a minimal issue with more than 2 rules) or at least i overlaps. In either case, Algorithm 1 would delete only at most $|\mathcal{B}| - 1 - 2i$ formulas (as either the free formulas do not need to be deleted or the overlap allows to resolve at least two minimal issues by deleting one formula). Consequently, for an optimal number of formulas $opt = \frac{|\mathcal{B}|}{2} - i$ that must be deleted to resolve all minimal issues, Algorithm 1 would delete at most $sol = |\mathcal{B}| - 1 - 2i$ formulas, thus, $sol \leq 2 * opt$. ☐

Corollary 42. *Let a rule base $\mathcal{B}$, the maximum number of elements that need to be deleted in order to resolve all minimal issues is $\frac{|\mathcal{B}|}{2}$.*

Intuitively, for culpability measures that satisfy minimality [69], the bounded error of Algorithm 1 is the number of non-free formulas in a rule base $\mathcal{B}$ (nff), i.e., given a minimal number opt of elements needed to eliminate quasi-inconsistency in a rule base $\mathcal{B}$, the number of elements sol deleted by Algorithm 1 is $\leq opt + \mathit{nff}$.

The above discussion shows that the proposed algorithm can be used as a heuristic to resolve minimal issues with a relative performance guarantee of factor 2. Future work should further investigate determining resolution strategies.

Summary: Resolving Inconsistency based on Culpability Measurement

>We developed means for culpability measurement in BRM (incl. postulate Fact-Minimality)

>The proposed culpability ranking approach presents companies a prioritized list in which elements should be attended to

>A first approximation algorithm for restoring consistency while deleting a minimal number of elements was proposed

3.4 Algorithmic Considerations (for Minimal Issues)

A substantial part of applying our algorithm in an industrial use-case is the actual computation of minimal issues, which we discuss in the following.

The use-cases of run-time compliance management and post-execution compliance management build on classical minimal inconsistent sets, as facts are known during these phases, and classic-logical inference is possible here. Existing results and solvers can therefore directly be applied in this setting, cf. e.g. [128] for a library for (classical) inconsistency measurement.

On the contrary, analyzing rule sets at design time is a novel contribution conceptually, and thus we also propose new means to compute minimal issues. As a central object of study, we utilize a so-called *reactive entailment graph* (REG). In the following, we discuss the REG concept and our proposed algorithm.

3.4.1 Reactive Entailment Graph

▷ *This section corresponds to the works [28, 31] by the author.*

Definition 43 (Reactive Entailment Graph, [31])**.** Given a rule base $\mathcal{B}$, its reactive entailment graph (REG) is defined as a graph $G_{\mathcal{B}} = (A, E, l)$. A is a set of nodes, where $n \in A$ iff for any $r \in \mathcal{B} : \mathit{body}(r) = n$, and $\overline{n} \in A$ iff for any $r \in \mathcal{B} : \mathit{head}(r) = n$. E is the set of directed edges between elements in A, where $(b, h) \in E$ iff for any $r \in \mathcal{B} : b = \mathit{body}(r)$ and $h = \mathit{head}(r)$. l is a labelling function $l : A \rightarrow 2^{\mathcal{L}(\mathcal{B})}$ that assigns literals to a node as atomic propositions.

The reactive entailment graph is a graph representation of the rule form in (1.1) [31]. For example, the rule $a \rightarrow b$ is represented as two nodes a and b, related by an edge. A rule $a, e \rightarrow b$ can be represented as two nodes labelled a, e and b, related by an edge. Note that via the considered rule form in (1.1), there is always only one literal as a rule conclusion. For details on rule formalisms with different rule types, such as Declare, please see the below discussion.

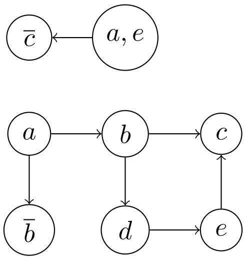
An central notion is that REG considers two "forms" of nodes: Given the general business rule form in (1.1), a premise can either entail a conclusion x, or, via negation, a premise can entail a conclusion $\neg x$. Here, we distinguish between so-called *permitting* and *prohibiting* deontic conclusions, where the form without negation "permits" x, and $\neg x$ can be seen as a "prohibition" of x. This is denoted in the REG via an overline symbol in the node label, e.g.$\overline{x}$.

We continue with an example clarifying the construction of a reactive entailment graph.

Example 44. Consider the exemplary rule base $\mathcal{B}_{15}$, with

$$\mathcal{B}_{15} = \{a \rightarrow b; b \rightarrow c; b \rightarrow d; a, e \rightarrow \neg c; a \rightarrow \neg b; d \rightarrow e; e \rightarrow c\}.$$

which yields the following reactive entailment graph:



This graph encodes the relations between business rules. For example, it can be seen that $a \rightarrow b$, and $a \rightarrow \bar{b}$ (i.e., $a \rightarrow \neg b$). This encodes that the activation a *permits* b, resp. *prohibits* b, which is quasi-inconsistent. The two forms of tasks are included to allow for a targeted scanning for potential inconsistencies, i.e., by searching for pairs of nodes n and n', where $n = \overline{n'}$.

For an edge $e \in E$, we denote the corresponding rule as $e^{\mathcal{R}}$. Given a path p, i.e., a sequence of edges in the REG, we denote the set of corresponding rules captured by p as $p^{\mathcal{R}}$. Also, we denote the REG of a rule base $\mathcal{B}$ as REG($\mathcal{B}$).

Regarding the construction of an REG for rules based on the general form in (1.1), note that the corresponding REG can grow exponentially with the number of literals (and nodes can have any label in $2^{\mathcal{L}(\mathcal{B})}$). Furthermore, note that the presented definition of an REG can also be extended to other rule formalisms. For example, in [31], an REG for declarative process models is presented. Here, edge types are introduced to capture specific intricacies of the Declare language. Also, in Declare, a core set of Declare rule only have one literal as a rule premise. This greatly lowers the computational costs for constructing an REG for a Declare rule base, as there are only $\mid \mathcal{L}(\mathcal{B}) \mid$ nodes in this case. We refer the reader to [31] for further details on reactive entailment graphs for Declare rule bases.

3.4.2 Algorithm for Computing Minimal Issues

▷ *This section corresponds to the works [28, 31] by the author.*
▷ *! An earlier version of Algorithm 2 was proposed in the Master Thesis by Matthias Deisen [40] (supervised by myself and Patrick Delfmann)*

Following Definition 12, quasi-inconsistency can only occur if [31]:

1. There is at least one literal Δ.
2. Δ is the conclusion of at least one rule r.
3. There exists a rule r' with conclusion $\neg\Delta$.

Furthermore, the constraints r and r' have to be activated simultaneously, thus

4. r and r' have to be activated by the activation sets x_1, resp. x_2, s.t. $x_1 \subseteq x_2$ (or by a chain of rules which themselves are activated by x_1, resp. x_2).

Algorithm 2 [31] computes the minimal issues of a rule base by exploiting the reactive entailment graph to search for subsets of the rule base satisfying 1-4. In the following, we explain our algorithm based on $\mathcal{B}_{15}$ and the corresponding REG($\mathcal{B}_{15}$) from Example 44.

```
Algorithm 2: Computation of minimal issues
    Input  : REG($\mathcal{B}$)
    Output: MinIssues($\mathcal{B}$)
1   $minIssues \leftarrow \emptyset$
2   $compConstraints = findElementsWithComplements(\mathsf{REG}(\mathcal{B}))$
3   foreach $r : compConstraints$ do
4       $\omega \leftarrow head(r)$
5       $\mathbf{P} = \bigcup_{a \in A} findPaths(a, \overline{\omega})$
6       foreach $P$:P do
7           $S \leftarrow source(P)$
8           $\mathbf{P'} = \bigcup_{s \subseteq S} findPaths(s, \omega)$
9           foreach $P'$:P' do
10              if $S \cup P^{\mathcal{R}} \cup P'^{\mathcal{R}} \models \perp$ then
11                  $minIssues \leftarrow minIssues \cup (p^{\mathcal{R}}, S, p'^{\mathcal{R}}, s)$
12  return $minIssues$
```

A set to store minimal issues (*minIssues*) is initialized in line 1. Then, an initial scan for inconsistencies is conducted by identifying those node n which are a complement to another node n' (All such nodes are stored in a set *compConstraints* in line 2). Note that if this set is empty, there is no quasi-inconsistency. In the example, there are two contradictory conclusions, namely b vs $\overline{b}$ and c vs $\overline{c}$ (cf. the corresponding REG in Example 44), thus $compConstraints = \{b, \overline{b}, c, \overline{c}\}$. We focus on c in the following, i.e., we assume the current iterated r_i in line 3 is c. The algorithm then continues to search for all self avoiding paths from all nodes of the REG to the inverse of the current r_i, stored in P (i.e., in our example the algorithm searches for all self avoiding paths from any node to $\overline{c}$ in the REG). Note that self avoiding paths are needed such that an infinite amount of paths is avoided due to potential loops[4]. In our example, these paths are:

$$Source : a, e \qquad\qquad Path : a, e \rightarrow \overline{c}$$

Subsequently, the algorithm iterates all found paths and stores the respective source in S. Then all self avoiding paths from every $s \subseteq S$ to the original ω are computed [5]. In the example, we would thus search for all paths from $a; e; a, e$ to c, i.e., we have

---

[4] Please note that in the implementation, this was implemented by a recursive search, which starts from the node $\overline{c}$ and recursively subsverses back all antecedents.

[5] This is necessary in regard to the subsumption condition in 4.). For example, consider a rule base $\{b \rightarrow a; b, d \rightarrow \neg a\}$. Assume $S = \{b, d\}$, and we are searching for paths that lead to the inverse of $\neg a$, i.e., to a. Only searching for a path $b, d \rightarrow a$ would not return any matches. However,

$$\begin{array}{ll}
Source : a & Path : a \rightarrow b, b \rightarrow c \\
Source : a & Path : a \rightarrow b, b \rightarrow d, d \rightarrow e, e \rightarrow c \\
Source : e & Path : e \rightarrow c
\end{array}$$

Here, the algorithm also verifies whether the union of all rules pertaining to a found path P, the rules pertaining to path P' and the activation set S are inconsistent, in which case we have found a minimal quasi-inconsistent subset. In the example, the verification in line 10 would thus be

$$\begin{array}{ll}
(1) & a, e \cup \{a, e \rightarrow \neg c\} \cup \{a \rightarrow b, b \rightarrow c\} \models \perp \\
(2) & a, e \cup \{a, e \rightarrow \neg c\} \cup \{a \rightarrow b, b \rightarrow d, d \rightarrow e, e \rightarrow c\} \models \perp \\
(3) & a, e \cup \{a, e \rightarrow \neg c\} \cup \{e \rightarrow c\} \models \perp
\end{array}$$

Note that we do not have to search for paths from all nodes of the REG to c, as for all other nodes in the REG, there is no path from that node to $\neg c$, thus the other comparisons in the example would all be similar to

$$b \cup \emptyset \cup \{b \rightarrow c\} \models \perp ?$$

(false in all cases)

Concluding the example, for the cases where the condition in line 10 is true, we have successfully found *minimal issues* based on the reactive entailment graph. While this algorithm computes minimal issues (i.e. *actual* issues), note that the algorithm can easily be extended to compute *potential* issues by augmenting multiple activation sets in line 10.

As with Algorithm 1, we now turn to the correctness of Algorithm 2.

Theorem 45. *Algorithm 2 is sound for the problem of computing all minimal issues if there are no self-contradictory (sequences of) rules*[6].

Proof. Assume Algorithm 2 is not sound. Then one of the following cases has to be true:

1. There exists an element of the returned set which is not a minimal issue in $\mathcal{B}$.

---

the shown rule base is quasi-inconsistent, which can be detected by also searching for the path $b \rightarrow a$. Note that the order in which rules are processed does not matter, as we consider both forms for complementary nodes (cf. line 3).

[6] A self-contradictory sequence is any sequence of rules $r_1, ..., r_n$, where $head(r_1)$ activates $r_2, ..., r_n$ and there exists a literal $l \in body(r_1)$ s.t. $l \cup head(r_n) \models \perp$ (or $l \cup head(r_1) \models \perp$).

2. The returned set does not contain all minimal issues of the input rule base $\mathcal{B}$.

Regarding case 1, observe that in line 3, the algorithm iterates over all nodes n s.t. there exists a node $\overline{n}$ in the REG. These are all pairs of conclusions for which an inconsistency can arise, i.e., a conclusion $n, \neg n$ can be entailed, c.f. also the above discussion. Then, for all these pairs of complementary conclusions $n, \overline{n}$, all self avoiding paths from any node in the REG to $\overline{n}$ are computed. Then, for all these paths, all self avoiding paths from the source of that path to n are computed (resp. all subsets of the source if the source is composed of multiple literals). In result, for any found combination of paths from any S to $\overline{n}$, and S' to n, we have that $S' \subseteq S$. Also, considering that a path on the REG pertains to a sequence of rules, we have that S' minimally activates the rule with conclusion n, resp. S minimally activates the rule with conclusion n'. Also, observe that we do not have any self-contradictions per assumption, thus, the respective chains of rules are S-consistent, resp. S'-consistent. Furthermore, as we considered pairs of contradictory conclusions, the activated rule sequences are S-inconsistent, as S activates two rules with contradictory conclusions (c.f. also line 10). This satisfies all conditions in Definition 12 and thus contradicts the above case 1. Case 2 is analogous, i.e., as all possible combinations of paths are considered (c.f. line 5 and 8), it is warranted that all possible minimal issues are found, which contradicts case 2. We reach a contradiction for both cases, thus, Algorithm 2 is sound.

□

Theorem 46. *Algorithm 2 is complete for the problem of computing all min. issues.*

Proof. First, observe that in case there are no minimal issues in a rule base $\mathcal{B}$, MinIssues($\mathcal{B}$) $= \emptyset$. Thus, there is no case we failure should be reported. An empty set for storing minimal issues (*minIssues*) is initialized in line 1. No alterations are made to this set from lines 2-10. In line 11, tuples are added to the set *minIssues*, c.f. the discussion on soundness (note that path finding is implemented via a breadth-first search which is complete [36]). Then, the set *minIssues* is returned in line 12 (which is either emtpy or contains minimal issues). Thus, Algorithm 2 is complete by invariance. □

Summary: Algorithmic Considerations

>The presented Algorithm 2 allows to compute minimal issues in design-time rule bases

3.5 Discussion: Handling Inconsistencies in Business Rule Bases

▷ *This section discusses the tools presented by the author and colleagues in [27, 30, 64].*

In this entire chapter, we have presented results for detecting (potential) inconsistencies in business rule bases, assessing these issues in general, pin-pointing the culprits of inconsistency, means for determining a resolution strategy, and an algorithm to (automatedly) resolve inconsistencies. We now conclude with a discussion of these results in the scope of BRM. To this aim, we propose a framework for handling inconsistencies in business rule bases, shown in Figure 3.7.

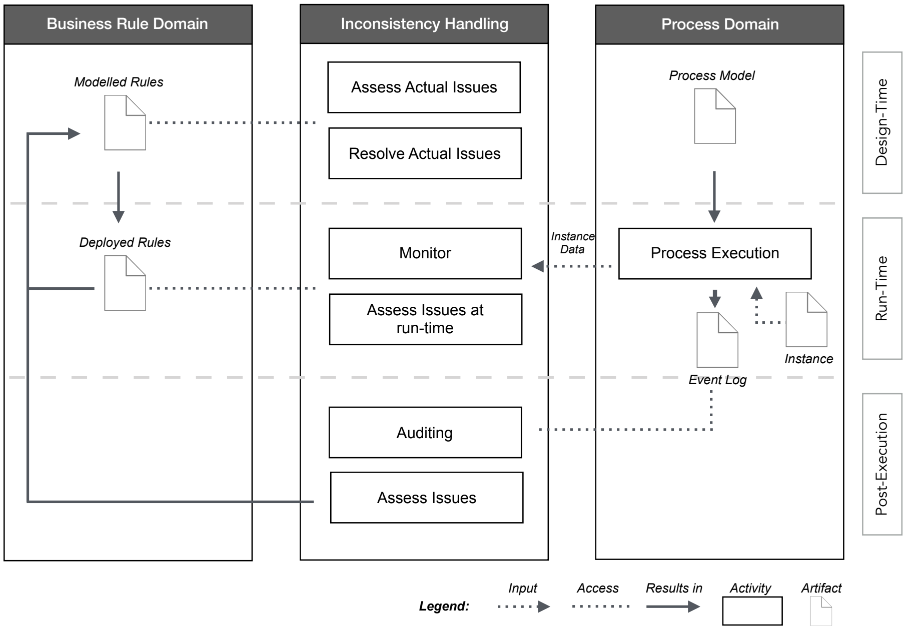


Figure 3.7: Overview of the proposed approach to handle inconsistencies in business rules.

The framework distinguishes between a business rule domain, the inconsistency handling domain and a process domain. It supports multiple compliance management phases and is designed to have a natural cycle throughout these phases. During design-time, it is ensured that the absolute minimum of potential inconsistencies, namely minimal issues, are removed. Then, during run-time, potential issues are monitored, and process execution can for example be stopped in case of violations. Allthough it is not within the scope of this thesis, it is worth noting that means for post-execution compliance could be integrated into the framework to inspect potential inconsistencies in the context of all process executions, which could be used for further insights for re-modelling. Also, the business rule domain and the process domain are connected by means of model query (not part of this thesis) [96]. We continue to discuss the individual steps of our framework in more detail.

At first, during design-time, the modelled rules can be analyzed for actual issues, i.e., the rule base can be verified for quasi-inconsistency. Here, quasi-inconsistency can be assessed by the proposed measures, and business rules can be prioritized by their degree of culpability. This provides valuable insights as a driver for inconsistency resolution, e.g., by means of a recommendation via a culpability ranking or automatically via the proposed resolution algorithm. It is important that all actual issues are removed, as they represent clear logical flaws in the business logic and impede using business rules correctly, e.g., to verify the process model for compliance at design-time.

After the rule base is no-longer quasi-inconsistent, the rule set can be deployed. This deployed rule set can thus be used to govern process execution. During run-time, instance data and its interrelation to the set of business rules should be monitored, due to the potential issues that may still be present in the deployed rule set. For example, it can be the case that unforseen fact combinations can lead to classic-logical inconsistency during run-time. Here, monitoring consistency of decision-making during run-time is important to mitigate potential compliance breaches. In case inconsistencies are detected during run-time, modelers can assess these problems. Then, experts could handle the case manually, or directly resolve the inconsistencies using the proposed means for analyzing inconsistencies at run-time.

After process execution, the event logs tracked by the system could be further used to gain comprehensive insights into the occurrences of potential inconsistencies across all executed processes, i.e., how often did certain potential inconsistencies actually become a problem w.r.t. to the observed case-input. Using results from drift detection, experts could gain further insights into the severity of potential inconsistencies. For example, if a potential issue actually became a problem only once in many cases, this might be negligible. On the contrary, if there is a recent rise of inconsistency occurrences, this potential issue can be seen as more problematic w.r.t. the business rule base. Inconsistency analysis from the auditing perspective is intended to provide further insights as a basis for refining the company rule base and to provide a retrospective assessment of the changes potentially made during run-time inconsistency handling, and could therefore be interesting for future work.

The proposed framework allows to iteratively develop business rules, which is in line with the iterative cycle of the general business rule management lifecycle shown in Figure 1.6. An advantage of the framework is that it supports all compliance management phases and is designed to have a natural, holistic cycle throughout these phases. During design-time, it is ensured that the absolute minimum of potential inconsistencies, namely minimal issues, are removed. Then, during run-time, potential issues are monitored. Last, post-execution analysis could also be used as a driver for re-modelling.

We conclude with an example that clarifies how inconsistency can be handled during the different compliance phases as proposed by our framework.

Example 47. Consider the exemplary rule base $\mathcal{B}_{16}$, with

$$\begin{aligned}\mathcal{B}_{16} = \{&a \to n; a \to \neg n\\ &b \to d; c \to \neg d\\ &x \to z; y \to \neg z\}.\end{aligned}$$

We see that there are three potential inconsistencies: the first line is an actual issue, and the two other lines are potential issues. At design-time, all actual (minimal) issues are to be resolved. To this aim, the analysis techniques or algorithm proposed can be applied. Assume the modeller has decided to eliminate the rule $a \to n$, leading to the deployable rule base

$$\begin{aligned}\mathcal{B}'_{16} = \{&a \to \neg n,\\ &b \to d; c \to \neg d\\ &x \to z; y \to \neg z\}.\end{aligned}$$

During run-time, consistent decision-making is monitored. For example, in any process instance where the facts b and c occur together, inconsistency can be detected. Assume we have the following three process instances, with different case-dependent facts:

b	c	b, c
$b \to d$	$b \to d$	$b \to d$
$c \to \neg d$	$c \to \neg d$	$c \to \neg d$
(a) Exemplary Instance A	(b) Exemplary Instance B	(c) Inconsistent Instance C

The instances (a) and (b) are unproblematic w.r.t. the business rules. In instance (c), process execution can be stopped, such that experts can inspect the case[7]. The means for run-time inconsistency measurement can also be applied to identify rules which should be re-modelled under the assumption of a given fact input.

Last, during post-execution auditing, event logs can be analyzed to inspect the actually observed behavior and fact occurrences, in order to further assess the severity of potential issues and re-model the rule base. For example, assume we have the following event log:

Case 1: axcacaxz
Case 2: zdzdbx
Case 3: axca**cbd**
Case 4: axz**yxzyxz**
Case 5: a**xcyz**
Case 6: zxza**xcyzyxz**

---

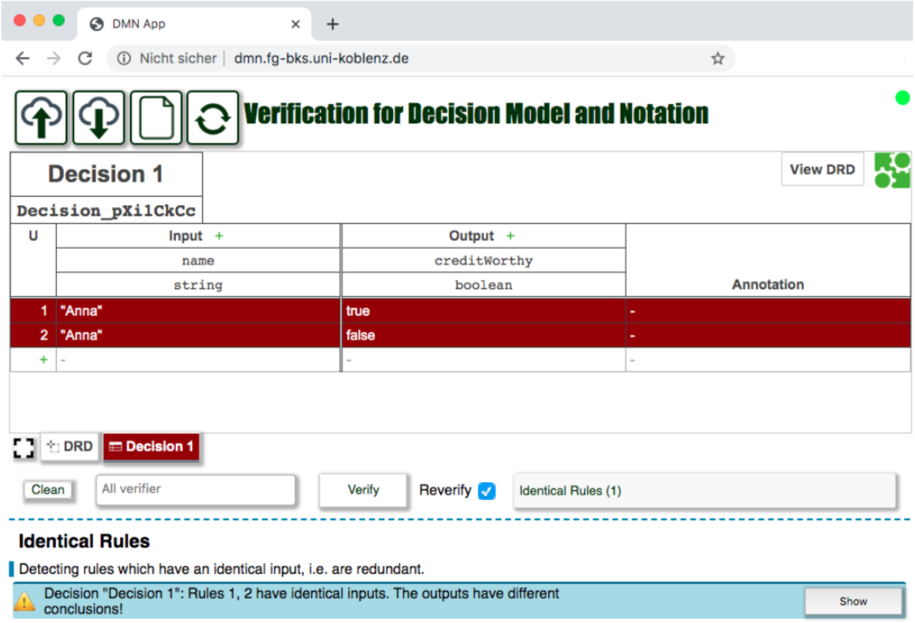
[7] We refer the reader to [27] (by the author) for further details on how the temporal order of fact occurences is also considered by means of a so-called decision execution history in that work.

During auditing, experts could analyze this log relative to the business rules in $\mathcal{B}'_{16}$. For the sake of the example, we assume a business rule $b \to d$ can be understood s.t. an activity b should be eventually followed by activity d (or "not eventually followed" in case of negation), which would be equivalent to the RESPONSE template in Declare. As can be seen, an inconsistency involving the rules $b \to d$ and $c \to \neg d$ only actually arises in one of the traces (highlighted in case 3). The inconsistency including the rules $x \to z$ and $y \to \neg z$ arises in multiple traces (highlighted in cases 4-6), and more often in those traces. Such insights can be used as a basis for re-modelling. For instance, the expert might use the provided means to determine that the inconsistency observed in trace 3 is very seldom and the corresponding business rules can be kept as-is, while the rule $x \to \neg z$ should be deleted.

The example showed how inconsistency can be handled during the different compliance phases in the proposed framework. As business rules can be added or changed, the proposed approach to handling inconsistency is an iterative process embedded within BRM, which is also shown in Figure 3.7.

As a result of this thesis, the components needed to implement the proposed framework were also developed. A general library was implemented to detect (quasi-)inconsistencies at design time, for the standards FCL, DMN and Declare (cf. Appendix A). Also, several graphical user interfaces were presented in [27, 30, 64].

Figure 3.9 shows a screenshot from a graphical user interface for design-time analysis of DMN rule bases presented in [27, 64].



Decision_pXi1CkCc

U	Input +	Output +	
	name	creditWorthy	
	string	boolean	Annotation
1	"Anna"	true	-
2	"Anna"	false	-
+	-	-	-

Identical Rules

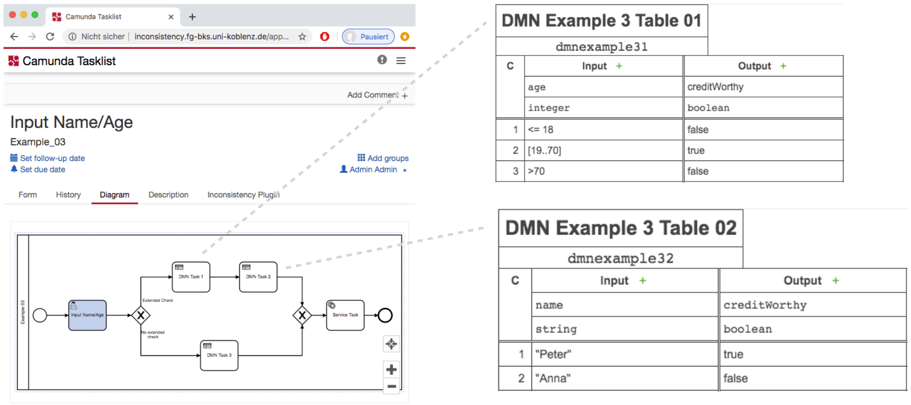
Detecting rules which have an identical input, i.e. are redundant.

⚠ Decision "Decision 1": Rules 1, 2 have identical inputs. The outputs have different conclusions!

Figure 3.9: Showcase of the design-time verification tool for DMN presented in [64].

The tool extends the open-source library bpmn.io[8]. The tool can be used to upload and verify DMN models (i.e. rule bases) directly in the browser. Based on the error detection, users can then re-model the DMN tables directly in the browser, or re-deploy them to a workflow management system.

Figure 3.10 shows an exemplary process to determine the creditworthyness of a customer, that has been integrated into the workflow management system Camunda[9]. In the example, the process instance is currently awaiting a user to manually input a name and age, e.g., of a customer application. Figure 3.10 also shows the decision tables associated with the business rule tasks 1 and 2. As can be seen, the creditworthiness of the customer is determined at multiple points in the process. This can however lead to inconsistent decision-making during run-time. An example of this is shown in Figure 3.11.



DMN Example 3 Table 01

dmnexample31

C	Input +	Output +
	age	creditWorthy
	integer	boolean
1	<= 18	false
2	[19..70]	true
3	>70	false

DMN Example 3 Table 02

dmnexample32

C	Input +	Output +
	name	creditWorthy
	string	boolean
1	"Peter"	true
2	"Anna"	false

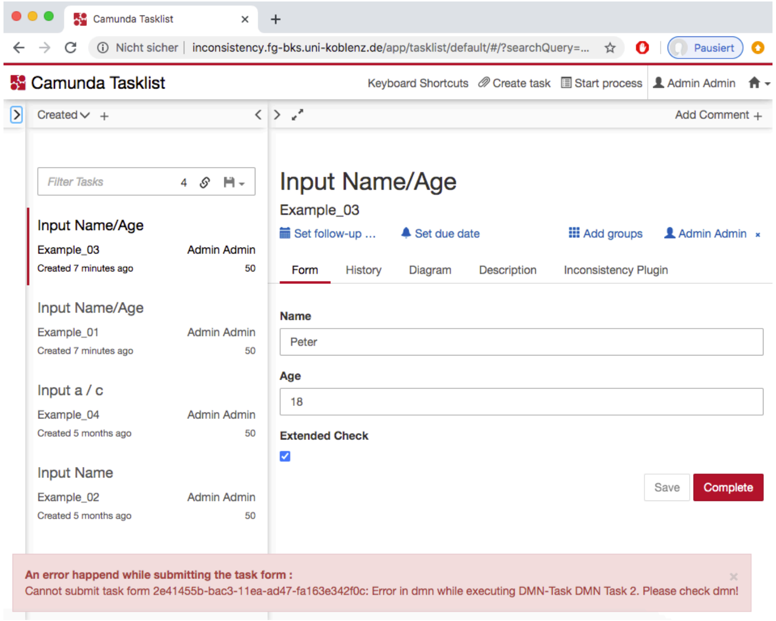
Figure 3.10: Exemplary process in Camunda.

Assume a modeller enters the name "Peter" and age "18" as depicted. Due to the contradictory decision logic, which could have for example been created by different modelers, an inconsistency in the automated decision-making arises during run-time. Here, the monitoring tool presented in [30] can detect such run-time inconsistencies. The tool displays an error message as shown, stops process execution such that no further compliance breaches are committed, and reverts the process instance (i.e., token) back to the last manual user task. A comprehensive assessment of rules responsible for the inconsistency is then presented to the user, exploiting the results for run-time inconsistency analysis presented in this thesis. In the exemplary tool, the user can view the inconsistency analysis and re-model (or re-deploy) the business rules directly in the browser. As depicted in Figure 3.12, the inconsistency analysis is also seamlessly integrated into the error reporting cockpit in Camunda, s.t. insights are made available to the company in the scope of reporting. From here, experts can directly examine so-called Camunda incidents and are presented with comprehensive insights.

---

[8] `http://bpmn.io`

[9] `https://camunda.com/`



Input Name/Age
Example_03 Admin Admin
Created 7 minutes ago 50

Input Name/Age
Example_01 Admin Admin
Created 7 minutes ago 50

Input a / c
Example_04 Admin Admin
Created 5 months ago 50

Input Name
Example_02 Admin Admin
Created 5 months ago 50

Input Name/Age
Example_03

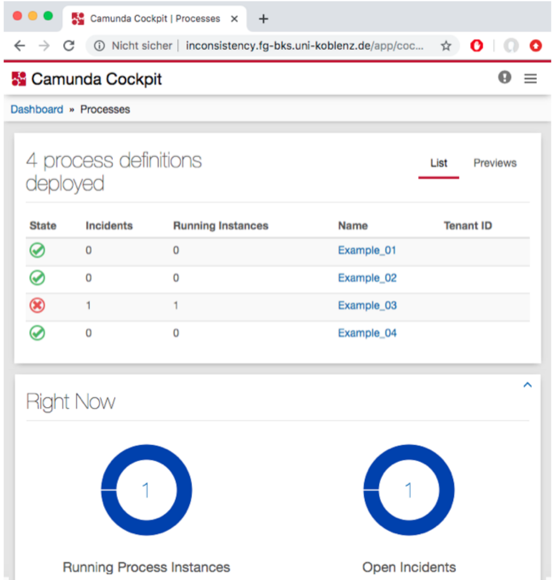
Name: Peter

Age: 18

Extended Check: ☑

An error happend while submitting the task form :
Cannot submit task form 2e41455b-bac3-11ea-ad47-fa163e342f0c: Error in dmn while executing DMN-Task DMN Task 2. Please check dmn!

Figure 3.11: Showcase of the run-time monitoring functionality presented in [30].



4 process definitions deployed

State	Incidents	Running Instances	Name	Tenant ID
✓	0	0	Example_01	
✓	0	0	Example_02	
✗	1	1	Example_03	
✓	0	0	Example_04	

Right Now

1 Running Process Instances

1 Open Incidents

Figure 3.12: Showcase of run-time error reporting feature in the monitoring tool presented in [30].

Summary: Handling Inconsistencies in Business Rule Bases

>Different compliance management phases require different means to detect and assess inconsistency, which have been proposed in this work

>We propose to utilize culpability measurement to pin-point and prioritize rules that should be re-modelled

>The proposed framework combining the presented results supports inconsistency handling in the scope of BRM

>The methods and techniques presented in this chapter have been implemented and can be used out-of-the-box for the rule standards of FCL, DMN and Declare, cf. Appendix A

4

Evaluation

In this chapter, we evaluate our proposed results, by means of analytical evaluation, run-time experiments, experiments with human participants and classification in the scope of existing evaluation frameworks.

4.1 Evaluation of Proposed Measures

The evaluation of inconsistency measures is predominantly driven by rationality postulates, i.e., properties that should be satisfied for specific measures [129]. In this work, we also follow this approach and evaluate the proposed (design-time/run-time) inconsistency measures, as well as the proposed adjusted Shapley inconsistency value in regard to their compliance with the identified postulates.

4.1.1 Inconsistency Measures

▷ *This section corresponds to the works [34, 35] by the author.*
▷ *! The Proofs for Propositions 52 and 53 were authored by Matthias Thimm for the above publication [35].*

In Sections 3.2.1 and 3.2.2, we motivated the use-cases of design-time rule management and run-time rule management, and proposed the postulates of QI, resp. RC, FE, RE, for these use-cases. In the following, we show the compliance of the proposed inconsistency measures with the respective postulates. We begin with an analysis of quasi-inconsistency measures $\mathcal{I}^Q$.

Example 48. We recall the rule bases $\mathcal{B}_1 = \{newCustomer \rightarrow creditWorthy; newCustomer \rightarrow \neg creditWorthy\}$ and $\mathcal{B}_5 = \{a \rightarrow b; b \rightarrow c; a \rightarrow \neg c\}$. Then, we recall the introduced measures, and that for the above $\mathcal{B}_1$ and $\mathcal{B}_5$ we expect a quasi-inconsistency assessment to yield a value > 0. Also, recall the rule base

$\mathcal{B}_7 = \{a \to b; b \to c; c \to d\}$. As $\mathcal{B}_7$ is not quasi-inconsistent, we expect a quasi-inconsistency assessment to yield a value of 0. Then we have

$$\begin{array}{ccc}
\mathcal{I}^Q_d(\mathcal{B}_1) = 1 & \mathcal{I}^Q_d(\mathcal{B}_5) = 1 & \mathcal{I}^Q_d(\mathcal{B}_7) = 0 \\
\mathcal{I}^Q_{\mathsf{MI}}(\mathcal{B}_1) = 1 & \mathcal{I}^Q_{\mathsf{MI}}(\mathcal{B}_5) = 1 & \mathcal{I}^Q_{\mathsf{MI}}(\mathcal{B}_7) = 0 \\
\mathcal{I}^Q_p(\mathcal{B}_1) = 2 & \mathcal{I}^Q_p(\mathcal{B}_5) = 3 & \mathcal{I}^Q_p(\mathcal{B}_7) = 0 \\
\mathcal{I}^Q_c(\mathcal{B}_1) = 1 & \mathcal{I}^Q_c(\mathcal{B}_5) = 1 & \mathcal{I}^Q_c(\mathcal{B}_7) = 0 \\
\mathcal{I}^Q_{mv}(\mathcal{B}_1) = 1 & \mathcal{I}^Q_{mv}(\mathcal{B}_5) = 1 & \mathcal{I}^Q_{mv}(\mathcal{B}_7) = 0
\end{array}$$

Thus, all approaches considered in this paper capture the desired behavior of QI.

Proposition 49 ([34]). *$\mathcal{I}^Q_d$, $\mathcal{I}^Q_{\mathsf{MI}}$ and $\mathcal{I}^Q_p$ satisfy QI, MO, and IN.*

Proof. We recall Definition 12 stating that a rule base $\mathcal{B}$ is quasi-inconsistent iff there exists an minimal issue. It follows that MinIssues($\mathcal{B}$) $\neq \emptyset$ for a quasi-inconsistent rule-base.

We now consider each measure $\mathcal{I}^Q_d$,$\mathcal{I}^Q_{\mathsf{MI}}$ and $\mathcal{I}^Q_p$ in turn. For this, we consider a rule base $\mathcal{B}$ (resp. $\mathcal{B}'$). The proofs for monotony and independence follow the original proofs in [69].

- We start with the measure $\mathcal{I}^Q_d$. QI clearly follows by definition. To show MO, consider $\mathcal{B} \cup \mathcal{B}'$ and proceed by cases: If $\mathcal{B} \cup \mathcal{B}'$ is not quasi-consistent, then $\mathcal{B}$ is also not quasi-consistent [34]. If $\mathcal{B} \cup \mathcal{B}'$ is quasi-inconsistent, then $\mathcal{I}^Q_d(\mathcal{B} \cup \mathcal{B}') = 1$ by definition (and $\mathcal{I}^Q_d(\mathcal{B} \cup \mathcal{B}') \geq \mathcal{I}^Q_d(\mathcal{B})$). So we have $\mathcal{I}^Q_d(\mathcal{B} \cup \mathcal{B}') \geq \mathcal{I}^Q_d(\mathcal{B})$ for both cases. To show IN, observe that any free rule $\alpha \in$ Free($\mathcal{B}$) is by definition not part of any minimal issue of $\mathcal{B}$. Then proceed by cases: Consider a free rule $\alpha \in \mathcal{B}$. If $\mathcal{B}$ is not quasi-inconsistent, then $\mathcal{I}^Q_d(\mathcal{B}) = \mathcal{I}^Q_d(\mathcal{B} \setminus \alpha)$. If $\mathcal{B}$ is quasi-inconsistent, and α is a free rule as by hypothesis, then $\mathcal{B} \setminus \alpha$ is still quasi-inconsistent. Then we have $\mathcal{I}^Q_d(\mathcal{B}) = \mathcal{I}^Q_d(\mathcal{B} \setminus \alpha)$ in both cases.

- We now consider $\mathcal{I}^Q_{\mathsf{MI}}$. For QI, observe that MinIssues($\mathcal{B}$) $\neq \emptyset$ if $\mathcal{B}$ is quasi-inconsistent. It follows that $\mid$ MinIssues($\mathcal{B}$) $\mid > 0$ in this case. To show MO, consider $\mathcal{B} \cup \mathcal{B}'$ and recall that MinIssues($\mathcal{B}$) $\subseteq$ MinIssues($\mathcal{B} \cup \mathcal{B}'$). Then we have that $\mathcal{I}^Q_{\mathsf{MI}}(\mathcal{B}) \leq \mathcal{I}^Q_{\mathsf{MI}}(\mathcal{B} \cup \mathcal{B}')$. This is analogous for IN. Consider a free rule $\alpha \in$ Free($\mathcal{B}$). By Definition, $\mid$ MinIssues($\mathcal{B}$) $\mid = \mid$ MinIssues($\mathcal{B} \setminus \alpha$) $\mid$, thus, $\mathcal{I}^Q_{\mathsf{MI}}(\mathcal{B}) = \mathcal{I}^Q_{\mathsf{MI}}(\mathcal{B} \setminus \alpha)$.

- We now consider $\mathcal{I}^Q_p$. The proof for QI is analogous to $\mathcal{I}^Q_{\mathsf{MI}}$, i. e., we have that MinIssues($\mathcal{B}$) $\neq \emptyset$ if $\mathcal{B}$ is quasi-inconsistent. The proof for MO is analogous to $\mathcal{I}^Q_{\mathsf{MI}}$, i. e., considering MinIssues($\mathcal{B}$) $\subseteq$ MinIssues($\mathcal{B} \cup \mathcal{B}'$), it follows that $\mathcal{I}^Q_p(\mathcal{B})$ cannot be larger than $\mathcal{I}^Q_p(\mathcal{B} \cup \mathcal{B}')$. The proof for IN is also analogous to $\mathcal{I}^Q_{\mathsf{MI}}$, i. e., a free rule $\alpha \in$ Free($\mathcal{B}$) is not part of any minimal issue by Definition, thus $\mathcal{I}^Q_p(\mathcal{B}) = \mathcal{I}^Q_p(\mathcal{B} \setminus \alpha)$.

□

Proposition 50. *$\mathcal{I}_c^Q$ and $\mathcal{I}_{mv}^Q$satisfy QI.*

Proof. This proof is analogous to $\mathcal{I}_{\mathsf{MI}}^Q$, i.e., observe that $\mathsf{MinIssues}(\mathcal{B}) \neq \emptyset$ if $\mathcal{B}$ is quasi-inconsistent. It follows that $\mid \mathsf{MinIssues}(\mathcal{B}) \mid > 0$ in this case. ☐

We now continue with rule-based inconsistency measures $\mathcal{I}^{RB}$.

Example 51. We recall the rule bases $\mathcal{B}_8 = \{a, \neg a\}, \mathcal{B}_9 = \{a; a \to b; a \to \neg b\}, \mathcal{B}_{10} = \{a; a \to b; a \to \neg b; c; \neg c\}, \mathcal{B}_{11} = \{a; a \to b; \neg b\}$, and that we expect a rule-based inconsistency assessment $\mathcal{I}^{RB}$ satisfying the postulates RC, FE, and RE to give

$$0 = \mathcal{I}^{RB}(\mathcal{B}_8) < \mathcal{I}^{RB}(\mathcal{B}_9) = \mathcal{I}^{RB}(\mathcal{B}_{10}) \qquad \text{and}$$
$$\mathcal{I}^{RB}(\mathcal{B}_{11}) < \mathcal{I}^{RB}(\mathcal{B}_9)$$

For our adapted measures, we have that

$\mathcal{I}_d^{RB}(\mathcal{B}_8) = 0$	$\mathcal{I}_d^{RB}(\mathcal{B}_9) = 1$	$\mathcal{I}_d^{RB}(\mathcal{B}_{10}) = 1$	$\mathcal{I}_d^{RB}(\mathcal{B}_{11}) = 1$
$\mathcal{I}_{\mathsf{MI}}^{RB}(\mathcal{B}_8) = 0$	$\mathcal{I}_{\mathsf{MI}}^{RB}(\mathcal{B}_9) = 1$	$\mathcal{I}_{\mathsf{MI}}^{RB}(\mathcal{B}_{10}) = 1$	$\mathcal{I}_{\mathsf{MI}}^{RB}(\mathcal{B}_{11}) = 1$
$\mathcal{I}_p^{RB}(\mathcal{B}_8) = 0$	$\mathcal{I}_p^{RB}(\mathcal{B}_9) = 2$	$\mathcal{I}_p^{RB}(\mathcal{B}_{10}) = 2$	$\mathcal{I}_p^{RB}(\mathcal{B}_{11}) = 1$
$\mathcal{I}_c^{RB}(\mathcal{B}_8) = 0$	$\mathcal{I}_c^{RB}(\mathcal{B}_9) = 1$	$\mathcal{I}_c^{RB}(\mathcal{B}_{10}) = 1$	$\mathcal{I}_c^{RB}(\mathcal{B}_{11}) = 1$
$\mathcal{I}_{mv}^{RB}(\mathcal{B}_8) = 0$	$\mathcal{I}_{mv}^{RB}(\mathcal{B}_9) = 1$	$\mathcal{I}_{mv}^{RB}(\mathcal{B}_{10}) = 2/3$	$\mathcal{I}_{mv}^{RB}(\mathcal{B}_{11}) = 1$

In result, we see that our alterations have improved all measures, s.t. they satisfy RC.

Proposition 52 ([35]). *$\mathcal{I}_d^{RB}$,$\mathcal{I}_{\mathsf{MI}}^{RB}$,$\mathcal{I}_c^{RB}$ satisfy RC and FE, and do not satisfy RE.*

Proof. For showing FE the following general observation will be useful:

$$\forall r \in \mathcal{B} : head(r) \neq \alpha \Longrightarrow \mathsf{MI}^{\setminus \mathcal{F}}(\mathcal{B}) = \mathsf{MI}^{\setminus \mathcal{F}}(\mathcal{B} \cup \{\overline{\alpha}\}) \tag{4.1}$$

To see this, let $M \in \mathsf{MI}^{\setminus \mathcal{F}}(\mathcal{B})$. Then obviously $M \in \mathsf{MI}^{\setminus \mathcal{F}}(\mathcal{B} \cup \{\overline{\alpha}\})$ as well. Let $M \in \mathsf{MI}^{\setminus \mathcal{F}}(\mathcal{B} \cup \{\overline{\alpha}\})$. Then $\overline{\alpha} \notin M$ because 1.) it cannot be that $\alpha \in M$ as this would violate the definition of $\mathsf{MI}^{\setminus \mathcal{F}}$ and 2.) there is no rule concluding α by assumption. It follows that if $M \in \mathsf{MI}^{\setminus \mathcal{F}}(\mathcal{B} \cup \{\overline{\alpha}\})$, then also $M \in \mathsf{MI}^{\setminus \mathcal{F}}(\mathcal{B})$.

We now consider each measure $\mathcal{I}_d^{RB}$,$\mathcal{I}_{\mathsf{MI}}^{RB}$,$\mathcal{I}_c^{RB}$ in turn.

- We start with the measure $\mathcal{I}_d^{RB}$ and RC. Assume $\mathcal{I}_d^{RB}(\mathcal{B}) = 0$. Then $\mathsf{MI}^{\setminus \mathcal{F}}(\mathcal{B}) = \emptyset$ and therefore for all $M \in \mathsf{MI}(\mathcal{B})$ there is $a \in \mathcal{A}$ with $a, \neg a \in M$. So for all consistent subsets $F' \subseteq \mathcal{F}(\mathcal{B})$, $\mathcal{R}(\mathcal{B}) \cup F'$ is consistent. The other direction is analogous.

  FE follows directly by definition and Equation (4.1).

  For RE consider $\mathcal{B} = \{a; b; a \to \neg a\}$. Then $b \to \neg b \notin \mathcal{B}$, $b \to \neg b \notin \mathsf{Free}(\mathcal{B} \cup \{b \to \neg b\})$ but $\mathcal{I}_d^{RB}(\mathcal{B} \cup \{\neg b\}) = 1 = \mathcal{I}_d^{RB}(\mathcal{B} \cup \{b \to \neg b\})$.

- We now consider $\mathcal{I}^{RB}_{\mathsf{MI}}$. The proof of RC is analogous to $\mathcal{I}^{RB}_{d}$.

  FE follows directly by definition and Equation (4.1).

  For RE consider $\mathcal{B} = \{a; \neg c; b \rightarrow c\}$. Then $a \rightarrow b \notin \mathcal{B}$, $a \rightarrow b \notin \mathsf{Free}(\mathcal{B} \cup \{a \rightarrow b\})$ but $\mathcal{I}^{RB}_{\mathsf{MI}}(\mathcal{B} \cup \{b\}) = 1 = \mathcal{I}^{RB}_{\mathsf{MI}}(\mathcal{B} \cup \{a \rightarrow b\})$.

- We now consider $\mathcal{I}^{RB}_{c}$ and RC. Assume $\mathcal{I}^{RB}_{c}(\mathcal{B}) = 0$. Then there is a three-valued interpretation v with $v(a) = b$ for no $a \in \mathcal{A}$ and

  $$v \models^3 \bigcup_{M \in \mathsf{MI}^{\setminus \mathcal{F}}(\mathcal{B})} M$$

  Assume $\mathsf{MI}^{\setminus \mathcal{F}}(\mathcal{B}) \neq \emptyset$ and let $M \in \mathsf{MI}^{\setminus \mathcal{F}}(\mathcal{B})$. Then $v \models^3 \alpha$ for each $\alpha \in M$ and as $v^{-1}(b) = \emptyset$, $v(\alpha) = t$. This is a contradiction as M is inconsistent. It follows $\mathsf{MI}^{\setminus \mathcal{F}}(\mathcal{B}) = \emptyset$ and therefore for all consistent sets $F' \subseteq \mathcal{F}(\mathcal{B})$, $\mathcal{R}(\mathcal{B}) \cup F'$ is consistent. On the other hand, if for all consistent sets $F' \subseteq \mathcal{F}(\mathcal{B})$, $\mathcal{R}(\mathcal{B}) \cup F'$ is consistent then $\mathsf{MI}^{\setminus \mathcal{F}}(\mathcal{B}) = \emptyset$ and v can be defined via $v(a) = t$ for all $a \in \mathcal{A}$ showing $\mathcal{I}^{RB}_{c}(\mathcal{B}) = 0$.

  FE follows directly by definition and Equation (4.1).

  For RE consider $\mathcal{B} = \{a; \neg c; b \rightarrow c\}$. Then $a \rightarrow b \notin \mathcal{B}$, $a \rightarrow b \notin \mathsf{Free}(\mathcal{B} \cup \{a \rightarrow b\})$ but $\mathcal{I}^{RB}_{c}(\mathcal{B} \cup \{b\}) = 1 = \mathcal{I}^{RB}_{c}(\mathcal{B} \cup \{a \rightarrow b\})$.

□

Proposition 53 ([35]). *$\mathcal{I}^{RB}_{p}$ satisfies RC, FE and RE.*

Proof. The postulates RC and FE follow from a similar reasoning as for $\mathcal{I}^{RB}_{\mathsf{MI}}$ in the proof of Proposition 52. For RE, let $B \rightarrow H \notin \mathcal{B}$ and $B \rightarrow H \notin \mathsf{Free}(\mathcal{B} \cup \{B \rightarrow H\})$. First observe that $\mathsf{MI}^{\setminus \mathcal{F}}(\mathcal{B}) \subseteq \mathsf{MI}^{\setminus \mathcal{F}}(\mathcal{B} \cup \{H \rightarrow B\})$. Consider then some $M \in \mathsf{MI}^{\setminus \mathcal{F}}(\mathcal{B} \cup \{H\}) \setminus \mathsf{MI}^{\setminus \mathcal{F}}(\mathcal{B})$. It follows $H \in M$. As $B \rightarrow H \notin \mathsf{Free}(\mathcal{B} \cup \{B \rightarrow H\})$ it follows that there is $X \subseteq \mathcal{B}$ such that $H \in \min(X \cup \{B \rightarrow H\})$ and the set of facts appearing in X is consistent (otherwise $B \rightarrow H$ could not be part of any $N \in \mathsf{MI}^{\setminus \mathcal{F}}(\mathcal{B} \cup \{B \rightarrow H)\})$. Then $M' = M \setminus \{H\} \cup X \cup \{B \rightarrow H\}$ is inconsistent. Let $M'' \subseteq M'$ be minimal inconsistent. It follows $M'' \in \mathsf{MI}^{\setminus \mathcal{F}}(\mathcal{B} \cup \{B \rightarrow H)\}$. This means that every $M \in \mathsf{MI}^{\setminus \mathcal{F}}(\mathcal{B} \cup \{H\}) \setminus \mathsf{MI}^{\setminus \mathcal{F}}(\mathcal{B})$ can be transformed to an $M'' \in \mathsf{MI}^{\setminus \mathcal{F}}(\mathcal{B} \cup \{B \rightarrow H)\}$ by replacing H with the rule $B \rightarrow H$ and further rules and facts to derive B. As this replacement introduces (at least) the rule $B \rightarrow H$ we get $\mathcal{I}^{RB}_{p}(\mathcal{B} \cup \{B \rightarrow H\}) > \mathcal{I}^{RB}_{p}(\mathcal{B} \cup \{H\})$. □

Proposition 54. *$\mathcal{I}^{RB}_{mv}$ satisfies RC, and does not satisfy FE and RE.*

Proof. The proof for postulate RC follows from a similar reasoning as for $\mathcal{I}^{RB}_{\mathsf{MI}}$ in the proof of Proposition 52. For FE and RE proceed by counterexample (cf. Example 51), e.g., $\mathcal{I}^{RB}_{mv}(\mathcal{B}_{10}) \neq \mathcal{I}^{RB}_{mv}(\mathcal{B}_{9})$, and $\mathcal{I}^{RB}_{mv}(\mathcal{B}_{11}) \not< \mathcal{I}^{RB}_{mv}(\mathcal{B}_{9})$.

□

4.1.2 Adjusted Shapley Value

▷ *This section corresponds to the work [35] by the author.*
▷ *! The proof for Proposition 57 was authored by Matthias Thimm for the above publication.*

To consider requirements in run-time culpability measurement, the adjusted Shapley value was proposed in this work.

Proposition 55 ([35]). *The adjusted Shapley value satisfies Distribution, Minimality and Fact-Minimality.*

Proof. We adress the three parts individually.

- For showing Distribution, let us rewrite the Definition of the adjusted Shapley inconsistency value as

$$S *_{\alpha}^{\mathcal{I}^{RB}}(\mathcal{B}) = \begin{cases} 0 & \text{if } \alpha \in \mathcal{F}(\mathcal{B}) \\ \sum_{B \subseteq \mathcal{B}} CoalPayoff_{\alpha,\mathcal{B}}^{\mathcal{I}^{RB}}(B) + \sum_{B \subseteq \mathcal{B}} AddPayoff_{\alpha,\mathcal{B}}^{\mathcal{I}^{RB}}(B) & \text{otherwise} \end{cases}$$

  We are now interested in the sum of all adjusted Shapley values (for all elements of a rule base $\mathcal{B}$).

$$\sum_{\alpha \in \mathcal{B}} S *_{\alpha}^{\mathcal{I}^{RB}}(\mathcal{B})$$

$$= \sum_{\alpha \in \mathcal{B}} \begin{cases} 0 & \text{if } \alpha \in \mathcal{F}(\mathcal{B}) \\ \sum_{B \subseteq \mathcal{B}} CoalPayoff_{\alpha,\mathcal{B}}^{\mathcal{I}^{RB}}(B) + \sum_{B \subseteq \mathcal{B}} AddPayoff_{\alpha}^{\mathcal{I}^{RB}}(B) & \text{otherwise} \end{cases}$$

$$= \sum_{\alpha \in \mathcal{B}} \sum_{B \subseteq \mathcal{B}} CoalPayoff_{\alpha,\mathcal{B}}^{\mathcal{I}^{RB}}(B) - \sum_{f \in \mathcal{F}(\mathcal{B})} \sum_{B \subseteq \mathcal{B}} CoalPayoff_{f,\mathcal{B}}^{\mathcal{I}^{RB}}(B) + \sum_{\alpha \in \mathcal{R}(\mathcal{B})} \sum_{B \subseteq \mathcal{B}} AddPayoff_{\alpha,\mathcal{B}}^{\mathcal{I}^{RB}}(B)$$

  Following [69], the first summand can be rewritten.

$$= \mathcal{I}^{RB}(\mathcal{B}) - \sum_{f \in \mathcal{F}(\mathcal{B})} \sum_{B \subseteq \mathcal{B}} CoalPayoff_{f,\mathcal{B}}^{\mathcal{I}^{RB}}(B) + \sum_{\alpha \in \mathcal{R}(\mathcal{B})} \sum_{B \subseteq \mathcal{B}} AddPayoff_{\alpha,\mathcal{B}}^{\mathcal{I}^{RB}}(B)$$

  Then

$$= \mathcal{I}^{RB}(\mathcal{B}) - \sum_{f \in \mathcal{F}(\mathcal{B})} \sum_{B \subseteq \mathcal{B}} CoalPayoff_{f,\mathcal{B}}^{\mathcal{I}^{RB}}(B) + \sum_{\alpha \in \mathcal{R}(\mathcal{B})} \sum_{B \subseteq \mathcal{B}} \begin{cases} 0 & \text{if } r \in \text{Free}(B) \\ \frac{\sum_{f \in \mathcal{F}(B)} CoalPayoff_{f,\mathcal{B}}^{\mathcal{I}^{RB}}(B)}{|r \in \mathcal{R}(B) \text{ s.t. } r \notin \text{Free}(B)|} & \text{otherwise} \end{cases}$$

$$= \mathcal{I}^{RB}(\mathcal{B}) - \sum_{f \in \mathcal{F}(\mathcal{B})} \sum_{B \subseteq \mathcal{B}} CoalPayoff_{f,\mathcal{B}}^{\mathcal{I}^{RB}}(B) + \sum_{r \in \text{Free}(\mathcal{R}(\mathcal{B}))} \sum_{B \subseteq \mathcal{B}} 0 + \sum_{r \notin \text{Free}(\mathcal{R}(\mathcal{B}))} \sum_{B \subseteq \mathcal{B}} \frac{\sum_{f \in \mathcal{F}(B)} CoalPayoff_{f,\mathcal{B}}^{\mathcal{I}^{RB}}(B)}{|r \in \mathcal{R}(B) \text{ s.t. } r \notin \text{Free}(B)|}$$

$$= \mathcal{I}^{RB}(\mathcal{B}) - \sum_{f \in \mathcal{F}(\mathcal{B})} \sum_{B \subseteq \mathcal{B}} CoalPayoff_{f,\mathcal{B}}^{\mathcal{I}^{RB}}(B) + \sum_{r \notin \text{Free}(\mathcal{R}(\mathcal{B}))} \sum_{B \subseteq \mathcal{B}} \frac{\sum_{f \in \mathcal{F}(B)} CoalPayoff_{f,\mathcal{B}}^{\mathcal{I}^{RB}}(B)}{|r \in \mathcal{R}(B) \text{ s.t. } r \notin \text{Free}(B)|}$$

$$= \mathcal{I}^{RB}(\mathcal{B}) - \sum_{f \in \mathcal{F}(\mathcal{B})} \sum_{B \subseteq \mathcal{B}} CoalPayoff_{f,\mathcal{B}}^{\mathcal{I}^{RB}}(B) + \sum_{B \subseteq \mathcal{B}} \sum_{f \in \mathcal{F}(B)} CoalPayoff_{f,\mathcal{B}}^{\mathcal{I}^{RB}}(B)$$

$$= \mathcal{I}^{RB}(\mathcal{B}) - \sum_{f \in \mathcal{F}(\mathcal{B})} \sum_{B \subseteq \mathcal{B}} CoalPayoff_{f,\mathcal{B}}^{\mathcal{I}^{RB}}(B) + \sum_{f \in \mathcal{F}(B)} \sum_{B \subseteq \mathcal{B}} CoalPayoff_{f,\mathcal{B}}^{\mathcal{I}^{RB}}(B)$$

$$= \mathcal{I}^{RB}(\mathcal{B})$$

- Minimality follows from a combination of Definitions 32, 31 and 34. Observe that any element in a rule base $\mathcal{B}$ is either a fact or a rule, thus we discuss this individually. For any fact f, $S *_{f}^{\mathcal{I}^{RB}} (\mathcal{B}) = 0$ via Definition 34. Thus we have that if a free formula α is a fact, $S *_{\alpha}^{\mathcal{I}^{RB}} (\mathcal{B}) = 0$. Continuing with rules, following the free-formula independence requirement, we have that if a rule r is a free-formula, then $\mathcal{I}^{RB}(\mathcal{B}) = \mathcal{I}^{RB}(\mathcal{B} \setminus r)$. Then we see that the last part $(\mathcal{I}^{RB}(\mathcal{B}) - \mathcal{I}^{RB}(\mathcal{B} \setminus r))$ in Definition 31 always equals out to zero for free formulas (This follows the original proof in [69]). Last, via Definition 32, we see that any rule which is a free formula automatically is assigned an additional payoff of zero. Concluding, the adjusted shapley value via Defintion 34 is zero for rules which are free formulas.
- Fact-minimality directly follows from Definition 34, i.e., facts are always assigned a value of zero.

☐

Proposition 56 ([35]). *Let $\mathcal{I}^{RB}$ be a rule-based inconsistency measure, $\mathcal{B}$ be a rule base and $r \in \mathcal{R}(\mathcal{B})$.*

- **Rule-Involvement.** *If $\mathcal{I}^{RB}$ satisfies RC and $r \in \mathcal{B}$ is a non-free rule, then $S *_{r}^{\mathcal{I}^{RB}} (\mathcal{B}) > 0$.*

Proof. A useful observation is that for a non-free formula α, all minimal inconsistent subsets that α is contained relate to coalitions, where removing α from that coalition resolves the inconsistency. Then we see that the last part $(\mathcal{I}^{RB}(\mathcal{B}) - \mathcal{I}^{RB}(\mathcal{B} \setminus r))$ in Definition 31 will be $\neq 0$ for those coalitions. Also, any non-free rule is at least part of one minimal inconsistent subset M_1. Thus, to show rule-involvement, we see that the smallest adjusted Shapley value a non-free rule can receive is $((m-1)!(n-m)!)/n!) + \sum_{f \in \mathcal{F}(M_1)}(((m-1)!(n-m)!)/n!)/(|r \in \mathcal{R}(M_1)$ s.t. $r \notin \text{Free}(M_1)|))$, where m is the cardinality of M_1, and n is the cardinality of $\mathcal{B}$. ☐

Proposition 57 ([35]). *Let $\mathcal{I}^{RB}$ be a rule-based inconsistency measure, and $\mathcal{B}$ be a rule base. Then, following [69], we have:*

- **Rule Consistency'.** *$\hat{S} *^{\mathcal{I}^{RB}} (\mathcal{B}) = 0$ iff $\mathcal{B}$ is rule consistent.*
- **Free formula independence'.** *If $\mathcal{I}^{RB}$ satisfies IN and α is a free formula of $\mathcal{B} \cup \{\alpha\}$, then $\hat{S} *^{\mathcal{I}^{RB}} (\mathcal{B} \cup \{\alpha\}) = \hat{S} *^{\mathcal{I}^{RB}} (\mathcal{B})$.*
- **Upper Bound.** *$\hat{S} *^{\mathcal{I}^{RB}} (\mathcal{B}) \leq \mathcal{I}^{RB}(\mathcal{B})$.*

Proof. To show Rule Consistency', observe that as dictated by Rule-Involvement, we have that $S *_{r}^{\mathcal{I}^{RB}} (\mathcal{B}) > 0$ for any non-free rule $r \in \mathcal{B}$. So, for an inconsistent rule base $\mathcal{B}'$, we have that $\hat{S} *^{\mathcal{I}^{RB}} (\mathcal{B}') > 0$. From the other side, a consistent rule base $\mathcal{B}$ only contains free formulas. Here, following minimality, we see that indeed $\hat{S} *^{\mathcal{I}^{RB}} (\mathcal{B}')$ would be 0. To show Free formula independence' observe that for free

α and $\mathcal{I}^{RB}$ satisfying IN we have $\mathcal{I}^{RB}(B) = \mathcal{I}^{RB}(B \cup \{\alpha\})$. Then for arbitrary $\gamma \in \mathcal{B} \cup \{\alpha\}$, $B \subseteq \mathcal{B} \cup \{\alpha\}$, $b = |B|$, and $n = |\mathcal{B}|$ we have

$$\mathcal{I}^{RB}(B) - \mathcal{I}^{RB}(B \setminus \{\gamma\}) = \mathcal{I}^{RB}(B \cup \{\alpha\}) - \mathcal{I}^{RB}((B \cup \{\alpha\}) \setminus \{\gamma\})$$

and therefore

$$\begin{aligned}
&CoalPayoff^{\mathcal{I}^{RB}}_{\gamma,\mathcal{B}\cup\{\alpha\}}(B) + CoalPayoff^{\mathcal{I}^{RB}}_{\gamma,\mathcal{B}\cup\{\alpha\}}(B \cup \{\alpha\}) \\
=&\frac{(b-1)!(n+1-b)!}{(n+1)!}(\mathcal{I}^{RB}(B) - \mathcal{I}^{RB}(B \setminus \{\gamma\})) + \frac{b!(n-b)!}{(n+1)!}(\mathcal{I}^{RB}(B \cup \{\alpha\}) - \mathcal{I}^{RB}((B \cup \{\alpha\}) \setminus \{\gamma\})) \\
=&(\frac{(b-1)!(n+1-b)!}{(n+1)!} + \frac{b!(n-b)!}{(n+1)!})(\mathcal{I}^{RB}(B) - \mathcal{I}^{RB}(B \setminus \{\gamma\})) \\
=&\frac{(b-1)!(n+1-b)! + b!(n-b)!}{(n+1)!}(\mathcal{I}^{RB}(B) - \mathcal{I}^{RB}(B \setminus \{\gamma\})) \\
=&\frac{(b-1)!(n-b)!(n+1-b+b)}{n!(n+1)}(\mathcal{I}^{RB}(B) - \mathcal{I}^{RB}(B \setminus \{\gamma\})) \\
=&\frac{(b-1)!(n-b)!}{n!}(\mathcal{I}^{RB}(B) - \mathcal{I}^{RB}(B \setminus \{\gamma\})) \\
=&CoalPayoff^{\mathcal{I}^{RB}}_{\gamma,\mathcal{B}}(B)
\end{aligned}$$

yielding $S *^{\mathcal{I}^{RB}}_{\gamma} (\mathcal{B}) = S *^{\mathcal{I}^{RB}}_{\gamma} (\mathcal{B} \cup \{\alpha\})$ for all γ. Also $S *^{\mathcal{I}^{RB}}_{\alpha} (\mathcal{B} \cup \{\alpha\}) = 0$ and therefore

$$\begin{aligned}
\hat{S} *^{\mathcal{I}^{RB}} (\mathcal{B} \cup \{\alpha\}) &= \max_{\gamma \in \mathcal{B} \cup \{\alpha\}} S *^{\mathcal{I}^{RB}}_{\gamma} (\mathcal{B} \cup \{\alpha\}) \\
&= \max_{\gamma \in \mathcal{B}} S *^{\mathcal{I}^{RB}}_{\gamma} (\mathcal{B}) \\
&= \hat{S} *^{\mathcal{I}^{RB}} (\mathcal{B})
\end{aligned}$$

Last, Upper Bound follows directly from Distribution, i.e., considering a knowledge base $\mathcal{B}$ and a rule-based inconsistency measure $\mathcal{I}^{RB}$, we see that the sum of $S *^{\mathcal{I}^{RB}}_{\alpha} (\mathcal{B})$ over all $\alpha \in \mathcal{B}$ equals $\mathcal{I}^{RB}(\mathcal{B})$. In turn, for any $\alpha_i \in \mathcal{B}$, $S *^{\mathcal{I}^{RB}}_{\alpha_i} (\mathcal{B})$ cannot be larger than $\mathcal{I}^{RB}(\mathcal{B})$. $\square$

Summary: Evaluation of Proposed Measures

$>$ All proposed quasi-inconsistency measures satisfy QI.

$>$ All proposed rule-based measures satisfy RC (see above for FE and RE).

$>$The adjusted Shapley value satisfies Distribution, Minimality and Fact-Minimality

---

4.2 Computational Complexity

▷ *This section corresponds to the work [34] by the author.*
▷ *! The technical results for Lemma/Proposition 58-62 in this section were authored by Matthias Thimm for the above publication. Due to excessive length,*

proofs are omitted - we refer the reader to [34].

Next to the compliance with rationality postulates, an evaluation technique for inconsistency measures is that of computational complexity [129]. For classical inconsistency measurement, existing results can be found in [131, 132]. As the setting of run-time inconsistency measurement is based on classic-logical inconsistency, these existing results are applicable here. On the contrary, the notion of quasi-inconsistency introduced in this thesis is a novel approach to the domain of inconsistency measurement. In the following, we therefore investigate various complexity aspects of assessing quasi-inconsistency in rule bases. Proofs can be found in [34]. We assume familiarity with basic concepts of computational complexity and basic complexity classes such as P and NP, see [93] for an introduction.

We start with analysing the complexity of verification tasks pertaining to issues.

Lemma 58 ([34]). *Let $\mathcal{B}$ be a rule base, $R_1, R_2 \subseteq \mathcal{B}$, and X_1, X_2 consistent sets of literals. Checking whether (R_1, X_1, R_2, X_2) is an issue can be done in polynomial time.*

Lemma 59 ([34]). *Let $\mathcal{B}$ be a rule base, $R_1, R_2 \subseteq \mathcal{B}$, and X_1, X_2 consistent sets of literals. Checking whether (R_1, X_1, R_2, X_2) is a minimal issue can be done in polynomial time.*

We now consider the central notion of quasi-inconsistency and the task to verify whether a rule base is quasi-inconsistent:

DEC-QI	**Input**:	rule base $\mathcal{B}$
	Output:	TRUE iff $\mathcal{B}$ is quasi-inconsistent

Proposition 60 ([34]). DEC-QI *is NP-complete. This remains true even for acyclic rule bases.*

For a rule $r \in \mathcal{B}$, we now consider checking whether r is contained in at least one minimal issue.

Lemma 61 ([34]). *Let $\mathcal{B}$ be a rule base and $r \in \mathcal{B}$. Checking whether there is a minimal issue (R_1, X_1, R_2, X_2) with $r \in R_1 \cup R_2$ is NP-complete.*

We now consider some counting problems related to issues:

#ISSUES	**Input**:	rule base $\mathcal{B}$
	Output:	$\lvert \mathsf{Issues}(\mathcal{B}) \rvert$
#MINISSUES	**Input**:	rule base $\mathcal{B}$
	Output:	$\lvert \mathsf{MinIssues}(\mathcal{B}) \rvert$

Proposition 62 ([34]). #ISSUES *and* #MINISSUES *are #P-complete.*[1]

---

[1] #P is the complexity class of counting problems where the problem of deciding whether a particular element has to be counted is in P.

Some other complexity results for individual inconsistency measures are provided in the extended version of [34] (c.f. Section 7).

We now turn to the computation complexity of the proposed algorithms. We start with Algorithm 2, which computes the minimal issues of a rule base. To compute the asymptotic costs, we consider a rule base $\mathcal{B}$ with a number of $|\mathcal{B}|$ formulas.

Note that Algorithm 2 takes as input the reactive entailment graph of $\mathcal{B}$. Observe that this graph can scale exponentially in size, c.f. the discussion in Section 3.4.1. Note also the discussion on the reduction of complexity while constructing an REG for Declare rule bases in Section 3.4.1.

Due to the general rule form in 3.1, computing the complementary constraints requires to scan all nodes in the REG that represent only one literal, and then looking up (constant) if there is a node that represents the complement to this literal. In result, finding the complementary literals requires at most $\mid \mathcal{L}(\mathcal{B}) \mid$ look ups. In line 3, a loop is entered which iterates over all complementary constraints, i.e., $\mid \mathcal{L}(\mathcal{B}) \mid$ iterations. In line 5, paths are computed (via a breadth first search). In the REG there can be at most $2*|\mathcal{B}|$ nodes, which would be the extreme case where no rule bodies and no rule heads are equal (then every rule would be transformed into two nodes). Here, the breadth first search has a maximum cost of $\mathcal{O}(\mid \text{nodes in } \mathsf{REG} \mid^2)$ if using an adjacency matrix, i.e., $\mathcal{O}(2*|\mathcal{B}|^2)$ [36]. The number of possible paths rises factorial. Thus, the loop in line 6 is perfomed at most $|\mathcal{B}|!$ times. The path search here is analogous to line 5. Then, in line 9, a factorial number of paths is again checked.

Given a rule base $\mathcal{B}$ with a number of $n =\mid \mathcal{B} \mid$ nodes and a number of $l =\mid \mathcal{L}(\mathcal{B}) \mid$ literals, the asypmtotic costs of Algorithm 2 are consequently

$$n + l * (n^2 + n! * (n^2 + n!)) = \mathcal{O}(n!^2)$$

We continue with Algorithm 1, which is an approximation algorithm for resolving quasi-inconsistency while deleting a minimal number of rules for a rule base $\mathcal{B}$. In line 1, the minimal issues of $\mathcal{B}$ are computed (c.f. the above discussion). In line 4, a while loop is entered, which loops until there are $<= 0$ minimal issues. Note that for every iteration, one element is deleted from the rule base in line 10. Thus, the loop in line 4 will be performed a maximum of n-1 times[2]. The body of the while loop contains a foreach loop, which is performed n times. Regarding the computation of the $\mathcal{C}_{\#}$ value in line 6, note that in the worst case all possible combinations of rules can be part of minimal issues, i.e., it could be necessary to look up an exponential number of rule sets [124]. Observe however that the specific culpability measure used can be changed, and this is not an inherent part of the run-time complexity of this algorithm. There exists other culpability measures with other run-times, which could be interchanged, c.f. e.g. [87] for a further discussion. In the while loop, observe that the set of minimal issues is again computed before starting the next iteration.

---

[2] Consider the worst case scenario for an arbitrary culpability measure, where all but one formula would be deleted.

Given a rule base $\mathcal{B}$ with a number of $n = |\mathcal{B}|$ nodes and a culpability measure with a run-time complexity of c, the asypmtotic costs of Algorithm 1 are consequently

$$n!^2 + (n-1) * (n * c + n!^2) = \mathcal{O}(n * n!^2 + n * c)$$

Summary: Computational Complexity

> Dec-QI is NP-complete.

> #MinIssues is #P-complete

4.3 Feasibility

To investigate the applicability of the proposed results in BRM, run-time experiments were conducted. This section presents results for measuring inconsistency in rule bases, and resolving inconsistencies based on culpability measurement.

4.3.1 Inconsistency Measurement in Business Rule Bases

▷ *This section corresponds to the works [27, 31] by the author.*

The proposed means to detect and analyze inconsistency in rule bases were originally implemented in the Master thesis by Matthias Deisen [40], and further developed in the works [27, 28, 30, 31, 64] by the author and colleagues. In the following, we present the results for solving quasi-inconsistency in design-time rule bases, as this is a novel concept. Note that run-time feasibility has already been shown in works such as [128], and is also discussed in [40].

The experiments were run on a machine with 3 GHz Intel Core i7 processor, 16 GB RAM (DDR3) under macOS High Sierra Version 10.13.6.

Evaluation with real-life data sets

To the best of our knowledge, no suitable real-life data sets for FCL and DMN were attainable (see below for synthetic data sets). For Declare, the following available data sets were used.

- BPI challenge 2017[3]. This data set contains an event log of a loan application process of a Dutch financial institute. The log contains 1,202,267 events corresponding to 31,509 loan application cases.
- BPI challenge 2018[4]. This data set contains an event log of a process at German federal ministries of agriculture and local departments. The log contains 2,514,266 events corresponding to 43,809 application cases.

---

[3] `https://www.win.tue.nl/bpi/doku.php?id=2017:challenge`

[4] `https://www.win.tue.nl/bpi/doku.php?id=2018:challenge`

- Sepsis 2016[5]. This data set contains an event log of a hospital process concerning the treatment of sepsis. The log contains around 1000 cases with 15,000 events.

We selected these data-sets because they provide recent data from real-life processes. Also, we selected these data-sets to analyze data of domains which are subject to a high degree of regulatory control and sensible to compliant process execution (e.g., financial-, government- and medical sector).

Declarative process models were mined from these logs using the Minerful tool [42]. As configuration for mining, the three parameters of *support*, *confidence* and *interest* can be considered by this tool. The support threshold indicates the minimum number of traces a constraint has to be fulfilled in for it to be included in the discovered model. Confidence scales the support by the ratio of traces in which the activation occurs; interest scales by the ratio of traces both the constrained tasks occur in. For the experiments in [31], we ran Minerful with a support of 75%, confidence of 12.5% and interest factor of 12.5%, as proposed in the experiment design by [42]. We then applied our implementation to a) compute all minimal issues, and b) compute the $\mathcal{C}_{\#}$ culpability measures for all constraints.

Log	**BPI Challenge '17**	**BPI Challenge '18**	**Sepsis '16**
Constraints	305	70	207
$\mathcal{I}^{Q}_{\mathsf{MI}}$ (or # of *MinIssues*)	28954	25303	7736
Runtime	27074ms	10930ms	4379ms

Table 4.1: Results of run-time experiments for the analyzed data-sets [31].

Table 4.1 shows an overview of the mined declarative process models and the corresponding number of detected minimal issues. For the largest model (BPI'17), roughly 29.000 minimal issues could be detected in around 27s.

Evaluation with synthetic data sets

For an evaluation of the other solvers, we performed run-time experiments with synthetic data sets. We present the results of the DMN solver in the following. Results for the FCL solver can be found in [40].

To evaluate this solver, the DMN generator by [99] was used to create synthetic decision tables. In this way, we analyzed a total of 300 synthetic decision tables. As parameters for generating these tables, we chose the number of table columns from {1,2,...,10}, and the number of table rows from {50,100,...,500} (i.e., 10x10 possible combinations). For each of the 100 possible combinations of rows and columns, we generated 3 decision tables with different random rules (i.e., 300 decision tables). The respective rules of these tables were randomly generated by using random integer conditions, with one of the operators from

---

[5] `https://data.4tu.nl/repository/uuid:915d2bfb-7e84-49ad-a286-dc35f063a460`

$\{=, [a..b], \leq, \geq, <, >\}$. We consequently applied our verification tool and computed the average run-time for each parameter configuration, which is shown in Figure 4.1.

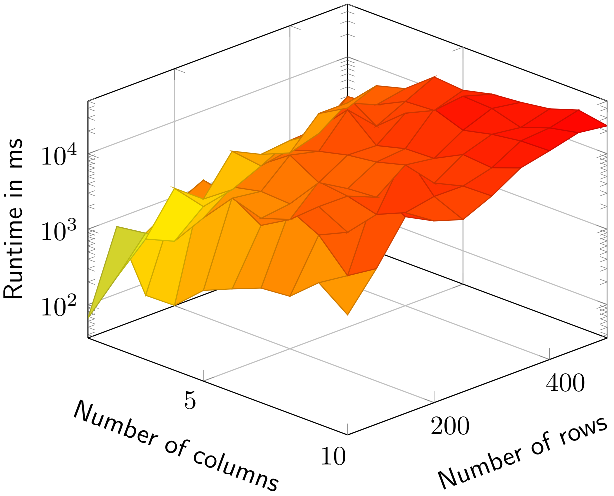


Figure 4.1: Run-time results for the DMN solver.

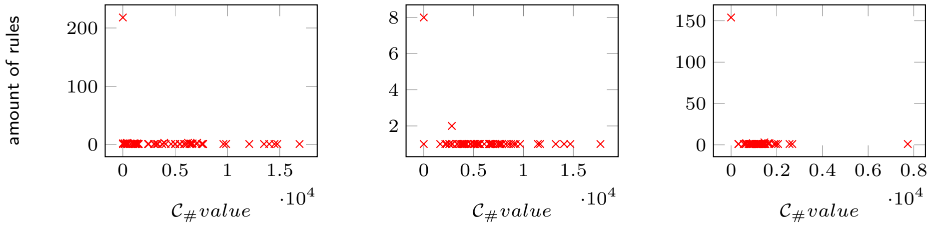
For the analyzed synthetic data, for the highest configuration, i.e. 10 columns and 450 rows, the run-time is roughly 26 seconds. Note that this configuration, i.e., size of the analyzed table, is based on the experiment setup in [11] and is in line with the size of business rule bases encountered in own research projects with industrial partners.

4.3.2 Resolution based on Culpability Measurement

▷ *This section corresponds to the works [28, 31] by the author.*

To evaluate our proposed algorithm for stepwise inconsistency resolution based on culpability measurement, we performed run-time experiments with the abovementioned real-life (Declare) data sets. For the three Declare rule bases described above, we computed the culpability values for all rules.

Figure 4.2 shows the distribution of the $\mathcal{C}_{\#}$ culpability values for the constraints mined from the respective logs. The x-axis shows the respective culpability value, while the y-axis shows the number of constraints with this value. Especially for the larger models BPI'17 and Sepsis'16, there are a high number of constraints with a culpability value of 0 (i.e. they are not part of any minimal issue), and only a few constraints which are highly responsible in the context of the overall inconsistency (i.e. they are part of many minimal issues). For instance, in the constraint set mined from the BPI'2017 log, there are around 200 constraints with a culpability of 0, which can thus be seen as unproblematic.



(a) BPI challenge 2017

(b) BPI challenge 2018

(c) Sepsis 2016

Figure 4.2: Distribution of culpability values for the constraints in the respective models, using the $\mathcal{C}_{\#}$ measure [31].

This equates to $\frac{2}{3}$ of all constraints. It is thus possible to identify those (roughly) other 100 constraints, which should be attended to. This piece of information provides valuable business intelligence which can be used as an effective driver for understanding inconsistencies in the context of resolution strategies.

Due to the culpability distribution in the analyzed data-sets, it was possible to identify a few rules that can be targeted to effectively resolve the overall inconsistency. We discuss this for the BPI 2017 log, which had the highest number of minimal issues.

From the BPI Challenge 2017 log, we mined three different process models, with the respective support factors of 75%, 85% and 95%. We then applied our algorithm for stepwise inconsistency resolution (Algorithm 1) to all three models. Table 4.2 summarizes our evaluation results.

Support Factor	**75%**	**85%**	**95%**
Constraints	305	70	207
Initial number of issues	28954	731	639
Deletions needed	5	1	1
Information Loss	1.63%	1.43%	0.48%
Runtime	101099ms	9148ms	4695ms

Table 4.2: Results of run-time experiments for the analyzed data-sets [28].

The largest model mined with a support factor of 75% (M1) comprised 305 constraints. The presented algorithm was able to resolve all minimal issues with deletion of only 5 constraints (information loss of 1.6%[6]). Figure 4.3 shows an overview of the number of issues remaining after each iteration of our algorithm for M1. This nicely shows that the number of issues is reduced roughly by half after only two iterations. Figure 4.3 also shows the $\mathcal{C}_{\#}$ culpability value of the respective constraint with highest culpability in each iteration. As can be seen, the first iteration resolves around 16000 of all 28954 MIS. Interestingly, for the other two models, only one element needed to be deleted, as this constraint was

---

[6] We define information loss as the number of elements deleted from a rule base, i.e., given a rule base $\mathcal{B}$ and a modified rule base $\mathcal{B}'$ (obtained by deleting elements from $\mathcal{B}$), information loss in this work is a function $L : \mathbb{B} \times \mathbb{B} \rightarrow \mathbb{N}$, defined via $L(\mathcal{B}, \mathcal{B}') = |\mathcal{B}| - |\mathcal{B}'|$.

part of all issues.

While our experiments could show a feasible run-time for the analyzed data sets, this is intuitively only an empirical analysis. We refer the reader to Section 4.2 for a discussion of run-time complexity.

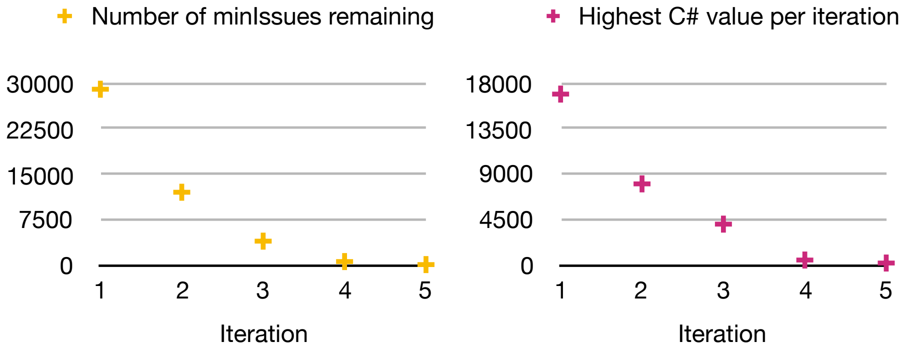


Figure 4.3: Number of minIssues remaining (left) and highest $\mathcal{C}_{\#}$ culpability for a constraint (right) during the iterations of resolving M1.

Summary: Feasibility Evaluation

> For the largest real-life data set (BPI'17), it was possible to compute roughly 29.000 minimal issues in just over 27s.

> Using the proposed Algorithm 1, it was possible to resolve this inconsistency in just over 100s, while deleting only 5 of 305 rules (loss: 1.63%).

4.4 Plausibility Evaluation

▷ *This section corresponds to the works [33, 89] by the author.*
▷ *! The results in the above publications were co-authored by Sabine Nagel*

To empirically evaluate the plausibility of our approach, we conducted experiments with human participants to investigate the effects of our framework on understanding inconsistencies in business rules. The aim of the experiments was to assess the cognitive effects of various aspects related to the proposed measures of inconsistency in the use-case of business rule organizing.

In total, 85 subjects participated in our experiments, which are divided into experiment 1 and 2 in the following. We will briefly recall the experiments, but refer the reader to [33, 89] for further details on the experiment design, settings and participants. We begin with experiment 1.

In experiment 1, the research question was how quantitative measures affect the understanding of inconsistencies in business rules. To this aim, the relation of quantitative measures and understanding was hypothesized as follows.

Hypothesis 1. Providing a quantification of inconsistency in business rules is associated with better understanding accuracy compared to manual analysis.

Hypothesis 2. Providing a quantification of inconsistency in business rules is associated with better understanding efficiency compared to manual analysis.

Hypothesis 3. Providing a quantification of inconsistency in business rules is associated with less mental effort needed for problem understanding compared to manual analysis.

In order to test our hypotheses, a single-factor experiment was conducted where participants had to answer questions regarding inconsistencies in business rules. The experiment setup is shown in Figure 4.4.

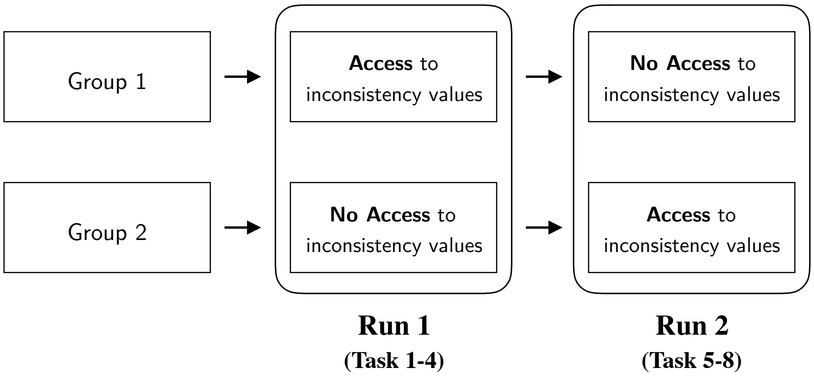
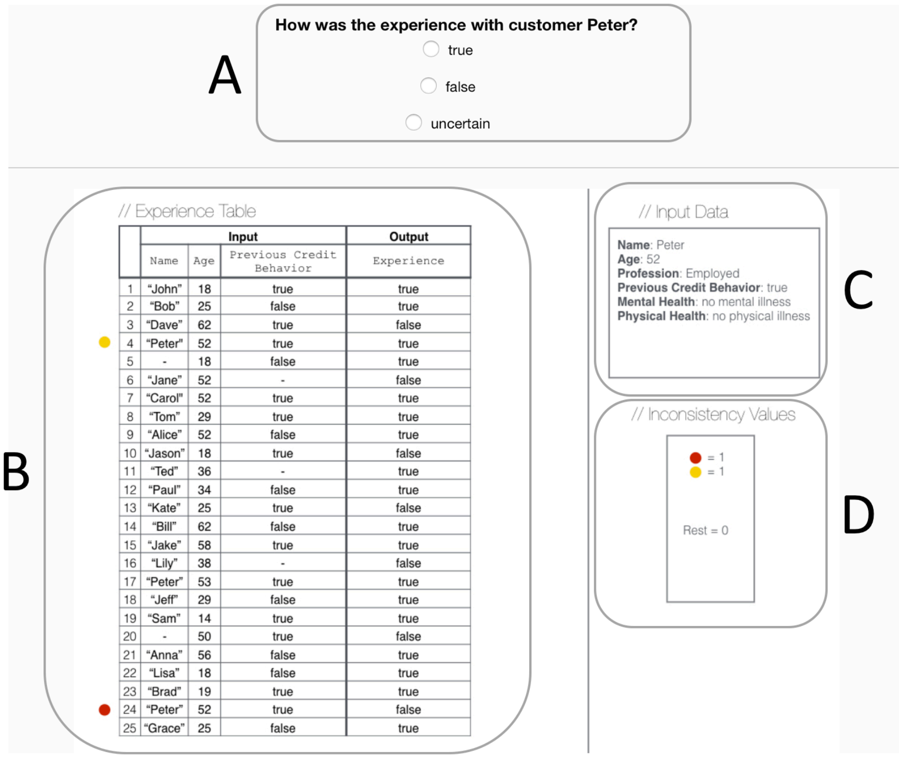


Figure 4.4: Experiment setup.

The considered factor is the use of quantitative measures, with factor levels "present" and "absent". Participants were split into two groups. The experiment was designed as a balanced single factor experiment with repeated measurement, based on an experiment design from [101], meaning that all participants used both factor levels. Thus, all participants conducted two test-runs (which comprises a block of comprehension questions), with both factor level configurations. For the two groups, the order of factor levels was switched to counteract a bias based on learning effects between the two test-runs. An example for a question in such a test-run is shown in Figure 4.5.

The respective question is shown in (A). In (B), business rules were shown in the DMN standard. If applicable, case data was provided in (C). Then, in area (D), quantitative insights were sometimes provided (depending on the factor configuration; otherwise, this area was left blank). As a concrete culpability measure, the $\mathcal{C}_{\#}$ measure was shown. All necessary prerequisites on this measure were also shown to the participants in a short tutorial.



How was the experience with customer Peter?

○ true

○ false

○ uncertain

// Experience Table

		Input			Output
		Name	Age	Previous Credit Behavior	Experience
	1	“John”	18	true	true
	2	“Bob”	25	false	true
	3	“Dave”	62	true	false
●	4	“Peter”	52	true	true
	5	-	18	false	true
	6	“Jane”	52	-	false
	7	“Carol”	52	true	true
	8	“Tom”	29	true	true
	9	“Alice”	52	false	true
	10	“Jason”	18	true	false
	11	“Ted”	36	-	true
	12	“Paul”	34	false	true
	13	“Kate”	25	true	false
	14	“Bill”	62	false	true
	15	“Jake”	58	true	true
	16	“Lily”	38	-	false
	17	“Peter”	53	true	true
	18	“Jeff”	29	false	true
	19	“Sam”	14	true	true
	20	-	50	true	false
	21	“Anna”	56	false	true
	22	“Lisa”	18	false	true
	23	“Brad”	19	true	true
●	24	“Peter”	52	true	false
	25	“Grace”	25	false	true

// Input Data

Name: Peter
Age: 52
Profession: Employed
Previous Credit Behavior: true
Mental Health: no mental illness
Physical Health: no physical illness

// Inconsistency Values

● = 1
● = 1

Rest = 0

Figure 4.5: Exemplary experiment question.

In our experiment, we distinguish between three types of measurements. To measure understanding accuracy, we use the percentage of correctly answered comprehension questions. To measure understanding efficiency, the time needed to complete a test-run was measured. Last, to measure mental effort, the eye-fixation duration was measured (which is the period of time where the eyes remain still and focused on a single location).

In the first experiment, 37 subjects participated. Please observe that our eye-tracking software could not track data for seven participants, thus N = 30 for the mental effort measurement.

After data collection, we compared the two groups using independent sample t-tests at the significance level of 0.05[7]. Table 4.3 shows our experiment results.

As all p-values were smaller than 0.05, we cannot reject our hypothesis. This indicates that quantitative measures are associated with better understanding accuracy, understanding efficiency and less mental effort in business rules management. In result, the quantitative measures provided as an outcome of our framework can induce positive cognitive effects in business rules management.

---

[7] Please refer to [89] for further details on the statistical pretests and parameters applied.

	Run 1		Run 2	
Group	1	2	1	2
N	19	18	19	18
Mean	0.95	0.49	0.62	0.94
Std. Dev.	0.13	0.21	0.19	0.16
p (1-tailed)	<0.0001		<0.0001	

Understanding Accuracy (%*100)

Run 1		Run 2	
1	2	1	2
19	18	19	18
38.89	86.06	74.47	32.32
23.72	32.25	31.45	25.25
<0.0001		<0.0001	

Understanding Efficiency (s)

Run 1		Run 2	
1	2	1	2
15	15	15	15
37.38	81.05	65.18	26.46
23.27	33.46	23.39	14.54
0.0002		<0.0001	

Objective Mental Effort (s)

Table 4.3: Overview of experiment results for understanding accuracy (Fraction of correct answers), understanding efficiency (task processing in Seconds), and objective mental effort (eye-fixation duration in Seconds) [89].

This can also be seen when looking at the collected eye-tracking data. Figure 4.6 shows the gaze distribution for an exemplary task. The red areas indicate a higher fixation duration.

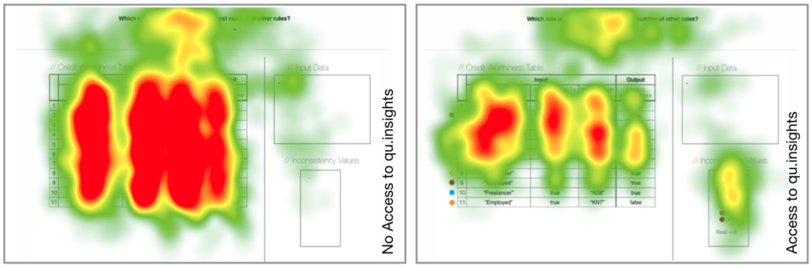
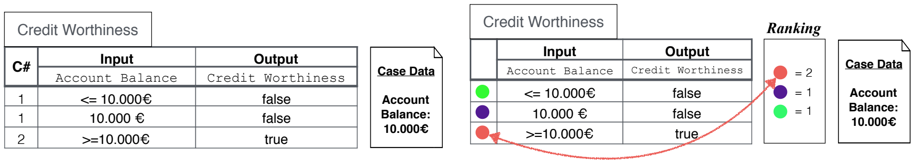


Figure 4.6: Comparison of Gaze-Distribution for a corresponding question (with different factor levels).

As can be seen, fixation duration without access to inconsistency values was significantly higher. Also, it can be seen that in cases where insights are provided, participants shortly read these values and could then directly find the necessary information to answer the question.

For the second experiment, the research question was how visualization techniques for inconsistency values affect the capability to analyze and interpret inconsistencies in automated decision-making.

We considered two ways to visualize values, namely an integrated visualization, and a (separate) ranked overview, as depicted in Figure 4.7.


Credit Worthiness

C#	Input Account Balance	Output Credit Worthiness
1	<= 10.000€	false
1	10.000 €	false
2	>=10.000€	true

Case Data
Account Balance: 10.000€

Credit Worthiness

	Input Account Balance	Output Credit Worthiness
🟢	<= 10.000€	false
🟣	10.000 €	false
🔴	>=10.000€	true

Ranking
🔴 = 2
🟣 = 1
🟢 = 1

Case Data
Account Balance: 10.000€

(a) Integrated Visualization

(b) Ranking-Based Visualization

Figure 4.7: Comparison of the considered visualization approaches.

In the integrated visualization, inconsistency values are shown directly within the table, i.e., as a column of the DMN table. In [33], we have proposed a separate, ranked overview visualization. This ranking allows to prepare information following the culpability ranking. In this second experiment, the goal was thus to investigate which visualization technique is better in the setting of BRM.

Interestingly, following the literature, both approaches have advantages and disadvantages, and there are even contradicting results from cognitive psychology that should be considered. A central aspect in this context is that of cognitive load theory [136], explained as follows.

In the integrated visualization, the inconsistency values and business rules are presented in a unified way. Following [136,137], this can lower the extraneous cognitive load for processing the information, due to a minimization of the so-called split-attention effect (which occurs if two related pieces of information have to be linked mentally).

On the other hand, as there can be multiple tables and inconsistency can be distributed, an expert might have to switch between tables. Here, the ranking-based overview allows to present the inconsistency values in a ranked manner, which can lower the amount of information that needs to be processed. Following [94, 123], other factors such as elaborative encodings can also promote a better information processing in this ranking-based visualization. However, a disadvantage of the ranking-based overview is that inconsistency values and rules are separated and need to be linked (which could introduce cognitive load).

From the literature, it is therefore not clear whether the lower amount of information that needs to be processed outweighs the potential additional extraneous cognitive load. Therefore, this second experiment was conducted.

The experiment design was similar to the first experiment, i.e., there were two groups, and two factor levels, i.e. integrated visualization and ranked visualization. Again, participants completed two test runs with different factor levels. We refer the reader to [33] for further details on the experiment design.

Our Hypothesis for the second experiment were as follows.

Hypothesis 4. A ranking-based visualization of inconsistency values is associated with a better understanding efficiency compared to an integrated visualization.

Hypothesis 5. A ranking-based visualization of inconsistency values is associated with less objective mental effort compared to an integrated visualization.

Hypothesis 6. A ranking-based visualization of inconsistency values is associated with less perceived mental effort compared to an integrated visualization.

As measurements, the task processing time was used to measure understanding efficiency, the eye-fixation duration for objective mental effort, and furthermore a direct questionnaire for perceived mental effort.

In the second experiment, 48 subjects participated. After data collection, we compared the two groups using independent sample t tests [8]. Table 4.4 shows our experiment results.

---

[8] We refer the reader to [33] for details on the applied statistical tests.

	Run 1		Run 2		Run 1		Run 2	
Group	1(i)	2(r)	1(r)	2(i)	1(i)	2(r)	1(r)	2(i)
N	24	24	24	24	21	21	21	21
Mean	34.54	29.04	23.09	25.02	30.95	22.68	19.58	19.78
Std. Dev.	8.71	12.89	8.47	16.24	16.22	23.73	15.22	26.97
p (1-tailed)	**0.048**		0.3		**0.004**		0.47	

Understanding Efficiency

Objective mental Effort

Table 4.4: Overview of experiment results for understanding efficiency (task processing in Seconds), and objective mental effort (eye-fixation duration in Seconds). i = integrated visualization, r = ranking-based visualization [33].

For the measured data, a ranking-based visualization of inconsistency metrics is (partially) associated with a better understanding efficiency compared to an integrated visualization (i.e., on average better, and significantly better in Run 1). Also, a ranking-based visualization of inconsistency metrics is (partially) associated with less objective mental effort compared to an integrated visualization. For the perceived mental effort, we asked participants to specify which visualization they found easier (or whether they found them equally difficult). To recall, all participants went through both visualizations during the experiments. In total, 52% of subjects stated that the ranking-based visualization was easier, 29% stated the integrated approach was easier, and 18% stated they found it equally helpful. We coded these results to statistically compare these subjective indications (each visualization was awarded 2 points if it was selected to be easier, and both received 1 point if "equal" was selected). Here, we then compared both visualizations with the Wilcoxon signed-rank test[9]. The p value is 0.039, thus we cannot reject our hypothesis. Thus, for the measured data, a ranking-based visualization of inconsistency metrics was associated with less perceived mental effort compared to an integrated visualization.

To summarize, the benefits of having to process less information via the ranked visualization seem to outweigh the cognitive costs of information assimilation (split-attention effect).

This can also be seen when looking at the collected eye-tracking data. Figure 4.8 shows the gaze distribution for an exemplary task. The red areas indicate a higher fixation duration. As can be seen, for the exemplary task, participants had on average less fixation duration via the ranking-based visualization than with the integrated visualization. The post experiment interviews also confirmed the gist of the visualized data. Here, there was a consensus amongst participants, that the ranking allowed to find relevant rules more efficiently. In many cases, having the guidance of a ranking was perceived as pleasant by individual participants.

---

[9] The Wilcoxon signed-rank test is used for dependent samples

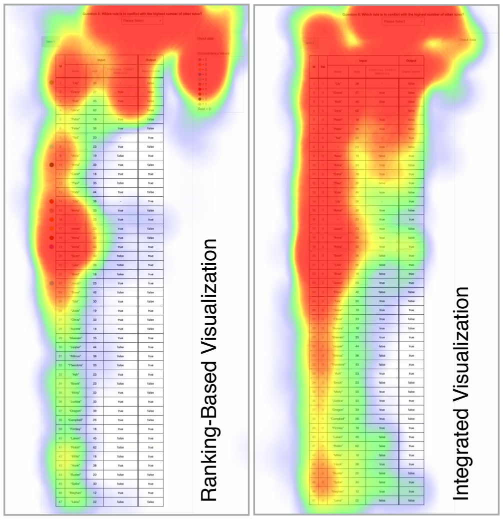


Figure 4.8: Comparison of Gaze-Distribution for a corresponding question (with different factor levels).

Summary: Plausibility Evaluation

$>$ Quantitative measures were associated with better understanding accuracy, better understanding efficiency, and less mental effort needed for understanding inconsistencies in business rules.

$>$ A ranking-based visualization, i.e., providing experts the proposed culpability ranking, was (partially) associated with better understanding efficiency and less objective/perceived mental effort.

4.5 Framework- and Meta-Evaluation

▷ *This section uses the taxonomy proposed in [32] by the author.*

As a main result of this thesis, we distilled the developed methods and techniques into a novel framework for handling inconsistency in the scope of business rule organizing. In this section, we evaluate our framework in the context of related work and provide a meta-evaluation, which evaluates the sufficiency of our evaluation efforts.

The scope of our framework follows the thesis scope shown in Figure 1.7, i.e., it solves a problem in BRM to support compliance management.

Regarding compliance management, important dimensions which need to be addressed were identified in the *regulatory compliance management framework* by EL Kharbili (2012). Table 4.5 shows how our results address these individual dimensions, divided into the components of modeling, compliance checking, compliance analysis and compliance enactment.

	Modeling	Verification	Monitoring	Auditing	Reporting	Explanation	Recovery	Resolution
Proposed Framework	n/a	✓	✓	o	✓	✓	o	o

Table 4.5: Compliance with the regulatory compliance management framework proposed in [44] (✓= support, o = initial ideas discussed in this work, yet future work needed).

To recall, the business rule lifefycle as used in this work (cf. Figure 1.6) contains the phases of elicitation, authoring and organizing. In the framework by El Kharbili (2012), elicitation and authoring as defined in this work are terminologicaly referred to as *modeling*, hence this dimension is beyond the scope of this thesis (n/a).

The second main group of dimensions in [44] is *compliance checking*, i.e. (design-time) verification, (run-time) monitoring and (post-execution) auditing. As discussed and shown in Table 4.5, our results support verification and monitoring by providing methods and techniques for ensuring a consistent rule set, cf. Sections 3.2.1-3.2.3. Initial ideas for auditing were discussed, yet, this is beyond the scope of this thesis and could be interesting for future work.

The third main group of dimensions in [44] is *compliance analysis*, i.e. reporting and explanation. Reporting refers to generating a documentation of detected errors, while explanation refers to the ability to identify the root-causes of problems. As shown in Figure 3.12, a proof-of-concept tool reporting dashboard for Camunda was developed in the scope of this work. By means such as culpability measurement, our results support these two dimensions, as companies can be presented with a careful analysis of the detected errors.

The last main group of dimension in [44] is *compliance enactment*, i.e., recovery and resolution. Recovery refers to the ability of reacting to the occurrence of errors during run-time, while resolution refers to eliminating inconsistencies through re-modelling. Our tool in [30] reacts to the occurrence of inconsistencies by stopping process execution, in order to mitigate potential compliance breaches due to inconsistent decision-making during run-time. Other means to offer continuous support after the occurrence of errors can easily be adapted from [84]. Also, the results on resolving inconsistencies based on culpability measurement presented in Section 3.3 provide initial ideas towards the resolution dimension in Table 4.5. However, the results presented in this work are to be seen as initial approximation algorithms, and more work is needed for approaches for compliance enactment in BRM.

Interim Result 4.1 (Compliance Management). Our framework supports the dimensions as shown in Figure 4.5 in the compliance management framework from [44]. In our opinion, this is an important indicator that our framework is suitable for an application within the domain of compliance management.

Regarding rule organizing, important dimensions which should be addressed were identified in [32] by the author of this thesis. Table 4.6 classifies our framework in comparison to the business rule organizing approaches identified in the survey in [32].

As a main contribution, our framework supports multiple compliance management phases, which has strongly been neglected in the past based on our findings. Here, recent works such as [63, 105] strongly advocate supporting multiple compliance management phases, which we thus see as an advantage of our approach. As discussed in Section 3.2.3, considering the intersections of different phases can also provide further benefits as opposed to employing different strategies individually.

Furthermore, our framework provides capabilities for the detection and analysis, as well as initial means for the resolution of errors within the scope of rule organizing. Regarding these capabilities, especially the detection of all (quasi-) inconsistent rule sets at design-time, as well as the presented means for measuring inconsistency in rule bases can be seen as primary contributions of this thesis (cf. Section 3.2.1).

All our results have been implemented, both as a general library which supports the rule formalisms FCL, DMN and Declare, as well as user-friendly solutions via web-based tools [27, 30, 64]. Our results can therefore be used ”out-of-the-box” by companies seeking to implement rule organizing. As we base our results on the general rule form 1.1 proposed in [58], our results can easily be extended to other rule formalisms rooted in such premise/conclusion structures.

Interim Result 4.2 (Business Rule Organizing). Our framework supports multiple dimensions as identified in [32]. This contribution is thus a next step towards a multi-phase framework for business rule organizing.

One point we would like to mention is that regarding the supported compliance management strategies, our findings shown in Table 4.6 indicate a strong focus on design-time compliance management. While we agree that this is an extremely important phase, our own results as well as works such as [84, 85] clearly indicate that certain errors cannot be detected a priori. This raises the need for advanced monitoring techniques [44, 63, 82].

Regarding such a compliance monitoring, the evaluation framework by LY ET AL. (2013) identifies important requirements for run-time compliance monitoring approaches. Table 4.7 shows how our results address these individual requirements.

	BPC Strategy			Capabilities			Applicability	
Literature	Design-Time Compliance	Run-time Compliance	Post-Execution Compliance	Detection	Quantitative Analysis	Resolution	Rule Formalism	Tool
Kardasis et al. (2004)	X						Text	
Fu et al. (2004)	X			X			n/a	X
Lin et al. (2005)	X						FOL	
Bajec et al. (2005)	X						Text	X
Hicks (2007)	X			X			Text	X
Cheng et al. (2009)	X			X			n/a	
Governatori et al. (2009)	X			X		A	FCL	
Governatori et al. (2010)	X			X		A	FCL	
Levy et al. (2010)	X			X			RIF	X
Maggi et al. (2011)		X		X	X		LTL	
Maggi et al. (2011)		X		X	X		LTL	
Decker et al. (2013)	X			X	X		FOL	
da Silva et al. (2013)	X						DRL	X
Maggi et al. (2013)	X			X		A	LTL	X
Berstell-Silva et al. (2014)	X			X		A	FOL	
Cuzzocrea et al. (2014)	X			X	X		FOL	
Santos et al. (2014)	X			X		A	SBVR	X
Zhang al. (2014)	X			X		A	n/a	X
Cemus et al. (2015)	X						n/a	
Gomez et al. (2015)		X		X			n/a	
Olivieri et al. (2015)	X			X		A	FCL	
Agli et al. (2016)	X			X		A	ILOG	X
Burgstaller et al. (2016)	X						SBVR	
Calvanese et al. (2016)	X			X			DMN	X
De Smedt et al. (2016)	X			X			LTL	
Di Ciccio et al. (2016)	X			X		A	LTL	X
Gomez et al. (2016)		X		X			n/a	X
Houari et al. (2016)	X			X			n/a	X
Batoulis et al. (2017)	X			X			DMN	
Calvanese et al. (2017)	X			X			DMN	
Di Ciccio et al. (2017)	X			X		A	LTL	X
Anand et al. (2018)	X			X			SBVR	X
Batoulis et al. (2018)	X			X		A	DMN	X
Calvanese et al. (2018)	X			X		A	DMN	X
De Smedt et al. (2018)	X			X			LTL	
Ezekiel et al. (2018)	X			X			DRL	X
This Work	**X**	**X**		**X**	**X**	**SA**	**FCL, DMN, LTL**	**X**

**Post-Execution Compliance:* Possible for the proposed framework but not part of this thesis.
Resolution: SA (Semi-Automated), A (Automated).
Rule Formalisms: TEXT (textual description/natural language), N/A (Non-standard or individual rule formalisms, if-then-structures), FOL (First-order logic), FCL (Formal Contract Language), RIF (Rule Interchange Format), DRL (Drools Rule Language), LTL (Linear Temporal Logic, resp. DECLARE), SBVR (Semantics of Business Vocabulary and Business Rules), ILOG (IBM Ilog), DMN (Decision Model and Notation).

Table 4.6: Comparison of the proposed framework to existing business rule organizing approaches [32].

	Temporal constraints	Data constraints	Organizational Constraints	Non-Atomic Constraints	Supporting Activity Orderings	Multi-Instance Support	Reactive Management	Proactive Management	Root-Cause Analysis	Compliance Degree
Proposed framework	✓	✓	✓	o	✓	o	✓	✓	✓	✓

Table 4.7: Compliance with the requirements for compliance monitoring approaches proposed in [82] (o = partial support).

The first three requirements deal with supported rule types. As our results are based on a general rule form in 1.1, such rule types can be captured by means of atomic propositions. Also, temporal constraint support was specifically implemented via Declare (LTL).

Next, our results support explicit non-atomic constraints via Declare, e.g. such as the exemplary rule "*An order creation cannot be completed until the customer registration is completed*" from [82]. Based only on atomic events, a support for implicit non-atomic constraints is currently not possible, e.g. such as the exemplary rule "*Activity check project can be executed only while the project is under preparation*" from [82]. However, this can be supported by introducing artifical start- and end- events to event logs.

The developed tools currently do not support multi-instance monitoring during run-time. However, as discussed in Section 3.2.3, the proposed means for post-execution auditing allow multi-instance support (in the referenced section referred to as different "cases").

Continuing, the proof-of-concept presented in [30] shows a tool which can automatically stop process execution during monitoring, i.e. supports reactive management. Specific actions for reacting to inconsistencies in the DMN standard were also proposed by the author and colleagues in [64]. Regarding proactive management, our results offer support in the form of detecting all potential inconsistencies at design-time. This could in theory be used to proactively mitigate all potential issues at design-time. A limitation of our approach is that potential issues are currently not addressed at the hand-off between design-time and run-time compliance. That is, currently, potential issues are kept as-is during design-time, and only actual issues are removed here. While potential issues are monitored later and can also be analyzed, one could also try to already assess and resolve potential issues at design-time. A problem here is however that there can be a vast number of potential issues, as this is basically any combination of facts such that a rule base becomes $F-$inconsistent. Simply computing and presenting potential issues at design-time is therefore not feasible due to the vast amount of information that can be expected. Here, future work could focus on means to filter and prioritize potential issues with preemptive diagnostics. For example, the system could recommend to the user that a certain combination of facts will cause severe problems – an expert could then assess whether this fact combination is likely to be expected later during run-time. Likewise, historic event data could be used to compute a probability for the occurrence of certain fact combinations. Then, cost functions could be defined that consider a trade-off between an expected likelihood of the occurrence of fact combinations and the expected severity of inconsistencies arising from this input, which would allow to further resolve potential issues at design-time.

Last, our results support the requirement of root-cause analysis via culpability measurement. Also, quantifying inconsistency and therefore providing a compliance degree is a core contribution of this work.

Interim Result 4.3 (Compliance Monitoring). Our framework supports 8/10 requirements for compliance monitoring approaches identified in [82], and partially supports the two others. Strategies for conquering limitations in future work were identified. In our opinion, this is an important indicator for the suitability of using our approach for run-time compliance management.

To conclude, in this chapter, we have presented our evaluation results. The question which arises in this context is however: *"What makes a good evaluation?"*. In this thesis, we will not provide a definite answer to this question, however, we will try to provide some contribution on this matter:

In our review on rule organizing approaches [32], we also analyzed how the individual works evaluated their approaches. This allowed us to capture the state-of-the-art evaluation methods/strategies. In that publication, we consequently proposed a guideline for evaluating rule organizing approaches, namely by means of *demonstration, case-studies, run-time experiments, comparison, analytical evaluation* and *(human) experiments*. Table 4.8 shows the evaluation methods addressed in this thesis in regard to the guidelines proposed in [32].

	Demonstration	Case-Study	Run-Time Experiments	Comparison	Analytical Evaluation	Plausibility Experiments
This work	✓	✓	✓	✓	✓	✓

Table 4.8: Evaluation methods conducted for this research in the scope of the guidelines for evaluating business rule organizing approaches proposed in [32].

Throughout this thesis, the proof-of-concept implementations as well as methods and techniques are referenced for demonstration. Also, the tools and techniques were evaluated by means of complexity analysis and run-time experiments (cf. Section 4.3). Here, real-life data-sets could also be analyzed to provide evidence based on actual case-studies. Our results and framework were compared to other works in the taxonomy developed in [32] (cf. Table 4.6). Regarding analytical evaluation, this was a main evaluation method used for evaluating the proposed inconsistency measures. Based on the survey by Thimm (2018), the (analytical) evaluation of inconsistency measures can be performed on the dimensions of *compliance with rationality postulates*, *expressivity* (essentially: how many different values can a measure yield based on different types of knowledge bases) and *computational complexity.* A main focus in this work was the evaluation based on the compliance with rationality postulates (cf. Section 4.1). Also, we evaluated the computational complexity of different aspects regarding (quasi-)inconsistency (cf. Section 4.2). Future work could further investigate the expressivity of inconsistency measures, however, we see this as part of a different scope. Instead, a focus in this thesis was set on experiments with human participants (cf. Section 4.4) to investigate the plausibility of applying results from inconsistency measurement to the domain of business rule management.

Interim Result 4.4 (Meta-Evaluation). Following the guideline for evaluating business rule organizing approaches in [32], we are confident that we have applied all state-of-the-art evaluation techniques.

Intuitively, it is always possible to conduct further experiments and continue the research. However, based on the various peer-review processes for the individual publications, we believe that the current extent of our evaluation can be seen as adequate by the standards of the relevant scientific communities.

5

Conclusion

The main results of this thesis are methods and techniques for the detection, analysis and resolution of inconsistencies in business rule bases (Goal of design $D1$). These results have been implemented as a general library which supports the rule standards FCL, DMN and Declare, as well as several proof-of-concept prototypes with browser-based interfaces (Goal of design $D2$).

The contribution of the developed artifacts in the context of related work could be identified in the course of this thesis (Goal of understanding $U1$). Importantly, the feasibility of the developed artifacts was also iteratively evaluated by means of analytical evaluation, proof-of-concept implementations, complexity analysis, analysis of real-life data-sets and plausibility experiments, to warrant a rigorous design science research approach (Goal of understanding $U2$).

To answer our main research question how inconsistencies in business rule bases can be detected, analyzed and resolved in the scope of BRM, we addressed a series of subsidiary research questions. In particular, we discussed the question

> *SRQ1: How must current results from inconsistency measurement be extended to make them applicable in BRM?*

As an important outcome, we showed that an application of current results from inconsistency measurement to the domain of business rule management is not feasible. This results from a series of conceptual mismatches between these two fields, such as different granularities within the considered knowledge representation formalisms (propositions vs facts/rules) and special requirements during the different compliance management phases. For the latter, an important outcome was that the classic inconsistency measurement is not feasible for design-time rule bases, as no facts are known during this phase. To counteract this problem, we introduced the notion of quasi-inconsistency, which presents a novel conceptual problem for the discipline of inconsistency measurement. Also, we developed new means to detect (quasi-)inconsistencies in business rule bases. Following a detection of inconsistencies, we then discussed the question:

SRQ2: How can inconsistency be measured in business rules?

As a further central outcome, we brought forward a series of rationality postulates to capture the desired behavior required from inconsistency measures with regard to the individual compliance management phases. We showed that classical measures do not satisfy these postulates and can therefore not be plausibly applied for the use-cases of measuring inconsistency in business rule bases. Subsequently, we developed new means for the intended use-cases and showed their compliance to the proposed rationality postulates. Furthermore, we developed means to allow for an analysis of individual business rules via culpability measurement. The development of these analysis capabilities then lead to an investigation of the following question:

SRQ3: How can quantitative measures be used to serve as a basis for inconsistency resolution?

Following the approach of stepwise inconsistency resolution proposed by GRANT AND HUNTER (2011), we proposed stepwise resolution of inconsistency via culpability measurement. To this aim, highly problematic business rules are iteratively removed from the rule base to restore consistency. The order in which rules are removed is determined based on a culpability ranking, which ranks business rules by their degree of culpability w.r.t. individual culpability measures.

As dictated by the applied design science research approach and the goal of design $U2$, the evaluation of the developed artifacts was of central importance. Here, a particular focus was set on the following question:

SRQ4: How feasible is the application of inconsistency analysis in business rule bases, and what is the added-value for BRM?

Embedded within the applied research method, the developed means and implemented results were iteratively evaluated. To further investigate the added-value of our results for BRM, we also conducted experiments with human participants, where we evaluated the cognitive effects of providing human modelers with quantitative measures, resp. a culpability-ranking. Our results broadly indicate that access to our developed artifacts are associated with a higher understanding accuracy, higher understanding efficiency and less mental effort needed to understand inconsistencies in business rules.

Finally, we discussed the question:

SRQ5: How can a BRO approach supporting multiple BPC strategies by the proposed means be created?

To this aim, we proposed a framework for handling inconsistencies in business rule management, by distilling the above results. As discussed in Section 3.5, our framework supports the BPC phases of design-time and run-time compliance management, and supports a broad selection of capabilities as identified in [32]. Here, we also presented implementations of our methods and techniques, allowing to use our results "out-of-the-box".

This thesis bridges the gap between the scientific fields of inconsistency measurement and business rule management, and closes multiple research gaps regarding the need for novel methods and techniques for (quantitative) assesments of errors in business rule bases, raised in works such as [4,9,11,21,42,81,106,120]. From a practitioner's point of view, our results can be used to support business rule organizing. Here, the presented means to handle inconsistency in business rule bases facilitate sustainable business rules management and can be seen as an important prerequisite for ensuring compliant business processes.

Our work also opens several new research streams for the fields of knowledge representation and reasoning and business rule management.

Preemptive Diagnostics. So far, many rule organizing approaches focus on individual compliance management phases. However, germane to recent suggestions in literature [63, 84, 105], we showed that considering the intersections of different phases can yield novel insights, further helping companies to understand issues within the modelled business rules. During the design-phase, our results allow to compute all potential inconsistencies. Currently, our framework considers mainly actual issues during design-time, and rather monitors potential issues during run-time. Here, future work could focus on means to filter and prioritize potential issues with preemptive diagnostics. For example, the system could recommend to the user that a certain combinations of facts will cause severe problem – an expert could then assess whether this fact combination is likely to be expected later during run-time. Likewise, historic event data could be used to compute a probability for the occurrence of certain fact combinations. Then, cost functions could be defined that consider a trade-off between an expected likelihood of the occurrence of fact combinations and the expected severity of inconsistencies arising from this input, which would allow to further assess or automatedly resolve potential issues during design-time.

Measuring Inconsistency in Sets of Knowledge Bases. Further regarding potential issues, we discussed a measure to assess the fraction of how many potential issues transformed into actual issues during run-time (from a post-execution auditing perspective). This is heavily based on not only considering individual run-time perspectives, but also the relation of different process instances during auditing. Essentially, this relates to measuring inconsistency over multiple knowledge bases from a post-execution perspective, where these bases have a shared set of atoms. This could be an interesting research stream for the field of inconsistency measurement and allows for the development of novel inconsistency and culpability measures, as already drafted in this work. In general, it seems also considering novel means for rule organizing from an auditing perspective could be highly beneficial. By assessing the relations of individual process instances, novel insights could be generated to guide experts in the re-modelling process. The proposed framework for handling inconsistencies in business rule bases can easily be extended with a post-execution perspective, thus, future work should investigate such means for rule organizing from a post-execution perspective, in order to extend the proposed framework into a holistic framework for BRO, supporting all compliance management strategies.

Evaluating the Suitability of Inconsistency Measures. A problem for applying results from inconsistency measurement as a driver for inconsistency resolution in business rule bases is the current dispute on how the "severity of inconsistency" should be quantified. To clarify, there exist many proposals for inconsistency measures, which may yield completely differing assessments for the same knowledge base. From Thimm (2017), we for example see that for every pair of inconsistency measures $\mathcal{I}, \mathcal{I}'$, we can find a pair of knowledge bases $\mathcal{K}_a, \mathcal{K}_b$, s.t. $\mathcal{I}(\mathcal{K}_a) > \mathcal{I}(\mathcal{K}_b)$, but $\mathcal{I}'(\mathcal{K}_a) < \mathcal{I}'(\mathcal{K}_b)$. It is therefore not clear, which exact measures provide insights that are perceived as valuable by employees, and thus which exact measures should be used in the scope of BRO. Further plausibility evaluations could yield results regarding the perceived value of inconsistency measures and could be used as a basis for determining suitable inconsistency measures wrt. specific application contexts. Here, exploiting research approaches from other scientific fields, such as experimental research with human participants, could allow to develop new evaluation dimensions for inconsistency measures (cf. the discussion on potential limitations of current evaluation dimensions in [129]).

Extending the notion of inconsistency. Traditionally, the field of inconsistency measurement is geared towards the quantitative assessment of classic-logical inconsistency. Measuring inconsistency in other formalisms has already been proposed in recent works (cf. e.g. [62]), and a direct extension of this work could be to investigate inconsistency for defeasible logic (i.e. assessing inconsistencies in the presence of superiority relations) as already prototypically implemented in the presented library. Additionally, future work could also consider inconsistency measurement from a completely different conceptual lense. In this work, we have already presented the novel concept of quasi-inconsistency, which in a sense allows to quantify the degree of a novel "type" of inconsistency. Here, further investigating the domain of BRM could yield additional application scenarios. For example, in the DMN standard, business rule bases do not only comprise the rules themselves, but also a meta-level model, which specifies the in- and output of the individual rules and their relations. As maintaining DMN meta-models and corresponding rules is again a human modelling task, "inconsistency" can arise between the meta-level and the rule level. For instance, any atoms appearing in DMN rules should be defined on a meta-level, however, it can occur that new atoms are added to rule bodies but forgotten on a meta-level, or vice-versa. Such a modeling error however makes the DMN model "inconsistent" and impedes a sound usage. In [64], the author and colleagues presented an initial catalogue of "inconsistency types" that can arise between the meta-level and the rule-level in the DMN standard (cf. the above example). It seems such human modeling errors, allthough they are not "classical" inconsistencies, are a current challenge for companies and need to be addressed [9, 10, 32, 64, 72, 120]. By extending the notion of inconsistency, further means for assessing such human modeling errors could be developed in future work, extending both the fields of business rule management and inconsistency measurement.

Bibliography

[1] van der Aalst, W.M.P.: Process mining: discovery, conformance and enhancement of business processes. Springer (2011)

[2] van der Aalst, W.M.P., La Rosa, M., Santoro, F.M.: Business process management. Business & Information Systems Engineering **58**(1), 1–6 (2016)

[3] Agli, H., Bonnard, P., Gonzales, C., Wuillemin, P.: Business rules uncertainty management with probabilistic relational models. In: Alferes, J.J., Bertossi, L.E., Governatori, G., Fodor, P., Roman, D. (eds.) Proceedings of the 10th International Symposium on Rules and Rule Markup Languages for the Semantic Web (RuleML 2016), Stony Brook, USA, July 6-9, 2016. Lecture Notes in Computer Science, vol. 9718, pp. 53–67. Springer (2016)

[4] Akhigbe, O., Amyot, D., Richards, G.: Information technology artifacts in the regulatory compliance of business processes: a meta-analysis. In: Benyoucef, M., Weiss, M., Mili, H. (eds.) Proceedings of the 6th International Conference on E-Technologies (MCETECH2015), Montreal, Canada, May 17-19, 2015. pp. 89–104. Springer (2015)

[5] Anand, K., Chittimalli, P.K., Naik, R.: An automated detection of inconsistencies in sbvr-based business rules using many-sorted logic. In: Calimeri, F., Hamlen, K.W., Leone, N. (eds.) Proceedings of the 20th International Symposium on Practical Aspects of Declarative Languages (PADL 2018), Los Angeles, USA, January 8-9, 2018. Lecture Notes in Computer Science, vol. 10702, pp. 80–96. Springer (2018)

[6] Bajec, M., Krisper, M.: Issues and challenges in business rule-based information systems development. In: Bartmann, D., Rajola, F., Kallinikos, J., Avison, D.E., Winter, R., Ein-Dor, P., Becker, J., Bodendorf, F., Weinhardt, C. (eds.) Proceedings of the 13th European Conference on Information Systems (ECIS 2005), Regensburg, Germany, May 26-28, 2005. pp. 887–898 (2005)

[7] Bajec, M., Krisper, M.: A methodology and tool support for managing business rules in organisations. Information Systems **30**(6), 423–443 (2005)

[8] Baskerville, R., Lyytinen, K., Sambamurthy, V., Straub, D.: A response to the design-oriented information systems research memorandum. European journal of information systems **20**(1), 11–15 (2011)

[9] Batoulis, K., Nesterenko, A., Repitsch, G., Weske, M.: Decision management in the insurance industry: Standards and tools. In: Brambilla, M., Hildebrandt, T.T. (eds.) Proceedings of the BPM 2017 Industry Track colocated with the 15th International Conference on Business Process Management (BPM 2017), Barcelona, Spain, September 10-15, 2017. CEUR Workshop Proceedings, vol. 1985, pp. 52–63. CEUR-WS.org (2017)

[10] Batoulis, K., Weske, M.: Soundness of decision-aware business processes. In: Carmona, J., Engels, G., Kumar, A. (eds.) Business Process Management Forum, Barcelona, Spain, September 10-15, 2017. Lecture Notes in Business Information Processing, vol. 297, pp. 106–124. Springer (2017)

[11] Batoulis, K., Weske, M.: Disambiguation of DMN decision tables. In: Abramowicz, W., Paschke, A. (eds.) Proceedings of the 21st International Conference on Business Information Systems (BIS 2018), Berlin, Germany, July 18-20, 2018. Lecture Notes in Business Information Processing, vol. 320, pp. 236–249. Springer (2018)

[12] Becker, J., Holten, R., Knackstedt, R., Niehaves, B.: Forschungsmethodische positionierung in der wirtschaftsinformatik: epistemologische, ontologische und linguistische leitfragen. Tech. rep., Arbeitsberichte des Instituts für Wirtschaftsinformatik, Westfälische Wilhelms-Universität Münster (2003)

[13] Berstel-Da Silva, B.: Verification of business rules programs. Springer (2014)

[14] Besnard, P.: Inconsistency measuring over multisets of formulas. In: Grant, J., Martinez, M.V. (eds.) Measuring Inconsistency in Information. College Publications (2018)

[15] Blatt, J.: Prototyping a Verification Tool for Decision Model and Notation. Master's thesis, University of Koblenz-Landau (2020), `https://kola.opus.hbz-nrw.de/opus45-kola/frontdoor/index/index/docId/2049`

[16] Bona, G.D., Grant, J., Hunter, A., Konieczny, S.: Towards a unified framework for syntactic inconsistency measures. In: McIlraith, S.A., Weinberger, K.Q. (eds.) Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), New Orleans, USA, February 2-7, 2018. pp. 1803–1810. AAAI Press (2018)

[17] Brewka, G., Thimm, M., Ulbricht, M.: Strong inconsistency. Artificial Intelligence **267**, 78–117 (February 2019)

[18] Burattin, A., Maggi, F.M., van der Aalst, W.M.P., Sperduti, A.: Techniques for a posteriori analysis of declarative processes. In: Chi, C., Gasevic, D., van den Heuvel, W. (eds.) Proceedings of the 16th IEEE International Enterprise Distributed Object Computing Conference (EDOC 2012), Beijing, China, September 10-14, 2012. pp. 41–50. IEEE Computer Society (2012)

[19] Burgstaller, F., Steiner, D., Schrefl, M.: Modeling context for business rule management. In: Kornyshova, E., Poels, G., Huemer, C., Wattiau, I., Matthes, F., Sanz, J.L.C. (eds.) Proceedings of the 18th IEEE Conference on Business Informatics (CBI 2016), Paris, France, August 19 - September 1, 2016. pp. 262–271. IEEE (2016)

[20] Calvanese, D., Dumas, M., Laurson, Ü., Maggi, F.M., Montali, M., Teinemaa, I.: Semantics and analysis of DMN decision tables. In: Rosa, M.L., Loos, P., Pastor, O. (eds.) Proceedings of the 14th International Conference on Business Process Management (BPM 2016), Rio de Janeiro, Brazil, September 18-22, 2016. Lecture Notes in Computer Science, vol. 9850, pp. 217–233. Springer (2016)

[21] Calvanese, D., Dumas, M., Laurson, Ü., Maggi, F.M., Montali, M., Teinemaa, I.: Semantics, analysis and simplification of DMN decision tables. Information Systems **78**, 112–125 (2018)

[22] Calvanese, D., Dumas, M., Maggi, F.M., Montali, M.: Semantic DMN: formalizing decision models with domain knowledge. In: Costantini, S., Franconi, E., Woensel, W.V., Kontchakov, R., Sadri, F., Roman, D. (eds.) Proceedings of the 13th International Conference on Rules and Reasoning (RR 2017), London, UK, July 12-15, 2017. Lecture Notes in Computer Science, vol. 10364, pp. 70–86. Springer (2017)

[23] Cemus, K., Cerný, T., Donahoo, M.J.: Evaluation of approaches to business rules maintenance in enterprise information systems. In: Nadimi, E.S., Cerný, T., Kim, S., Wang, W. (eds.) Proceedings of the 15th International Conference on Research in Adaptive and Convergent Systems (RACS 2015), Prague, Czech Republic, October 9-12, 2015. pp. 324–329. ACM (2015)

[24] Cheng, M., Huang, C.: A novel approach for treating uncertain rule-based knowledge conflicts. Journal of Information Science and Engineering **25**(2), 649–663 (2009)

[25] Chomicki, J., Lobo, J., Naqvi, S.: Conflict resolution using logic programming. IEEE Transactions on Knowledge and Data Engineering **15**(1) (2003)

[26] Ciccio, C.D., Maggi, F.M., Montali, M., Mendling, J.: Ensuring model consistency in declarative process discovery. In: Motahari-Nezhad, H.R.,

Recker, J., Weidlich, M. (eds.) Proceedings of the 13th International Conference on Business Process Management (BPM 2015), Innsbruck, Austria, August 31 - September 3, 2015. Lecture Notes in Computer Science, vol. 9253, pp. 144–159. Springer (2015)

[27] Corea, C., Blatt, J., Delfmann, P.: A tool for decision logic verification in DMN decision tables. In: Depaire, B., Smedt, J.D., Dumas, M., Fahland, D., Kumar, A., Leopold, H., Reichert, M., Rinderle-Ma, S., Schulte, S., Seidel, S., van der Aalst, W.M.P. (eds.) Proceedings of the Dissertation Award, Doctoral Consortium, and Demonstration Track at BPM 2019 co-located with the 17th International Conference on Business Process Management (BPM 2019), Vienna, Austria, September 1-6, 2019. CEUR Workshop Proceedings, vol. 2420, pp. 169–173. CEUR-WS.org (2019)

[28] Corea, C., Deisen, M., Delfmann, P.: Resolving inconsistencies in declarative process models based on culpability measurement. In: Ludwig, T., Pipek, V. (eds.) Proceedings der 14. Internationalen Tagung der Wirtschaftsinformatik (WI 2019), Siegen, Germany, February 24-27, 2019. pp. 139–153. AISeL (2019)

[29] Corea, C., Delfmann, P.: Supporting business rule management with inconsistency analysis. In: van der Aalst, W.M.P., Casati, F., Conforti, R., de Leoni, M., Dumas, M., Kumar, A., Mendling, J., Nepal, S., Pentland, B.T., Weber, B. (eds.) Proceedings of the Dissertation Award, Demonstration, and Industrial Track at BPM 2018 co-located with the 16th International Conference on Business Process Management (BPM 2018), Sydney, Australia, September 9-14, 2018. CEUR Workshop Proceedings, vol. 2196, pp. 141–147. CEUR-WS.org (2018)

[30] Corea, C., Delfmann, P.: A tool to monitor consistent decision-making in business process execution. In: van der Aalst, W.M.P., Casati, F., Conforti, R., de Leoni, M., Dumas, M., Kumar, A., Mendling, J., Nepal, S., Pentland, B.T., Weber, B. (eds.) Proceedings of the Dissertation Award, Demonstration, and Industrial Track at BPM 2018 co-located with the 16th International Conference on Business Process Management (BPM 2018), Sydney, Australia, September 9-14, 2018. CEUR Workshop Proceedings, vol. 2196, pp. 76–80. CEUR-WS.org (2018)

[31] Corea, C., Delfmann, P.: Quasi-inconsistency in declarative process models. In: Hildebrandt, T.T., van Dongen, B.F., Röglinger, M., Mendling, J. (eds.) Business Process Management Forum, Vienna, Austria, September 1-6, 2019. Lecture Notes in Business Information Processing, vol. 360, pp. 20–35. Springer (2019)

[32] Corea, C., Delfmann, P.: A taxonomy for busines rule organizing approaches in regard to business process compliance. Journal of Conceptual Modeling (EMISAJ) (2020)

[33] Corea, C., Nagel, S., Delfmann, P.: Effects of visualization techniques on understanding inconsistencies in automated decision-making. In: Bui, T. (ed.) Proceedings of the 53rd Hawaii International Conference on System Sciences (HICSS 2020), Maui, USA, January 7-10, 2020. pp. 198–207. ScholarSpace (2020)

[34] Corea, C., Thimm, M.: On quasi-inconsistency and its complexity. Artificial Intelligence (2020)

[35] Corea, C., Thimm, M.: Towards inconsistency measurement in business rule bases. In: Proceedings of the 24th European Conference on Artificial Intelligence (ECAI 2020), Santiago de Compostela, Spain, 2020 (2020)

[36] Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to algorithms. MIT press (2009)

[37] Cuzzocrea, A., Decker, H., Muñoz-Escoí, F.D.: Scalable uncertainty-tolerant business rules. In: Polycarpou, M.M., de Leon Ferreira de Carvalho, A.C.P., Pan, J., Wozniak, M., Quintián, H., Corchado, E. (eds.) Proceedings of the 9th International Conference on Hybrid Artificial Intelligence Systems (HAIS 2014), Salamanca, Spain, June 11-13, 2014. Lecture Notes in Computer Science, vol. 8480, pp. 179–190. Springer (2014)

[38] De Bona, G., Hunter, A.: Localising iceberg inconsistencies. Artificial Intelligence **246**, 118–151 (2017)

[39] Decker, H., Muñoz-Escoí, F.D.: Inconsistency-tolerant business rules in distributed information systems. In: Demey, Y.T., Panetto, H. (eds.) Proceedings of the Confederated International Workshops On the Move to Meaningful Internet Systems (OTM 2013), Graz, Austria, September 9-13, 2013, Proceedings. Lecture Notes in Computer Science, vol. 8186, pp. 322–331. Springer (2013)

[40] Deisen, M.: Quasi-Inconsistency Library for Business Rule Management. Master's thesis, University of Koblenz-Landau (2019), `https://kola.opus.hbz-nrw.de/frontdoor/index/index/docId/1813`

[41] Delfmann, P., Steinhorst, M., Dietrich, H.A., Becker, J.: The generic model query language gmql–conceptual specification, implementation, and runtime evaluation. Information Systems **47**, 129–177 (2015)

[42] Di Ciccio, C., Maggi, F.M., Montali, M., Mendling, J.: Resolving inconsistencies and redundancies in declarative process models. Information Systems **64**, 425–446 (2017)

[43] Dumas, M., La Rosa, M., Mendling, J., Reijers, H.A.: Fundamentals of business process management. Springer (2013)

[44] El Kharbili, M.: Business process regulatory compliance management solution frameworks: A comparative evaluation. In: Ghose, A., Ferrarotti, F. (eds.) Proceedings of the 8th Asia-Pacic Conference on Conceptual Modelling (APCCM) 2012, Melbourne, Australia, January 2012. CRPIT, vol. 130, pp. 23–32. Australian Computer Society (2012)

[45] Ezekiel, K., Vassilev, V.T., Ouazzane, K., Patel, Y.: Adaptive business rules framework for workflow management. Business Process Management Journal **25**(5), 948–971 (2019)

[46] Fu, G., Shao, J., Embury, S.M., Gray, W.A.: Algorithms for analysing related constraint business rules. Data and Knowledge Engineering **50**(2), 215–240 (2004)

[47] Gadi, T.: Methods of detection and resolution of anomalies in business rule: Comparative study. International Journal of Computer Applications **975**, 88–94 (2015)

[48] Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., Bouchachia, A.: A survey on concept drift adaptation. ACM computing surveys (CSUR) **46**(4), 1–37 (2014)

[49] Gamut, L.: Logic, language, and meaning, volume 1: Introduction to logic, vol. 1. University of Chicago Press (1991)

[50] Gelfond, M., Lifschitz, V.: Classical negation in logic programs and disjunctive databases. New Generation Computing **9**, 365–385 (1991)

[51] Gelfond, M., Leone, N.: Logic programming and knowledge representation - The A-Prolog perspective. Artificial Intelligence **138**(1-2), 3–38 (2002)

[52] Gericke, A., Robert, W.: Entwicklung eines bezugsrahmens fuer konstruktionsforschung und artefaktkonstruktion in der gestaltungsorientierten wirtschaftsinformatik. In: Wissenschaftstheorie und gestaltungsorientierte Wirtschaftsinformatik, pp. 195–210. Springer (2009)

[53] Gómez-López, M.T., Gasca, R.M., Pérez-Álvarez, J.M.: Compliance validation and diagnosis of business data constraints in business processes at runtime. Information Systems **48** (2015)

[54] Governatori, G.: Representing business contracts in ruleml. International Journal of Cooperative Information Systems **14**, 181–216 (2005)

[55] Governatori, G., Maher, M.J.: Annotated defeasible logic. Theory and Practice of Logic Programming **17**(5-6), 819–836 (2017)

[56] Governatori, G., Milosevic, Z.: A formal analysis of a business contract language. International Journal of Cooperative Information Systems **15**(04), 659–685 (2006)

[57] Governatori, G., Olivieri, F., Scannapieco, S., Cristani, M.: Designing for compliance: Norms and goals. In: Olken, F., Palmirani, M., Sottara, D. (eds.) Proceedings of the 5th International Symposium on Rule-Based Modeling and Computing on the Semantic Web (RuleML 2011), Ft. Lauderdale, USA, November 3-5, 2011. Lecture Notes in Computer Science, vol. 7018, pp. 282–297. Springer (2011)

[58] Graham, I.: Business rules management and service oriented architecture: a pattern language. John wiley & sons (2007)

[59] Grant, J., Hunter, A.: Measuring consistency gain and information loss in stepwise inconsistency resolution. In: Liu, W. (ed.) Proceedings of the 11th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU 2011), Belfast, UK, June 29-July 1, 2011. Lecture Notes in Computer Science, vol. 6717, pp. 362–373. Springer (2011)

[60] Grant, J., Hunter, A.: Analysing inconsistent information using distance-based measures. International Journal of Approximate Reasoning **89**, 3–26 (2017)

[61] Grant, J., Martinez, M.V. (eds.): Measuring Inconsistency in Information. College Publications (2018)

[62] Grant, J., Parisi, F.: Measuring inconsistency in a general information space. In: Herzig, A., Kontinen, J. (eds.) Proceedings of the 11th International Symposium on Foundations of Information and Knowledge Systems (FoIKS 2020), Dortmund, Germany, February 17-21, 2020. Lecture Notes in Computer Science, vol. 12012, pp. 140–156. Springer (2020)

[63] Hashmi, M., Governatori, G., Lam, H., Wynn, M.T.: Are we done with business process compliance: State of the art and challenges ahead. Knowledge and Information Systems **57**(1), 79–133 (2018)

[64] Hasic, F., Corea, C., Blatt, J., Delfmann, P., Serral, E.: Decision model change patterns for dynamic system evolution. Knowledge and Information Systems (2020)

[65] Hevner, A.R., March, S.T., Park, J., Ram, S.: Design science in information systems research. MIS quarterly pp. 75–105 (2004)

[66] Hicks, R.C.: The no inference engine theory - performing conflict resolution during development. Decision Support Systems **43**(2) (2007)

[67] Houari, N.S., Taghezout, N.: A novel collaborative approach for business rules consistency management. In: Liu, S., Delibasic, B., Oderanti, F.O. (eds.) Proceedings of the 2nd International Conference on Decision Support Systems (ICDSST 2016), Plymouth, UK, May 23-25, 2016. Lecture Notes in Business Information Processing, vol. 250, pp. 152–164. Springer (2016)

[68] Hunter, A., Konieczny, S.: Shapley inconsistency values. In: Doherty, P., Mylopoulos, J., Welty, C.A. (eds.) Proceedings of the 10th International Conference on Principles of Knowledge Representation and Reasoning, Lake District, UK, June 2-5, 2006. pp. 249–259. AAAI Press (2006)

[69] Hunter, A., Konieczny, S.: On the measure of conflicts: Shapley inconsistency values. Artificial Intelligence **174**(14), 1007–1026 (2010)

[70] Hunter, A., Konieczny, S., et al.: Measuring inconsistency through minimal inconsistent sets. KR **8**, 358–366 (2008)

[71] Imgrund, F., Malorny, M., Janiesch, C.: Eine literaturanalyse zur integration von business rules und business process management. In: Leimeister, J.M., Brenner, W. (eds.) Proceedings der 13. Internationalen Tagung der Wirtschaftsinformatik (WI 2017), St.Gallen, Switzerland, February 12-15, 2017. pp. 211–225. AISel (2017)

[72] Janssens, L., Bazhenova, E., Smedt, J.D., Vanthienen, J., Denecker, M.: Consistent integration of decision (DMN) and process (BPMN) models. In: España, S., Ivanovic, M., Savic, M. (eds.) Proceedings of the CAiSE 2016 Forum co-located with the 28th International Conference on Advanced Information Systems Engineering (CAiSE 2016), Ljubljana, Slovenia, June 13-17, 2016. CEUR Workshop Proceedings, vol. 1612, pp. 121–128. CEUR-WS.org (2016)

[73] Karami, N., Iijima, J.: A dynamic knowledge approach for dynamic business rules modeling pp. 96–102 (2006)

[74] Kardasis, P., Loucopoulos, P.: Expressing and organising business rules. Information and Software Technology **46**(11), 701–718 (2004)

[75] Kuechler, B., Vaishnavi, V.: On theory development in design science research: anatomy of a research project. European Journal of Information Systems **17**(5), 489–504 (2008)

[76] Laudon, K.C., Laudon, J.P., Schoder, D.: Wirtschaftsinformatik: Eine Einführung. Pearson Deutschland GmbH (2010)

[77] Lévy, F., Guissé, A., Nazarenko, A., Omrane, N., Szulman, S.: An environment for the joint management of written policies and business rules. In: Proceeding sof the 22nd IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2010), Arras, France, October 27-29, 2010. pp. 142–149. IEEE Computer Society (2010)

[78] Lin, B., Zhang, Y.: The management and validation of business rules in telecom operators settlement and apportion system. In: Proceedings of the 8th International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (ACIS 2007). IEEE (2007)

[79] Lin, L., Embury, S.M., Warboys, B.C.: Facilitating the implementation and evolution of business rules. In: Proceedings of the 21st IEEE International Conference on Software Maintenance (ICSM'05). IEEE (2005)

[80] van der Linden, E., Smit, K., Berkhout, M., Zoet, M.: The business rule type jungle: An explorative analysis. In: Pucihar, A., Borstnar, M.K., Bons, R.W.H., Seitz, J., Cripps, H., Vidmar, D. (eds.) Proceedings of the 32nd Bled eConference: Humanizing Technology for a Sustainable Society, Bled, Slovenia, June 16-19, 2019. pp. 24–36. University of Maribor Press / Association for Information Systems (2019)

[81] Lu, R., Sadiq, S., Governatori, G.: Measurement of compliance distance in business processes. Information Systems Management **25**(4), 344–355 (2008)

[82] Ly, L.T., Maggi, F.M., Montali, M., Rinderle-Ma, S., van der Aalst, W.M.P.: A framework for the systematic comparison and evaluation of compliance monitoring approaches. In: Gasevic, D., Hatala, M., Nezhad, H.R.M., Reichert, M. (eds.) Proceedings of the 17th IEEE International Enterprise Distributed Object Computing Conference (EDOC 2013), Vancouver, Canada, September 9-13, 2013. pp. 7–16. IEEE Computer Society (2013)

[83] Maggi, F.M., Bose, R.P.J.C., van der Aalst, W.M.P.: A knowledge-based integrated approach for discovering and repairing declare maps. In: Salinesi, C., Norrie, M.C., Pastor, O. (eds.) Proceedings of the 25th International Conference on Advanced Information Systems Engineering (CAiSE 2013), Valencia, Spain, June 17-21, 2013. Lecture Notes in Computer Science, vol. 7908, pp. 433–448. Springer (2013)

[84] Maggi, F.M., Montali, M., Westergaard, M., van der Aalst, W.M.P.: Monitoring business constraints with linear temporal logic: An approach based on colored automata. In: Rinderle-Ma, S., Toumani, F., Wolf, K. (eds.) Proceedings of the 9th International Conference on Business Process Management (BPM 2011), Clermont-Ferrand, France, August 30 - September 2, 2011. Lecture Notes in Computer Science, vol. 6896, pp. 132–147. Springer (2011)

[85] Maggi, F.M., Westergaard, M., Montali, M., van der Aalst, W.M.P.: Runtime verification of ltl-based declarative process models. In: Khurshid, S., Sen, K. (eds.) Proceedings of the 2nd International Conference on Runtime Verification (RV 2011), San Francisco, USA, September 27-30, 2011. Lecture Notes in Computer Science, vol. 7186, pp. 131–146. Springer (2011)

[86] March, S.T., Smith, G.F.: Design and natural science research on information technology. Decision support systems **15**(4), 251–266 (1995)

[87] McAreavey, K., Liu, W., Miller, P.: Computational approaches to finding and measuring inconsistency in arbitrary knowledge bases. International Journal of Approximate Reasoning **55**(8), 1659–1693 (2014)

[88] Meghanathan, R.N., van Leeuwen, C., Nikolaev, A.R.: Fixation duration surpasses pupil size as a measure of memory load in free viewing. Frontiers in human neuroscience **8**, 1063 (2015)

[89] Nagel, S., Corea, C., Delfmann, P.: Effects of quantitative measures on understanding inconsistencies in business rules. In: Bui, T. (ed.) Proceedings of the 52nd Hawaii International Conference on System Sciences (HICSS 2019), Maui, USA, January 8-11, 2019. pp. 146–155. ScholarSpace (2019)

[90] Namiri, K., Stojanovic, N.: Towards A formal framework for business process compliance. In: Bichler, M., Hess, T., Krcmar, H., Lechner, U., Matthes, F., Picot, A., Speitkamp, B., Wolf, P. (eds.) Proceedings der Multikonferenz Wirtschaftsinformatik (MKWI 2008), München, Germany, February 26-28, 2008. GITO-Verlag (2008)

[91] Nelson, M.L., Rariden, R.L., Sen, R.: A lifecycle approach towards business rules management. In: Sprague, R. (ed.) Proceedings of the 41st Hawaii International International Conference on Systems Science (HICSS 2008), Waikoloa, USA, January 7-10, 2008. pp. 113–121. IEEE Computer Society (2008)

[92] Nelson, M.L., Sen, R.: Business rules management in healthcare: A lifecycle approach. Decision Support Systems **57**, 387–394 (2014)

[93] Papadimitriou, C.: Computational Complexity. Addison-Wesley (1994)

[94] Park, B., Flowerday, T., Brünken, R.: Cognitive and affective effects of seductive details in multimedia learning. Computers in Human Behavior **44**, 267–278 (2015)

[95] Peffers, K., Tuunanen, T., Rothenberger, M.A., Chatterjee, S.: A design science research methodology for information systems research. Journal of management information systems **24**(3), 45–77 (2007)

[96] Polyvyanyy, A.: Process Querying Methods. Springer (2020)

[97] Potyka, N., Thimm, M.: Probabilistic reasoning with inconsistent beliefs using inconsistency measures. In: Proceedings of the 24th International Joint Conference on Artificial Intelligence (2015)

[98] Priest, G.: The logic of paradox. Journal of Philosophical logic pp. 219–241 (1979)

[99] Ramberger, H.H.: Converting Decision Model and Notation Decision Tables to Formal Contract Language. Master's thesis, University of Koblenz-Landau (2018)

[100] Rayner, K.: Eye movements in reading and information processing: 20 years of research. Psychological bulletin **124**(3), 372 (1998)

[101] Reijers, H.A., Mendling, J., Dijkman, R.M.: Human and automatic modularizations of process models to enhance their comprehension. Information Systems **36**(5), 881–897 (2011)

[102] Ross, R.G.: Principles of the business rule approach. Addison-Wesley Professional (2003)

[103] Royce, G.: Integration of a business rules engine to manage frequently changing workflow: A case study of insurance underwriting workflow p. 495 (2007)

[104] Saake, G., Sattler, K.: Algorithmen und Datenstrukturen (in Java). dpunkt (2012)

[105] Sackmann, S., Kühnel, S., Seyffarth, T.: Using business process compliance approaches for compliance management with regard to digitization: Evidence from a systematic literature review. In: Weske, M., Montali, M., Weber, I., vom Brocke, J. (eds.) Proceedings of the 16th International Conference on Business Process Management (BPM 2018), Sydney, Australia, September 9-14, 2018. Lecture Notes in Computer Science, vol. 11080, pp. 409–425. Springer (2018)

[106] Sadiq, S., Governatori, G.: Managing regulatory compliance in business processes. In: Handbook on Business Process Management 2, pp. 159–175. Springer (2015)

[107] Sadiq, S., Governatori, G., Namiri, K.: Modeling control objectives for business process compliance. In: Alonso, G., Dadam, P., Rosemann, M. (eds.) Proceedings of the 5th International Conference on Business Process Management (BPM 2007), Brisbane, Australia, September 24-28, 2007. Lecture Notes in Computer Science, vol. 4714, pp. 149–164. Springer (2007)

[108] Sadiq, S.W.: A roadmap for research in business process compliance. In: Abramowicz, W., Maciaszek, L.A., Wecel, K. (eds.) Proceedings of the Business Information Systems Workshops, Poznań, Poland, June 15-17, 2011. Lecture Notes in Business Information Processing, vol. 97, pp. 1–4. Springer (2011)

[109] dos Santos Guimarães, D., Lima, P., Correa, A.L., Schmitz, E.A., Alencar, A.J.: A method for verifying the consistency of business rules using alloy. In: Reformata, M. (ed.) Proceedings of the 26th International Conference on Software Engineering and Knowledge Engineering (SEKE 2013), Vancouver, Canada, July 1-3, 2013. pp. 381–386. Knowledge Systems Institute Graduate School (2014)

[110] Saralaya, S., Saralaya, V., D'Souza, R.: Compliance management in business processes. In: Digital Business, pp. 53–91. Springer (2019)

[111] Schlosser, S., Baghi, E., Otto, B., Österle, H.: Toward a functional reference model for business rules management. In: Sprague, R. (ed.) Proceedings of the 47th Hawaii International Conference on System Sciences (HICSS 2014), Waikoloa, USA, January 6-9, 2014. pp. 3837–3846. IEEE Computer Society (2014)

[112] Shapley, L.S.: A value for n-person games. Contributions to the Theory of Games **2**(28), 307–317 (1953)

[113] da Silva, C.F., Baptista, L., da Cunha, P.R., Melo, P.E.: Empowering end-users to manage business rules: The case of a graphical environment built for a telco. In: Stjepandic, J., Rock, G., Bil, C. (eds.) Proceedings of the 19th International Conference on Concurrent Engineering (ISPE 2012), Trier, Germany, September 3-7, 2012, pp. 1193–1204. Springer (2012)

[114] Smedt, J.D., Weerdt, J.D., Serral, E., Vanthienen, J.: Improving understandability of declarative process models by revealing hidden dependencies. In: Nurcan, S., Soffer, P., Bajec, M., Eder, J. (eds.) Proceedings of the 28th International Conference on Advanced Information Systems Engineering (CAiSE 2016), Ljubljana, Slovenia, June 13-17, 2016. Lecture Notes in Computer Science, vol. 9694, pp. 83–98. Springer (2016)

[115] Smedt, J.D., Weerdt, J.D., Serral, E., Vanthienen, J.: Discovering hidden dependencies in constraint-based declarative process models for improving understandability. Information Systems **74**, 40–52 (2018)

[116] Smirnov, S., Reijers, H.A., Weske, M., Nugteren, T.: Business process model abstraction: a definition, catalog, and survey. Distributed and Parallel Databases **30**(1), 63–99 (2012)

[117] Smit, K., Versendaal, J., Zoet, M.: Identifying challenges in BRM implementations regarding the verification and validation capabilities at governmental institutions. In: Alias, R.A., Ling, P.S., Bahri, S., Finnegan, P., Sia, C.L. (eds.) Proceedings of the 21st Pacific Asia Conference on Information Systems (PACIS 2017), Langkawi, Malaysia, July 16-20, 2017. p. 201 (2017)

[118] Smit, K., Zoet, M.: A business rules management reference process for the dutch government. In: Alias, R.A., Ling, P.S., Bahri, S., Finnegan, P., Sia, C.L. (eds.) Proceedings of the 21st Pacific Asia Conference on Information Systems (PACIS 2017), Langkawi, Malaysia, July 16-20, 2017. p. 199 (2017)

[119] Smit, K., Zoet, M.: Applying the decision model and notation in practice: A method to design and specify business decisions and business logic. In: Proceedings of the Mediterranean Conference on Information Systems (2018)

[120] Smit, K., Zoet, M., Berkhout, M.: Verification capabilities for business rules management in the dutch governmental context. In: Proceedings of the International Conference on Research and Innovation in Information Systems (ICRIIS 2017). IEEE (2017)

[121] Smit, K., Zoet, M., Versendaal, J.: Identifying challenges in business rules management implementations regarding the elicitation, design, and specification capabilities at dutch governmental institutions. Journal of Information Technology Theory and Application (JITTA) **19**(2) (2018)

[122] Taylor, J.: Achieving decision consistency across the soa-based enterprise using business rules management systems. In: Proceedings of the International Conference on Web Information Systems Engineering. Springer (2005)

[123] Thalmann, M., Souza, A.S., Oberauer, K.: How does chunking help working memory? Journal of Experimental Psychology: Learning, Memory, and Cognition **45**(1), 37 (2019)

[124] Thimm, M.: On the expressivity of inconsistency measures. Artificial Intelligence **234**, 120–151 (2016)

[125] Thimm, M.: Stream-based inconsistency measurement. International Journal of Approximate Reasoning **68**, 68–87 (2016)

[126] Thimm, M.: Measuring inconsistency with many-valued logics. International Journal of Approximate Reasoning **86**, 1–23 (July 2017)

[127] Thimm, M.: On the compliance of rationality postulates for inconsistency measures: A more or less complete picture. KI-Künstliche Intelligenz **31**(1), 31–39 (2017)

[128] Thimm, M.: The tweety library collection for logical aspects of artificial intelligence and knowledge representation. Künstliche Intelligenz **31**(1), 93–97 (2017)

[129] Thimm, M.: On the evaluation of inconsistency measures. In: Grant, J., Martinez, M.V. (eds.) Measuring Inconsistency in Information. College Publications (2018)

[130] Thimm, M.: Inconsistency measurement. In: Amor, N.B., Quost, B., Theobald, M. (eds.) Proceedings of the 13th International Conference on Scalable Uncertainty Management (SUM 2019), Compiègne, France, December 16-18, 2019. Lecture Notes in Computer Science, vol. 11940, pp. 9–23. Springer (2019)

[131] Thimm, M., Wallner, J.P.: On the complexity of inconsistency measurement. Artificial Intelligence **275**, 411–456 (October 2019)

[132] Thimm, M., Wallner, J.P.: Some complexity results on inconsistency measurement. In: Baral, C., Delgrande, J.P., Wolter, F. (eds.) Proceedings of the 15th International Conference on Principles of Knowledge Representation and Reasoning (KR 2016), Cape Town, South Africa, April 25-29, 2016. pp. 114–124. AAAI Press (2016)

[133] Ulbricht, M., Thimm, M., Brewka, G.: Measuring strong inconsistency. In: Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI'18) (2018)

[134] Vaishnavi, V., Kuechler, W., Petter, S.: Design science research in information systems. January **20**, 2004 (2004)

[135] Vanthienen, J., Mues, C., Wets, G., Delaere, K.: A tool-supported approach to inter-tabular verification. Expert systems with applications **15**(3-4), 277–285 (1998)

[136] Wang, W., Indulska, M., Sadiq, S.W.: Cognitive efforts in using integrated models of business processes and rules. In: España, S., Ivanovic, M., Savic, M. (eds.) Proceedings of the CAiSE'16 Forum at the 28th International Conference on Advanced Information Systems Engineering (CAiSE 2016), Ljubljana, Slovenia, June 13-17, 2016. CEUR Workshop Proceedings, vol. 1612, pp. 33–40. CEUR-WS.org (2016)

[137] Wang, W., Indulska, M., Sadiq, S.W., Weber, B.: Effect of linked rules on business process model understanding. In: Carmona, J., Engels, G., Kumar, A. (eds.) Proceedings of the 15th International Conference on Business Process Management (BPM 2017), Barcelona, Spain, September 10-15, 2017. Lecture Notes in Computer Science, vol. 10445, pp. 200–215. Springer (2017)

[138] Weiden, M., Hermans, L., Schreiber, G., van der Zee, S.: Classification and representation of business rules. In: Schacher, M. (ed.) Proceedings of the 1st European Business Rules Conference, Zürich, Switzerland, June 26-27, 2002. pp. 14–28. OTS (2002)

[139] Weidlich, M., Mendling, J., Weske, M.: Efficient consistency measurement based on behavioral profiles of process models. IEEE Transactions on Software Engineering **37**(3), 410–429 (2010)

[140] Weske, M.: Business process management architectures. In: Business Process Management, pp. 333–371. Springer (2012)

[141] Winter, R.: Design science research in europe. European Journal of Information Systems **17**(5), 470–475 (2008)

[142] Xiao, G., Ma, Y.: Inconsistency measurement based on variables in minimal unsatisfiable subsets. In: Proceedings of the European Conference on Artificial Intelligence 2012 (ECAI'12) (2012)

[143] Yeshchenko, A., Ciccio, C.D., Mendling, J., Polyvyanyy, A.: Comprehensive process drift detection with visual analytics. In: Laender, A.H.F., Pernici, B., Lim, E., de Oliveira, J.P.M. (eds.) Proceedings of the 38th International Conference on Conceptual Modeling (ER 2019), Salvador, Brazil, November 4-7, 2019. Lecture Notes in Computer Science, vol. 11788, pp. 119–135. Springer (2019)

[144] Zhang, Q., Zeng, G., Xiao, C., Yue, Y.: A rule conflict resolution method based on vague set. Soft Computing **18**(3) (2014)

List of Figures

List of Tables

6

Overview of Publications

In regard to the dissertation regulations of the faculty of computer science at the University of Koblenz-Landau ("Promotionsordnung", in the following *PO*), this thesis is a cumulative dissertation and relates to the publications shown in Table 6.1. The author percentage quantifies the contribution by the author and is derived from the explanation provided in the Disclaimer.

ID	**Title**	**Outlet**	**Ranking**	**Adjusted Ranking**	**Author %**
[35]	Towards Inconsistency Measurement in Business Rule Bases	ECAI 2020	A	A	0.8
[33]	Effects of Visualization Techniques on Understanding Inconsistencies in Automated Decision-Making	HICSS 2020	A	A	0.5
[27]	A Tool for Decision-Logic Level Verification in DMN Decision Tables	BPM 2019 (Demo-Track)	A	A	0.8
[31]	Quasi-Inconsistency in Declarative Process Models	BPM 2019 (Forum)	A	B[1]	0.9
[28]	Resolving Inconsistencies in Declarative Process Models based on Culpability Measurement (**Winner of the Best-Paper Award**)	WI 2019	C	C	0.8
[89]	Effects of Quantitative Measures on Understanding Inconsistencies in Business Rules	HICSS 2019	A	A	0.4
[29]	Supporting Business Rule Management with Inconsistency Analysis	BPM 2018 (Industry-Track)	A	B[2]	0.9
[30]	A Tool to Monitor Consistent Decision-Making in Business Process Execution	BPM 2018 (Demo-Track)	A	A	0.9
[32]	A Taxonomy of Business Rule Organizing Approaches	EMISA (**Journal**)	C	C	0.9
[34]	On Quasi-Inconsistency and its Complexity	AIJ (**Journal**)	A*	A*	0.6
[64]	Decision Model Change Patterns for Dynamic System Evolution	KAIS (**Journal**)	B	B	0.2

Table 6.1: Overview of the published works corresponding to this thesis.

---

[1] As the BPM Forum is a workshop-like format, the ranking of this work was adjusted to B in accordance with the supervisors of this thesis.

[2] As the paper-length in the industry track is half of the other research tracks, the ranking of this work was adjusted to B in accordance with the supervisors of this thesis.

Part I of this thesis presented an overview of this thesis, including problem statement, research questions, research approach, results, evaluation and conclusion. Figure 6.1 furthermore details the specific interrelations of the publications corresponding to this thesis, in the scope of the thesis contributions (cf. Section 1.4).

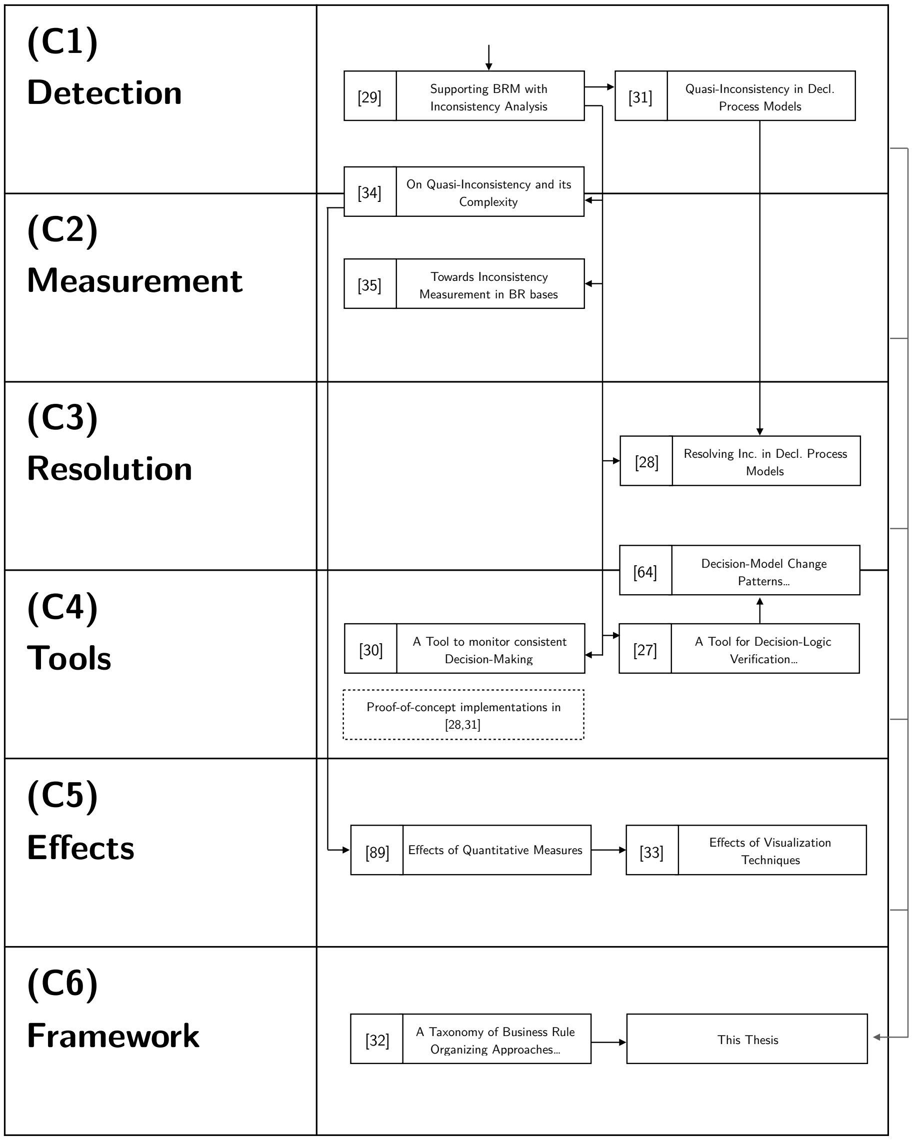


Figure 6.1: Interrelation of the publications relating to this work.

The starting point was an industry-track paper [29], which drafted the idea for applying inconsistency measurement to the domain of business rules management. From this paper, obstacles impeding a straightforward applications were identified, leading to the development of means of detecting and analyzing inconsistency at design-time [31, 34], and run-time [35]. The culpability ranking proposed in [29] was also used to develop means for inconsistency resolution based on culpability measurement [28] (using the culpability measures proposed in [31]). These methods and techniques were implemented and evaluated in several works [28, 31]. Also, user-friendly tools/frontends utilizing the developed means were presented in [27, 30]. The tool in [27] was later extended with further resolution functionalities in [64]. In parallel, the plausibility of applying quantitative measures in a business domain was evaluated in [89]. After some promising results regarding the usage of quantitative measures in general, the experiments were extended to evaluate the effects of the specific visualization techniques of such measures (e.g. in a ranking-based overview) in [33]. The research in general was guided by the literature review in [32]. The resulting taxonomy classified all other presented results in the context of business rule organizing, resulting in this thesis.

7

Published Contributions

The published works are now presented in full.

7.1 Towards Inconsistency Measurement in Business Rule Bases

Source	Towards Inconsistency Measurement in Business Rule Bases
Authors	Carl Corea and Matthias Thimm
Outlet	ECAI 2020 - 24th European Conference on Artificial Intelligence, June 10-12, 2020, Santiago the Compostela, Spain
Ranking	A
Status	Published

Towards Inconsistency Measurement in Business Rule Bases

Carl Corea[1] and **Matthias Thimm**[2]

Abstract. We investigate the application of inconsistency measures to the problem of analysing business rule bases. Due to some intricacies of the domain of business rule bases, a straightforward application is not feasible. We therefore develop some new rationality postulates for this setting as well as adapt and modify existing inconsistency measures. We further adapt the notion of *inconsistency values* (or *culpability measures*) for this setting and give a comprehensive feasibility study.

1 Introduction

Business rules have gained major attention in the context of business process management and compliance management [16, 21, 10]. Here, business rules are used to encode policies and laws as a declarative business logic, aimed to ensure that company activities comply with such regulatory controls. For example, the company should only conduct activities as constrained by the set of business rules. Otherwise, the behavior might violate legal regulations, which could result in sensitive fines, or criminal prosecution. Using business rules to verify the compliance of company activities comes with increased demands on the quality of the business rules themselves. However, as company rule bases are usually maintained by multiple modelers, and in an incremental manner, modeling errors can occur frequently [16, 1, 17]. For instance, a recent case study with a large insurance company revealed that 27% of analysed rules contained modeling errors [1]. Hence, the maintenance of business rule bases is recognized as an important challenge for companies [16, 4, 17, 14] .

A potential problem here is that of *inconsistency*, i. e., rules that contradict each other. For example, consider the following business rule base $\mathcal{B}_1$ (we will formalize syntax and semantics later)

$$\begin{aligned}\mathcal{B}_1 = \{&\mathit{platinumCustomer}, \mathit{mentalCondition},\\ &\mathit{platinumCustomer} \rightarrow \mathit{creditWorthy},\\ &\mathit{mentalCondition} \rightarrow \neg\mathit{creditWorthy} \quad \}\end{aligned}$$

with the intuitive meaning that we have a (platinum) customer who has a mental condition and two general rules stating that 1.) platinum customers are credit worthy, and 2.) a customer with a mental condition is not credit worthy. $\mathcal{B}_1$ is inconsistent in the classic logical sense, as it entails the contradictory conclusions $creditWorthy$ and $\neg creditWorthy$. Therefore, this rule base cannot be used to draw meaningful conclusions or to correctly regulate process execution.

To counteract such problems, companies need to be supported with means for the detection and analysis of inconsistencies in business rules, such that experts can resolve inconsistencies. The field of *inconsistency measurement* [9, 20] is about analysing inconsistency in logic-based knowledge representations and therefore represents a good candidate for this use-case. In general, an inconsistency measure $\mathcal{I}$ is a function that assigns a non-negative real value to knowledge base $\mathcal{K}$, quantifying the inconsistency in $\mathcal{K}$ with the informal meaning that a higher value reflects a higher degree of inconsistency.

Applying existing inconsistency measures to business rule bases seems straightforward, however, we can identify a conceptional mismatch. In the classical setting of inconsistency measurement—that of propositional logic—, knowledge bases are constituted of propositional formulas, where these formulas do not have a distinguishable level of granularity. On the other hand, business rule bases distinguish between *facts* and *rules*. That is, facts have a different conceptual quality as their veracity is unconditionally assumed [7].[3] However, assuming facts as indisputable has strong implications for applying results from inconsistency measurement to this use-case. For example, reconsider the above rule base $\mathcal{B}_1$. As mentioned, this rule base is inconsistent, but we can see that by removing the fact $mentalCondition$ it becomes consistent. However, the facts $mentalCondition$, and $platinumCustomer$ are provided by a given case input and have to be kept as-is, even in the scope of inconsistency handling. For instance, one cannot change the mental condition of a customer just to make the set of business rules consistent. Consequently, methods are needed to analyze inconsistency based on a distinction between facts and rules, such that companies can be supported in re-modeling the business rules.

In this work, we develop means for this use-case as follows:

1. We first investigate inconsistency measures for business rule bases in Section 3. To this aim, we propose new postulates that specify expected behavior of inconsistency measures in the business rule base use-case. We show that existing means do not satisfy these requirements and consequently propose new adaptations.
2. Then, in Section 4, we investigate element-based inconsistency measures, which are useful to pin-point problematic elements in the context of inconsistency handling. Again, we show that existing means are not suitable for our use-case and propose adaptations for a plausible application.

Preliminaries are presented in Section 2. Also, we provide an application example for our proposed means in Section 5, and conclude in Section 6. An extended version of this paper with proofs of technical results can be found online[4].

---

[1] University of Koblenz-Landau, Germany, `ccorea@uni-koblenz.de`

[2] University of Koblenz-Landau, Germany, `thimm@uni-koblenz.de`

[3] We acknowledge there could be contradictions between facts due to data errors, however, in this work, we assume a consistent set of non-negotiable facts is evaluated against a humanly modelled rule set.

[4] `https://arxiv.org/abs/1911.08872`

$$\mathcal{I}_d(\mathcal{K}) = \begin{cases} 1 & \text{if } \mathcal{K} \models \perp \\ 0 & \text{otherwise} \end{cases}$$

$$\mathcal{I}_{\mathsf{MI}}(\mathcal{K}) = |\mathsf{MI}(\mathcal{K})|$$

$$\mathcal{I}_p(\mathcal{K}) = |\bigcup_{M \in \mathsf{MI}(\mathcal{K})} M|$$

$$\mathcal{I}_c(\mathcal{K}) = \min\{|v^{-1}(b) \cap \mathcal{A}| \mid v \models^3 \mathcal{K}\}$$

Figure 1. Definitions of the considered inconsistency measures.

2 Preliminaries

To formalise business rule bases, we rely on a simple (monotonic) logic programming language, cf. [6]. For that, we consider a finite set $\mathcal{A}$ of atoms. Let $\mathcal{L}$ be the corresponding set of literals, i. e., atoms and negations of atoms. We abbreviate $\overline{\neg a} = a$ and $\overline{a} = \neg a$ for an atom a. A *(business) rule base* $\mathcal{B}$ is then a set of rules of the form

$$r : \quad l_1, \ldots, l_m \to l_0. \tag{1}$$

with $l_0, \ldots, l_m \in \mathcal{L}$. Let $\mathbb{B}$ be the set of all such rule bases. We abbreviate $head(r) = l_0$ and $body(r) = \{l_1, \ldots, l_m\}$. If $body(r) = \emptyset$ we call r a *fact* and simply write l_0 instead of $\to l_0$. For a rule base $\mathcal{B}$ let $\mathcal{F}(\mathcal{B}) \subseteq \mathcal{B}$ denote the facts in $\mathcal{B}$ and $\mathcal{R}(\mathcal{B}) \subseteq \mathcal{B}$ denote the rules in $\mathcal{B}$.

Example 1. We recall $\mathcal{B}_1$ from Section 1. Then we have

$$\begin{aligned} \mathcal{F}(\mathcal{B}_1) &= \{mentalCondition, platinumCustomer\} \\ \mathcal{R}(\mathcal{B}_1) &= \{platinumCustomer \to creditWorthy, \\ &\qquad mentalCondition \to \neg creditWorthy\}. \end{aligned}$$

A set M of literals is *closed* wrt. $\mathcal{B}$ if for every rule of the form 1, if $l_1, \ldots, l_m \in M$ then $l_0 \in M$. The *minimal model* M of a rule base $\mathcal{B}$, denoted by $\min(P)$ is the smallest (wrt. set inclusion) closed set of literals. A set M of literals is called consistent if it does not contain both a and $\neg a$ for an atom a. A program P is called consistent if its minimal model is consistent.

An inconsistency measure [9, 20] is a function $\mathcal{I} : \mathbb{B} \to \mathbb{R}^{\infty}_{\geq 0}$, where the semantics of the value are defined such that a higher value reflects a higher degree, or severity, of inconsistency. As the concept of a "severity" of inconsistency is not easily characterisable, numerous inconsistency measures have been proposed, see [19] for an overview. A baseline is the drastic inconsistency measure $\mathcal{I}_d$ [12], which only differentiates between inconsistent and consistent knowledge bases.

In general, all other measures can be divided into formula-centric measures, and atom-centric measures [11].[5] In this work, we therefore consider measures representative of these two groups, namely the MI-inconsistency measure [12], the *problematic* inconsistency measure [8] and the *contension* measure [8], as shown in Figure 1 and explained below.

Formula-centric measures take into account the (number of) formulas responsible for the overall inconsistency. A central approach to measure inconsistency here is derived from minimal inconsistent subsets. Let $\mathcal{B}$ be a rule base, the minimal inconsistent subsets MI of $\mathcal{B}$ are defined via

$$\mathsf{MI}(\mathcal{B}) = \{M \subseteq \mathcal{B} \mid M \models \perp, \forall M' \subset M : M' \not\models \perp\}.$$

Example 2. We recall $\mathcal{B}_1$. Then we have

$$\begin{aligned} \mathsf{MI}(\mathcal{B}_1) &= \{M_1\} \\ M_1 &= \{platinumCustomer, \\ &\qquad platinumCustomer \to contractuallyCapable, \\ &\qquad mentalCondition, \\ &\qquad mentalCondition \to \neg contractuallyCapable\} \end{aligned}$$

The MI-inconsistency measure $\mathcal{I}_{\mathsf{MI}}$ counts the number of minimal inconsistent subsets. A similar version is the problematic inconsistency measure $\mathcal{I}_p$ [8], which counts the number of distinct formulas appearing in any inconsistent subset.

Atom-centric measures take into account the propositional variables involved in the overall inconsistency. The contension measure $\mathcal{I}_c$ quantifies inconsistency by utilizing three-valued interpretations. Here, a three-valued interpretation is a function $v : \mathcal{A} \to \{b, t, f\}$, which assigns every atom to either b, f or t, where t and f correspond to the classic logical TRUE and FALSE, and b denotes that there exist conflicting truth values. Assuming the *truth order* $\prec_T$ with $f \prec_T b \prec_T t$, the function v is extended to arbitrary formulas as follows: $v(\alpha \wedge \beta) = \min_{\prec_T}(v(\alpha), v(\beta))$, $v(\alpha \wedge \beta) = \max_{\prec_T}(v(\alpha), v(\beta))$, $v(\neg\alpha) = t$ if $v(\alpha) = f$, $v(\neg\alpha) = f$ if $v(\alpha) = t$, and $v(\neg\alpha) = b$ if $v(\alpha) = b$. We say an interpretation v satisfies a formula α, denoted by $v \models^3 \alpha$, if $v(\alpha) = t$ or $v(\alpha) = b$. Then the contension measure quantifies inconsistency by seeking an interpretation that assigns b to a minimal number of propositions.

Example 3. Considering again $\mathcal{B}_1$, we see that

$$\mathcal{I}_d(\mathcal{B}_1) = 1 \quad \mathcal{I}_{\mathsf{MI}}(\mathcal{B}_1) = 1 \quad \mathcal{I}_p(\mathcal{B}_1) = 4 \quad \mathcal{I}_c(\mathcal{B}_1) = 1.$$

For all considered measures, we see that the measures inherently do not distinguish between facts and rules. Considering our use-case where facts have a different assumption of veracity than rules, this might impede a plausible application. We consequently investigate inconsistency measurement with a distinction between indisputable facts and rules.

3 Measures of Inconsistency with indisputable facts

Consider the following exemplary rule bases $\mathcal{B}_2$, $\mathcal{B}_3$, $\mathcal{B}_4$, defined via

$$\begin{aligned} \mathcal{B}_2 &= \{a, \neg a\} \\ \mathcal{B}_3 &= \{a, a \to b, a \to \neg b\} \\ \mathcal{B}_4 &= \{a, a \to b, a \to \neg b, c, \neg c\}. \end{aligned}$$

In a business rule management use-case, we are interested only in inconsistencies comprising at least one business rule, as this indicates a human modelling error in the set of business rules. Analysing only such modeling errors is an important basis for re-modelling. In turn, we will not consider inconsistencies such as in $\mathcal{B}_2$ (this can be handled by existing results from inconsistency measurement), but want to develop new "rule-based" inconsistency measures, in the following denoted as $\mathcal{I}^{RB}$, which can specifically assess actual modeling errors, i. e., inconsistencies including at least one rule. To this aim, we propose the property of rule-consistency that should be satisfied by rule-based inconsistency measures.

---

[5] We acknowledge there are hybrid forms and some outliers (cf. the discussion in [2]), but limit our discussion to these two main perspectives due to space limitations.

Rule Consistency **(RC)** $\mathcal{I}^{RB}(\mathcal{B}) = 0$ if and only if for all consistent sets $F' \subseteq \mathcal{F}(\mathcal{B})$, $\mathcal{R}(\mathcal{B}) \cup F'$ is consistent.

The above rationality postulate is a weakening of the classical postulate *consistency* [18], which requires $\mathcal{I}(\mathcal{B}) = 0$ if and only if $\mathcal{B}$ is consistent. Following RC, measures should assess the degree of inconsistency for $\mathcal{B}_2$ as 0. Vice versa, in case there is at least one inconsistency which touches at least one rule, the returned inconsistency value should not be 0.

In any case however, we want to ensure that only conflicts including at least one rule are valued towards the quantification of inconsistency. Atoms appearing only as facts should not alter the degree of inconsistency for rule-based measures, if they do not contradict any rules. We subsequently define the property of fact elision.

Fact Elision **(FE)** If $\forall r \in \mathcal{B} : head(r) \neq \alpha$ then $\mathcal{I}^{RB}(\mathcal{B}) = \mathcal{I}^{RB}(\mathcal{B} \cup \{\overline{\alpha}\})$.

This postulate is closely related to the postulate safe-formula independence (SI) [18], which states that formulas which do not share the signature with the existing propositions of a knowledge base, i. e., safe formulas, should not alter the degree of inconsistency. The proposed postulate is a weakening of SI, and states that a formula only needs to be safe w.r.t. the business rules. That is, even if an added formula is *not* safe w.r.t. facts in the knowledge base, this should not alter the score. Consequently, for any $\mathcal{I}^{RB}$, $\mathcal{I}^{RB}(\mathcal{B}_3 \cup c)$ should be equal to $\mathcal{I}^{RB}(\mathcal{B}_4 \cup c)$.

Last, we consider a further aspect of rule-based inconsistency measures. Consider again the rule base $\mathcal{B}_3$ and the rule base $\mathcal{B}_5$, defined as

$$\mathcal{B}_3 = \{a, a \rightarrow b, a \rightarrow \neg b\} \qquad \mathcal{B}_5 = \{a, a \rightarrow b, \neg b\}.$$

In a traditional setting of inconsistency measurement, one could argue that both rule bases are equally inconsistent. However, we see that the inconsistency in $\mathcal{B}_5$ can only be resolved in one way in our setting—namely by modifying or deleting the rule $a \rightarrow b$—as the given facts a and $\neg b$ are indisputable. On the contrary, the inconsistency in $\mathcal{B}_3$ is caused by contradicting rules, thus, this inconsistency is more complex to handle and requires attention by domain experts. To identify such cases, we introduce a third, optional property of rule emphasis. For that, a formula $a \in \mathcal{B}$ is called a *free formula*, if $a \notin M, \forall M \in \mathsf{MI}(\mathcal{B})$. We denote the free formulas of $\mathcal{B}$ as $\mathsf{Free}(\mathcal{B})$.

Rule Emphasis **(RE)** If $B \rightarrow H \notin \mathcal{B}$ and $B \rightarrow H \notin \mathsf{Free}(\mathcal{B} \cup \{B \rightarrow H\})$ then $\mathcal{I}(\mathcal{B} \cup \{B \rightarrow H\}) > \mathcal{I}(\mathcal{B} \cup \{H\})$.

This postulate states that adding a rule to a rule base, where this rule is not a free formula, should increase the inconsistency more than adding only the head of that rule, i. e. as a fact. This postulate ensures that measures valuate the conflicts involving contradictory rules as more significant than a conflict resulting from a rule and a non-negotiable fact (as the former type of inconsistency might be more complex to resolve than the latter). Note that RE is similar in spirit to the classical postulate *penalty*, which requires inconsistency to strictly increase whenever a non-free formula is added, cf. [18].

Example 4. For the rule bases $\mathcal{B}_2$, $\mathcal{B}_3$, $\mathcal{B}_4$, and $\mathcal{B}_5$ from before, we expect a rule-based inconsistency assessment $\mathcal{I}^{RB}$ satisfying the postulates RC, FE, and RE to give

$$\begin{aligned} 0 = \mathcal{I}^{RB}(\mathcal{B}_2) < \mathcal{I}^{RB}(\mathcal{B}_3) = \mathcal{I}^{RB}(\mathcal{B}_4) \qquad & \text{and} \\ \mathcal{I}^{RB}(\mathcal{B}_5) < \mathcal{I}^{RB}(\mathcal{B}_3) \qquad & \end{aligned}$$

However, for the considered inconsistency measures we get

$$\begin{array}{llll} \mathcal{I}_d(\mathcal{B}_2) = 1 & \mathcal{I}_d(\mathcal{B}_3) = 1 & \mathcal{I}_d(\mathcal{B}_4) = 1 & \mathcal{I}_d(\mathcal{B}_5) = 1 \\ \mathcal{I}_{\mathsf{MI}}(\mathcal{B}_2) = 1 & \mathcal{I}_{\mathsf{MI}}(\mathcal{B}_3) = 1 & \mathcal{I}_{\mathsf{MI}}(\mathcal{B}_4) = 2 & \mathcal{I}_{\mathsf{MI}}(\mathcal{B}_5) = 1 \\ \mathcal{I}_p(\mathcal{B}_2) = 2 & \mathcal{I}_p(\mathcal{B}_3) = 3 & \mathcal{I}_p(\mathcal{B}_4) = 5 & \mathcal{I}_p(\mathcal{B}_5) = 3 \\ \mathcal{I}_c(\mathcal{B}_2) = 1 & \mathcal{I}_c(\mathcal{B}_3) = 1 & \mathcal{I}_c(\mathcal{B}_4) = 2 & \mathcal{I}_c(\mathcal{B}_5) = 1 \end{array}$$

We see that none of the considered measures is capable of capturing the desired outcome. Specifically, we see that for the above measures (in the following abbreviated as $\mathcal{I}$ by a slight misuse of notation):

- $\mathcal{I}(\mathcal{B}_2) > 0$ for all measures, thus violating RC.
- $\mathcal{I}(\mathcal{B}_3) \neq \mathcal{I}(\mathcal{B}_4)$ for all measures except $\mathcal{I}_d$, thus broadly violating FE.
- $\mathcal{I}(\mathcal{B}_5) \not< \mathcal{I}(\mathcal{B}_3)$ for all measures, thus violating RE.

Regarding $\mathcal{I}(\mathcal{B}_2) > 0$, this is intuitive, as all considered measures satisfy the postulate of *consistency* (CO)[12], which demands that the returned value should only be zero iff the rule base is consistent. As a result, we have the following:

Proposition 5. CO is incompatible with RC.

Following from Proposition 5, virtually all existing inconsistency measures cannot be used as rule-based inconsistency measures, as they uniformly satisfy CO (cf. [18]) and thus broadly violate the proposed rationality postulates as motivated from the business use-case. This impedes using existing results in a company context and calls for an adaptation of measures to fit this use-case. In the following, we therefore propose rule-based versions of the original measures.

3.1 A baseline for rule-based measures

As a baseline measure, the rule-based drastic measure is geared to distinguish between inconsistent and rule-consistent rule bases. To recall, a rule base is not rule-consistent if it contains at least one minimal inconsistent subset, that itself contains at least one rule. To verify this condition, we consider only those minimal inconsistent subsets that do not contain two complementary facts $a, \neg a$. Formally, define $\mathsf{MI}^{\setminus\mathcal{F}}(\mathcal{B}) = \{M \in \mathsf{MI}(\mathcal{B}) \mid \neg\exists a \in \mathcal{A} : a, \neg a \in M\}$. If $a, \neg a \in M$ we also call M a pure fact set (note that indeed $a, \neg a \in M$ implies $M = \{a, \neg a\}$).

Example 6. We recall $\mathcal{B}_2$. Then we have

$$\mathsf{MI}(\mathcal{B}_2) = \{\{a, \neg a\}\} \qquad \mathsf{MI}^{\setminus\mathcal{F}}(\mathcal{B}_2) = \emptyset$$

We are now ready to define a baseline for rule-based measures.

Definition 7. The rule-based *drastic inconsistency measure* $\mathcal{I}_d^{RB} : \mathbb{B} \rightarrow \mathbb{R}^{\infty}_{\geq 0}$ is defined as

$$\mathcal{I}_d^{RB}(\mathcal{B}) = \begin{cases} 1 & \text{iff } \mathsf{MI}^{\setminus\mathcal{F}}(\mathcal{B}) \neq \emptyset \\ 0 & \text{otherwise} \end{cases}$$

for $\mathcal{B} \in \mathbb{B}$.

In other words, the rule-based drastic measure is 1 if and only if a rule base contains at least one inconsistent subset which is not simply a pair of two complementary facts $a, \neg a$ (and 0 otherwise).

Example 8. We recall the business rule bases $\mathcal{B}_2$ and $\mathcal{B}_5$, and consider a consistent rule base $\mathcal{B}_6$

$$\begin{aligned} \mathcal{B}_2 &= \{a, \neg a\} \\ \mathcal{B}_5 &= \{a, a \rightarrow b, \neg b\} \\ \mathcal{B}_6 &= \{a, a \rightarrow b, a \rightarrow c, d\}. \end{aligned}$$

Then, $\mathcal{I}_d^{RB}(\mathcal{B}_5) = 1$, and $\mathcal{I}_d^{RB}(\mathcal{B}_2) = \mathcal{I}_d^{RB}(\mathcal{B}_6) = 0$.

3.2 Rule-Based inconsistency measures based on formulas

We continue using the introduced notion of $\mathsf{MI}^{\setminus\mathcal{F}}$.

Definition 9. The rule-based MI-*inconsistency measure* $\mathcal{I}^{RB}_{\mathsf{MI}} : \mathbb{B} \to \mathbb{R}^{\infty}_{\geq 0}$ is defined as

$$\mathcal{I}^{RB}_{\mathsf{MI}}(\mathcal{B}) = |\mathsf{MI}^{\setminus\mathcal{F}}(\mathcal{B})|$$

for $\mathcal{B} \in \mathbb{B}$.

Considering only minimal inconsistent subsets without pure fact sets ensures satisfying the requirements of RC and FE, as pure fact MI are omitted.

Example 10. We recall the business rule bases $\mathcal{B}_2, \mathcal{B}_3$ and $\mathcal{B}_5$

$$\mathcal{B}_2 = \{a, \neg a\} \quad \mathcal{B}_3 = \{a, a \to b, a \to \neg b\} \quad \mathcal{B}_5 = \{a, a \to b, \neg b\}.$$

Then

$$\begin{aligned}
\mathsf{MI}(\mathcal{B}_2) &= \{\{a, \neg a\}\} \\
\mathsf{MI}^{\setminus\mathcal{F}}(\mathcal{B}_2) &= \emptyset \\
\mathsf{MI}(\mathcal{B}_3) &= \{\{a, a \to b, a \to \neg b\}\} \\
\mathsf{MI}^{\setminus\mathcal{F}}(\mathcal{B}_3) &= \{\{a, a \to b, a \to \neg b\}\} \\
\mathsf{MI}(\mathcal{B}_5) &= \{\{a, a \to b, \neg b\}\} \\
\mathsf{MI}^{\setminus\mathcal{F}}(\mathcal{B}_5) &= \{\{a, a \to b, \neg b\}\}
\end{aligned}$$

and thus $\mathcal{I}^{RB}_{\mathsf{MI}}(\mathcal{B}_2) = 0$ and $\mathcal{I}^{RB}_{\mathsf{MI}}(\mathcal{B}_3) = \mathcal{I}^{RB}_{\mathsf{MI}}(\mathcal{B}_5) = 1$.

Next, in the original problematic inconsistency measure, the idea is to count the number of formulas contributing in any MI. Thus, in our use-case, an intuitive adaptation is to consider only all problematic rules.

Definition 11. The rule-based *problematic inconsistency measure* $\mathcal{I}^{RB}_{p} : \mathbb{B} \to \mathbb{R}^{\infty}_{\geq 0}$ is defined as

$$\mathcal{I}^{RB}_{p}(\mathcal{B}) = |\bigcup_{M \in \mathsf{MI}^{\setminus\mathcal{F}}(\mathcal{B})} M \setminus \mathcal{F}(M)|$$

for $\mathcal{B} \in \mathbb{B}$.

Example 12. We continue Example 10 and recall

$$\begin{aligned}
\mathsf{MI}^{1}_{\mathcal{B}_3} &= \{a, a \to b, a \to \neg b\} \\
\mathsf{MI}^{1}_{\mathcal{B}_5} &= \{a, a \to b, \neg b\}
\end{aligned}$$

Then we have

$$\begin{aligned}
\mathsf{MI}^{1}_{\mathcal{B}_3} \setminus \mathcal{F}(M_1) &= \{a \to b, a \to \neg b\} \\
\mathsf{MI}^{1}_{\mathcal{B}_5} \setminus \mathcal{F}(M_1) &= \{a \to b\}
\end{aligned}$$

and thus $\mathcal{I}^{RB}_{p}(\mathcal{B}_3) = 2$ and $\mathcal{I}^{RB}_{p}(\mathcal{B}_5) = 1$.

3.3 Rule-based inconsistency measures based on multi-valued semantics

Again, for an adaptation of the contension measure, it is necessary to eliminate conflicts resulting from fact contradictions. Therefore, given a rule base $\mathcal{B}$ we propose to only consider $\mathsf{MI}^{\setminus\mathcal{F}}(\mathcal{B})$.

Definition 13. The rule-based *contension inconsistency measure* $\mathcal{I}^{RB}_{c} : \mathbb{B} \to \mathbb{R}^{\infty}_{\geq 0}$ is defined as

$$\mathcal{I}^{RB}_{c}(\mathcal{B}) = min\{|v^{-1}(b) \cap \mathcal{A}| \mid v \models^{3} \bigcup_{M \in \mathsf{MI}^{\setminus\mathcal{F}}(\mathcal{B})} M\}$$

for $\mathcal{B} \in \mathbb{B}$, with $\mathcal{I}^{RB}_{c}(\emptyset) = 0$.

Example 14. We recall the business rule bases $\mathcal{B}_2, \mathcal{B}_3, \mathcal{B}_4$ and $\mathcal{B}_5$

$$\begin{aligned}
&\mathcal{B}_2 = \{a, \neg a\} && \mathcal{B}_3 = \{a, a \to b, a \to \neg b\} \\
&\mathcal{B}_4 = \{a, a \to b, a \to \neg b, c, \neg c\} && \mathcal{B}_5 = \{a, a \to b, \neg b\}.
\end{aligned}$$

Then

$$\begin{aligned}
\bigcup_{M \in \mathsf{MI}^{\setminus\mathcal{F}}(\mathcal{B}_2)} M &= \emptyset \\
\bigcup_{M \in \mathsf{MI}^{\setminus\mathcal{F}}(\mathcal{B}_3)} M &= \{a, a \to b, a \to \neg b\} \\
\bigcup_{M \in \mathsf{MI}^{\setminus\mathcal{F}}(\mathcal{B}_4)} M &= \{a, a \to b, a \to \neg b\} \\
\bigcup_{M \in \mathsf{MI}^{\setminus\mathcal{F}}(\mathcal{B}_5)} M &= \{a, a \to b, \neg b\}.
\end{aligned}$$

Then consider $v_1 : \{a, b\} \to \{b, t, f\}$, defined via

$$v_1(a) = t \qquad v_1(b) = b$$

Then we have $v_1 \models^{3} \alpha$ for all formulas α in the considered unions of $\mathsf{MI}^{\setminus\mathcal{F}}$. Also, there is no interpretation that assigns b to fewer propositions, and thus $\mathcal{I}^{RB}_{c}(\mathcal{B}_3) = \mathcal{I}^{RB}_{c}(\mathcal{B}_4) = \mathcal{I}^{RB}_{c}(\mathcal{B}_5) = 1$ and $\mathcal{I}^{RB}_{c}(\mathcal{B}_2) = 0$.

3.4 Analysis

We now investigate the compliance of the adapted measures with the proposed rationality postulates. We would like to remind the reader that the proofs of the technical results can be found in an extended version referenced above.

Example 15. Recall that for the rule bases $\mathcal{B}_2, \mathcal{B}_3, \mathcal{B}_4, \mathcal{B}_5$ we expect a rule-based inconsistency assessment $\mathcal{I}^{RB}$ satisfying the postulates RC, FE, and RE to give

$$\begin{aligned}
0 = \mathcal{I}^{RB}(\mathcal{B}_2) < \mathcal{I}^{RB}(\mathcal{B}_3) &= \mathcal{I}^{RB}(\mathcal{B}_4) \qquad \text{and} \\
\mathcal{I}^{RB}(\mathcal{B}_5) &< \mathcal{I}^{RB}(\mathcal{B}_3)
\end{aligned}$$

For our adapted measures we get

$$\begin{aligned}
&\mathcal{I}^{RB}_{d}(\mathcal{B}_2) = 0 && \mathcal{I}^{RB}_{d}(\mathcal{B}_3) = 1 && \mathcal{I}^{RB}_{d}(\mathcal{B}_4) = 1 && \mathcal{I}^{RB}_{d}(\mathcal{B}_5) = 1 \\
&\mathcal{I}^{RB}_{\mathsf{MI}}(\mathcal{B}_2) = 0 && \mathcal{I}^{RB}_{\mathsf{MI}}(\mathcal{B}_3) = 1 && \mathcal{I}^{RB}_{\mathsf{MI}}(\mathcal{B}_4) = 1 && \mathcal{I}^{RB}_{\mathsf{MI}}(\mathcal{B}_5) = 1 \\
&\mathcal{I}^{RB}_{p}(\mathcal{B}_2) = 0 && \mathcal{I}^{RB}_{p}(\mathcal{B}_3) = 2 && \mathcal{I}^{RB}_{p}(\mathcal{B}_4) = 2 && \mathcal{I}^{RB}_{p}(\mathcal{B}_5) = 1 \\
&\mathcal{I}^{RB}_{c}(\mathcal{B}_2) = 0 && \mathcal{I}^{RB}_{c}(\mathcal{B}_3) = 1 && \mathcal{I}^{RB}_{c}(\mathcal{B}_4) = 1 && \mathcal{I}^{RB}_{c}(\mathcal{B}_5) = 1
\end{aligned}$$

We see that our alterations have improved all measures wrt. the rationality postulates, in particular wrt. RC.

Table 1 and Table 2 summarize our results regarding the compliance with the rationality postulates motivated by our use case.

Proposition 16. $\mathcal{I}^{RB}_{d}, \mathcal{I}^{RB}_{\mathsf{MI}}, \mathcal{I}^{RB}_{c}$ satisfy RC and FE, and do not satisfy RE.

Proposition 17. $\mathcal{I}^{RB}_{p}$ satisfies RC, FE and RE.

$\mathcal{I}$	RC	FE	RE
$\mathcal{I}_d$	✗	✗	✗
$\mathcal{I}_{\mathsf{MI}}$	✗	✗	✗
$\mathcal{I}_p$	✗	✗	✗
$\mathcal{I}_c$	✗	✗	✗

Table 1. Compliance with rationality postulates of the *original* inconsistency measures

$\mathcal{I}^{RB}$	RC	FE	RE
$\mathcal{I}_d^{RB}$	✓	✓	✗
$\mathcal{I}_{\mathsf{MI}}^{RB}$	✓	✓	✗
$\mathcal{I}_p^{RB}$	✓	✓	✓
$\mathcal{I}_c^{RB}$	✓	✓	✗

Table 2. Compliance with rationality postulates of the *proposed* rule-based inconsistency measures

4 Inconsistency Values with Indisputable Facts

So far we considered inconsistency measures that assess the *entire* rule base. In the context of inconsistency handling, this is, however, often not sufficient. Companies need to pin-point those formulas in their rule bases that contribute towards the overall inconsistency, e. g. as a basis for inconsistency resolution. As a manual analysis of formulas can quickly become unfeasible, the field of inconsistency measurement also studies so-called *inconsistency values*. These are essentially functions which assign a numerical value to individual formulas of a rule base, with the intuition that a higher value indicates a higher blame which a resp. formula carries in the context of the overall inconsistency[6].

As with inconsistency measures, there have been numerous proposals for specific inconsistency values, see e. g. [13] for a nice overview. In this work, we consider the Shapley inconsistency value as proposed in [11], as it is a generalized measure which can be parametrized with arbitrary inconsistency measures. The Shapley inconsistency value uses notions from game theory to determine the blame—also referred to as payoff—that each formula carries w.r.t. the assessment of an arbitrary inconsistency measure. We can consequently directly plug in our proposed rule-based measures. For the following discussion, we assume all rule-based inconsistency measures used to derive inconsistency values satisfy RC. Also, we assume the used inconsistency measures satisfy the two basic properties of monotony and free-formula independence [12], which are usual desirable properties satisfied by most measures [18].

Monotony **(MO)** If $\mathcal{B} \subseteq \mathcal{B}'$ then $\mathcal{I}(\mathcal{B}) \leq \mathcal{I}(\mathcal{B}')$
Free-formula independence **(IN)** If $\alpha \in \mathsf{Free}(\mathcal{B})$ then
$\mathcal{I}(\mathcal{B}) = \mathcal{I}(\mathcal{B} \setminus \{\alpha\})$

MO demands that the addition of information cannot decrease the degree of inconsistency. IN states that free formulas should not affect the degree of inconsistency.

We are now ready to plug rule-based inconsistency measures into the original Shapley inconsistency value.

Definition 18. Let $\mathcal{I}^{RB}$ be a rule-based inconsistency measure, $\mathcal{B}$ be a rule base and $\alpha \in \mathcal{B}$. Then, the Shapley inconsistency value of α w.r.t. $\mathcal{I}^{RB}$, denoted $S_\alpha^{\mathcal{I}^{RB}}$ is defined via

$$S_\alpha^{\mathcal{I}^{RB}}(\mathcal{B}) = \sum_{B \subseteq \mathcal{B}} \frac{(b-1)!(n-b)!}{n!}(\mathcal{I}^{RB}(\mathcal{B}) - \mathcal{I}^{RB}(\mathcal{B} \setminus \alpha))$$

where b is the cardinality of B, and n is the cardinality of $\mathcal{B}$.

In the following, we consider all elements α of a rule base as a vector $(\alpha_1, \alpha_2, ...\alpha_n)$, and denote $S^{\mathcal{I}^{RB}}(\mathcal{B})$ as the vector of corresponding Shapley inconsistency values, i. e., $S^{\mathcal{I}^{RB}}(\mathcal{B}) = (S_{\alpha_1}^{\mathcal{I}^{RB}}(\mathcal{B}), S_{\alpha_2}^{\mathcal{I}^{RB}}(\mathcal{B}), ..., S_{\alpha_n}^{\mathcal{I}^{RB}}(\mathcal{B}))$. In turn, the Shapley inconsistency value based on the proposed rule-based inconsistency measures satisfies some desirable properties. This result is adapted from [11], the proofs are analogous.

Proposition 19. Let $\mathcal{I}^{RB}$ be a rule-based inconsistency measure, $\mathcal{B}$ be a rule base and $\alpha \in \mathcal{B}$. Then, $S^{\mathcal{I}^{RB}}(\mathcal{B})$ satisfies:

- **Distribution.** $\sum_{\alpha \in \mathcal{B}} S_\alpha^{\mathcal{I}^{RB}}(\mathcal{B}) = \mathcal{I}^{RB}(\mathcal{B})$
- **Minimality.** If $\mathcal{I}^{RB}$ satisfies IN and α is a free formula of $\mathcal{B}$, then $S_\alpha^{\mathcal{I}^{RB}}(\mathcal{B}) = 0$

Example 20. Consider the rule base $\mathcal{B}_7 = \{a, a \to b, a \to \neg b, \neg a\}$. Then, for the Shapley inconsistency values w.r.t. $\mathcal{I}_d^{RB}$ and $\mathcal{I}_{\mathsf{MI}}^{RB}$, we have that $S_a^{\mathcal{I}_d^{RB}}(\mathcal{B}) = S_a^{\mathcal{I}_{\mathsf{MI}}^{RB}}(\mathcal{B}_7) = \frac{1}{12} + \frac{1}{4} = \frac{1}{3}$. Also, we have that $S_{\neg a}^{\mathcal{I}_d^{RB}}(\mathcal{B}_7) = S_{\neg a}^{\mathcal{I}_{\mathsf{MI}}^{RB}}(\mathcal{B}_7) = 0$. Thus, we have $S^{\mathcal{I}_d^{RB}}(\mathcal{B}_7) = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3}, 0)$ and $S^{\mathcal{I}_{\mathsf{MI}}^{RB}}(\mathcal{B}_7) = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3}, 0)$.

What is nice about Example 20 is that it shows that the gist of RC of the rule-based inconsistency measures transfers to the element-based assessment: Only those formulas that are part of at least one MI which itself contains at least one rule are assigned blame. Still, we see the following problem in using the above Shapley inconsistency value for measuring culpability in our setting. In Example 20, we see that the blame is equally distributed over $\{a, a \to b, a \to \neg b\}$ in both assessments. However, facts are viewed as indisputable. This has strong implications for element-based culpability, the simplest one being that facts should not be deleted. In turn, they should also not be assigned with any blame value, as they have to be kept as-is. To capture this requirement, we therefore propose a new property of fact-minimality.

- **Fact-Minimality** $S_f^{\mathcal{I}^{RB}}(\mathcal{B}) = 0$ for any fact f in $\mathcal{B}$.

This is an extension of the minimality property, which is necessary for the intended use-case of viewing facts as indisputable, i. e., facts should not be associated with any blame (value) towards the overall inconsistency. As evidenced by Example 20, the Shapley inconsistency value does not satisfy fact-minimality. Therefore, it is not plausible to apply the Shapley inconsistency value in our use case. We therefore propose an adjusted Shapley inconsistency value. The intuition of our approach is as follows.

The original Shapley inconsistency value essentially assigns responsibilities to a number of formulas (or players) in a coalition. Currently, there is no distinction between facts and rules. Following our use case, the idea is to shift the blame from facts to all rules which are part of the inconsistency for that coalition. We now introduce some notation on this matter for later clarification.

Definition 21. Let $\mathcal{I}^{RB}$ be a rule-based inconsistency measure, $\mathcal{B}$ be a rule base and $\alpha \in \mathcal{B}$. Then, the individual Shapley inconsistency coalition value of α w.r.t. $\mathcal{I}^{RB}$ (in a coalition $B \subseteq \mathcal{B}$), is defined via

$$\mathit{CoalPayoff}_{\alpha,\mathcal{B}}^{\mathcal{I}^{RB}}(B) = \frac{(b-1)!(n-b)!}{n!}(\mathcal{I}^{RB}(B) - \mathcal{I}^{RB}(B \setminus \alpha))$$

where b is the cardinality of B, and n is the cardinality of $\mathcal{B}$.

In our use case, blame should only be assigned to rules. Accordingly, the share of the blame that falls upon facts from any coalition should be shifted away from the facts and equally distributed among the blamable rules, i. e., the rules which contribute towards the inconsistency for that coalition. We denote the blame that is shifted from facts to the individual blamable rules, as an *additional payoff*.

---

[6] Note that measures that determine inconsistency values are also referred to as *culpability measures*.

Definition 22. Let $\mathcal{I}^{RB}$ be a rule-based inconsistency measure and $\mathcal{B}$ be a rule base, denote the additional blame for a *rule* $r \in \mathcal{R}(\mathcal{B})$ in any coalition $B \subseteq \mathcal{B}$ as

$$AddPayoff^{\mathcal{I}^{RB}}_{r,\mathcal{B}}(B) = \begin{cases} 0 & \text{if } r \in \mathsf{Free}(B) \\ \frac{\sum_{f \in \mathcal{F}(B)} CoalPayoff^{\mathcal{I}^{RB}}_{f,\mathcal{B}}(B)}{|r' \in \mathcal{R}(B) \text{ s.t. } r' \notin \mathsf{Free}(B)|} & \text{otherwise} \end{cases}$$

Example 23. Consider again $\mathcal{B}_3 = \{a, a \to b, a \to \neg b\}$ and $\mathcal{B}_5 = \{a, a \to b, \neg b\}$. According to the original Shapley value, the blame is evenly distributed for both rule bases. In $\mathcal{B}_3$, given the premise of indisputable facts, a is not to blame for the overall inconsistency. Rather, the blame value of a should be evenly distributed among $a \to b$ and $a \to \neg b$, as both these formulas evenly contribute towards the inconsistency. Next, in $\mathcal{B}_5$, both a and $\neg b$ are not to blame in our use case. Here, the blame values of a and $\neg b$ should be transferred to $a \to b$. This is directly in line with the intuition of RE, as for $\mathcal{B}_5$, the only way to resolve the inconsistency would be to remove $a \to b$ (thus all blame is relocated to that rule), but for $\mathcal{B}_3$, one can delete either of the two rules (thus the blame is distributed among both rules). In result, for each coalition, the blame is shifted from facts to the blameable rules via the additional payoff.

We are now ready to define the adjusted Shapley inconsistency value.

Definition 24. Let $\mathcal{I}^{RB}$ be a rule-based inconsistency measure, $\mathcal{B}$ be a rule base and $\alpha \in \mathcal{B}$. Then, the adjusted Shapley inconsistency value of α w.r.t. $\mathcal{I}^{RB}$, denoted $S{*}^{\mathcal{I}^{RB}}_{\alpha}$ is defined via

$$S *^{\mathcal{I}^{RB}}_{\alpha}(\mathcal{B}) = \begin{cases} 0 & \text{if } \alpha \in \mathcal{F}(\mathcal{B}) \\ \sum\limits_{B \subseteq \mathcal{B}} (CoalPayoff^{\mathcal{I}^{RB}}_{\alpha,\mathcal{B}}(B) + AddPayoff^{\mathcal{I}^{RB}}_{\alpha,\mathcal{B}}(B)) & \text{otherwise} \end{cases}$$

The adjusted Shapley value assigns the value of 0 to all facts, and computes the blame value of all rules taking into consideration the additional payoff.

Example 25. Consider the rule bases $\mathcal{B}_3 = \{a, a \to b, a \to \neg b\}$ and $\mathcal{B}_5 = \{a, a \to b, \neg b\}$. Then for the *adjusted* Shapley inconsistency values w.r.t. $\mathcal{I}^{RB}_d$, we have that $S^{\mathcal{I}^{RB}_d}_a(\mathcal{B}_3) = 0, S^{\mathcal{I}^{RB}_d}_{a \to b}(\mathcal{B}_3) = \frac{1}{3}(+\frac{1}{3}/2) = \frac{1}{2}$, and $S^{\mathcal{I}^{RB}_d}_{a \to \neg b}(\mathcal{B}_3) = \frac{1}{3}(+\frac{1}{3}/2) = \frac{1}{2}$. Also, we have that $S^{\mathcal{I}^{RB}_d}_a(\mathcal{B}_5) = 0, S^{\mathcal{I}^{RB}_d}_{\neg b}(\mathcal{B}_5) = 0$, and $S^{\mathcal{I}^{RB}_d}_{a \to \neg b}(\mathcal{B}_3) = \frac{1}{3}(+\frac{2}{3}) = 1$. Thus, we have that $S^{\mathcal{I}^{RB}_d}(\mathcal{B}_3) = (0, \frac{1}{2}, \frac{1}{2})$ and $S^{\mathcal{I}^{RB}_d}(\mathcal{B}_5) = (0, 1, 0)$, which is directly in line with RE.

Proposition 26. The adjusted Shapley value satisfies Distribution, Minimality and Fact-Minimality.

Proposition 26 shows that our approach follows the same gist as the original Shapley inconsistency value, but shifts the blame from facts to blamable rules as necessary in our use-case. Next to Distribution and (fact-)Minimality, we can identify a further property for this quantitative assessment $S *^{\mathcal{I}^{RB}}_{\alpha}(\mathcal{B})$.

Proposition 27. Let $\mathcal{I}^{RB}$ be a rule-based inconsistency measure, $\mathcal{B}$ be a rule base and $r \in \mathcal{R}(\mathcal{B})$.

- **Rule-Involvement.** If $\mathcal{I}^{RB}$ satisfies RC and $r \in \mathcal{B}$ is a non-free rule then $S *^{\mathcal{I}^{RB}}_{r}(\mathcal{B}) > 0$.

This ensures that rules which are part of any inconsistency have an inconsistency value greater 0.

Next to properties for individual formula assessments, we can also identify properties of the distribution of blame in the vector $S{*}^{\mathcal{I}^{RB}}$.

Definition 28. Let a rule base $\mathcal{B}$, define $\hat{S} *^{\mathcal{I}^{RB}}(\mathcal{B}) = max_{\alpha \in \mathcal{B}} S *^{\mathcal{I}^{RB}}_{\alpha}(\mathcal{B})$.

Proposition 29. Let $\mathcal{I}^{RB}$ be a rule-based inconsistency measure, and $\mathcal{B}$ be a rule base. Then, following [11], we have:

- **Rule Consistency'.** $\hat{S} *^{\mathcal{I}^{RB}}(\mathcal{B}) = 0$ iff $\mathcal{B}$ is rule consistent.
- **Free formula independence'.** If $\mathcal{I}^{RB}$ satisfies IN and α is a free formula of $\mathcal{B} \cup \{\alpha\}$, then $\hat{S} *^{\mathcal{I}^{RB}}(\mathcal{B} \cup \{\alpha\}) = \hat{S} *^{\mathcal{I}^{RB}}(\mathcal{B})$.
- **Upper Bound.** $\hat{S} *^{\mathcal{I}^{RB}}(\mathcal{B}) \leq \mathcal{I}^{RB}(\mathcal{B})$.

The first property states that the highest adjusted Shapley inconsistency value can only be 0 in case of a (rule) consistent rule base. The second property states that adding free formulas should not increase any individual values. The last property describes an upper bound.

From the above discussion, we have shown that the proposed adjusted Shapley inconsistency value can be used to assess the distribution of blame for individual rules, relative to the overall inconsistency. Also, the blame (value) will be higher for rules with a higher blame. This information can be useful for companies, e. g. for prioritizing which rules to attend to first. To this aim, the elements of a rule base can be ranked by their adjusted Shapley inconsistency value, similar to the approach in [4].

Definition 30 ([4]). Let $\mathcal{I}^{RB}$ be a rule-based inconsistency measure and $\mathcal{B}$ be a rule base, define the adjusted Shapley culpability ranking over all rules $\alpha \in \mathcal{B}$, denoted $C(\mathcal{B})$ via $\langle \alpha_1, ..., \alpha_n \rangle$, s.t. $S *^{\mathcal{I}^{RB}}_{\alpha_1}(\mathcal{B}) \geq ... \geq S *^{\mathcal{I}^{RB}}_{\alpha_n}(\mathcal{B})$.

Observe that the culpability ranking ranks elements of a rule base by their adjusted Shapley inconsistency values. As dictated by *Fact-Minimality*, all facts have a value of 0 and rank last (or may be ommitted entirely during inspection). Rather, the ranking represents the blame of all rules in prioritized order and thus provides valuable insights towards inconsistency resolution. In a setting where facts are non-negotiable, it is necessary to adjust the Shapley inconsistency value, as otherwise blame would also be assigned to facts, which could render an undesirable recommendation of deleting a fact.

To show the usefulness of such a culpability ranking and recap the need for the measure adjustments made in this paper, we close with an application example.

5 Application Example

Consider the rule base $\mathcal{B}'_1$, defined via

$$\begin{aligned} \mathcal{B}'_1 = \{&customer, mentalCondition, platinumCustomer, \\ &customer \to contractuallyCapable, \\ &mentalCondition \to \neg contractuallyCapable, \\ &mentalCondition \to \neg platinumCustomer\} \end{aligned}$$

with the intuitive meaning that we have a (platinum) customer who also has a mental condition, and three rules stating that 1) all customers are generally contractually capable, 2) a person with a mental condition is not contractually capable, and 3) a person with a mental condition is not a platinum customer. We see that $\mathcal{B}'_1$ is inconsistent.

In this section, we assume a company needs to analyze $\mathcal{B}'_1$ with the aim of resolving the inconsistency. Furthermore, we assume the facts were provided by a new case input and are thus non-negotiable.

To begin, $\mathcal{B}'_1$ yields

$$\begin{aligned}
&\mathsf{MI}(\mathcal{B}'_1) = \{M_1, M_2\}\\
&\quad M_1 = \{customer, customer \rightarrow contractuallyCapable,\\
&\qquad\quad mentalCondition,\\
&\qquad\quad mentalCondition \rightarrow \neg contractuallyCapable\}\\
&\quad M_2 = \{platinumCustomer, mentalCondition,\\
&\qquad\quad mentalCondition \rightarrow \neg platinumCustomer\}
\end{aligned}$$

Then we have

$$\begin{array}{llll}
\mathcal{I}_d(\mathcal{B}'_1) = 1 & \mathcal{I}_{\mathsf{MI}}(\mathcal{B}'_1) = 2 & \mathcal{I}_p(\mathcal{B}'_1) = 6 & \mathcal{I}_c(\mathcal{B}'_1) = 1\\
\mathcal{I}^{RB}_d(\mathcal{B}'_1) = 1 & \mathcal{I}^{RB}_{\mathsf{MI}}(\mathcal{B}'_1) = 2 & \mathcal{I}^{RB}_p(\mathcal{B}'_1) = 3 & \mathcal{I}^{RB}_c(\mathcal{B}'_1) = 1
\end{array}$$

As can be expected, the rule-based versions of the drastic-, the MI-, and the contension-measure do not differ from their original counterpart, as we do not have any fact contradictions. However, $\mathcal{I}_p$ is highly confusing to modelers in our scenario, as it suggests there are 6 problematic pieces of information. Correctly—w.r.t. the use-case—our adapted $\mathcal{I}^{RB}_p$ counts 3 problematic pieces of information. Note that the original measures would be even more confusing in our use-case in the presence of fact contradictions (cf. e. g. the example in Section 3), and would provide only very limited insights towards re-modelling and improving business rules. This becomes even more apparent for inconsistency values:

Assume the company now wants to pin-point elements of the rule base which are responsible for the overall inconsistency as a basis for inconsistency resolution.

Part 1 (Inconsistency handling with existing means). We recall $\mathcal{B}'_1$. Then we have that

$$\begin{aligned}
S^{\mathcal{I}^{RB}_d}_{customer}(\mathcal{B}'_1) &= \frac{1}{12},\\
S^{\mathcal{I}^{RB}_d}_{mentalCondition}(\mathcal{B}'_1) &= \frac{3}{12} + \frac{1}{6},\\
S^{\mathcal{I}^{RB}_d}_{platinumCustomer}(\mathcal{B}'_1) &= \frac{1}{6},\\
&etc.
\end{aligned}$$

(Analogously for $\mathcal{I}^{RB}_{\mathsf{MI}}$). Thus, we have $S^{\mathcal{I}^{RB}_d}(\mathcal{B}'_1) = (\frac{1}{12}, 0.41\overline{6}, \frac{1}{6}, \frac{1}{12}, \frac{1}{12}, \frac{1}{6})$ and $S^{\mathcal{I}^{RB}_{\mathsf{MI}}}(\mathcal{B}'_1) = (\frac{1}{4}, 0.58\overline{3}, \frac{1}{3}, \frac{1}{4}, \frac{1}{4}, \frac{1}{3})$. For both assessments, a recommendation based on the original Shapley value would strongly suggest to delete the fact $mentalCondition$ first. Here, this is not an acceptable recommendation, as one cannot delete the fact that the customer has a mental condition in our setting. Rather, the rules should be deleted or modified. However, even if one would skip the first recommendation based on the original Shapley values, the ranking also does not further distinguish between the remaining facts and rules of the individual MI. We see that the recommendation based on the original Shapley inconsistency value is not plausible and provides very limited value for companies. We will therefore now consider an assessment via our proposed means.

Part 2 (Inconsistency handling with the proposed means). Consider again $\mathcal{B}'_1$. Then we have that

$$S*^{\mathcal{I}^{RB}_d}_{mentalCondition}(\mathcal{B}'_1) = 0,$$

etc. for all facts. Also, we have that

$$\begin{aligned}
&S*^{\mathcal{I}^{RB}_d}_{customer \rightarrow contractuallyCapable}(\mathcal{B}'_1) && = 0.2\overline{2}\\
&S*^{\mathcal{I}^{RB}_d}_{mentalCondition \rightarrow \neg contractuallyCapable}(\mathcal{B}'_1) && = 0.2\overline{2}\\
&S*^{\mathcal{I}^{RB}_d}_{mentalCondition \rightarrow \neg platinumCustomer}(\mathcal{B}'_1) && = 0.5\overline{5}
\end{aligned}$$

(Analogously for $\mathcal{I}^{RB}_{\mathsf{MI}}$). Thus, we have $S^{\mathcal{I}^{RB}_d}(\mathcal{B}'_1) = (0, 0, 0, 0.2\overline{2}, 0.2\overline{2}, 0.5\overline{5})$ and $S^{\mathcal{I}^{RB}_{\mathsf{MI}}}(\mathcal{B}'_1) = (0, 0, 0, 0.5\overline{5}, 0.5\overline{5}, 0.8\overline{8})$. A recommendation based on a culpability ranking using these adjusted Shapley values proposes to attend to $mentalCondition \rightarrow \neg platinumCustomer$ first. This makes sense in our example, as this is the only rule in the resp. MI, thus the only option is to delete (or alter) this rule in M_2. Then, the recommendation suggests to attend to the remaining two rules with an equal value. This also follows our use-case, as an expert has two possible options in M_1.

6 Conclusion

From our discussion, we see that although the field of inconsistency measurement would be a good candidate for supporting companies, a straightforward application is not plausible due to the assumption of non-negotiable facts in the company setting. To this aim, the adapted means presented in this paper are a first step towards allowing for a plausible application of inconsistency measurement in the scope of business rule bases. Future work should also investigate inconsistency measurement for knowledge bases partitioned into general non-negotiable and negotiable parts (regardless of the granularity of the respective information), for example through a general study of inconsistency measurement in defeasible logics [15]. Based on recent studies [1, 17, 16, 4, 5], approaches for inconsistency handling are needed from a business perspective and could thus also be an interesting application domain for future work. We would like to point out one specific result from the case-study in [17], namely that companies are not only facing the problem of inconsistent rules, but also the problem of identical (redundant) rules (which could for example result from collaborative modeling or a lack of oversight). As most research in inconsistency measurement is based on sets, applying these results to multi-sets of (business) rules should be further examined, cf. also a recent discussion in [3].

Acknowledgements: The research reported here was supported by the Deutsche Forschungsgemeinschaft (grant DE 1983/9-1).

REFERENCES

[1] Kimon Batoulis, Alexey Nesterenko, Günther Repitsch, and Mathias Weske, 'Decision management in the insurance industry: Standards and tools', in *International Conference on Business Process Management*. Springer, (2017).

[2] Philippe Besnard, 'Forgetting-based inconsistency measure', in *International Conference on Scalable Uncertainty Management*, pp. 331–337. Springer, (2016).

[3] Philippe Besnard, 'Inconsistency measuring over multisets of formulas', in *Measuring Inconsistency in Information*, eds., John Grant and Maria Vanina Martinez, College Publications, (2018).

[4] Carl Corea and Patrick Delfmann, 'Supporting business rule management with inconsistency analysis', in *Proceedings of the BPM 2018 Industry Track co-located with the 16th International Conference on Business Process Management*, (2018).

[5] Claudio Di Ciccio, Fabrizio Maria Maggi, Marco Montali, and Jan Mendling, 'Resolving inconsistencies and redundancies in declarative process models', *Information Systems*, **64**, 425–446, (2017).

[6] M. Gelfond and V. Lifschitz, 'Classical negation in logic programs and disjunctive databases', *New Generation Computing*, **9**, 365–385, (1991).

[7] Ian Graham, *Business rules management and service oriented architecture: a pattern language*, John wiley & sons, 2007.

[8] John Grant and Anthony Hunter, 'Measuring consistency gain and information loss in stepwise inconsistency resolution', in *European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, pp. 362–373. Springer, (2011).

[9] *Measuring Inconsistency in Information*, eds., John Grant and Maria Vanina Martinez, College Publications, 2018.

[10] Mustafa Hashmi, Guido Governatori, Ho-Pun Lam, and Moe Thandar Wynn, 'Are we done with business process compliance: state of the art and challenges ahead', *Knowledge and Information Systems*, **57**(1), 79–133, (2018).

[11] Anthony Hunter and Sébastien Konieczny, 'On the measure of conflicts: Shapley inconsistency values', *Artificial Intelligence*, **174**(14), 1007–1026, (2010).

[12] Anthony Hunter, Sébastien Konieczny, et al., 'Measuring inconsistency through minimal inconsistent sets.', *KR*, **8**, 358–366, (2008).

[13] Kevin McAreavey, Weiru Liu, and Paul Miller, 'Computational approaches to finding and measuring inconsistency in arbitrary knowledge bases', *International Journal of Approximate Reasoning*, **55**(8), 1659–1693, (2014).

[14] Matthew L Nelson and Ravi Sen, 'Business rules management in healthcare: A lifecycle approach', *Decision Support Systems*, **57**, 387–394, (2014).

[15] Donald Nute, 'Defeasible logic', in *Handbook of logic in artificial intelligence and logic programming*, volume 3, 353–395, Oxford University Press, (1994).

[16] Shazia Sadiq and Guido Governatori, 'Managing regulatory compliance in business processes', in *Handbook on Business Process Management 2*, 159–175, Springer, (2015).

[17] Koen Smit, Martijn Zoet, and Matthijs Berkhout, 'Verification capabilities for business rules management in the dutch governmental context', in *Research and Innovation in Information Systems (ICRIIS), 2017 International Conference on*, pp. 1–6. IEEE, (2017).

[18] Matthias Thimm, 'On the compliance of rationality postulates for inconsistency measures: A more or less complete picture', *KI-Künstliche Intelligenz*, **31**(1), 31–39, (2017).

[19] Matthias Thimm, 'On the evaluation of inconsistency measures', in *Measuring Inconsistency in Information*, eds., John Grant and Maria Vanina Martinez, College Publications, (February 2018).

[20] Matthias Thimm, 'Inconsistency measurement', in *Proceedings of the 13th International Conference on Scalable Uncertainty Management (SUM'19)*, (2019).

[21] Wil M. P. van der Aalst, Marcello La Rosa, and Flávia Maria Santoro, 'Business process management', *Business & Information Systems Engineering*, **58**(1), 1–6, (2016).

A Proofs of technical results

Proposition 16. $\mathcal{I}_d^{RB}, \mathcal{I}_{\mathsf{MI}}^{RB}, \mathcal{I}_c^{RB}$ satisfy RC and FE, and do not satisfy RE.

Proof. For showing FE the following general observation will be useful:

$$\forall r \in \mathcal{B} : head(r) \neq \alpha \Longrightarrow \mathsf{MI}^{\setminus\mathcal{F}}(\mathcal{B}) = \mathsf{MI}^{\setminus\mathcal{F}}(\mathcal{B} \cup \{\overline{\alpha}\}) \tag{2}$$

To see this, let $M \in \mathsf{MI}^{\setminus\mathcal{F}}(\mathcal{B})$. Then obviously $M \in \mathsf{MI}^{\setminus\mathcal{F}}(\mathcal{B} \cup \{\overline{\alpha}\})$ as well. Let $M \in \mathsf{MI}^{\setminus\mathcal{F}}(\mathcal{B} \cup \{\overline{\alpha}\})$. Then $\overline{\alpha} \notin M$ because 1.) it cannot be that $\alpha \in M$ as this would violate the definition of $\mathsf{MI}^{\setminus\mathcal{F}}$ and 2.) there is no rule concluding α by assumption. It follows $M \in \mathsf{MI}^{\setminus\mathcal{F}}(\mathcal{B})$.

We now consider each measure $\mathcal{I}_d^{RB}, \mathcal{I}_{\mathsf{MI}}^{RB}, \mathcal{I}_c^{RB}$ in turn.

- We start with the measure $\mathcal{I}_d^{RB}$ and RC. Assume $\mathcal{I}_d^{RB}(\mathcal{B}) = 0$. Then $\mathsf{MI}^{\setminus\mathcal{F}}(\mathcal{B}) = \emptyset$ and therefore for all $M \in \mathsf{MI}(\mathcal{B})$ there is $a \in \mathcal{A}$ with $a, \neg a \in M$. So for all consistent subsets $F' \subseteq \mathcal{F}(\mathcal{B})$, $\mathcal{R}(\mathcal{B}) \cup F'$ is consistent. The other direction is analogous.
  FE follows directly by definition and Equation (2).
  For RE consider $\mathcal{B} = \{a; b; a \to \neg a\}$. Then $b \to \neg b \notin \mathcal{B}$, $b \to \neg b \notin \mathsf{Free}(\mathcal{B} \cup \{b \to \neg b\})$ but $\mathcal{I}_d^{RB}(\mathcal{B} \cup \{\neg b\}) = 1 = \mathcal{I}_d^{RB}(\mathcal{B} \cup \{b \to \neg b\})$.
- We now consider $\mathcal{I}_{\mathsf{MI}}^{RB}$. The proof of RC is analogous to $\mathcal{I}_d^{RB}$.
  FE follows directly by definition and Equation (2).
  For RE consider $\mathcal{B} = \{a; \neg c; b \to c\}$. Then $a \to b \notin \mathcal{B}$, $a \to b \notin \mathsf{Free}(\mathcal{B} \cup \{a \to b\})$ but $\mathcal{I}_{\mathsf{MI}}^{RB}(\mathcal{B} \cup \{b\}) = 1 = \mathcal{I}_{\mathsf{MI}}^{RB}(\mathcal{B} \cup \{a \to b\})$.
- We now consider $\mathcal{I}_c^{RB}$ and RC. Assume $\mathcal{I}_c^{RB}(\mathcal{B}) = 0$. Then there is a three-valued interpretation v with $v(a) = b$ for no $a \in \mathcal{A}$ and

  $$v \models^3 \bigcup_{M \in \mathsf{MI}^{\setminus\mathcal{F}}(\mathcal{B})} M$$

  Assume $\mathsf{MI}^{\setminus\mathcal{F}}(\mathcal{B}) \neq \emptyset$ and let $M \in \mathsf{MI}^{\setminus\mathcal{F}}(\mathcal{B})$. Then $v \models^3 \alpha$ for each $\alpha \in M$ and as $v^{-1}(b) = \emptyset$, $v(\alpha) = t$. This is a contradiction as M is inconsistent. It follows $\mathsf{MI}^{\setminus\mathcal{F}}(\mathcal{B}) = \emptyset$ and therefore for all consistent sets $F' \subseteq \mathcal{F}(\mathcal{B})$, $\mathcal{R}(\mathcal{B}) \cup F'$ is consistent. On the other hand, if for all consistent sets $F' \subseteq \mathcal{F}(\mathcal{B})$, $\mathcal{R}(\mathcal{B}) \cup F'$ is consistent then $\mathsf{MI}^{\setminus\mathcal{F}}(\mathcal{B}) = \emptyset$ and v can be defined via $v(a) = t$ for all $a \in \mathcal{A}$ showing $\mathcal{I}_c^{RB}(\mathcal{B}) = 0$.
  FE follows directly by definition and Equation (2).
  For RE consider $\mathcal{B} = \{a; \neg c; b \to c\}$. Then $a \to b \notin \mathcal{B}$, $a \to b \notin \mathsf{Free}(\mathcal{B} \cup \{a \to b\})$ but $\mathcal{I}_c^{RB}(\mathcal{B} \cup \{b\}) = 1 = \mathcal{I}_c^{RB}(\mathcal{B} \cup \{a \to b\})$.

□

Proposition 17. $\mathcal{I}_p^{RB}$ satisfies RC, FE and RE.

Proof. The postulates RC and FE follow from a similar reasoning as for $\mathcal{I}_{\mathsf{MI}}^{RB}$ in the proof of Proposition 17. For RE, let $B \to H \notin \mathcal{B}$ and $B \to H \notin \mathsf{Free}(\mathcal{B} \cup \{B \to H\})$. First observe that $\mathsf{MI}^{\setminus\mathcal{F}}(\mathcal{B}) \subseteq \mathsf{MI}^{\setminus\mathcal{F}}(\mathcal{B} \cup \{H \to B\})$. Consider then some $M \in \mathsf{MI}^{\setminus\mathcal{F}}(\mathcal{B} \cup \{H\}) \setminus \mathsf{MI}^{\setminus\mathcal{F}}(\mathcal{B})$. It follows $H \in M$. As $B \to H \notin \mathsf{Free}(\mathcal{B} \cup \{B \to H\})$ it follows that there is $X \subseteq \mathcal{B}$ such that $H \in \min(X \cup \{B \to H\})$ and the set of facts appearing in X is consistent (otherwise $B \to H$ could not be part of any $N \in \mathsf{MI}^{\setminus\mathcal{F}}(\mathcal{B} \cup \{B \to H)\})$. Then $M' = M \setminus \{H\} \cup X \cup \{B \to H\}$ is inconsistent. Let $M'' \subseteq M'$ be minimal inconsistent. It follows $M'' \in \mathsf{MI}^{\setminus\mathcal{F}}(\mathcal{B} \cup \{B \to H)\}$. This means that every $M \in \mathsf{MI}^{\setminus\mathcal{F}}(\mathcal{B} \cup \{H\}) \setminus \mathsf{MI}^{\setminus\mathcal{F}}(\mathcal{B})$ can be transformed to an $M'' \in \mathsf{MI}^{\setminus\mathcal{F}}(\mathcal{B} \cup \{B \to H)\}$ by replacing H with the rule $B \to H$ and further rules and facts to derive B. As this replacement introduces (at least) the rule $B \to H$ we get $\mathcal{I}_p^{RB}(\mathcal{B} \cup \{B \to H\}) > \mathcal{I}_p^{RB}(\mathcal{B} \cup \{H\})$. □

Proposition 26. The adjusted Shapley value satisfies Distribution, Minimality and Fact-Minimality.

Proof. We adress the three parts individually.

- For showing Distribution, let us rewrite the Definition of the adjusted Shapley inconsistency value as

  $$S*_{\alpha}^{\mathcal{I}^{RB}}(\mathcal{B}) = \begin{cases} 0 & \text{if } \alpha \in \mathcal{F}(\mathcal{B}) \\ \sum_{B \subseteq \mathcal{B}} CoalPayoff_{\alpha,\mathcal{B}}^{\mathcal{I}^{RB}}(B) + \sum_{B \subseteq \mathcal{B}} AddPayoff_{\alpha,\mathcal{B}}^{\mathcal{I}^{RB}}(B) & \text{otherwise} \end{cases}$$

  We are now interested in the sum of all adjusted Shapley values (for all elements of a rule base $\mathcal{B}$).

  $$\begin{aligned} \sum_{\alpha \in \mathcal{B}} S*_{\alpha}^{\mathcal{I}^{RB}}(\mathcal{B}) &= \sum_{\alpha \in \mathcal{B}} \begin{cases} 0 & \text{if } \alpha \in \mathcal{F}(\mathcal{B}) \\ \sum_{B \subseteq \mathcal{B}} CoalPayoff_{\alpha,\mathcal{B}}^{\mathcal{I}^{RB}}(B) + \sum_{B \subseteq \mathcal{B}} AddPayoff_{\alpha}^{\mathcal{I}^{RB}}(B) & \text{otherwise} \end{cases} \\ &= \sum_{\alpha \in \mathcal{B}} \sum_{B \subseteq \mathcal{B}} CoalPayoff_{\alpha,\mathcal{B}}^{\mathcal{I}^{RB}}(B) - \sum_{f \in \mathcal{F}(\mathcal{B})} \sum_{B \subseteq \mathcal{B}} CoalPayoff_{f,\mathcal{B}}^{\mathcal{I}^{RB}}(B) + \sum_{\alpha \in \mathcal{R}(\mathcal{B})} \sum_{B \subseteq \mathcal{B}} AddPayoff_{\alpha,\mathcal{B}}^{\mathcal{I}^{RB}}(B) \end{aligned}$$

  Following [11], the first summand can be rewritten.

  $$= \mathcal{I}^{RB}(\mathcal{B}) - \sum_{f \in \mathcal{F}(\mathcal{B})} \sum_{B \subseteq \mathcal{B}} CoalPayoff_{f,\mathcal{B}}^{\mathcal{I}^{RB}}(B) + \sum_{\alpha \in \mathcal{R}(\mathcal{B})} \sum_{B \subseteq \mathcal{B}} AddPayoff_{\alpha,\mathcal{B}}^{\mathcal{I}^{RB}}(B)$$

Then

$$
\begin{aligned}
&= \mathcal{I}^{RB}(\mathcal{B}) - \sum_{f \in \mathcal{F}(\mathcal{B})} \sum_{B \subseteq \mathcal{B}} \mathit{CoalPayoff}^{\mathcal{I}^{RB}}_{f,\mathcal{B}}(B) + \sum_{\alpha \in \mathcal{R}(\mathcal{B})} \sum_{B \subseteq \mathcal{B}} \begin{cases} 0 & \text{if } r \in \mathsf{Free}(B) \\ \frac{\sum_{f \in \mathcal{F}(B)} \mathit{CoalPayoff}^{\mathcal{I}^{RB}}_{f,\mathcal{B}}(B)}{|r \in \mathcal{R}(B) \text{ s.t. } r \notin \mathsf{Free}(B)|} & \text{otherwise} \end{cases} \\
&= \mathcal{I}^{RB}(\mathcal{B}) - \sum_{f \in \mathcal{F}(\mathcal{B})} \sum_{B \subseteq \mathcal{B}} \mathit{CoalPayoff}^{\mathcal{I}^{RB}}_{f,\mathcal{B}}(B) + \sum_{r \in \mathsf{Free}(\mathcal{R}(\mathcal{B}))} \sum_{B \subseteq \mathcal{B}} 0 + \sum_{r \notin \mathsf{Free}(\mathcal{R}(\mathcal{B}))} \sum_{B \subseteq \mathcal{B}} \frac{\sum_{f \in \mathcal{F}(B)} \mathit{CoalPayoff}^{\mathcal{I}^{RB}}_{f,\mathcal{B}}(B)}{|r \in \mathcal{R}(B) \text{ s.t. } r \notin \mathsf{Free}(B)|} \\
&= \mathcal{I}^{RB}(\mathcal{B}) - \sum_{f \in \mathcal{F}(\mathcal{B})} \sum_{B \subseteq \mathcal{B}} \mathit{CoalPayoff}^{\mathcal{I}^{RB}}_{f,\mathcal{B}}(B) + \sum_{r \notin \mathsf{Free}(\mathcal{R}(\mathcal{B}))} \sum_{B \subseteq \mathcal{B}} \frac{\sum_{f \in \mathcal{F}(B)} \mathit{CoalPayoff}^{\mathcal{I}^{RB}}_{f,\mathcal{B}}(B)}{|r \in \mathcal{R}(B) \text{ s.t. } r \notin \mathsf{Free}(B)|} \\
&= \mathcal{I}^{RB}(\mathcal{B}) - \sum_{f \in \mathcal{F}(\mathcal{B})} \sum_{B \subseteq \mathcal{B}} \mathit{CoalPayoff}^{\mathcal{I}^{RB}}_{f,\mathcal{B}}(B) + \sum_{B \subseteq \mathcal{B}} \sum_{f \in \mathcal{F}(B)} \mathit{CoalPayoff}^{\mathcal{I}^{RB}}_{f,\mathcal{B}}(B) \\
&= \mathcal{I}^{RB}(\mathcal{B}) - \sum_{f \in \mathcal{F}(\mathcal{B})} \sum_{B \subseteq \mathcal{B}} \mathit{CoalPayoff}^{\mathcal{I}^{RB}}_{f,\mathcal{B}}(B) + \sum_{f \in \mathcal{F}(B)} \sum_{B \subseteq \mathcal{B}} \mathit{CoalPayoff}^{\mathcal{I}^{RB}}_{f,\mathcal{B}}(B) \\
&= \mathcal{I}^{RB}(\mathcal{B})
\end{aligned}
$$

- Minimality follows from a combination of Definitions 22, 21 and 24. Observe that any element in a rule base $\mathcal{B}$ is either a fact or a rule, thus we discuss this individually. For any fact f, $S *^{\mathcal{I}^{RB}}_{f}(\mathcal{B}) = 0$ via Definition 24. Thus we have that if a free formula α is a fact, $S *^{\mathcal{I}^{RB}}_{\alpha}(\mathcal{B}) = 0$. Continuing with rules, following the free-formula independence requirement, we have that if a rule r is a free-formula, then $\mathcal{I}^{RB}(\mathcal{B}) = \mathcal{I}^{RB}(\mathcal{B} \setminus r)$. Then we see that the last part $(\mathcal{I}^{RB}(\mathcal{B}) - \mathcal{I}^{RB}(\mathcal{B} \setminus r))$ in Definition 21 always equals out to zero for free formulas (This follows the original proof in [11]). Last, via Definition 22, we see that any rule which is a free formula automatically is assigned an additional payoff of zero. Concluding, the adjusted shapley value via Defintion 24 is zero for rules which a free formulas.
- Fact-minimality directly follows from Definition 24, i.e., facts are always assigned a value of zero.

□

Proposition 27. Let $\mathcal{I}^{RB}$ be a rule-based inconsistency measure, $\mathcal{B}$ be a rule base and $r \in \mathcal{R}(\mathcal{B})$.

- **Rule-Involvement.** If $\mathcal{I}^{RB}$ satisfies RC and $r \in \mathcal{B}$ is a non-free rule, then $S *^{\mathcal{I}^{RB}}_{r}(\mathcal{B}) > 0$.

Proof. A useful observation is that for a non-free formula α, all minimal inconsistent subsets that α is contained relate to coalitions, where removing α from that coalition resolves the inconsistency. Then we see that the last part $(\mathcal{I}^{RB}(\mathcal{B}) - \mathcal{I}^{RB}(\mathcal{B} \setminus r))$ in Definition 21 will be $\neq 0$ for those coalitions. Also, any non-free rule is at least part of one minimal inconsistent subset M_1. Thus, to show rule-involvement, we see that the smallest adjusted Shapley value a non-free rule can receive is $((m-1)!(n-m)!)/n!) + \sum_{f \in \mathcal{F}(M_1)}(((m-1)!(n-m)!)/n!)/(|r \in \mathcal{R}(M_1) \text{ s.t. } r \notin \mathsf{Free}(M_1)|))$, where m is the cardinality of M_1, and n is the cardinality of $\mathcal{B}$. □

Proposition 28. Let $\mathcal{I}^{RB}$ be a rule-based inconsistency measure, and $\mathcal{B}$ be a rule base. Then, following [11], we have:

- **Rule Consistency'.** $\hat{S} *^{\mathcal{I}^{RB}}(\mathcal{B}) = 0$ iff $\mathcal{B}$ is rule consistent.
- **Free formula independence'.** If $\mathcal{I}^{RB}$ satisfies IN and α is a free formula of $\mathcal{B} \cup \{\alpha\}$, then $\hat{S} *^{\mathcal{I}^{RB}}(\mathcal{B} \cup \{\alpha\}) = \hat{S} *^{\mathcal{I}^{RB}}(\mathcal{B})$.
- **Upper Bound.** $\hat{S} *^{\mathcal{I}^{RB}}(\mathcal{B}) \leq \mathcal{I}^{RB}(\mathcal{B})$.

Proof. To show Rule Consistency', observe that as dictated by Rule-Involvement, we have that $S *^{\mathcal{I}^{RB}}_{r}(\mathcal{B}) > 0$ for any non-free rule $r \in \mathcal{B}$. So, for an inconsistent rule base $\mathcal{B}'$, we have that $\hat{S} *^{\mathcal{I}^{RB}}(\mathcal{B}') > 0$. From the other side, a consistent rule base $\mathcal{B}$ only contains free formulas. Here, following minimality, we see that indeed $\hat{S} *^{\mathcal{I}^{RB}}(\mathcal{B}')$ would be 0. To show Free formula independence' observe that for free α and $\mathcal{I}^{RB}$ satisfying IN we have $\mathcal{I}^{RB}(B) = \mathcal{I}^{RB}(B \cup \{\alpha\})$. Then for arbitrary $\gamma \in \mathcal{B} \cup \{\alpha\}$, $B \subseteq \mathcal{B} \cup \{\alpha\}$, $b = |B|$, and $n = |\mathcal{B}|$ we have

$$\mathcal{I}^{RB}(B) - \mathcal{I}^{RB}(B \setminus \{\gamma\}) = \mathcal{I}^{RB}(B \cup \{\alpha\}) - \mathcal{I}^{RB}((B \cup \{\alpha\}) \setminus \{\gamma\})$$

and therefore

$$\begin{aligned}
&\mathit{CoalPayoff}^{\mathcal{I}^{RB}}_{\gamma,\mathcal{B}\cup\{\alpha\}}(B) + \mathit{CoalPayoff}^{\mathcal{I}^{RB}}_{\gamma,\mathcal{B}\cup\{\alpha\}}(B\cup\{\alpha\})\\
=&\frac{(b-1)!(n+1-b)!}{(n+1)!}(\mathcal{I}^{RB}(B)-\mathcal{I}^{RB}(B\setminus\{\gamma\}))+\frac{b!(n-b)!}{(n+1)!}(\mathcal{I}^{RB}(B\cup\{\alpha\})-\mathcal{I}^{RB}((B\cup\{\alpha\})\setminus\{\gamma\}))\\
=&(\frac{(b-1)!(n+1-b)!}{(n+1)!}+\frac{b!(n-b)!}{(n+1)!})(\mathcal{I}^{RB}(B)-\mathcal{I}^{RB}(B\setminus\{\gamma\}))\\
=&\frac{(b-1)!(n+1-b)!+b!(n-b)!}{(n+1)!}(\mathcal{I}^{RB}(B)-\mathcal{I}^{RB}(B\setminus\{\gamma\}))\\
=&\frac{(b-1)!(n-b)!(n+1-b+b)}{n!(n+1)}(\mathcal{I}^{RB}(B)-\mathcal{I}^{RB}(B\setminus\{\gamma\}))\\
=&\frac{(b-1)!(n-b)!}{n!}(\mathcal{I}^{RB}(B)-\mathcal{I}^{RB}(B\setminus\{\gamma\}))\\
=&\mathit{CoalPayoff}^{\mathcal{I}^{RB}}_{\gamma,\mathcal{B}}(B)
\end{aligned}$$

yielding $S *^{\mathcal{I}^{RB}}_{\gamma}(\mathcal{B}) = S *^{\mathcal{I}^{RB}}_{\gamma}(\mathcal{B}\cup\{\alpha\})$ for all γ. Also $S *^{\mathcal{I}^{RB}}_{\alpha}(\mathcal{B}\cup\{\alpha\}) = 0$ and therefore

$$\begin{aligned}
\hat{S} *^{\mathcal{I}^{RB}}(\mathcal{B}\cup\{\alpha\}) &= \max_{\gamma\in\mathcal{B}\cup\{\alpha\}} S *^{\mathcal{I}^{RB}}_{\gamma}(\mathcal{B}\cup\{\alpha\})\\
&= \max_{\gamma\in\mathcal{B}} S *^{\mathcal{I}^{RB}}_{\gamma}(\mathcal{B})\\
&= \hat{S} *^{\mathcal{I}^{RB}}(\mathcal{B})
\end{aligned}$$

Last, Upper Bound follows directly from Distribution, i.e., considering a knowledge base $\mathcal{B}$ and a rule-based inconsistency measure $\mathcal{I}^{RB}$, we see that the sum of $S *^{\mathcal{I}^{RB}}_{\alpha}(\mathcal{B})$ over all $\alpha \in \mathcal{B}$ equals $\mathcal{I}^{RB}(\mathcal{B})$. In turn, for any $\alpha_i \in \mathcal{B}$, $S *^{\mathcal{I}^{RB}}_{\alpha_i}(\mathcal{B})$ cannot be larger than $\mathcal{I}^{RB}(\mathcal{B})$. $\square$

7.2 Effects of Visualization Techniques on Understanding Inconsistencies in Automated Decision-Making

Source	Effects of Visualization Techniques on Understanding Inconsistencies in Automated Decision-Making
Authors	Carl Corea, Sabine Nagel and Patrick Delfmann
Outlet	HICSS 2020 - 53rd Hawaii International Conference on System Sciences, January 8-10, 2020, Maui, USA
Ranking	A
Status	Published

Effects of Visualization Techniques on Understanding Inconsistencies in Automated Decision-Making

Carl Corea
Institute for IS Research
University of Koblenz-Landau
ccorea@uni-koblenz.de

Sabine Nagel
Institute for IS Research
University of Koblenz-Landau
snagel@uni-koblenz.de

Patrick Delfmann
Institute for IS Research
University of Koblenz-Landau
delfmann@uni-koblenz.de

Abstract

The automation of business processes and decision-making has received major interest from practice and academia. As automation allows us to execute more processes (cases), monitoring automated decision-making is currently evolving into a big data analytics problem for companies. Thus, not only monitoring insights themselves, but also an effective use of such insights become important. In this context, the speed and ability to interpret data is closely related to the visualization of metrics and data. While various approaches for quantitative insights on automated decision-making have been proposed, there is currently no evidence as to how the specific visualization of such metrics helps companies to create more value from their data. In this report, we therefore present the results of an empirical experiment analyzing the cognitive effects of different visualization techniques for quantitative insights on understanding inconsistencies in automated decision-making data.

1. Introduction

Business process automation is a central challenge for today's businesses [1, 25]. In regard to the current digital transformation, company activities have to be digitalized and automated in order to stay competitive [11]. Thus, major efforts are currently being directed towards executing business processes via workflow management systems (WFMS) [30]. Such systems can automatedly perform predefined business processes, e.g., retail or customer service processes, to increase efficiency and reduce the manual effort needed in company activities. A central challenge here is automated decision-making, to allow WFMS to handle processes autonomously.

However, monitoring correct and consistent decision-making in WFMS is currently evolving into a big data analytics problem for companies [6, 25]. This becomes apparent in the scope of the *7V model* [13, 19], which indicates the main attributes of big data. For example, the *volume* of data created by WFMS is increasing rapidly. The online retailer Zalando, which handles customer processes with a WFMS [25], reports that 31 million cases were executed by their system only in the first quarter of 2019, which is nearly a 50% increase since the first quarter of 2017[1]. Also, as company processes span across systems and organizations, process data often includes heterogeneous and unstructured data, such as scanned documents. Thus, data *variety* and *veracity* also become increasingly challenging for companies [7, 23, 25]. Last, the *velocity* of data creation increases through automation. WFMS track processes in real-time, leading to shorter periods in which companies have to monitor decision-making.

In order to ensure consistent decision-making, companies must understand and utilize process case data to detect errors in automated decisions [6]. Such an understanding can be an important driver in creating value through innovation, e.g., by minimizing mistakes, improving WFMS, and streamlining business processes [6, 12, 19]. To support companies in this aim, the scientific field of inconsistency measurement has evolved and proposed so-called inconsistency measures, which are *metrics* that can help to identify those cases where inconsistencies have occurred in the decision-making [6, 8]. Also, inconsistency measures provide quantitative insights that can help to assess the severity of inconsistency and thus prioritize cases for the analysis by experts [6, 12, 24].

Due to the unique big data challenges arising in the context of WFMS, not only these quantitative metrics themselves, but also an effective *use* becomes even more important [19, 27]. In this context, the *speed and ability to interpret data* is closely related to a dimension of the 7V model, namely the *visualization* of metrics and data.

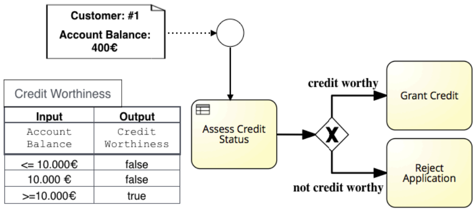
---

[1] https://zln.do/31wcTqz

While various approaches for inconsistency metrics have been proposed, there is currently no evidence as to how the specific *visualization* of such insights helps companies to create more *value* from their data. In this work, we thus investigate how visualization techniques for inconsistency metrics affect the capability to analyze and interpret inconsistencies in automated decision-making. To this aim, we introduce different visualization techniques for inconsistency metrics in section 2 and hypothesize the relation of these approaches and understanding inconsistencies in section 3. To verify our hypotheses, we conducted an empirical experiment using neurophysiological measurement, presented in sections 4-5. Our study suggests that providing users with a separate, ranked overview of metrics (cf. section 2) is associated with better understanding efficiency and less mental effort required to handle cases compared to other visualization techniques.

2. Background and Related Work

WFMS are systems that allow companies to integrate process- and decision logic, subsequently allowing them to execute business processes (semi-) automatically. If a new process (i.e., a case) is started, the WFMS conducts all tasks as predefined in the process model sequentially. During this traversal of the process model, decision-making is performed using business rules, which govern how the process should be executed based on the case data. Figure 1 shows an exemplary business process in the Business Process Model and Notation (BPMN)[2] and corresponding business rules in the Decision Model and Notation (DMN) standard[3].



Input	Output
Account Balance	Credit Worthiness
<= 10.000€	false
10.000 €	false
>=10.000€	true

Figure 1. Exemplary business process and business rules

DMN allows formalizing rules with so-called decision tables. The rules in the shown table can be read such that "*if*" the account balance is <= 10.000, "*then*" the customer is not credit worthy, and so forth. In Figure 1, given the shown case input, we thus see that the system can automatedly reject the request, reducing manual effort in decision-making.

However, business rules are created by human modelers, mostly collaboratively and incrementally [16, 17]. As suggested by a wealth of recent research, modeling errors can frequently occur in this setting [2, 3, 6, 24, 26]. For example, in a recent case study with a large insurance company, the authors in [2] found that 27% of rules had modeling errors.

In turn, such modeling errors can result in inconsistent decision-making [6]. For example, considering the business rules in Figure 1, the customer input of 10.000€ would result in contradictory conclusions due to erroneous modeling. Here, all rule conditions overlap, yielding contradicting conclusions. This flawed decision-making could thus result in compliance breaches, as the WFMS could perform unallowed activities. It is therefore essential for companies to monitor consistent decision-making during process execution as a driver for innovative and sustainable development of WFMS, e.g., by re-modeling business rules and improving operations [6, 10, 24].

Implementing such compliance monitoring is, however, becoming more and more difficult for companies due to the increasing amounts of data which are generated by WFMS [25]. Such data amounts are challenging in the scope of gaining business intelligence insights, as well as creating value from data and insights. For example, during compliance monitoring, where a case can consist of heterogeneous or even unstructured data, experts must analyze the entire case to correctly understand and resolve modeling errors.

To support companies in this aim, inconsistency metrics have been proposed, providing insights on inconsistencies in decision-making during run-time. Examples include [6] for an overview and [8, 29] for some recent surveys. An important family of such metrics are *culpability measures*, which aim to quantify the severity of inconsistency for *individual* business rules with a *numerical value*. The intuition is that a higher value reflects a higher severity of inconsistency. An exemplary culpability measure is the $C_{\#}$ measure [6], which assesses the culpability of an individual rule r by counting the number of other rules that contradict the respective rule r.

Example 1. *Revisiting the business rules from Figure 1 and a case input of 10.000€, this would mean that rule 3 would have a C#-value of 2, as the conclusion of rule 3 conflicts the two other rules. Respectively, rules 1 and 2 would each have a C#-value of 1, as they both individually contradict rule 3.*

---

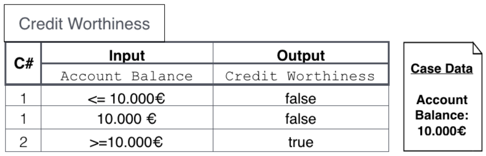
[2] https://www.omg.org/spec/BPMN/2.0/

[3] https://www.omg.org/spec/DMN/

Thus, culpability measures can pinpoint the actual causes of inconsistencies in WFMS cases as a driver for resolving modeling errors [6]. Nagel et al. (2019) could show that such inconsistency metrics are associated with a better understanding of inconsistencies. In light of the unique big data challenges in WFMS, not only these (quantitative) insights themselves but also the *speed of their use* become important to foster the innovative and sustainable development of WFMS [19]. Thus, the actual visualization of metrics becomes a critical success factor in creating value from case data, as it can impact how information can be analyzed and interpreted.

So far, there are two different visualization approaches for inconsistency metrics that have been proposed, namely an *integrated* visualization and a *ranked overview* visualization [6].

In the integrated approach, the culpability values are displayed directly *within* the respective decision table, i.e., next to the individual business rules. Figure 2 shows an example of such an *integrated* visualization for Example 1.



Credit Worthiness

C#	Input	Output
	Account Balance	Credit Worthiness
1	<= 10.000€	false
1	10.000 €	false
2	>=10.000€	true

Case Data

Account Balance: 10.000€

Figure 2. Integrated visualization

On the other hand, Corea et al. (2018) propose a separate *ranked overview* of inconsistency metrics. Instead of integrating the respective culpability values directly in the table, a reference (e.g., a colored dot) is used to point to a *separate*, *ranked* overview, where all rules are presented in a sorted list. Figure 3 shows such a ranked visualization for Example 1. Due to the ranking, it can be directly seen that rule 3 is the most problematic one and should be attended to first. Intuitively, such a ranked overview can provide benefits for larger tables, or if problems are distributed across different tables.

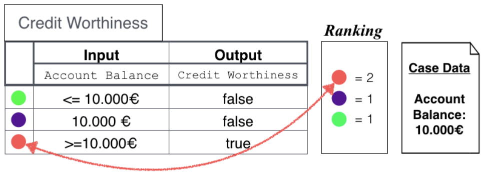


Figure 3. Ranking-based visualization

Regarding which of these visualization techniques is “better” to display big data insights, both approaches have advantages and disadvantages, and there are several contradicting aspects from the field of cognitive psychology that must be considered. An important aspect here is *cognitive load theory*, which describes the relation between cognitive load during information processing and the performance of understanding information [31].

In the *integrated* approach, the metric information and the rules are visually presented in a unified manner. This can lower the extraneous cognitive load for processing the information, due to a minimization of the so-called split-attention effect [31, 32, 33]. This effect can occur when two related pieces of information are visually distributed and need to be linked mentally by the expert, which is not the case in the integrated approach.

However, while this would advocate an integrated visualization, this approach can lead to a higher amount of case information which needs to be processed. It is important to realize that business rules are usually stored in multiple tables in practice. Thus, all tables need to be checked, and it is not sufficient to “simply” sort rules by their culpability values in the respective tables in an integrated visualization.

On the contrary, a *ranked overview* allows the expert to quickly comprehend which rules (and which decision tables) should be attended to first. Such a *recommendation* can thus guide modelers in handling cases and lower the amount of information that has to be processed. Also, results from cognitive psychology suggest that factors such as coherence or elaborative encoding of information can promote more efficient processing of information [20, 28].

Yet again, while this would advocate using a ranked visualization, the resulting split of information can be expected to introduce more cognitive load during information assimilation [31, 33]. As both proposed approaches have advantages and disadvantages, a further investigation based on literature is difficult at this point. Currently, there exists no *empirical* evidence investigating which of these visualization techniques helps companies to create more value from their data.

3. Research Aim

Following authors such as Surbakti et al. (2019) or Olszak & Zurada (2019), the effectiveness and speed of using insights into big data is a key factor in creating value for companies. As these abilities are closely linked to how information can be processed, this work aims at investigating how different visualization techniques for quantitative insights affect the effectiveness of their use and thus, also help companies to create more value from their data.

Accordingly, we derive the following research question:

RQ: How do visualization techniques for inconsistency metrics affect the capability to analyze and interpret inconsistencies in automated decision-making?

As introduced in the previous section, two visualization approaches have been proposed so far, namely an *integrated* approach and a *ranking-based* approach. Here, the ranking-based approach has been introduced more recently and makes use of culpability measures to present users with a recommendation. As a result, especially in real-life cases where experts would have to scan multiple and large tables, such a recommendation could potentially increase *efficiency* by guiding modelers. Accordingly, our first aim is to investigate how a ranking-based visualization affects understanding efficiency, i.e., the time needed for modelers to understand inconsistencies based on inconsistency metrics.

Hypothesis 1: A ranking-based visualization of inconsistency metrics is associated with a better understanding efficiency compared to an integrated visualization.

Also, while an integrated approach could potentially lower cognitive load due to a reduction of the split-attention effect, experts might struggle to gain a more holistic oversight of problems. Here, a ranked overview could potentially lower the mental effort needed to understand inconsistencies, as the ranking can be used to present a holistic prioritization of problems. Accordingly, our second aim is to investigate how a ranking-based visualization affects the objective mental effort needed for understanding inconsistencies.

Hypothesis 2a: A ranking-based visualization of inconsistency metrics is associated with less objective mental effort compared to an integrated visualization.

Next to the objective mental effort, the visualization might also affect the perceived ease of use, which could be used as a driver for the development of visualization techniques. Therefore, our third aim is to investigate the effects of visualization on perceived ease of use, i.e., the perceived mental effort needed to understand inconsistencies.

Hypothesis 2b: A ranking-based visualization of inconsistency metrics is associated with less perceived mental effort compared to an integrated visualization.

Despite the potential advantages of the ranking-based visualization, it is currently not clear whether the advantages of a ranked overview outweigh the potential cognitive costs that can be expected due to the split-attention effect [33]. Thus, empirical evidence is needed. Consequently, we opted for an experimental research approach to test these hypothesized relations.

We consequently follow the experimental research methodology as proposed by Neuman [18], as it is highly suitable for the investigation of causal relations between independent variables and their effects. Here, independent variables are manipulated in a controlled environment in order to assess the effects that follow the manipulation. We thus see this research methodology as highly appropriate, as the visualization techniques can be seen as the independent variables which the researcher can manipulate, and the methodology thus allows us to assess the effects of these techniques. The dimensions of understanding efficiency and mental effort are based on the experiment design in [32].

4. Experiment

We conducted an empirical experiment[4] to test our hypothesis. In the following, we describe our experiment design and the measures used to verify our hypotheses.

4.1. Experiment Design

In order to empirically evaluate the effects of the two approaches of displaying insights into WFMS data, we confronted participants with cases and corresponding questions covering inconsistencies.

We designed our experiment as a single factor experiment, as this allows us to assess the effects of a single factor on a shared variable [22]. In our case, the considered factor is the visualization approach of the inconsistency values, with the factor levels being “integrated-” and “ranking-based” visualization.

The experiment was split into two runs (i.e., models), which each represent a separate domain. We divided our participants into two groups randomly and tested them for both domains. Here, each domain was tested with different factor configurations for the

---

[4] The experiment can be downloaded from https://bit.ly/2KfsBAU

two groups, one seeing the inconsistency metrics directly in the table and the other one having them visualized in a separate ranking. As a result of this, the experiment is balanced with repeated measurement. This has the advantage of the participants using both factor levels without being exposed to the same cases twice. This repeated measurement thus allows each participant to generate more data, which can lead to more precise, powerful results [4] and, therefore, increases generalizability. As the order of factor levels was reversed between groups, we could also counterbalance a potential distortion following [32].

Figure 4 gives an overview of the overall experiment design. In the first run, which contains tasks 1-4, Group 1 was exposed to the inconsistency metrics being shown directly in the tables (integrated visualization), while group 2 had them visualized in a separate ranking (ranking-based visualization). This situation was inverted in the second run, where Group 1 was provided with the inconsistency metrics in a separate ranking, while they were displayed directly in the tables for Group 2.

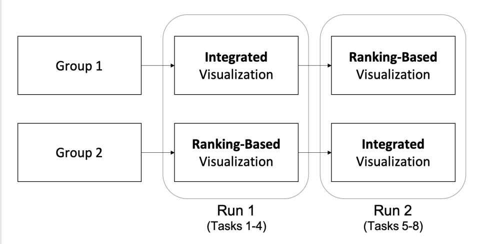


Figure 4. Experiment design

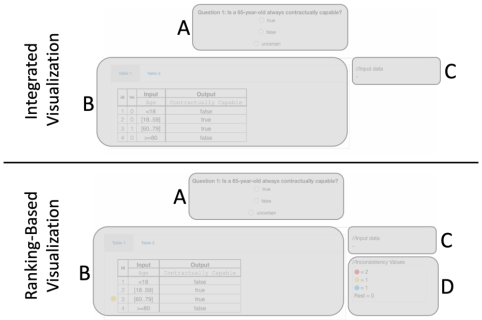
The entire experiment and the questions were formulated in English to warrant comprehension in the scope of reproducible research.

4.2. Experiment Structure & Instrumentation

Before exposing the participants to actual cases and corresponding questions, we presented them with an introduction to the experiment. These introductory slides covered all topics that were needed to be able to perform the tasks, such as the basics of decision management and DMN tables, as well as the quantitative measures and their two visualization approaches used in the experiment. The tutorial also covered an exemplary task to ensure that the participants became familiar with the quantitative measures and their use. The participants were able to go through the introductory slides at their own speed, as an understanding of the used concepts was crucial for the success of the experiment.

As shown in Figure 4, each run comprised four different tasks, each containing a case and a corresponding question. We designed the cases to cover basic types of inconsistencies in WFMS cases; however, there were no dependencies between cases.

In general, each task was divided into three, resp. four, areas as illustrated in Figure 5, which shows the structure of an exemplary task for both factor levels.



Question 1: Is a 65-year-old always contractually capable?
○ true
○ false
○ uncertain

//Input data
-

id	Val	Input	Output
		Age	Contractually Capable
1	0	<18	false
2	0	[18..59]	true
3	1	[60..79]	true
4	0	>=80	false

Question 1: Is a 65-year-old always contractually capable?
○ true
○ false
○ uncertain

//Input data
-

id	Input	Output
	Age	Contractually Capable
1	<18	false
2	[18..59]	true
● 3	[60..79]	true
4	>=80	false

//Inconsistency Values
● = 2
● = 1
● = 1
Rest = 0

Figure 5. Exemplary task structure

The question and the corresponding response options are displayed at the top of the screen (A). Below, the current case is shown, which is divided into two areas. On the left, the first DMN table is shown (B). If the case consists of multiple tables, the user can switch between them using the tabs displayed above the table. The different tables in one case may be directly related, e.g., inconsistencies can be distributed among multiple tables. However, there is also the possibility of a table being irrelevant to the current question. Furthermore, the participant was shown case data, which is needed to answer the questions. The case data can either be implied by the question or listed in the corresponding box (C). Depending on the current factor level, the inconsistency values are either displayed to the right of the row ID, or in a separate, ordered ranking (D).

There were four different types of questions:

- Questions that asked about the existence of inconsistencies in the current case (e.g., "Is there an inconsistency in James' case?")
- Content-related questions (e.g., "Is the person in this case contractually capable?"). The questions could either be answered with "true", "false", or – if no conclusion could be made due to an existing inconsistency – "uncertain".
- Questions that asked for the rule with the highest/lowest number of inconsistencies (e.g., "Which rule is in conflict with the highest number of other rules for this case?").

- Questions that asked for a number of rules that contradict a specific number of other rules (e.g., “How many rules are in conflict with EXACTLY three other rules in this case?”)

4.3. Measurements

To test our hypotheses, we used three different types of measurements.

Understanding efficiency was measured using the time from the point a case was displayed until the participant clicked on an answer. Next, we used the eye-fixation duration to measure objective mental effort. The eye-fixation duration is defined as the period of time where the eyes are focused on a specific location and therefore remain still. As vision is suppressed as soon as the eyes are moved, new information can only be captured during fixation [21]. Thus, fixation duration is an indication for the time it takes a participant to process the respective information. In turn, a higher fixation duration indicates a higher objective measure for mental effort [14]. In addition to the objective measures, we also asked the participants which run, i.e., which visualization approach they found easier to work with, in order to measure the perceived mental effort. We also conducted semi-structured interviews with all participants after the experiment, to gain further insights into the reasons behind their choices and their general impressions.

4.4. Settings

We used the cloud-based eye tracking software EYEVIDO lab[5] to create our study. While the introductory slides were added as pdf files, we implemented the tasks as HTML files. We also verified the clear visibility of all task components from a distance of at least 60cm during a pre-test (c.f. a further description of the pre-test in section 4.5.)

As already depicted in section 4.2, the screen was divided into up to four parts: the question area at the top of the screen and the corresponding case as well as the inconsistency metrics below. As most cases consisted of more than one table, the participants were able to click on the corresponding tab to switch between tables when needed. Furthermore, some tables required scrolling as their size prevented them from being displayed in full size.

We used a *myGaze n 30Hz* eye tracker in combination with a 22-inch screen and a resolution of 1680 x 1050. The experiment was conducted in an IT lab at the University of Koblenz-Landau. To keep the lighting situation consistent, the blinds were closed, and the ceiling light was the only source of light.

4.5. Participants

In order to ensure the understandability and readability of the introductory slides and experiment questions, we conducted a pre-test with 8 Ph.D. students prior to the actual experiment. During the pretest, we found that one major problem was the lack of trust in the inconsistency metrics, as many participants preferred to double-check the implication of the quantitative insights manually. As a result of this, we refined the introductory slides in order to build more trust in the values, as our focus was an investigation of different visualization approaches, and the general advantage of the values themselves has already been evaluated in [15].

A total of *48* undergraduate and graduate students from the school of Computer Science at the University of Koblenz-Landau participated in our experiment. Please observe that this school does not only include traditional computer science but also includes institutes focusing on business informatics or business administration. Thus, the participants represented diverse fields. Although the introductory covered all relevant concepts, all participants had a general knowledge of business process management based on their study programs.

The assignment of the 48 students into the two groups was performed at random. Furthermore, no incentive was offered, so participation was voluntary.

5. Results

To test our hypotheses, we assessed the experiment data with the measures described in section 4.3. Given that our experiment was set up as a between-subject design experiment, we statistically compared the respective measures of the individual groups. Here, we proceeded as follows.

First, we checked if each dependent variable could be assumed to be normally distributed, using the Shapiro-Wilcox test at a significance level of 0.05. Then, if the data could be assumed to be normally distributed, we verified whether the dependent variables had an equal variance, using Levene’s test[6] at a significance level of 0.05. Given a normal distribution and equal variance, we ran an independent-sample t-test. If data could not be assumed to be

---

[5] https://www.eyevido.de

[6] Levene’s test is used to analyze the variance for a variable measured for at least two groups

normally distributed, we used the Wilcoxon-Mann-Whitney test[7]. For all comparison tests, we assumed the commonly used significance level of 0.05.

For Hypothesis 1, the measured time needed to answer the questions was normally distributed and met the assumption of equal variance for both models. We consequently ran an independent-sample t-test between groups 1 and 2 for both models, using the time needed to answer all four questions in one run as the dependent variable.

Table 1. Test of Hypothesis 1 (understanding efficiency)

	Run 1		**Run 2**	
Group	1 (*integrated*)	2 (*ranked*)	1 (*r*)	2 (*i*)
N	24	24	24	24
Mean	34.54	29.04	23.09	25.02
SD	8.71	12.89	8.46	16.24
p (1-tailed)	**0.048**		0.30	

As can be seen in Table 1, the group using a ranked overview had a significantly better understanding efficiency in run 1, which partially supports our hypotheses. For run 2, the difference was not significant, yet the group using the ranked visualization was still better on average.

Conclusion 1: For the measured data, a ranking-based visualization of inconsistency metrics is (partially) associated with a better understanding efficiency compared to an integrated visualization.

For Hypothesis 2a, the eye-fixation duration measured during the experiments was also normally distributed and met the assumption of equal variance for both models. Accordingly, we ran an independent sample t-test, using the eye-fixation duration needed for answering all four questions in one run as the dependent variable. For 6 participants, eye-tracking data could not be tracked due to an unknown system failure, thus N = 42 for comparing fixation duration.

Table 2. Test of Hypothesis 2a (objective mental effort)

	Run 1		**Run 2**	
Group	1 (*i*)	2 (*r*)	1 (*r*)	2 (*i*)
N	21	21	21	21
Mean	30.95	22.86	19.58	19.78
SD	16.22	23.73	15.22	26.97
p (1-tailed)	**0.004**		0.47	

As depicted in Table 2, the group using a ranked overview had a significantly lower eye-fixation duration in model 1, which partially supports Hypothesis 2a. For model 2, the difference was again not significant, yet the eye-fixation duration was still lower on average for the group using a ranked overview.

Conclusion 2: For the measured data, a ranking-based visualization of inconsistency metrics is (partially) associated with less objective mental effort compared to an integrated visualization.

Table 3 shows the results of the perceived mental effort, as provided by the participants.

Table 3. Perception of mental effort

	Group 1	**Group 2**	**Total**
Ranking is easier	14	11	25 (52%)
Integration is easier	5	9	14 (29%)
Equal	5	4	9 (18%)

To recall, both groups performed two runs and were exposed to both visualization techniques. Overall, the ranking-based overview was perceived as easier by the majority of participants. To statistically compare the perceived mental effort, we coded the answers by participants as follows: If the participant selected that the ranking based visualization was easier, the ranking based visualization was assigned 2 points. Vice versa, if the participant selected that the integrated visualization was easier, the integrated visualization was assigned 2 points. If a participant indicated that both visualization techniques were perceived to require equal mental effort, both the ranking-based and integrated visualization were assigned 1 point. We then compared the average perceived mental effort between the two visualization forms. As the coded data is ordinal (i.e., possible answers were 0, 1 or 2), the data could not be assumed to be normally distributed. Since for the perceived mental effort *every* participant provided his or her answer based on *both* models, we ran the Wilcoxon signed-rank test[8] for the differences in the perceived mental effort for both visualizations.

Table 4. Test of Hypothesis 2b (perceived mental effort)

	N	**Coded mean**	**SD**	**P (1-tailed)**
Ranking	48	1.22	0.88	**0.039**
Integrated	48	0.77	0.88	

---

[7] The Wilcoxon-Mann-Whitney test is used to compare the distribution data from two samples

[8] The Wilcoxon signed-ranked test is used for dependent samples

The perceived mental effort for the ranking-based visualization was significantly lower than for the integrated visualization, which supports our hypothesis.

Conclusion 3: For the measured data, a ranking-based visualization of inconsistency metrics is associated with less perceived mental effort compared to an integrated visualization.

6. Discussion

For run 1, our results fully support Hypotheses 1 and 2a, which suggests that the ranked overview visualization was associated with a better understanding efficiency and less objective mental effort needed for handling cases in this run. In run 2, the difference in group performances was not large enough to be statistically significant. Still, the group which used the ranking-based visualization had on average a better understanding efficiency and less mental effort than the group using the integrated visualization in run 2. We assume that the performances could have been affected by participants handling similar cases in run 1 and run 2, i.e., a learning effect could have affected the measured performance in run 2. However, the similar structure of run 1 and run 2 was necessary to mitigate a bias based on different questions. Intuitively, this would not be the case in practice, where experts face far more complex and individual cases. Thus, based on the statistical significance in run 1 and the fact that participants using a ranking-based visualization in run 2 were still on average better, we cannot reject our hypotheses. Also, our results fully support Hypothesis 2b (which assessed both runs), indicating that perceived mental effort was significantly lower for the ranking-based visualization, compared to an integrated visualization of metrics.

Figure 6 shows the distribution of average *time* needed to complete a run, as well as the corresponding average *fixation duration*.

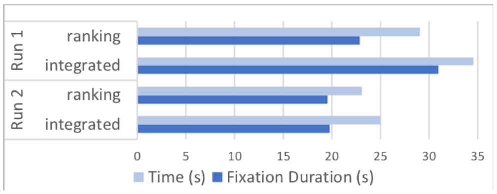


Figure 6. Distribution of time and fixation duration (units in seconds)

As can be seen, participants using a ranking-based visualization took less time and had a lower fixation duration in both runs. Especially in run 1, the proportion of fixation duration relative to total time needed was also much higher for the integrated visualization (89% vs. 79%). Thus, the participants spent a higher fraction of time trying to understand information for the integrated visualization as opposed to the ranking-based visualization.

As mentioned, a split of information when using a separate ranking can increase the extraneous cognitive load during information assimilation [31, 32, 33]. Yet, in an integrated approach, more information has to be processed due to a lack of guidance. This can be visualized using the collected eye tracking data. Figure 7 shows a heatmap of the participants' gaze distribution. The colored areas indicate areas of visual focus, where the red color ("heat") represents a higher fixation duration. As can be seen, participants tried to process a higher amount of information in the integrated approach. Thus, the benefits of having to process less information seem to outweigh the cognitive costs of information assimilation (split-attention effect).

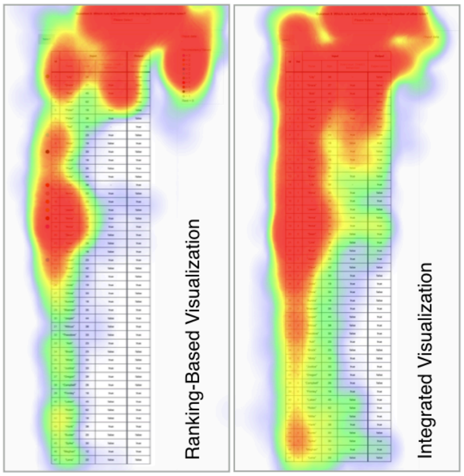


Figure 7. Heatmaps of gaze distribution for different visualization techniques

In the post-experiment interviews with the participants, we were also able to gain further insights into the reasons for the perceived mental effort and their general impression regarding the different visualization approaches. Most participants stated that they felt relieved that the ranking allowed them to find important information more efficiently. For example, participants described that they could focus on specific rows more effectively and were therefore often able to skip irrelevant parts or even entire tables when having access to the ranking.

Interestingly, some participants stated that despite being sure about understanding the provided metric information, they did not feel confident in using it right away. This resulted in the participants trying to re-verify the metric assessment manually. However, the participants stated an increase of trust during the course of the experiment. Although the general trust in the inconsistency metrics seems to have improved compared to the pre-test (due to the refinement of the introductory slides), trust in the insights seems to be an important issue when creating value from big data.

7. Conclusion

In this work, we investigated the relation of visualization techniques on understanding inconsistencies in automated decision-making. To this aim, we conducted and evaluated an empirical experiment using neurophysiological measurement. Our results indicate that a ranking-based visualization was (partially) associated with a better understanding efficiency and lower mental effort in handling inconsistent cases.

Our empirical insights contribute to the development of future WFMS and tools. Recommender systems and elaborate rankings based on quantitative metrics should be developed, as the benefits of having to process less information seem to outweigh the cognitive costs of information assimilation (split-attention effect). Also, this study affirms that the specific visualization of big data insights can help companies to use these insights more effectively. This capability thus represents an added value of big data and business intelligence in regard to innovation, i.e., improving WFMS or streamlining processes.

In the scope of the 7V model, the visualization component is often overlooked. More research is needed that investigates how the visualization of big data insights can help to foster innovation and help companies to create value from their data.

Due to the research methodology, our study is not without limitations. Intuitively, the results are dependent on the participant composition (e.g., number of participants or personality traits) which could affect external validity. Our results are based on experiment data from *48* participants, which we see as comparable to other related studies in the field, e.g. [22] (*19* participants), [14] (*23* participants), [9] (*28* participants), [32] (*50* participants) or [5] (*75* participants). Also, English was not the native language of the participants, which could affect internal validity. However, all participants indicated a sufficient understanding of the English language as a precondition for participating in the experiment. A further limitation is that the participant group comprised students, as opposed to domain experts, which could also affect external validity. However, the study curricula include highly practice-based and related subjects, such as business process management or business administration. Also, as the cases included basic inconsistencies, a transferability to other domains seems plausible. In future work, we aim to conduct experiments with professionals and larger sample sizes.

Bearing these limitations in mind, we could, however, observe that participants using a ranking-based visualization of data insights could on average handle cases in a lower amount of time and with less mental effort needed. Therefore, our findings can help companies to create more value from their data by using these insights on visualization effects to guide the future development of WFMS and monitoring tools.

Future work should investigate more elaborate rankings and recommendation systems. For example, participants indicated that switching between the ranking and information in the tables, such as the row number, was perceived as difficult. Here, future work should investigate how to integrate more information into the ranking and exploit amenities of the integrated approach to develop advanced rankings or hybrid approaches. Also, some participants suggested providing additional guidance through further visual elements in the ranking, e.g., more elaborate or graphical encodings.

Our results affirm that the visualization of big data insights is a key factor in creating value from data for companies. The effects of visualization techniques should more strongly be included as a driver for the development of information systems. In this way, specific visualization techniques can be used to foster economic value by means of helping companies to understand their data, thus enabling innovation through improved decision-making and more sustainable WFMS management.

8. Acknowledgements

This research is part of the research project “Handling Inconsistencies in Business Process Modeling”, which is funded by the German Research Association (reference number: DE1983/9-1). We thank Wei Wang, Marta Indulska, Shazia Sadiq and Barbara Weber for inspiration to our research design.

9. References

[1] Aalst, W.M.P. van der, M.L. Rosa, and F.M. Santoro, “Business Process Management”, *Business & Information Systems Engineering 58*(1), 2016, pp. 1–6.

[2] Batoulis, K., and A. Nesterenko, “Decision Management in the Insurance Industry: Standards and Tools”, *BPM*

2017 Industry Track co-located with the 15th International Conference on Business Process Management, Springer International Publishing (2017), 11.

[3] Calvanese, D., M. Dumas, Ü. Laurson, F.M. Maggi, M. Montali, and I. Teinemaa, "Semantics, analysis and simplification of DMN decision tables", *Information Systems 78*, 2018, pp. 112–125.

[4] Charness, G., U. Gneezy, and M.A. Kuhn, "Experimental methods: Between-subject and within-subject design", *Journal of Economic Behavior & Organization 81*(1), 2012, pp. 1–8.

[5] Chen, T., W. Wang, M. Indulska, and S. Sadiq, "Business Process and Rule Integration Approaches-An Empirical Analysis", *International Conference on Business Process Management*, Springer (2018), 37–52.

[6] Corea, C., and P. Delfmann, "Supporting Business Rule Management with Inconsistency Analysis", *Proceedings of the BPM*, (2018), 09–14.

[7] Goes, P.B., "Big data and IS research", *Mis Quarterly 38*(3), 2014, pp. iii–viii.

[8] Grant, J., and M.V. Martinez, *Measuring Inconsistency in Information*, College Publications, 2018.

[9] Haji, F.A., D. Rojas, R. Childs, S. de Ribaupierre, and A. Dubrowski, "Measuring cognitive load: performance, mental effort and simulation task complexity", *Medical Education 49*(8), 2015, pp. 815–827.

[10] Hashmi, M., G. Governatori, H.-P. Lam, and M.T. Wynn, "Are we done with business process compliance: state of the art and challenges ahead", *Knowledge and Information Systems*, 2018, pp. 1–55.

[11] Imgrund, F., M. Fischer, C. Janiesch, and A. Winkelmann, "Approaching digitalization with business process management", *Proceedings of the MKWI*, 2018, pp. 1725–1736.

[12] Lu, R., S. Sadiq, and G. Governatori, "Measurement of Compliance Distance in Business Processes", *Information Systems Management 25*(4), 2008, pp. 344–355.

[13] Manyika, J., M. Chui, B. Brown, et al., *Big data: The next frontier for innovation, competition, and productivity*, McKinsey Global Institute, 2011.

[14] Meghanathan, R.N., C. van Leeuwen, and A.R. Nikolaev, "Fixation duration surpasses pupil size as a measure of memory load in free viewing", *Frontiers in Human Neuroscience 8*, 2015.

[15] Nagel, S., C. Corea, and P. Delfmann, "Effects of Quantitative Measures on Understanding Inconsistencies in Business Rules", *Proceedings of the 52nd Hawaii International Conference on System Sciences*, (2019).

[16] Nelson, M.L., R.L. Rariden, and R. Sen, "A lifecycle approach towards business rules management", *Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS 2008)*, IEEE (2008), 113–113.

[17] Nelson, M.L., and R. Sen, "Business rules management in healthcare: A lifecycle approach", *Decision Support Systems 57*, 2014, pp. 387–394.

[18] Neuman, W.L., *Basics of social research: Qualitative and quantitative approaches*, Pearson Education, Inc., 2007.

[19] Olszak, C., and J. Zurada, "Big Data-driven Value Creation for Organizations", *Proceedings of the 52nd Hawaii International Conference on System Sciences*, (2019).

[20] Park, B., T. Flowerday, and R. Brünken, "Cognitive and affective effects of seductive details in multimedia learning", *Computers in Human Behavior 44*, 2015, pp. 267–278.

[21] Rayner, K., "Eye Movements in Reading and Information Processing: 20 Years of Research", *Psychological Bulletin 124*(3), 1998, pp. 372–422.

[22] Reijers, H.A., J. Mendling, and R.M. Dijkman, "Human and automatic modularizations of process models to enhance their comprehension", *Information Systems 36*(5), 2011, pp. 881–897.

[23] Sackmann, S., S. Kuehnel, and T. Seyffarth, "Using Business Process Compliance Approaches for Compliance Management with Regard to Digitization: Evidence from a Systematic Literature Review", *International Conference on Business Process Management*, Springer (2018), 409–425.

[24] Sadiq, S., and G. Governatori, "Managing Regulatory Compliance in Business Processes", In *Handbook on Business Process Management*. 2015, 23.

[25] Schrepfer, M., M. Kunze, G. Obst, and J. Siegeris, "Why are process variants important in process monitoring? The case of Zalando SE", In *Business Process Management Cases*. Springer, 2018, 431–448.

[26] Smit, K., M. Zoet, and M. Berkhout, "Verification capabilities for business rules management in the Dutch governmental context", *2017 International Conference on Research and Innovation in Information Systems (ICRIIS)*, IEEE (2017), 1–6.

[27] Surbakti, F.P.S., W. Wang, M. Indulska, and S. Sadiq, "Factors influencing effective use of big data: A research framework", *Information & Management*, 2019.

[28] Thalmann, M., A.S. Souza, and K. Oberauer, "How does chunking help working memory?", *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 2018.

[29] Thimm, M., "On the Compliance of Rationality Postulates for Inconsistency Measures: A More or Less Complete Picture", *KI - Künstliche Intelligenz 31*(1), 2017, pp. 31–39.

[30] Van Der Aalst, W., K.M. Van Hee, and K. van Hee, *Workflow management: models, methods, and systems*, MIT press, 2004.

[31] Wang, W., *Integrating Business Process Models and Rules*, Springer, 2019.

[32] Wang, W., M. Indulska, S. Sadiq, and B. Weber, "Effect of Linked Rules on Business Process Model Understanding", *International Conference on Business Process Management*, Springer International Publishing (2017), 200–215.

[33] Wang, W., M. Indulska, and S.W. Sadiq, "Cognitive Efforts in Using Integrated Models of Business Processes and Rules.", *CAiSE Forum*, (2016), 33–40.

7.3 A Tool for Decision-Logic Level Verification in DMN Decision Tables

Source	A Tool for Decision-Logic Level Verification in DMN Decision Tables
Authors	Carl Corea, Jonas Blatt and Patrick Delfmann
Outlet	BPM 2019 (Demo-Track) - 17th International Conference on Business Process Management, September 3-6, 2019, Vienna, Austria
Ranking	A
Status	Published

A Tool for Decision Logic Verification in DMN Decision Tables

Carl Corea, Jonas Blatt, and Patrick Delfmann

University of Koblenz-Landau, Germany
{`ccorea,jonasblatt,delfmann`}`@uni-koblenz.de`

Abstract. The Decision Model and Notation (DMN) is a popular standard to model company decision logic. Here, decision tables can be used to specify decision logic by the means of business rules. As these tables are modelled and maintained in an incremental and collaborative manner, this raises the need to verify the correctness of DMN decision tables. In this report, we therefore present a tool which allows to analyze the decision logic in DMN decision tables at design-time. Our tool implements all so-called verification capabilities from the recently proposed "business rule management capability framework" by Smit et al. [10], and also allows to detect errors distributed among multiple tables.

Keywords: DMN, Decision Logic Analysis, Camunda

1 Introduction

The Decision Model and Notation (DMN)[1] is an OMG standard for the representation of operational decision logic. Here, decisions can be expressed by the means of so-called decision tables. Fig. 1 shows an exemplary decision table. The columns of the table represent the input (income and assets) and output (credit worthiness). The rows constitute individual rules. The rules in Fig. 1 can be understood such that: *if* the income is ≤20 and the assets are >50, *then* the customer is not creditworthy (etc.).

U	Income (n)	Assets (n)	Creditworthy?
1	≤ 20	>50	false
2	[20..50]	>50	false
3	[30..40]	>100	true

Fig. 1: Exemplary decision table with modelling errors (units in thousands)

While DMN provides a standard on how to *represent* decisions, the actual content of the business rules is still the responsibility of the modeler. In turn,

---

[1] `https://www.omg.org/spec/DMN/About-DMN/`

the modelled decision logic can potentially contain errors such as inconsistencies or redundancies. An example of this is shown in Figure 1. First, the income conditions for rules 1 and 2 are overlapping (e.g., for an income of exactly 20). Furthermore, rule 2 completely subsumes rule 3. However, the respective outputs are inconsistent, meaning that this modelling error must be attended to by experts before rules can be merged.

In practice, such modelling errors can occur frequently, as decision tables are mostly maintained by multiple modelers. For example, Batoulis and Weske [1] reported on a case study with a large insurance company, where those authors found that 27% of analyzed rules contained overlaps. This calls for (automated) means to support companies in the verification of decision logic [1].

In this context, Smit et al. [10] have recently proposed the business rule management (BRM) capability framework. This framework identifies specific decision logic level verification capabilities, derived from qualitative research with industrial partners. Thus, those authors present a comprehensive set of verification capabilities actually needed in practice. Therefore, our tool implements these verification capabilities, which are: *identical rule verification, equivalent rule verification, subsumed rule verification, interdeterminism verification, partial reduction verification, overlapping condition verification* and *missing rule verification* (We will discuss these capabilities in in Section 2).

		Detection capabilities						
Literature		Identical Rules	Equivalent Rules	Subsumed Rules	Inter-determinism	Overlapping Conditions	Partial Reduction	Missing Rules
Calvanese et al. (2016)	[5]	X	o	X		X		X
Laurson et al. (2016)	[9]	X		X		X	o	X
Batoulis et al. (2017)	[2]	X		X		X		X
Calvanese et al. (2017)	[7]	X	o	X		X		X
Batoulis et al. (2018)	[3]	X		X		X		o
Batoulis et al. (2018)	[4]	X		X		X		
Calvanese et al. (2018)	[6]	X	o	X		X	o	X
Corea et al. (2018)	[8]	X			X	X		
This work		X	X	X	X	X	X	X

Table 1: Overview of capabilities from [10] covered by existing approaches.
(X = Full support, o = partial support/not aligned with [10])

Table 1 shows an overview of DMN decision logic verification approaches that have been introduced to the BPM community in recent years. To the best of our knowledge, our tool is the first to offer all decision logic level verification capabilities by Smit et al. [1][2]. Most prominently, we extend the works of Laurson and Maggi [9] by covering the capabilities missing in their approach. Also, those authors do not distinguish between identical rules, subsumed rules and overlapping rules, but denote all these error types as overlaps. Here our approach

---

[2] Please note that we do not implement "unnecessary facts verification", as this is geared towards analyzing case-dependent facts and is beyond the scope of this report.

distinguishes errors aligned along the definition in Smit et al. [10], to provide a more fine-granular understanding for companies as a basis for resolution. Also, to the best of our knowledge, whereas existing tools only allow to analyze individual DMN decision tables, our tool is the first to allow checking *multiple* tables at once. In case that different modelers have created tables, inconsistencies or overlaps between them can be analyzed.

2 Tool Description

Our tool integrates camunda-dmn[3], which is a Java library for DMN by Camunda. Our project can be viewed at `https://gitlab.uni-koblenz.de/fg-bks/br-verification-tool`. Also, an online-demo[4] and screencast[5] are available.

Our browser-based tool allows to upload and analyze DMN decision tables. The analysis is based on the framework by Smit et al. [10]. As mentioned, this framework was derived based on interviews with industrial partners, i.e. the capabilities reflect analysis tasks needed in practice. Our tool therefore implements all of the verification capabilities proposed in [10] as follows:

- **Identical rule verification.** Detecting rules, which have an identical input, i.e. are redundant.
- **Equivalent rule verification.** Detecting rules, which are not identical, but still semantically equivalent. Here, our tool can detect equivalent rules, based on synonym relations.
- **Subsumed rule verification.** Detecting individual rules, which are subsumed by other rules, i.e. they are not necessary.
- **Interdeterminism verification.** Detecting rules, which will *always* be activated together, but have differing or contradicting conclusions. For example, rules must not yield that a customer is both credit worthy, and not credit worthy, as this is logically inconsistent.
- **Partial reduction verification.** Checking whether ranges can be combined to simplify decision tables.
- **Overlapping condition verification.** Detecting whether there are any overlaps in rule conditions.
- **Missing rule verification.** Detecting whether there are any missing business rules, e.g., gaps in condition ranges.

Our tool also allows an analysis of multiple decision tables at once, e.g. the tool can be used to find identical rules which are distributed across multiple tables. A more detailed explanation and examples for the individual verification capabilities can be found in the supplementary documentation[6].

Figure 2 shows an actual usage example. For this example, we uploaded the decision table shown in Figure 1. The tool provides an overview of all errors

---

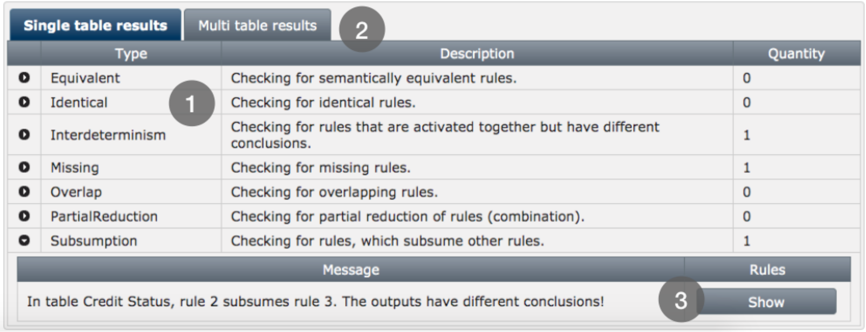
[3] `https://github.com/camunda/camunda-engine-dmn`

[4] `http://inconsistency.fg-bks.uni-koblenz.de:8090/`

[5] `https://youtu.be/yTXTKi3s6LM`

[6] `https://gitlab.uni-koblenz.de/fg-bks/br-verification-tool/wikis/home`

found in the tables (1). There are 2 main tabs, namely for errors found *within* single tables, or errors including *multiple* tables (2). Users can browse the errors and click on the "show" button (3) to highlight the concerned rules. Users can also edit and re-verify the tables directly in the browser. We would like to remind the reader of the available online-demo and screencast referenced above.



	Type	Description	Quantity
●	Equivalent	Checking for semantically equivalent rules.	0
●	Identical	Checking for identical rules.	0
●	Interdeterminism	Checking for rules that are activated together but have different conclusions.	1
●	Missing	Checking for missing rules.	1
●	Overlap	Checking for overlapping rules.	0
●	PartialReduction	Checking for partial reduction of rules (combination).	0
●	Subsumption	Checking for rules, which subsume other rules.	1

Message	Rules
In table Credit Status, rule 2 subsumes rule 3. The outputs have different conclusions!	Show

Fig. 2: Usage example: Results of analyzing the table in Figure 1.

3 Maturity and Outlook

For evaluation, we performed run-time experiments. To this aim, we analyzed a total of 300 synthetic decision tables. As parameters for generating these tables, we chose the number of table columns from {1,2,...,10}, and the number of table rows from {50,100,...,500} (i.e., 10x10 possible combinations). For each of the 100 possible combinations of rows and columns, we generated 3 decision tables with different random rules (i.e., 300 decision tables). The respective rules of these tables were randomly generated by using random integer conditions, with one of the operators from $\{=, [a..b], \leq, \geq, <, >\}$. These random conditions allow to create synthetic decision tables with actual errors, which are meant to be analyzed by our tool, such as redundancies or condition overlaps. We consequently applied our verification tool and computed the average run-time for each parameter configuration, which is shown in Figure 3. Our experiments were run on a Windows 10 PC with i7 processor, 16GB DDR4 RAM and 512 GB SSD memory. As can be seen in Figure 3, the run-time for analyzing 500 rules with 10 columns averages to roughly 5s. Thus, for our analyzed data-sets, our tool allowed for a feasible analysis.

To conclude, the tool presented in this work allows to analyze multiple DMN decision tables. The tool currently supports the unique hit policy, which will be extended in future work. Regarding the verification capabilities, our tool implements the capabilities proposed in the practice-based framework by [10], and thus extends existing works such as [9]. Hence, our tool supports companies in decision logic verification and facilitates sustainable business rules management. In future work, we aim to apply our tool to industrial data-sets.

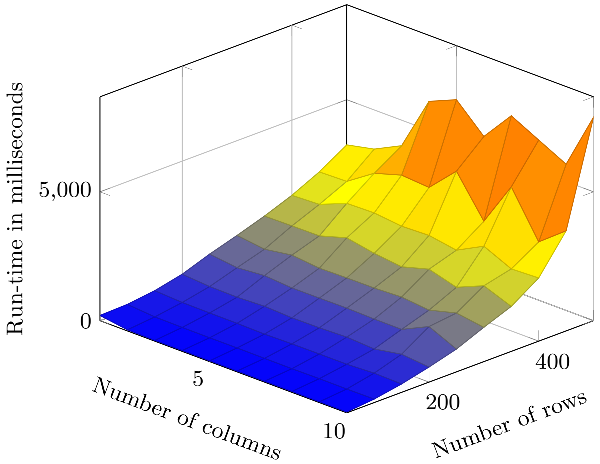


Fig. 3: Run-time for the analysis of synthetic decision tables with up to 10 columns and 500 rows.

References

1. Batoulis, K., Nesterenko, A., Repitsch, G., Weske, M.: Decision management in the insurance industry: Standards and tools. In: Proceedings of the BPM 2017 Industry Track (2017)
2. Batoulis, K., Weske, M.: A tool for checking soundness of decision-aware business processes. In: Proceedings of the BPM Demo Track (2017)
3. Batoulis, K., Weske, M.: Disambiguation of DMN decision tables. In: Business Information Systems - 21st International Conference (2018)
4. Batoulis, K., Weske, M.: A tool for the uniqueification of DMN decision tables. In: Proceedings of the BPM Demo Track (2018)
5. Calvanese, D., Dumas, M., Laurson, Ü., Maggi, F.M., Montali, M., Teinemaa, I.: Semantics and analysis of dmn decision tables. In: International Conference on Business Process Management. Springer (2016)
6. Calvanese, D., Dumas, M., Laurson, Ü., Maggi, F.M., Montali, M., Teinemaa, I.: Semantics, analysis and simplification of dmn decision tables. Information Systems **78** (2018)
7. Calvanese, D., Dumas, M., Maggi, F.M., Montali, M.: Semantic dmn: Formalizing decision models with domain knowledge. In: International Joint Conference on Rules and Reasoning. Springer (2017)
8. Corea, C., Delfmann, P.: A tool to monitor consistent decision-making in business process execution. In: Proceedings of the BPM Demo Track (2018)
9. Laurson, Ü., Maggi, F.M.: A tool for the analysis of dmn decision tables. In: Proceedings of the BPM Demo Track (2016)
10. Smit, K., Zoet, M., Berkhout, M.: Verification capabilities for business rules management in the dutch governmental context. In: Research and Innovation in Information Systems (ICRIIS), 2017 International Conference on. IEEE (2017)

7.4 Quasi-Inconsistency in Declarative Process Models

Source	Quasi-Inconsistency in Declarative Process Models
Authors	Carl Corea and Patrick Delfmann
Outlet	BPM Forum 2019 - 17th International Conference on Business Process Management, September 3-6, 2019, Vienna, Austria
Ranking	A
Status	Published

Quasi-Inconsistency in Declarative Process Models

Carl Corea and Patrick Delfmann

Institute for Information System Research, University of Koblenz-Landau, Germany
{ccorea,delfmann}@uni-koblenz.de

Abstract. The field of declarative process discovery comprises techniques for mining declarative constraint sets from event logs. While current techniques verify the relation of individual constraints to the log, they do not consider the interrelation between constraints. This can lead to logical contradictions between the discovered constraints. In this work, we introduce a new form of such contradictions entitled implicit inhibitors. In short, these are sets of constraints which will always be activated together, but demand contradicting reactions. In turn, such constraint sets can be denoted as quasi-inconsistent, as the contained constraints are unsatisfiable should they be activated together. We introduce a structured approach to detect and analyze quasi-inconsistencies in declarative process models and evaluate our approach through formal analysis and run-time experiments on real-life data-sets.

Keywords: Declarative Constraints · Implicit Inhibition · Declare

1 Introduction

Declarative process models consist of constraints which specify the behavior which company processes should adhere to. Process execution in declarative process models is thus all allowed behaviour within the set of constraints. The semantics of declarative constraints is mostly formalized with temporal logic, e.g. with modelling languages such as Declare [6], [14]. For example, the Declare constraint $\textsc{ChainResponse}(a, b)$ imposes that if a task a occurs, it must be directly followed by a task b. Likewise, $\textsc{Response}(a, b)$ states, that if a task a occurs, it must be eventually followed by a task b. When utilizing declarative models, companies face numerous current challenges: In the scope of process discovery, current discovery techniques can yield sets of constraints which are unusable or confusing to modelers [7]. Also, human modelling errors or merging models in the scope of company mergers can yield erroneous models [3].

As an example, consider the constraint sets $\mathcal{C}_1$ and $\mathcal{C}_2$, defined via

$$\begin{aligned}\mathcal{C}_1 = \{&\textsc{ChainResponse}(a, b) & \mathcal{C}_2 = \{&\textsc{ChainResponse}(a, b)\\ &\textsc{NotResponse}(a, b)\} & &\textsc{ChainResponse}(b, c)\\ & & &\textsc{NotResponse}(a, c)\}.\end{aligned}$$

In both cases, the task a is inhibited by (multiple) constraints which define which tasks must or must not follow. However, these constraints demand logically contradicting reactions to the occurence of task a. In turn, should the task a occur, the declarative process model cannot further be executed.

Motivation: Can't this be already solved with finite state automata? The observant reader might ask whether the above examples are not *inconsistencies* as defined in Di Ciccio et al. (2017), and thus could already be detected by existing approaches such as automata products. In short, the above examples are not inconsistencies, but rather *quasi*-inconsistencies, explained as follows.

Di Ciccio et al. (2017) have discussed the problem of inconsistent constraints that can be returned during process discovery. Those authors define inconsistency as a declarative process model which does not accept any execution trace, i.e. it is unsatisfiable. An example would be the constraint set $\mathcal{C}_3$, defined via

$$\begin{aligned}\mathcal{C}_3 = \{&\textsc{Participation}(a)\\ &\textsc{ChainResponse}(a,b)\\ &\textsc{NotResponse}(a,b)\}.\end{aligned}$$

As can be seen, $\mathcal{C}_3$ contains the constraint Participation(a), which states that the task a <u>must</u> occur in every execution trace. In result, the constraints $\textsc{ChainResponse}(a,b)$ and $\textsc{NotResponse}(a,b)$ must also always be activated in any execution trace. However, this constellation is inconsistent, i.e. there cannot exist a trace that satisfies the model in $\mathcal{C}_3$.

On the contrary, a model containing the constraints in $\mathcal{C}_1$ or $\mathcal{C}_2$ can accept an arbitrary amount of execution traces, namely any trace which does not contain the task a. For example, a trace "*bcbdbde*" would satisfy respective models in $\mathcal{C}_1$ or $\mathcal{C}_2$. Thus, these constraint sets are not inconsistent as defined in [7], but rather *quasi-inconsistent*. That is, certain tasks are implicitly inhibited by a set of contradictory constraints. Due to this different conceptualization, this implicit inhibition can however not be detected via automata products as in [7], as there can be a non-empty set of accepted traces. In result, this paper discusses a new form of problem in declarative process discovery. Intuitively, declarative process models should not contain sets of constraints as in $\mathcal{C}_1$ or $\mathcal{C}_2$, as the can potentially make models unusable. Yet, current discovery techniques can return such quasi-inconsistent constraint sets. Furthermore, as motivated above, such quasi-inconsistencies can currently not be detected by existing means such as automata products. This is underlined by our experiment results (cf. Section 5), where we analyzed real-life models and found more than 25.000 of such contradictory constraint sets as in $\mathcal{C}1$ or $\mathcal{C}_2$, which cannot be detected with the approach by Di Ciccio et al. (2017). In this work, we therefore introduce the notion of quasi-inconsistency and propose a first structured approach to detect and analyze all minimal implicit inhibition sets in declarative process models.

The remainder of this paper is as follows. Section 2 provides background information on declarative process models. In Section 3, we introduce the novel concept of *quasi-inconsistent subsets* and show how results from the scientific field of inconsistency measurement can be adapted to detect and analyze such subsets. In Section 4, we present an algorithm for the feasible computation of quasi-inconsistent subsets. The proposed capabilities for detection and analysis are evaluated in Section 5, followed by a conclusion in Section 6.

2 Background

Traditional process models define a clear imperative structure of how exactly company activities should be executed. To allow for more flexibility, declarative process models have received increasing attention [1],[6], [7]. Here, a declarative process model defines constraints which must be upheld or not be violated, and thus process execution is flexible within this set of constraints.

Definition 1 (Declarative Process Model). *A declarative process model is a tuple* $\boldsymbol{M} = (\boldsymbol{A}, \boldsymbol{T}, \boldsymbol{C})$*, where* $\boldsymbol{A}$ *is a set of tasks,* $\boldsymbol{T}$ *is a set of constraint templates, and* $\boldsymbol{C}$ *is the set of actual constraints, which instantiate the template elements in* $\boldsymbol{T}$ *with tasks in* $\boldsymbol{A}$*.*

In this paper, we consider Declare [14], which is a widely acknowledged declarative process modelling language and notation. Declare allows to define constraints by using predefined templates and passing tasks as parameters to respective templates, cf. the examples in Section 1. In this way, modelers can use the rather intuitive templates to define constraints, with the formal semantics "hidden" from the user. Formally, the semantics of Declare can be defined with temporal logic [1], [4]. This allows to use the amenities of temporal logic checking, as well as to create custom Declare constraint templates.

We define the semantics of Declare constraints with LTL_p [13], a linear-time temporal logic with past. An LTL_p formula is given by the grammar

$$\varphi ::= a|(\neg\varphi)|(\varphi_1 \wedge \varphi_2)|(\bigcirc\varphi)|(\varphi_1 \mathbf{U} \varphi_2)|(\ominus\varphi)|(\varphi_1 \mathbf{S} \varphi_2).$$

Each formula is built from atomic propositions $\in \mathbf{A}$ (relative to a declarative process model), and is closed under the boolean connectives, the unary temporal operators $\bigcirc$ (next) and $\ominus$ (previous), and the binary temporal operators $\mathbf{U}$ (until) and $\mathbf{S}$ (since). Given a declarative process model $\mathbf{M} = (\mathbf{A}, \mathbf{T}, \mathbf{C})$, a sequence t (with length n) of tasks in $\mathbf{A}$, where $t(i)$ denotes the i^{th} element of the sequence t, the semantics of LTL_p formulae are defined as follows:

$\mathrm{t,i} \models True/\mathrm{t,i} \not\models False$ $\quad$ $\mathrm{t,i} \models a$ iff $t(i) = a$

$\mathrm{t,i} \models \neg\varphi$ iff $t, i \not\models \varphi$ $\quad$ $\mathrm{t,i} \models \varphi_1 \wedge \varphi_2$ iff $t, i \models \varphi_1$ and $t, i \models \varphi_2$

$\mathrm{t,i} \models \bigcirc\varphi$ iff $i < n$ and $t, i+1 \models \varphi$ $\quad$ $\mathrm{t,i} \models \ominus\varphi$ iff $i > 1$ and $t, i-1 \models \varphi$

$\mathrm{t,i} \models \varphi_1 \mathbf{U} \varphi_2$ iff $t, j \models \varphi_2$ with $i \leq j \leq n,$ and $t, k \models \varphi_1$ for all k s.t.$i \leq k < j$

$\mathrm{t,i} \models \varphi_1 \mathbf{S} \varphi_2$ iff $t, j \models \varphi_2$ with $1 \leq j \leq i,$ and $t, k \models \varphi_1$ for all k s.t.$j < k \leq i$

From the above syntax and semantics, we furthermore derive $\varphi_1 \vee \varphi_2$ as $\neg(\neg\varphi_1 \wedge \neg\varphi_2)$, $\varphi_1 \rightarrow \varphi_2$ as $\neg\varphi_1 \vee \varphi_2$, $\Diamond\varphi$ as $True\mathbf{U}\varphi$ (which indicates that

φ will eventually hold true, possibly later and not directly following $t(i)$), $\diamondminus\varphi$ as *True* $\mathbf{S}\ \varphi$ (which indicates that φ holds true sometime before $t(i)$, but not necessarily directly before $t(i)$), and $\Box\varphi$ as $\neg\Diamond\neg\varphi$ (which indicates that there is no future $t(i)$ which does not satisfy φ).

Based on such LTL$_p$ formulae, the semantics of individual Declare constraints can be defined. For instance, the exemplary constraints used in $\mathcal{C}_1$ and $\mathcal{C}_2$ are defined as ChainResponse$(a,b) \equiv \Box(a \rightarrow \bigcirc b)$, NotChainResponse$(a,b) \equiv \Box(a \rightarrow \neg \bigcirc b)$, NotResponse$(a,b) \equiv \Box(a \rightarrow \neg\Diamond b)$. A standard set of Declare templates and corresponding semantics have been defined derived from the work of [8]. Please see [1], [7] or [12] for further details.

An interesting gist about such constrains is that Declare seems to capture *activation-response* relations between tasks. For instance, ChainResponse(a,b) can be interpreted such that, *if* there is an activation a, then this entails a reaction $\bigcirc b$. Therefore, following [2], we use the notion of reactive constraints, which make the the activation and reaction semantics of LTL$_p$ constraints explicit.

Definition 2 (Reactive Constraints [2]). *Given a declarative process model* $\boldsymbol{M} = (\boldsymbol{A}, \boldsymbol{T}, \boldsymbol{C})$*, and a constraint* $\in \boldsymbol{C}$ *with activation* α *and reaction* φ*, a reactive constraint (RCon)* Ψ *is a pair* (α, φ)*. We denote* $\Psi = (\alpha, \varphi)$ *as* $\alpha \Rightarrow \varphi$*. We say that* α *activates the constraint and the reaction* φ*.*

Table 1 provides an overview of Declare constraints used in this work, as well as the corresponding RCon and activation. Please refer to [2],[7] for a further discussion and classification of activations in Declare constraints.

Constraint	**Reactive Constraint**	**Activation**
Response(a,b)	$a \Rightarrow \Diamond b$	a
ChainResponse(a,b)	$a \Rightarrow \bigcirc b$	a
AlternateResponse(a,b)	$a \Rightarrow \bigcirc(\neg a\ \mathbf{U}\ b)$	a
Precedence(a,b)	$b \Rightarrow \diamondminus a$	b
ChainPrecedence(a,b)	$b \Rightarrow \ominus a$	b
AlternatePrecedence(a,b)	$b \Rightarrow \ominus(\neg b\ \mathbf{S}\ a)$	b
NotResponse(a,b)	$a \Rightarrow \neg\Diamond b$	a
NotChainResponse(a,b)	$a \Rightarrow \neg \bigcirc b$	a
NotPrecedence(a,b)	$b \Rightarrow \neg\diamondminus a$	b
NotChainPrecedence(a,b)	$b \Rightarrow \neg\ominus a$	b

Table 1: Reactive Constraints corresponding to exemplary Declare constraints

In result, a quasi-inconsistency is present if we have a constraint set containing multiple RCons with the same activation, but contradictory reactions. In the following, we will show how such quasi-inconsistencies in declarative process models can be detected and analyzed.

3 Detecting and Assessing Quasi-Inconsistencies

3.1 Detection

As declarative constraints are inherently of reactive nature, they underly the principle of ex falso quodlibet: no conclusions can be made without knowledge of activation. As motivated in Section 1, this means that the exemplary constraint sets $\mathcal{C}_1$ and $\mathcal{C}_2$ are not inconsistent per se, as it is not known whether these constraints will actually be activated (i.e., there is no constraint like $\textsc{Participation}(a)$ which dictates the occurrence of a task a in an execution). In turn, there can be an arbitrary amount of traces that satisfy models as in $\mathcal{C}_1$ and $\mathcal{C}_2$, thus it is not possible to detect quasi-inconsistency by the existing means of automata products as in [7], which detects inconsistency as an empty set of acceptable input traces.

Thus, we present a novel means for detecting quasi-inconsistency. In the following, we use the RCon representation, but sometimes provide specific DECLARE templates for readability. Furthermore, let a constraint c, we denote $out(c)$ as the outcome of a constraint, i.e. φ of the respective RCon.

Definition 3 (Individual Constraint Activation). *A set of activations A activates an individual constraint $c : a \Rightarrow \varphi$ iff $a \in A$.*

Quasi-inconsistencies can arise, if we have a set of activations A', such that A' activates at least two different constraints, and these constraints have contradictory outcomes, e.g. in example $\mathcal{C}_1$, the activation set {a} activates two contradictory constraints. However, as the conclusions of some constraints might be an activation to other constraints themselves via transitive relations, the activation set A' might activate a multitude of constraints. In order to analyze quasi-inconsistencies, all these activated constraints must be considered.

Definition 4 (Constraint Set activation). *A set of activations A activates a set of constraints C iff $\forall c \in C : A \cup \{out(c) | c \in C\}$ activates c.*

Example 1. Consider the constraint set $\mathcal{C}_4$, defined via

$$\mathcal{C}_4 = \{a \Rightarrow b, b \Rightarrow c, c \Rightarrow d\}.$$

For each individual constraint, the activation set is simply the premise of the constraint, i.e. a is the activation set of the individual constraint $a \Rightarrow b$. Furthermore, the activation a also activates the entire set of constraints in $\mathcal{C}_4$ via the transitive relations.

Given a declarative set of constraints, the introduced notions allow to define *quasi-inconsistent subsets*.

Definition 5 (Quasi-Inconsistent Subset). *For a constraint set C, the set of quasi-inconsistent subsets QI is defined as a set of pairs $(\boldsymbol{A}, \boldsymbol{C})$, s.t.*

1. $\boldsymbol{C} \subseteq C$

2. $\boldsymbol{A}$ *activates* $\boldsymbol{C}$
3. $\boldsymbol{A} \cup \boldsymbol{C} \models \perp$

To clarify, we consider a set of activations $\mathbf{A}$, which activate $\mathbf{C}$. Then, the entirety of all activations and activated constraints is inconsistent. Our proposition of quasi-inconsistent subsets allows to determine the "inconsistent subsets" of arbitrary declarative constraints sets, by augmenting activations and thus determining those constraints which will a) always be activated together, and b) yield an inconsistency, should they be activated. Consequently, we define minimal quasi-inconsistent subsets analogously.

Definition 6 (Minimal Quasi-Inconsistent Subset). *For a constraint set* C*, the set of minimal quasi-inconsistent subsets* MQI *is defined as set of pairs* $t = (\boldsymbol{A}, \boldsymbol{C})$*, s.t.*

1. t *is a quasi-inconsistent subset in* C
2. *for any* $t' \subset t$*, where exactly one element is deleted from exactly one of the sets in* t*,:* $t' \not\models \perp$

A minimal quasi-inconsistent subset is a quasi-inconsistent subset which is minimal w.r.t. set inclusion, i.e., removing exactly one constraint resolves the quasi-inconsistency. As we are mostly interested in the distinct constraints which are quasi-inconsistent to each other, we use M^C to denote the set of constraints $\mathbf{C}$ from any $M \in \mathsf{MQI}$.

Example 2. Consider the following Declare constraint set $\mathcal{C}_5$, defined via

$$\begin{array}{llll} \mathcal{C}_5 = \{ & \textsc{ChainResponse}(a,b), & \textsc{Response}(b,d), & \textsc{NotChainPrecedence}(a,b) \\ & \textsc{ChainResponse}(d,e), & \textsc{NotResponse}(a,b), & \textsc{ChainResponse}(e,c) \\ & \textsc{ChainResponse}(b,c), & \textsc{Response}(a,b) & \textsc{NotResponse}(a,c) \quad \} \end{array}$$

Then[1],

$$\begin{aligned} \mathsf{MQI}(\mathcal{C}_5) &= \{M_1, M_2, M_3, M_4, M_5, M_6, M_7\} \\ M_1^C &= \{\textsc{NotChainPrecedence}(a,b), \textsc{ChainResponse}(a,b)\} \\ M_2^C &= \{\textsc{ChainResponse}(a,b), \textsc{NotResponse}(a,b)\} \\ M_3^C &= \{\textsc{Response}(a,b), \textsc{NotResponse}(a,b)\} \\ M_4^C &= \{\textsc{Response}(a,b), \textsc{NotResponse}(a,c), \textsc{ChainResponse}(b,c)\} \\ M_5^C &= \{\textsc{ChainResponse}(a,b), \textsc{ChainResponse}(b,c), \textsc{NotResponse}(a,c)\} \\ M_6^C &= \{\textsc{Response}(a,b), \textsc{Response}(b,d), \textsc{ChainResponse}(d,e), \\ &\qquad \textsc{ChainResponse}(e,c), \textsc{NotResponse}(a,c)\} \\ M_7^C &= \{\textsc{ChainResponse}(a,b), \textsc{Response}(b,d), \textsc{ChainResponse}(d,e), \\ &\qquad \textsc{ChainResponse}(e,c), \textsc{NotResponse}(a,c)\} \\ M_1 &= (\{a\}, \{\textsc{NotChainPrecedence}(a,b), \textsc{ChainResponse}(a,b)\}) \end{aligned}$$

---

[1] $M_2 - M_7$ are omitted due to space restrictions, but are analogously to M_1 (all with activation set $\{a\}$)

This example shows the minimal quasi-inconsistent subsets of the constraint set $\mathcal{C}_5$. As can be seen, all such subsets implicitly inhibit certain tasks in an unsatisfiable way. Thus, in the example, should the task a occur, the resp. model is unsatisfiable and can thus not be used for simulation or to govern compliant process execution. Intuitively, declarative process models should therefore not contain such inhibiting subsets. Through our novel definition of quasi-inconsistent subsets, we are able to detect all such problematic subsets within a set of constraints. Furthermore, our definition of quasi-inconsistent subsets enables a further assessment of resp. subsets.

3.2 Analysis

In order to understand potential inconsistencies, companies should be provided with a careful analysis of detected quasi-inconsistencies. To this aim, results from the scientific field of inconsistency measurement can be adapted [10]. Inconsistency measurement is a discipline concerned with the analysis of inconsistent information. Here, the central object of study are quantitative measures, which allow to assign a numerical value to (elements of) a constraint set, with the informal meaning that a higher value reflects a higher degree of inconsistency. These measures can be distinguished into so-called *inconsistency measures*, and *culpability measures*. The former is used to assess the inconsistency of the entire constraint set, while the latter is used to assess the degree of blame that individual constraints carry in the context of the overall inconsistency. As some of these measures are based on set-theoretic principles, we propose to adopt these measures to analyze quasi-inconsistent subsets as follows.

Quasi-Inconsistency Measures. Let $\mathfrak{C}$ denote the universe of all declarative constraint sets. Then, an inconsistency measure $\mathcal{I}$ is a function

$$\mathcal{I} : \mathfrak{C} \rightarrow [0, \infty)$$

which assigns a non-negative real value to a constraint set, with the informal meaning that a higher value reflects a higher severity of inconsistency.

Following recent surveys [16,17], there are four measures based on minimal inconsistent subsets which have been proposed, namely the *MI-inconsistency measure*, the $\mathsf{MI}^{\mathcal{C}}$*-inconsistency measure*, the *problematic inconsistency measure* and the *mv-inconsistency measure*. Currently these measures are only defined for inconsistencies (and not for quasi-inconsistencies). To analyze the degree of quasi-inconsistency of declarative process models, these can easily be adapted to fit the use-case of quasi inconsistencies. For further evaluation, we present this for the example of the *MI-inconsistency measure*. We omit a detailled discussion of all measures due to space limitations.

Let $\mathcal{C}$ be a set of constraints and $\mathcal{A}(\mathcal{C})$ denote the tasks in a set $\mathcal{C}$. Then, the adapted versions of the abovementioned measures are defined as follows.

Definition 7 (MQI-inconsistency measure). *Define the* MQI-inconsistency measure *via*

$$\mathcal{I}^{Q}_{\mathsf{MI}}(\mathcal{C}) = |\mathit{MQI}(\mathcal{C})|$$

This measure counts the number of minimal quasi-inconsistent subsets in $\mathcal{C}$.

Example 3. We revisit the constraint set $\mathcal{C}_5$ from Example 2. Then

$$\mathcal{I}^{Q}_{\mathsf{MI}}(\mathcal{C}_4) = 7$$

Culpability Measures. Next to assessing the degree of inconsistency for an entire constraint set, results from inconsistency measurement also allow to quantify the degree of inconsistency for individual constraints. This allows to pin-point constraints with a high degree of blame for the overall inconsistency. Let $\mathfrak{c}$ denote the universe of all possible constraints, and $\mathfrak{C}$ the universe of declarative constraint sets. Then, a culpability measure $\mathit{\Gamma}$ is a function

$$\mathit{\Gamma} : \mathfrak{C} \times \mathfrak{c} \to [0, \infty)$$

which assigns a non-negative number to a mapping of an individual constraint to a constraint set, and can thus assess the culpability that an individual constraint represents w.r.t. the constraint set. There are two culpability measures based on minimal inconsistent subsets which have been proposed, namely the *cardinality based culpability measure* $\mathit{\Gamma}_{\#}$, and the *normalized culpability measure* $\mathit{\Gamma}_c$ [11]. Again, these can be easily adopted for the use-case at hand, which we show for the cardinality-based culpability measure.

Definition 8 (Cardinality-Based Culpability Measure). *Define the* cardinality based culpability measure $\mathit{\Gamma}^{Q}_{\#}$ *via*

$$\mathit{\Gamma}^{Q}_{\#}(\mathcal{C}, \alpha) = |M \in \mathit{MQI}(\mathcal{C}) | \alpha \in M^{C}|$$

This measure counts the number of minimal quasi-inconsistent subsets that a constraint α appears in.

Example 4. We revisit the constraint set $\mathcal{C}_5$ from Example 2. Then

(i) $\mathit{\Gamma}^{Q}_{\#}(\mathcal{C}_5, \textsc{ChainResponse}(a, b)) = 4$
(ii) $\mathit{\Gamma}^{Q}_{\#}(\mathcal{C}_5, \textsc{NotChainPrecedence}(a, b) = 1$
(iii) $\mathit{\Gamma}^{Q}_{\#}(\mathcal{C}_5, \textsc{NotResponse}(a, b) = 2$
(iv) $\mathit{\Gamma}^{Q}_{\#}(\mathcal{C}_5, \textsc{ChainResponse}(b, c) = 2$
(v) $\mathit{\Gamma}^{Q}_{\#}(\mathcal{C}_5, \textsc{NotResponse}(a, c) = 4$
(vi) $\mathit{\Gamma}^{Q}_{\#}(\mathcal{C}_5, \textsc{Response}(b, d) = 2$
(vii) $\mathit{\Gamma}^{Q}_{\#}(\mathcal{C}_5, \textsc{ChainResponse}(d, e) = 2$
(viii) $\mathit{\Gamma}^{Q}_{\#}(\mathcal{C}_5, \textsc{ChainResponse}(e, c) = 2$
(ix) $\mathit{\Gamma}^{Q}_{\#}(\mathcal{C}_5, \textsc{Response}(a, b) = 3$

Culpability measures provide quantitative insight that can help companies to understand and resolve problems in their models [15]. The intuition here is that a higher culpability reflects a higher degree of blame that an individual constraint carries in the context of the overall inconsistency [9]. For example, the $\mathit{\Gamma}^{Q}_{\#}$ is essentially a scoring function which quantifies how many quasi-inconsistent subsets can be resolved, if a constraint is deleted.

4 Computation of Quasi-Inconsistent Subsets

The basis for the proposed detection and analysis are quasi-inconsistent subsets. In the following, we therefore propose an novel approach for the feasible computation of MQI. Algorithm 1 shows our approach to compute minimal quasi-inconsistent subsets for declarative constraints. As a central object of study, we utilize reactive constraints to construct a so-called *reactive entailment graph.*

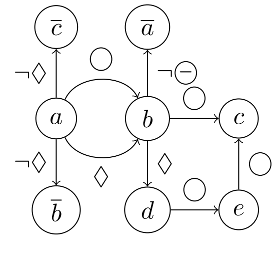
4.1 Reactive Entailment Graph

Definition 9 (Reactive Entailment Graph). *Given a declarative process model* $\boldsymbol{M} = (\boldsymbol{A}, \boldsymbol{T}, \boldsymbol{C})$*, its reactive entailment graph (REG) is defined as a graph* $G = (A, E, \tau, n)$*, where* $A = \boldsymbol{A} \cup \overline{\boldsymbol{A}}$ *are the tasks in* M *in two forms (with and without overline symbol),* $E \subseteq A \times A$ *is the set of directed edges between tasks in* A*,* τ *is a function* $\tau : E \rightarrow \boldsymbol{T}$ *assigning an individual edge in* E *to a template type in* $\boldsymbol{T}$*, and* n *is a function* $n : E \rightarrow \mathbb{N}$ *which assign a natural number to an edge to allow for multiple edges between the same tasks in* A*.*

The reactive entailment graph is a graph representation of reactive constraints. For example, given the declarative constraint ChainResponse(a, b), this can be represented as two nodes a and b, related by an edge of type ○. In the following, we ommit edge numbering for simplicity.

An important detail is that we include two "forms" of tasks, explained as follows. As can be seen in Table 1, one could argue there are essentially two types of declarative constraints. First, there are constraints such as ChainResponse, which are aimed to ensure that, should some event occur, then another event *must* occur (in a certain way). Then, there are other constraints such as NotChainResponse, which are aimed to ensure that, should some event occur, then another event *must not* occur (in a certain way). The reactive entailment graph captures these two types of *demanding* and *prohibiting* constraints, with the intuition that the overlined form of a task relates to a prohibition and vice versa. Then, the edges, respectively the edge types convey information on how exactly a task is demanded or prohibited, w.r.t a node which is the activation.

Example 5. We revisit the exemplary constraint set from Example 2. Then, this yields the following reactive entailment graph:



This graph encodes the relations between tasks of a declarative process model, as well as their relation type. For example, it can be seen that $a \Rightarrow \Diamond b$, and $a \Rightarrow \neg\Diamond\overline{b}$. This encodes that the activation a *demands* task b, resp. *prohibits* a later occurence of task b. An advantage of including two forms of tasks to encode the demanding, resp. prohibiting, nature of reactive constraints is an efficient way to scan the REG for potential inconsistencies, by searching for pairs of nodes n and n', where $n = \overline{n'}$, as will be discussed in the subsequent section.

The graph relations can be transformed back into the original constraints, where if $(\alpha, t_i) \in E$, then the original reactive constraint c_i is defined as $c_i = \alpha \Rightarrow \tau(\alpha, t_i)t_i$. For an edge $e \in E$, we denote the corresponding constraint as e^C. Given a path p being a sequence of edges in the REG, we denote the set of corresponding constraints captured by p as p^C.

4.2 Algorithm for computing MQI in declarative constraint sets

Following Definition 5, quasi-inconsistency can only occur if

1. There is at least one task Δ
2. Δ is the $\overline{\text{outcome}}$ of at least two constraints c_1 and c_2
3. $out(c_1) = \overline{out(c_2)}$

Furthermore, the constraints c_1 and c_2 have to be activated simultaneously, thus

4. c_1 and c_2 have the same activation set a

Algorithm 1 computes MQI of declarative constraint sets by exploiting the reactive entailment graph to search for subsets satisfying 1-4. In the following, we explain our algorithm based on the constraint set from Example 2 and the corresponding REG from Example 5.

Algorithm 1: Computation of minimal quasi-inconsistent subsets

```
Input  : Set of constraints $\mathbf{C}$
Output: $\mathsf{MQI}(\mathbf{C})$
1 $qmis \leftarrow \emptyset$;
2 $compConstraints = findComplements(C)$;
3 foreach $n$:$compConstraints$ do
4     $\alpha \leftarrow n.activation$;
5     $\omega \leftarrow n.reactionTask$ ;
6     $\mathbf{P} = findPaths(\alpha, \overline{\omega}) \cup findPaths(\overline{\omega}, \alpha)$;
7     foreach $P$:$\mathbf{P}$ do
8         if $\alpha \cup n \cup P^C \models \bot$ then
9             $mis \leftarrow mis \cup n \cup P^C$;
```

In line 1, a set to store minimal quasi-inconsistent subsets ($mqis$) is initialized. Then, we start by identifying all nodes n' of the REG which are a complement to another node n'' (line 2). In the example, there are three such cases, namely a vs $\overline{a}$, b vs $\overline{b}$ and c vs $\overline{c}$ (cf. the corresponding REG). Due to space limitations, we focus on $\overline{c}$ in the following, i.e., we assume the current iterated node $n_i = \overline{c}$. Its activation α is its predecessor in the REG, here a. The algorithm subsequently search for all shortest paths from α to the inverse of the current n_i via a breadth-first search, stored in P (i.e., in our example the algorithm searches for all shortest paths from a to c, and from c to a in the REG). We store possible paths from $\overline{n_i}$ to α, to cope with constraints such as precedence. Also, note that these can be transitive paths with multiple hops. As can be seen in the REG in Example 5, there are four paths from a to c. Subsequently, the algorithm verifies whether the constraints pertaining to a found path P contradict the original constraint $c_i = a \Rightarrow \neg\Diamond\overline{c}$. To this aim, we verify if $\alpha \cup c_i \cup P^C \models\perp$, in which case we have found a minimal quasi-inconsistent subset. In the example, the conditions verified in line 8 are respectively:

(1) $a \cup \textsc{NotResponse}(a,c) \cup \{\textsc{ChainResponse}(a,b), \textsc{ChainResponse}(b,c)\} \models\perp$
(2) $a \cup \textsc{NotResponse}(a,c) \cup \{\textsc{Response}(a,b), \textsc{ChainResponse}(b,c)\} \models\perp$
(3) $a \cup \textsc{NotResponse}(a,c) \cup \{\textsc{ChainResponse}(a,b), \textsc{Response}(b,d),$ $\textsc{ChainResponse}(d,e), \textsc{ChainResponse}(e,c)\} \models\perp$
(4) $a \cup \textsc{NotResponse}(a,c) \cup \{\textsc{Response}(a,b), \textsc{Response}(b,d),$ $\textsc{ChainResponse}(d,e), \textsc{ChainResponse}(e,c)\} \models\perp$

Note that the activation α is augmented in line 8 to allow for this detection of quasi-inconsistent subsets via Definition 5. Concluding the example, as all 4 cases return true, we have successfully found four *mqis* based on the reactive entailment graph (cf. the formalization of these four *mqis* in Example 2, specifically $M_4 - M_7$).

5 Evaluation

We implemented an MQI-solver for Declare constraints. Our implementation takes as input a DECLARE constraint set C and returns as output $\mathsf{MQI}(C)$ and the introduced (quasi) inconsistency measures. We then performed run-time experiments on the following real-life data sets:

- BPI challenge 2017[2]. This data set contains an event log of a loan application process of a Dutch financial institute. The log is constituted of 1,202,267 events corresponding to 31,509 loan application cases.
- BPI challenge 2018[3]. This data set contains an event log of a process at the level of German federal ministries of agriculture and local departments. The log comprises 2,514,266 events corresponding to 43,809 application cases.
- Sepsis 2016[4]. This data set contains an event log of a hospital process concerning the treatment of sepsis, which is a life threatening condition. The log contains around 1000 cases with 15,000 events.

---

[2] `https://www.win.tue.nl/bpi/doku.php?id=2017:challenge`
[3] `https://www.win.tue.nl/bpi/doku.php?id=2018:challenge`
[4] `https://data.4tu.nl/repository/uuid:915d2bfb-7e84-49ad-a286-dc35f063a460`

We selected these data-sets because they provide recent data from real-life processes. Also, we selected these data-sets to analyze data of domains which are subject to a high degree of regulatory control and sensible to compliant process execution (e.g., financial-, government- and medical sector).

From these logs, we mined declarative process models using Minerful, which is a state-of-the-art tool for declarative constraint discovery [7]. As configuration for mining, we considered the three parameters of *support*, *confidence* and *interest*. The support threshold indicates the minimum number of traces a constraint has to be fulfilled in for it to be included in the discovered model. Confidence scales the support by the ratio of traces in which the activation occurs, resp. interest scales by the ratio of traces both the constrained tasks occur in. We ran Minerful with a support of 75%, confidence of 12.5% and interest factor of 12.5%, as proposed in the experiment design by [7]. We then applied our implementation to a) compute all minimal quasi-inconsistent subsets, and b) compute the $\mathcal{I}^{Q}_{\mathsf{MI}}$ quasi-inconsistency measures, as well as the $\mathit{\Gamma}^{Q}_{\#}$ culpability measures for all constraints. The experiments were run on a machine with 3 GHz Intel Core i7 processor, 16 GB RAM (DDR3) under macOS High Sierra Version 10.13.6.

Log	**BPI Challenge '17**	**BPI Challenge '18**	**Sepsis '16**
Constraints	305	70	207
$\mathcal{I}^{Q}_{\mathsf{MI}}$ (or # of *mqis*)	28954	25303	7736
Runtime	27074ms	10930ms	4379ms

Table 2: Results of run-time experiments for the analyzed data-sets

Table 2 shows an overview of the resp. mined constraints, as well as the number of detected *mqis*, and resp. quasi-inconsistency measures. For the model mined from the BPI'17 log, nearly 29.000 *mqis* were detected. The largest *mqi* had 17 elements. Here, the REG could efficiently be used to detect this subset via a path-based search. In the BPI'18 model, the largest *mqi* contained 22 elements. Also, 62 of the 70 discovered constraints were part of the overall inconsistency (as opposed to only 87/305 constraints in the BPI'17 log). Interestingly, only 70 constraints still lead to a high amount of *mqis*. In the Sepsis'16 log, there were roughly 7.700 *mqis*.

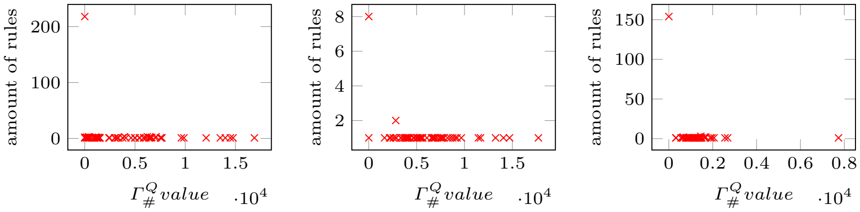
Example 6. For illustration, the following shows an actual *mqi* that we detected in the BPI'17 model.

$\textsc{CoExistence}(A\_Accepted, A\_Concept)$,
$\textsc{ChainResponse}(A\_Concept, W\_Validateapplication)$.
$\textsc{ChainResponse}(W\_Validateapplication, W\_PersonalLoancollection)$,
$\textsc{CoExistence}(A\_Accepted, A\_CreateApplication)$,
$\textsc{Response}(A\_CreateApplication, O\_Sent)$,
$\textsc{NotCoExistence}(W\_PersonalLoancollection, O\_Sent)$

This actual constraint set returned by the discovery algorithm is quasi-inconsistent. First, *A_Accepted* and *A_Concept* are constrained to appear together. Then, *A_Concept* transitively entails *W_PersonalLoancollection* via two ChainResponse constraints. Also, *A_Accepted* and *A_CreateApplication* are constrained to appear together. Then, because *A_CreateApplication* occurs, the task *O_Sent* must occur later. However, the last constraint demands that *O_Sent* and *W_PersonalLoancollection* never occur together, both of which are however entailed. In result, this is a quasi-inconsistent subset with the activation *A_Accepted*. Note that the discovery algorithm however did not return a constraint such as Participation(*A_Accepted*). Thus, this set of constraints returned by the miner is not inconsistent per se and thus cannot be detected as problematic with existing approaches. Yet, we argue that such a set of constraints should not be contained in any declarative process model, as it is highly confusing and potentially makes the model unusable in practice. Here, our approach allows to detect such problematic sets of constraints as quasi-inconsistent subsets. Table 2 shows that a high number of these *mqis* was actually returned by the miner for all three analyzed logs. As identifying such amount of problematic subsets manually is unfeasible, our approach therefore contributes a feasible means to detect problematic constraints and thus to improve model quality.

In the scope of identifying the actual causes of inconsistency, culpability measures can be used to quantify the degree of blame that individual constraints carry [9]. For the three discovered models, we therefore computed the respective $\mathit{\Gamma}^{Q}_{\#}$ values for all constraints.

Figure 3 shows the distribution of the $\mathit{\Gamma}^{Q}_{\#}$ culpability values for the constraints mined from the respective logs. The x-axis shows the respective culpability value, while the y-axis shows the number of constraints with this value. What can be seen for all analyzed models is that we have a high number of constraints with a culpability value of 0 (i.e. they are not part of any *mqi*), and only a few number of individual constraints which are highly responsible in the context of the overall inconsistency (i.e. they are part of many *mqi*). For example, in the constraint set mined from the BPI'2017 log, there are around 200 constraints with a culpability of 0, which can thus be seen as unproblematic.



(a) (a) BPI challenge 2017 (b) (b) BPI challenge 2018 (c) (c) Sepsis 2016

Fig. 3: Distribution of culpability values for the constraints in the respective models, using the $\mathit{\Gamma}^{Q}_{\#}$ measure.

This equates to $\frac{2}{3}$ of all constraints. It is thus possible to identify those (roughly) 100 constraints, which should be attended to. We argue that this is a valuable piece of business intelligence and increases efficiency in managing constraints. Here, the corresponding culpability ranking is a further driver for understanding inconsistencies in the context of resolution strategies. That is, for all the considered models, a few number of constraints can be identified that have the highest culpability values. Thus, these constraints can be strategically targeted first to allow for an effective inconsistency resolution. This is evident in the model mined from the Sepsis 2016 log. There was one specific constraint which was part of all *mqis*, namely $\textsc{Response}(\mathit{AdmissionNC}, \mathit{ReleaseA})$. If one would delete this constraint, all quasi-inconsistencies would be resolved. This information could therefore be exploited for effective resolution means. As a further example, the model derived from the BPI'17 log contained the constraint $\textsc{Response}(\mathit{A\_Incomplete}, \mathit{O\_Accepted})$, which had the highest $\mathit{\Gamma}^{Q}_{\#}$ value of 16890, meaning one could eliminate over 60% of all *mqis* while deleting only one constraint.

To summarize, due to the distribution of culpability values, it would be possible to resolve all quasi-inconsistent subsets through targeting selected constraints via the culpability ranking and deleting only these few elements. This would allow to mitigate all *potential* inconsistencies, i.e. implicit inhibition sets, with a low amount of information loss. As mentioned, this is clearly shown for the model of the Sepsis'16 log, where it would be possible to resolve all *mqis* while deleting only one constraint. In result, we argue that our analysis capabilities by the means of culpability measures provide valuable business insights that can be used as a basis for an informed resolution strategy.

6 Related Work

Our work is related to the discipline of business rules management, i.e., ensuring a consistent set of business rules. In this context, companies have to be supported with means to ensure design-time compliance of declarative process models. While there are some approaches that are aimed to solve problems as discussed in this work by design, i.e. during modelling, this work is related to works that assess an existing set of constraints. This is relevant when existing constraints have to be analyzed, which can be often the case, e.g. analyzing the constraints discovered in process discovery, analyzing a previously modelled set of constraints or analyzing a merged set of constraints after company mergers. A closely related work is that by Di Ciccio et al. (2017), who focus on resolving redundancies and inconsistencies in declarative process models. However, as discussed in the introduction, those authors define inconsistency as a model which cannot accept any traces, i.e. it is unsatisfiable. To detect such inconsistencies, those authors represent declarative constraints as finite state automata $\mathcal{A}$, and denote $\mathcal{L}(\mathcal{A})$ as the set of strings accepted by $\mathcal{A}$. Then, those authors can detect inconsistent constraints by identifying those constraint sets that are unsatisfiable via automata products, i.e. $\mathcal{L}(\mathcal{A}') = \emptyset$. As motivated in our introduction, quasi-inconsistent constraint sets can still accept an arbitrary set of traces. Thus, quasi-inconsistency cannot be detected by existing means. Our contribution relative to [7] can be seen in the analysis of the BPI'17 log, which was also analyzed by those authors. Where our approach found nearly 29.000 *potential* inconsistencies, those authors reported 2 inconsistencies. While not inconsistent per se, we argue that quasi-inconsistent sets of constraints such as in $\mathcal{C}_1$ or $\mathcal{C}_2$ should still not be contained in declarative process models, as they can potentially make the model unusable and are highly confusing to modelers. Here, to the best of our knowledge, our approach is the first to offer a tractable solution for detecting all sets of potentially contradictory constraints, i.e. minimal implicit inhibition sets, in declarative process models.

7 Conclusion

In this work, we presented the novel concept of quasi-inconsistencies in declarative process models. As quasi-inconsistencies *potentially* make the model unusable, it is important to detect such problems. Here, we proposed a first approach for such a detection. Through the proposed inconsistency measures, companies are presented with quantitative insights regarding model quality. Element-based culpability measures furthermore allow to prioritize problematic constraints and pin-point individual constraints which should be attended to. Through a computation of MQI based on the reactive entailment graph, our approach is applicable to arbitrary reactive constraints.

Future work could be directed towards the integration of our results, especially the proposed analysis capabilities. In the scope of process discovery, inconsistency measures could be used to as a metric to evaluate the quality of discovered constraint sets. For example, users could define a threshold of allowed quasi-inconsistency. Also, the quantitative insights provided by culpability measures could be used to pin-point the actual causes of quasi-inconsistency, and could thus be integrated into existing methods for resolving errors in declarative process models, e.g. [7], or as a basis for cost-analysis in trace alignment [5].

References

1. Burattin, A., Maggi, F.M., Sperduti, A.: Conformance checking based on multi-perspective declarative process models. Expert Sys. with Applications **65** (2016)
2. Cecconi, A., Di Ciccio, C., De Giacomo, G., Mendling, J.: Interestingness of traces in declarative process mining: The janus ltlpf approach. In: International Conference on Business Process Management (2018)
3. Corea, C., Delfmann, P.: Supporting business rule management with inconsistency analysis. In: Proceedings of the BPM 2018 Industry Track (2018)
4. De Giacomo, G., De Masellis, R., Montali, M.: Reasoning on ltl on finite traces: Insensitivity to infiniteness. In: AAAI (2014)
5. De Giacomo, G., Maggi, F.M., Marrella, A., Patrizi, F.: On the disruptive effectiveness of automated planning for ltl f-based trace alignment. In: Thirty-first AAAI conference on artificial intelligence (2017)

6. van Der Aalst, W.M., Pesic, M., Schonenberg, H.: Declarative workflows: Balancing between flexibility and support. Computer Science-Research and Development **23**(2), 99–113 (2009)
7. Di Ciccio, C., Maggi, F.M., Montali, M., Mendling, J.: Resolving inconsistencies and redundancies in declarative process models. Information Systems **64** (2017)
8. Dwyer, M.B., Avrunin, G.S., Corbett, J.C.: Patterns in property specifications for finite-state verification. In: Proceedings of the 21st international conference on Software engineering (1999)
9. Grant, J., Hunter, A.: Measuring consistency gain and information loss in stepwise inconsistency resolution. In: Proceedings of the 11th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (2011)
10. Grant, J., Martinez, M.V.: Measuring Inconsistency in Information. College Publications (2018)
11. Hunter, A., Konieczny, S., et al.: Measuring inconsistency through minimal inconsistent sets. KR **8** (2008)
12. Maggi, F.M., Di Ciccio, C., Di Francescomarino, C., Kala, T.: Parallel algorithms for the automated discovery of declarative process models. Information Systems **74** (2018)
13. Markey, N.: Past is for free: on the complexity of verifying linear temporal properties with past. Acta Informatica **40**(6-7) (2004)
14. Pesic, M.: Constraint-based workflow management systems: shifting control to users (2008)
15. Sadiq, S., Governatori, G.: Managing regulatory compliance in business processes. In: Handbook on Business Process Management 2, pp. 265–288. Springer (2015)
16. Thimm, M.: On the expressivity of inconsistency measures. Artificial Intelligence **234** (2016)
17. Thimm, M.: On the evaluation of inconsistency measures. In: Grant, J., Martinez, M.V. (eds.) Measuring Inconsistency in Information, Studies in Logic, vol. 73. College Publications (February 2018)

7.5 Resolving Inconsistencies in Declarative Process Models based on Culpability Measurement

Source	Resolving Inconsistencies in Declarative Process Models based on Culpability Measurement
Authors	Carl Corea, Matthias Deisen and Patrick Delfmann
Outlet	WI 2019 - 14. Internationale Tagung der Wirtschaftsinformatik, February 23-27, 2019, Siegen, Germany
Ranking	C
Status	Published
Comment	**Winner of the Best-Paper Award**

Resolving Inconsistencies in Declarative Process Models based on Culpability Measurement

Carl Corea, Matthias Deisen, and Patrick Delfmann

Institute for Information System Research, University of Koblenz-Landau, Germany
`{ccorea,deisen,delfmann}@uni-koblenz.de`

Abstract. Contrary to traditional process models, declarative process models define a set of declarative constraints to specify the behavior which a process should adhere to. In the scope of process mining, declarative process discovery aims to derive such constraint sets from event logs. Here, a problem for current discovery techniques is that of inconsistency. That is, dependent of certain event log characteristics, the derived constraint set may contain contradictory constraints. This in turn however makes the discovered model unusable, as contradictory constraints make it impossible to execute declarative process models, thus hampering previous process discovery efforts. In this work, we present an approach for resolving inconsistencies in declarative process models, based on methods from the scientific field of inconsistency measurement. We introduce our approach algorithm and evaluate its feasibility with real life data sets of the BPI Challenge 2017.

Keywords: Declarative Process Models, Inconsistency Resolution, Declare

1. Introduction

Process Discovery is a key part of Process Mining and comprises techniques for the automated derivation of process models [18, 20]. The creation of processes models is performed by the means of discovery algorithms and techniques, which are applied to entail process models based on observed behaviour, for example that in event logs [6, 17]. Recent efforts have been directed towards so-called declarative process models [17]. Where procedural process models provide an imperative description of how exactly company processes should be performed, declarative process models consist of a set of constraints, that must not be violated. Hence, contrary to the explicitly confined process execution in procedural process models, process execution in declarative process models is all allowed behaviour within the set of constraints.

As the declarative constraint set defines how processes can be executed, the quality of this set is of central importance to companies, in order to ensure a correct and compliant process execution. A potential problem here is that of inconsistency [6]. That is, the set of constraints must not contain any contradictory constraints, as this can make it impossible to perform the process such that it satisfies the set of declarative constraints, i.e. the declarative process model cannot be executed.

Ensuring the consistency of declarative process models in the scope of process discovery is widely recognized as a challenging task [6, 8, 15]. In cases of inconsistency, methods are needed to provide a careful analysis and (semi-) automated resolution capabilities, as a manual analysis and resolution is not feasible in practice due to the potential size of declarative process models.

In this paper, we describe an approach that can automatically resolve inconsistencies in declarative process models. The approach applies quantitative measures from the scientific field of inconsistency measurement [10]. Such measures allow to assign a numerical value to individual constraints, with the informal meaning that a higher value reflects a higher degree of inconsistency. In result, this quantitative assessment is used to pin-point and delete causes of inconsistency in the declarative process model while ensuring a low amount of information loss. We implemented our approach and conducted an evaluation of data sets of the Business Process Intelligence Challenge 2017.

The remainder of this work is structured as follows. In Section 2, we provide a motivational example and introduce preliminaries. Next, Section 3 introduces our approach algorithm to resolve inconsistencies in declarative process models. The evaluation of our implemented algorithm is presented in Section 4. Last, we discuss our approach in the context of related work in Section 5 and conclude in Section 6. As a design choice, we base our approach on the Declare formalisms, which is a widely acknowledged modelling language for declarative process models [6, 12, 17]. Our approach can however be extended to arbitrary declarative process modelling languages as defined subsequently.

2. Background

This section provides an example for the necessity of resolving inconsistencies in declarative process models. Also, prerequisites for the Declare modelling language and culpability measurement are introduced.

2.1. Declare Modelling Language and Motivational Example

Declarative process models can be defined as a set of constraints, modelling the allowed behaviour that processes should adhere to.

Definition 1 (Declarative Process Model). A declarative process model is a tuple $\boldsymbol{M} = (\boldsymbol{A}, \boldsymbol{T}, \boldsymbol{C})$, where $\boldsymbol{A}$ is a set of *tasks*, $\boldsymbol{T}$ is a set of pre-defined constraint *templates*, and $\boldsymbol{C}$ is the set of actual *constraints*, which instantiate the template elements in $\boldsymbol{T}$ with activities in $\boldsymbol{A}$.

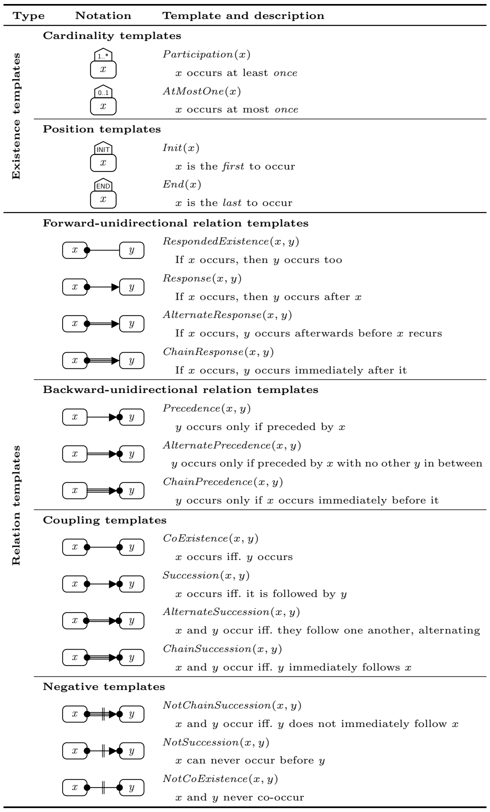
In this paper, we consider Declare, which is a popular declarative modelling language based on a set of constraint templates. Such templates can be used to define declarative constraints by passing tasks as parameters. For instance, the template *init(x)* indicates that every process instance must start with a task *x*. Declare can be used to define the *relationships* between two individual tasks, by the means of so-called relationship templates, each taking two input parameters. For example, *Response(x,y)* indicates that for any process instance, if *x* occurs, then later *y* must occur. The template *ChainResponse(x,y)* denotes that for any process instance, if *x* occurs, then *y* must occur *immediately* after *x*, i.e., a directly follows relation. Due to space limitations, we ommit a further detailed discussion of all relationship templates. An overview and a short description is provided in Figure 1. Please see [6, 17] for a further discussion.

The template relationships can be hierarchically ordered. Figure 2 shows the subsumption hierarchy of relational Declare templates considered in this work. For example, a *ChainResponse* between tasks *a* and *b* is also a *Response* between *a* and *b*.

Three important templates for the remainder of our discussion are the so-called *negative* relationship templates *NotCoExistence(x,y)*, *NotSuccession(x,y)* and *NotChainSuccession(x,y)*. *NotCoExistence(x,y)* states that tasks *x* and *y* must never appear in the same process instance. *NotSuccession(x,y)* states that *y* must never occur after *x* in a process instance (regardless of how many tasks are in-between *x* and *y*). *NotChainSuccession(x,y)* states that *y* must not *directly* follow *x* for any process instance. These three negative templates can cause inconsistency of the overall declarative process model in combination with other templates. For example, *ChainSuccession(a,b)* and *NotChainSuccession(a,b)* must not appear simultaneously in the same constraint set, as this is a logical contradiction. As the set of Declare templates is pre-defined, it is possible to investigate which combinations of templates can cause inconsistencies. Here we distinguish between three types of Declare inconsistencies, namely *trivial inconsistencies*, *generalization-based inconsistencies* and *path-based inconsistencies*.

- **Trivial Inconsistencies.** We define trivial inconsistencies as the co-existence of any negative constraint and its direct complement, i.e. with the same parameters, in the same constraint set. That is, any constraint-set with the templates *CoExistence(x,y)* and *NotCoExistence(x,y)*, OR, *Succession(x,y)* and *NotSuccession(x,y)*, OR, *ChainSuccession(x,y)* and *NotChainSuccession(x,y)* is trivially inconsistent.

- **Generalization-based Inconsistencies.** The subsumption hierarchy entails that generalization relations can also impose inconsistencies in combination with negative templates. For instance, every *ChainSuccession(x,y)* is also a *Succession* between *x* and *y*. Therefore, the two constraints *ChainSuccession(x,y)* and *NotSuccession(x,y)* are contradictory to each other. Based on the subsumption hierarchy shown in Figure 2, all possible combinations of generalization-based inconsistencies can be defined. Intuitively, *NotCoExistence(x,y)* contradicts all other (non-negative) templates with the parameters *x* and *y*, as any possible occurrence of both *x,y* contradicts the *NotCoExistence* of *x* and *y*. *NotSuccession(x,y)* contradicts *Precedence(x,y), Succession(x,y), Response(x,y)* and all inheriting template types. Last, *NotChainSuccession(x,y)* contradicts *ChainPrecedence(x,y), ChainSuccession(x,y)* and *ChainResponse(x,y)*.



Type	Notation	Template and description
Existence templates	**Cardinality templates**	
	1..* x	*Participation*(x) x occurs at least *once*
	0..1 x	*AtMostOne*(x) x occurs at most *once*
	Position templates	
	INIT x	*Init*(x) x is the *first* to occur
	END x	*End*(x) x is the *last* to occur
Relation templates	**Forward-unidirectional relation templates**	
	x y	*RespondedExistence*(x, y) If x occurs, then y occurs too
	x y	*Response*(x, y) If x occurs, then y occurs after x
	x y	*AlternateResponse*(x, y) If x occurs, y occurs afterwards before x recurs
	x y	*ChainResponse*(x, y) If x occurs, y occurs immediately after it
	Backward-unidirectional relation templates	
	x y	*Precedence*(x, y) y occurs only if preceded by x
	x y	*AlternatePrecedence*(x, y) y occurs only if preceded by x with no other y in between
	x y	*ChainPrecedence*(x, y) y occurs only if x occurs immediately before it
	Coupling templates	
	x y	*CoExistence*(x, y) x occurs iff. y occurs
	x y	*Succession*(x, y) x occurs iff. it is followed by y
	x y	*AlternateSuccession*(x, y) x and y occur iff. they follow one another, alternating
	x y	*ChainSuccession*(x, y) x and y occur iff. y immediately follows x
	Negative templates	
	x y	*NotChainSuccession*(x, y) x and y occur iff. y does not immediately follow x
	x y	*NotSuccession*(x, y) x can never occur before y
	x y	*NotCoExistence*(x, y) x and y never co-occur

Figure 1. Overview of Declare templates (Taken from [6])

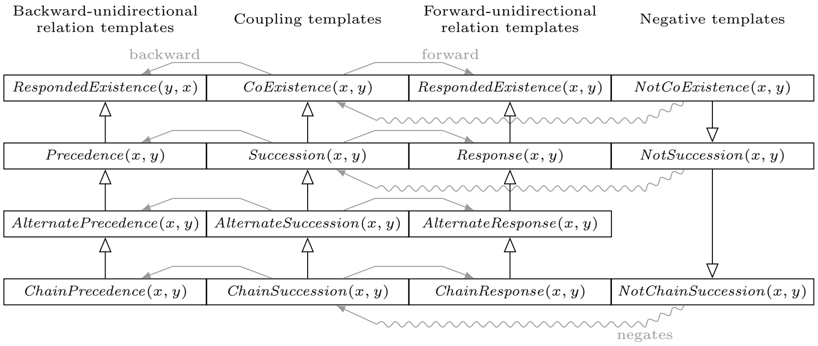


Figure 2. Declare Subsumption Hierarchy (Taken from [6])

- **Path-based Inconsistencies.** So far, we have only discussed the relation of templates with identical parameter signature, e.g. *(x,y)* for both templates. However, *transitive* relations between individual constraints must also be considered. Consider a declarative constraint set consisting of *ChainSuccession(a,b), ChainSuccession(b,c)* and *NotSuccession(a,c)*. The first two constraints state that for every process instance, *a* must directly be followed by *b*. Accordingly, *b* must be directly followed by *c*. Yet, the constraint *NotSuccession(a,c)* demands that *a* must never occur before *c* in the same process instance. Thus, the demands of the first two constraints are contradictory with *NotSuccession(a,c)*. Therefore, inconsistency can occur between groups of templates even if they don't have the same parameters. The definition of path-based inconsistencies will be further discussed in Section 3 based on the definition of so-called *task entailment graphs*.

To clarify, consider the following declarative process model D_1 based on Declare.

ChainSuccession(a,b) *(i)*
NotChainSuccession(a,b) *(ii)*
NotSuccession(a,b) *(iii)*
ChainSuccession(b,c) *(iv)*
NotSuccession(a,c) *(v)*
Response(b,d) *(vi)*
ChainResponse(d,e) *(vii)*
ChainResponse(e,c) *(viii)*
ChainResponse(a,b) *(ix)*

Figure 3. Exemplary Declare model D_1

First, the constraint set in Figure 3 has a trivial inconsistency, as *(i) ChainSuccession(a,b)* and *(ii) NotChainSuccession(a,b)* is contradictory. Also, there are generalization-based inconsistencies. *NotSuccession(a,b)* also entails *NotChainSuccession(a,b)*, therefore, line *(iii)* contradicts line *(i)*. This also yields a conflict between line (iii) and line (ix). Next, the constraints in lines *(i)* and *(iv)* constrain the tasks *abc* as a valid process instance. This path is in contradiction to *(v) NotSuccession(a,c)*, thus a path-based inconsistency. Also, the constraints in line *(i), (vi), (vii)* and *(viii)* define the sequence *abdec* as a valid process instance, which constitutes a path-based inconsistency to *(v) NotSuccession(a,c)*. Path-based inconsistencies also arise for *(ix)*, *(vi), (vii), (viii)* with *(v)*, and (ix), (iv) with (v).

The question may arise how a constraint set such as in Figure 3 can even exist in the first place. Following Di Ciccio et al. (2017), Declare models are mostly automatedly generated in the scope of declarative process discovery. A central problem here are certain completeness charactericstics in the underlying event logs. Recording errors or irregularities in process execution can lead to incomplete or distorted logs. To cope with such problems, most declarative process discovery algorithms introduce the notion of a *support factor*, that is, a parameter defining the fraction of traces in which a discovered template must occur in, in order to accept it in the resulting declarative constraint set [6]. Due to the mentioned problems of log completeness, it can make sense to lower this support factor. However, while this increases the amount of constraints, this does not ensure that these constraints are consistent in relation to each other. Thus, methods for post-evaluation in process discovery are needed to ensure that the returned constraint set is consistent.

2.2. Culpability Measurement

A scientific field concerned with the analysis of inconsistent information is the field of Inconsistency Measurement, cf. Grant and Martinez (2018). Here, a central object of study are quantitative measures, which allow to assign a numerical value to (elements of) a constraint set, with the informal meaning that a higher value reflects a higher degree of inconsistency, cf. Thimm (2016) for a survey.

A core notion of our approach are *culpability measures*, which are a type of mentioned quantitative measures [5]. Informally, culpability assesses the degree of blame that an individual constraints carries in the context of the overall inconsistency of the constraint set [11]. Let $\mathfrak{D}$ be the set of all declare constraint sets, and $\mathfrak{C}$ the set of individual constraints in a set $\in \mathfrak{D}$. Then, a culpability measure C is a function

$$C : \mathfrak{D} \times \mathfrak{C} \rightarrow [0, \infty) \tag{1}$$

which assigns a non-negative real value to a mapping of a declare constraint set and an individual constraint.

An example is the so-called $C_{\#}$ measure [11] which assesses the culpability of an constraint γ for a constraint set $\boldsymbol{D}$ via

$$C_{\#}(\boldsymbol{D}, \gamma) = | m \in \boldsymbol{MIS}(\boldsymbol{D}) \mid \gamma \in m | \tag{2}$$

This measure is based on so-called minimal inconsistent subsets of the constraint set $\boldsymbol{D}$. Given a constraint set $\boldsymbol{D}$, the minimal inconsistent subsets (***MIS***) of $\boldsymbol{D}$ are defined via

$$\boldsymbol{MIS}(\boldsymbol{D}) = \{\boldsymbol{D'} \subseteq \boldsymbol{D} \mid \boldsymbol{D'} \textit{ is inconsistent and minimal in terms of set inclusion}\} \tag{3}$$

This definition of minimal inconsistent subsets can be used to find inconsistencies in declarative constraint sets. We revisit the exemplary constraint set D_1 from Figure 3. Figure 4 shows the minimal inconsistent subsets of D_1.

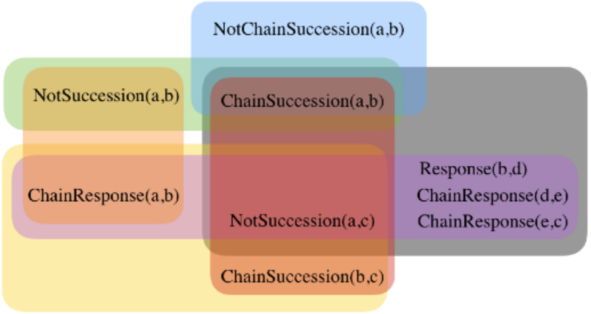


Figure 4. Minimal Inconsistent Subsets for Example D_1 from Figure 3

Based on these ***MIS***, the $C_\#$ measure counts the number of minimal inconsistent subsets that a constraint γ belongs to. Applying the $C_\#$ measure for the ***MIS*** shown in Figure 4 results in the quantification shown in Figure 5.

$C_\#(D_1, ChainSuccesion(a,b)) = 4$
$C_\#(D_1, NotChainSuccession(a,b)) = 1$
$C_\#(D_1, NotSuccession(a,b)) = 2$
$C_\#(D_1, ChainSuccession(b,c)) = 2$
$C_\#(D_1, NotSuccession(a,c)) = 4$
$C_\#(D_1, Response(b,d)) = 2$
$C_\#(D_1, ChainResponse(d,e)) = 2$
$C_\#(D_1, ChainResponse(e,c)) = 2$
$C_\#(D_1, ChainResponse(a,b)) = 3$

Figure 5. Constraint Culpability for Example D_1 from Figure 3

As a driver for inconsistency resolution, the authors in [5] introduced the notion of a *culpability ranking*, which orders the constraints by their degree of culpability. For the quantification shown in Figure 5, this leads to the following ranking

$$\langle i, v, ix, iii, iv, vi, vii, viii, ii \rangle \tag{4}$$

indicating a prioritization of which constraints should be attended to.

In the following Section, we discuss how this culpability ranking can be exploited for inconsistency resolution and show our algorithm, which includes the computation of minimal inconsistent subsets, culpability assessment and ranking.

3. Inconsistency Resolution Algorithm

3.1 Approach

To resolve inconsistency in a declarative constraint set, there are generally two possibilities [6]: One, a *new* constraint set is constructed, by iteratively moving elements of the old set to the new set, if their introduction does not cause an inconsistency. This technique can be compared to an optimisation problem. A potential disadvantage here is an information loss due to local optima [6]. A different possibility to resolve inconsistency is to use the original constraint set, and iteratively *delete* elements until the constraint set is consistent again. In this context, Grant and Hunter (2011) denote this approach as stepwise inconsistency resolution, and discuss its advantages related to less information loss. In order to resolve inconsistencies with the goal of mitigating information loss, we therefore base our approach on the latter approach of stepwise resolution. Let a constraint set $\boldsymbol{D}$, the deletion of a constraint γ from $\boldsymbol{D}$ is defined via $deletion(\gamma) = \boldsymbol{D} \setminus \{\gamma\}$. Stepwise inconsistency resolution by *deletion* is thus a sequence of deletions $S = <d_1,\ldots,d_n>$. We denote $d_\gamma(\mathbf{D})$ as the constraint set obtained from deleting the element γ from $\boldsymbol{D}$. We denote $\mathbf{\Gamma}_S = < \gamma_1,\ldots, \gamma_n>$ as the individual constraints deleted in the Sequence S via the deletions $d_1,\ldots,d_n$.

The challenge is to find suitable elements which to delete, such that the sequence *S* results in a low amount of information loss. The authors in [11] do not provide means to determine which elements should be deleted. Here, our algorithm extends the approach by [11] and utilizes the introduced culpability measure $C_{\#}$ to automatedly determine which elements to delete. Let a constraint γ with a $C_{\#}$ value of n_{γ}, then, deleting γ results in the resolution of n_{γ} minimal inconsistent subsets [9]. Thus, we exploit the culpability ranking to find the constraint with the highest culpability, which maximises the number of resolved minimal inconsistent subsets with loosing only one constraint. For example, as can be seen from Figure 4, deleting the constraint *ChainSuccession(a,b)* with the highest culpability value of 4 would result in the deletion of 4 minimal inconsistent subsets.

We iteratively perform the steps of *analysis*, *ranking* and *deletion* until the constraint set is consistent. In the following, we discuss the these steps of our algorithm in detail.

3.2 Algorithm

Our algorithm is shown in Figure 6.

Algorithm 1 Inconsistency Resolution

```
1:  procedure RESOLVEINCONSISTENCIES($constraintSet$)
2:      $mis \leftarrow computeMIS(constraintSet)$
3:      $tempMaxConstraint \leftarrow \varnothing$
4:      $tempValue \leftarrow 0$
5:      while $mis.size > 0$ do
6:          for each $c : constraintSet$ do
7:              $culpability \leftarrow \; | m \in mis \mid c \in m |$
8:              if $culpability > tempValue$ then
9:                  $tempMaxConstraint \leftarrow c$
10:                 $tempValue \leftarrow culpability$
11:         $constraintSet.delete(tempMaxConstraint)$
12:         $mis \leftarrow computeMIS(ConstraintSet)$
```

Figure 6. Approach Algorithm

The algorithm takes as input parameter a set of constraints. A first step is the computation of minimal inconsistent subsets in line 2. (cf. the below discussion). The algorithm then performs the steps of a) analyzing the constraint with the highest culpability (lines 6-10), and b) deleting the constraint with the highest culpability (line 11), in an iterative manner, while there are still inconsistencies (line 5). To clarify, the deletion of constraints intuitively bears the danger of information loss. Yet, deleting constraints might be necessary to restore consistency, i.e. if there exist inherently contradictory constraints. In this context, our approach offers a recommendation as to which element should be deleted, with the goal of maximizing consistency while deleting the lowest amount of constraints possible. We also store the deleted elements and present them to the user for further inspection after the fully automated resolution process.

A substantial part of our algorithm is the computation of minimal inconsistent subsets, shown in Figure 7. The function depicted in Figure 7 takes as input parameter the declarative constraint set, and returns the set of minimal inconsistent subsets. We initialize an empty set *mis* for storing minimal inconsistent subsets (line 2). Then, the negative template *types*, as well as the individual *constraints* in the *constraintSet* are defined (lines 3-4). In the following, we denote constraints of the types defined in line 3 as *negative constraints*. We then proceed to compute inconsistencies.

Trivial Inconsistencies (lines 6-9). We iterate over all negative constraints to verify if there is a corresponding complement template with the identical parameter signature. This is performed with the method *findRulesByTypeAndParams(type, params)*, which returns a set of sets. This function takes as input a specific template type and parameters (i.e. an ordered pair of tasks, e.g. *a,b*) and returns a set of sets, where the inner sets contain all corresponding constraints. As depicted in line 7, we pass the complement of the respective negative constraints as a parameter. Consequently, the function used in line 7 returns all trivial inconsistencies to the respective constraint *c*. A new minimal inconsistent subset containing a pair with the found constraint and constraint *c* is added to the set of minimal inconsistent subsets.

```
Algorithm 2 Computation of Minimal Inconsistent Subsets
1:  function COMPUTEMIS($constraintSet$)
2:      $mis \leftarrow \{\emptyset\}$
3:      $negativeTemplates \leftarrow \{NotCoExistence, NotSuccession, NotChainSuccession\}$
4:      $negativeConstraints \leftarrow getConstraintsInType(negativeTemplates)$
5:
6:      for each $n : negativeConstraints$ do
7:          $contradictions \leftarrow getRulesByTypeAndParams(n.type.complement, n.params)$
8:          for each $c : contradictions$ do
9:              $mis \leftarrow mis \cup \{n, c\}$
10:
11:     for each $n : negativeConstraints$ do
12:         for each $compType : n.type.complementSet$ do
13:             $contradictions \leftarrow getRulesByTypeAndParams(compType, n.params)$
14:             for each $c : contradictions$ do
15:                 $mis \leftarrow mis \cup \{n, c\}$
16:
17:     $graph = new\ ConstraintGraph(constraintSet)$
18:     for each $ns : getConstraintsByType(NotSuccession)$ do
19:         $pathSet \leftarrow computeAllPaths(graph, ns.params)$
20:         for each $p : pathSet$ do
21:             $mis \leftarrow mis \cup \{ns, p.getConstraints()\}$
        return mis
```

Figure 7. Computation of Minimal Inconsistent Subsets in Declare

Generalization-Based Inconsistencies (lines 11-15). Section 2 defines the complements to the negative template types *NotCoExistence*, *NotSuccession* and *NotChainSuccession*. We denote these complements as the *complementSet* of the respective negative template. For all negative constraints γ, we iterate over all constraints of a template type contained within the *complementSet* of γ, with identical parameter signature (line 12). To compute inconsistencies, we again use the function *findRulesByTypeAndParams*. Here, we pass as parameters the current complement type to the current negative constraint, and the parameters of the negative constraint (line 13). This yields all generalization-based inconsistencies to the current negative constraint. We add a set containing the negative constraint and the computed contradicting constraint to the set of minimal inconsistent subsets (line 15).

Path-Based Inconsistencies (lines 17-21). After the previous two steps, all constraints in contradiction to *NotCoExistence*- and *NotChainSuccession* constraints have been identified. The detection of all contradictions to templates of type *NotSuccession* requires a path-based perspective, due to inconsistencies arising through transitivity. To find path-based inconsistencies, we construct a graph-like structure consisting of the relations between individual Declare constraints (line 17). We denote this as a task entailment graph.

Definition 2 (Task Entailment Graph). Let the declarative process model $\boldsymbol{M}$, the task entailment graph of $\boldsymbol{M}$ is a tuple $\boldsymbol{G} = (\boldsymbol{A}, \boldsymbol{E}, \boldsymbol{t})$, where $\boldsymbol{A}$ is the set of tasks in $\boldsymbol{M}$, $\boldsymbol{E} \subseteq \boldsymbol{A} \times \boldsymbol{A}$ is the set of directed edges between activities, and $\boldsymbol{t}$ is a function $\boldsymbol{t}: \boldsymbol{E} \rightarrow \boldsymbol{T}$, which maps an individual edge to a template type from $\boldsymbol{T}$ in $\boldsymbol{M}$.

An edge $\boldsymbol{e} \in \boldsymbol{E} = (a,b)$ indicates that the task a entails the task b via the constraint $\boldsymbol{t(e)}$. To compute path-based inconsistencies, we iterate over all negative constraints of type *NotSuccession* (line 18). Let the parameters of the respective negative constraint be a and b, then, we compute all paths from a to b in the task entailment graph (line 19). Each path from a to b constitutes a contradiction to *NotSuccesion(a,b)*, thus the path is added to the set of minimal inconsistent subsets (line 21).

We revisit the example constraint set from the example in Figure 3. All non-negative constraints are used to construct the task entailment graph, shown in Figure 8. The graph can be understood, such that task *b* is entailed by task *a* via the template type *ChainSuccession*. The rest of the graph is constructed accordingly.

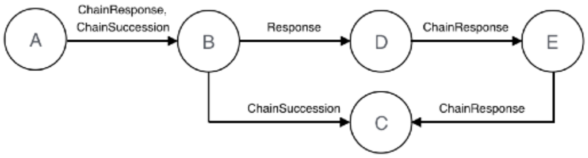


Figure 8. Task Entailment Graph for Example D_1 from Figure 3

The constraint set in this example also contained *NotSuccession(a,c)*. To search for a contradiction to this negative constraint, we verify if there are paths from *a* to *c*. In total, there are four possible paths from *a* to *c*, shown in Figure 9 (Note that the edge from task a to b in the graph has two types, due to the constraints *ChainResponse(a,b)* and *ChainSuccession(a,b)*). Via these four paths, *c* can transitively be entailed from *a.* Hence, all these paths contradict *NotSuccession(a,c)*. Accordingly, the constraint *NotSuccession(a,c)* and all constraints on the identified paths in the task entailment graph respectively constitute minimal inconsistent subset.

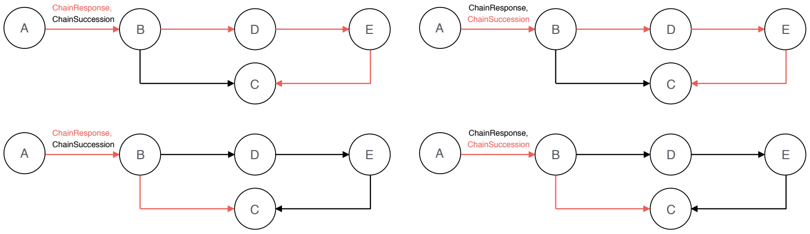


Figure 9. Paths found from task *a* to task *c*, for Example D_1 from Figure 3

In our algorithm, following the computed minimal inconsistent subset, the constraint with the highest $C_{\#}$ culpability is deleted. In the example, this is *ChainSuccession(a,b)* with a $C_{\#}$ value of 4. The proposed algorithm would therefore delete this constraint to generate $d_{ChainSuccession(a,b)}(D_1)$. Continuing the example, the element with the subsequent highest culpability would be *ChainResponse(a,b)* = 3. A deletion would consequently eliminate all remaining ***MIS***, with a total information loss of two constraints. The sequence $\mathbf{\Gamma}_S$ traces the constraints which were deleted for further inspection, i.e. here $\mathbf{\Gamma}_S$ = <*ChainSuccession(a,b) , ChainResponse(a,b)*>.

To conclude, our approach computes all minimal inconsistent subsets with the above algorithm. Then, we delete the constraint with the highest culpability value w.r.t. the $C_{\#}$ measure. This process is iteratively repeated until the set of constraint is consistent.

4. Evaluation

We implemented our approach to resolve inconsistencies in Declare models in Java. Our evaluation is based on data from the Business Processing Intelligence Challenge 2017 (BPI). In the scope of the BPI, real life event logs are provided.

The analyzed log[1] of the 2017 challenge is an event log of a Dutch financial institute and comprises 262.200 events in 13.087 cases. From this log, we mined a declarative process model in the Declare language using Minerful, which is a state-of-

---

[1] https://www.win.tue.nl/bpi/doku.php?id=2017:challenge

the-art tool for declarative process discovery [6]. We mined three different process models, with the respective support factors of 75%, 85% and 95%. We then applied our algorithm to all three models. Table 1 summarizes our evaluation results.

Table 1. Results from the application of our algorithm to the BPI challenge log

Support Factor	**75%**	**85%**	**95%**
Discovered Constraints	305	232	207
Initial number of MIS	28954	731	639
Deleted Elements needed	5	1	1
Information Loss	1,63 %	0,43%	0,48%
Runtime	101099ms	9148ms	4695ms

The declarative model mined with a support factor of 75% (*M1*) consisted of 305 constraints. We computed over 28000 inconsistencies in this constraint set, which shows that the lowering the support factor can result in inconsistent models, even for state-of-the-art discovery algorithms. Our algorithm was able to resolve all these subsets in *M1* by deleting a total of only 5 constraints, which equals a total information loss of 1.6% of all initial constraints. This was possible by iteratively determining the constraint with the highest culpability, as deleting this constraint warrants the highest number of resolved ***MIS*** for each iteration. Figure 10 shows an overview of the number of ***MIS*** remaining after each iteration of our algorithm for *M1*. As can be seen, in each of the first three iterations, the number of MIS is reduced roughly by half. Then, a significant reduction from roughly 4000 to 700 MIS can be achieved in iteration 4. Figure 10 also shows the $C_{\#}$ culpability value of the respective constraint with highest culpability in each iteration. As can be seen, the first iteration resolves around 16000 of all 28954 MIS.

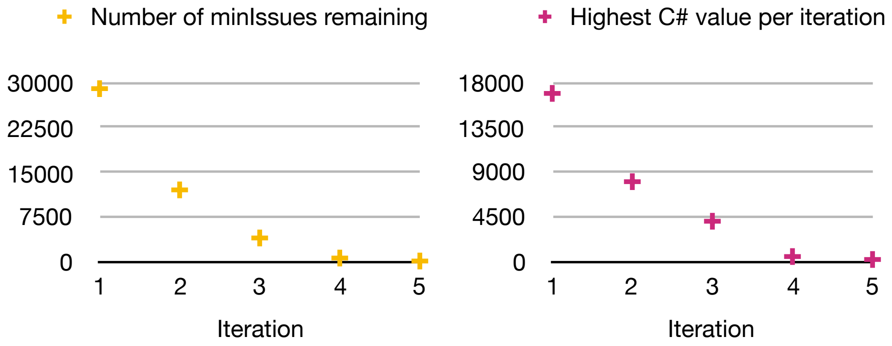


Figure 10. Number of MIS remaining (left) and highest C# culpability for a constraint (right) during the 5 algorithm iterations of resolving *M1*.

Specifically, for *M1*, the five constraints which needed to be deleted to completely resolve inconsistency are shown in Figure 11. The deletions as in Figure 11 can be presented to users. Should they deem this constraint must not be deleted under any circumstances, our approach can be extended with a „whitelist“.

Response(A_Incomplete, O_Accepted), $C_{\#} = 16890$
Precedence(W_Validateapplication, W_PersonalLoancollection), $C_{\#} = 8020$
Response(A_Incomplete, A_Pending), $C_{\#} = 3348$
Precedence(O_CreateOffer, W_PersonalLoancollection), $C_{\#} = 496$
Precedence(O_Created, W_PersonalLoancollection), $C_{\#} = 200$

Figure 11. Constraints deleted corresponding to Figure 10.

For the other two process models, mined with 85%, resp. 95%, support factor, we observed an interesting case. The algorithm was able to make the set of constraints consistent by deleting only one element. In both cases, this was the constraint *NotCoExistence(W_PersonalLoancollection, O_Sent)*. Thus, this constraint was part of all ***MIS*** in both models. Apparently, the support factor configuration yielded exactly this one constraint which makes the entire model unusable. This is a good case for our approach idea of determining the constraint that has the highest blame in the overall inconsistency. Based on culpability measurement, our algorithm could effectively resolve all inconsistencies in a low runtime, deleting only one constraint.

Our results show, that even for a low number of constraints, the number of inconsistencies can be rather large. This underlines the need for post-processing techniques in process discovery such as our approach. Our algorithm was able to resolve these inconsistencies.

5. Related Work

This work is related to consistency checking in Business Process Management (BPM) and declarative process discovery.

Consistency checking in BPM is widely recognized as a challenging task [2-6, 13, 15]. A core task here is to ensure the consistency and correctness of BPM related artifacts. We have focused on the area of verifying the consistency of declarative process model artifacts, respectively a resolution of inconsistency. There have been many proposals for the *qualitative* analysis of the reasons of inconsistencies [1, 7, 14]. However, works such as [13, 15, 19] point out the benefits of a *quantitative* analysis. The intuition here is that individual problems can have a different severity. Hence, using such a quantitative assessment as proposed offers a more sophisticated insight on how to resolve inconsistencies in declarative constraint sets [5, 13].

As a directly related work, we identify the report by Di Ciccio et al (2017). Those authors also proposed an approach to resolve inconsistencies in Declare constraint sets. Those authors however do not consider all ***MIS***, but rather build a maximal consistent *new* constraint set. As discussed, we adapt the approach by [9] of resolution by *deletion*. The authors in [6] point out, that the computation of inconsistencies by comparing all possible subsets of constraints is intractable. We agree, that this would be indeed unfeasible. To solve this problem, we therefore defined different inconsistency types in Declare, based on the set of pre-defined template types, which allows for informed comparisons and feasible computation. Our evaluation shows, that the computation of ***MIS*** can be performed in a feasible run-time.

Our algorithm promotes declarative process discovery. Due to the introduced notion of a support factor, existing process discovery techniques can yield inconsistent declarative process models [6, 17]. Our work contributes with an approach for automated resolution, allowing for effective post-processing techniques in process discovery.

6. Conclusion

In this work, we presented an approach for resolving inconsistencies in declarative process models. Our approach was implemented for process models in the Declare language. Our evaluation based on a real-life event log showed that it was possible to compute and resolve around 28.000 minimal inconsistent subsets in a feasible runtime. In the process model mined with 75% support factor in our evaluation, there were 305 constraints. However, only 87 constraints were part of any ***MIS*** (i.e. all other constraints had a $C_{\#}$ value of 0). We therefore argue it makes sense to only consider those constraints that contribute towards inconsistency, as in our approach. Deleting constraints with a $C_{\#}$ value of 0 cannot decrease inconsistency in Declare [9]. Deleting the element with the highest value resolves the most ***MIS***. To the best of our knowledge, this work is the first to incorporate culpability measurement.

In our evaluation, all models could be resolved with a deletion of max. 5 constraints, which equaled an information loss of under 2% in all models. This shows that our algorithm is aligned with our goal of ensuring a low amount of information loss. As a direct limitation, elements to delete are automatedly selected. Hence, it would be possible that knowledge which domain experts would not delete is automatedly deleted. To solve this problem, we store the sequence of deleted constraints for further inspection by domain experts. The aim of this paper was to present an approach for fully automated resolution approach. In future work, we will include the possibility to define constraints which must not be deleted. In case such „marked“ constraint would have the highest culpability value in an iteration run, it would simply be possible to delete the constraint with the next highest $C_{\#}$ value based on the computed culpability ranking.

A limitation of this work is, that our algorithm considers only the predefined set of Declare templates. While the majority of declarative discovery tools only focus on these templates, future work should be directed to find ***MIS*** in arbitrary constraints.

As a key learning, we observed that the support factor used during declarative process mining strongly impacts the resulting model. This should be considered by companies seeking to implement declarative process discovery. Automated approaches as presented in this work should be utilized to verify the consistency of the resulting models to allow for correct and compliant process execution.

References

1. Awad, A., Smirnov, S., and Weske, M. (2009). Towards resolving compliance violations in business process models. GRCIS. CEUR-WS. org.
2. Batoulis, K., Nesterenko, A., Repitsch, G., and Weske, M. (2017). Decision management in the insurance industry: Standards and tools. In Proceedings of the BPM 2017 Industry Track (BPM 2017), Barcelona, Spain, September 10-15, 2017., pages 52–63.
3. Batoulis, K. and Weske, M. (2017a). Soundness of decision-aware business processes. In International Conference on Business Process Management, pages 106–124. Springer.
4. Calvanese, D., Dumas, M., Laurson, U., Maggi, F. M., Montali, M., and Teinemaa, I. (2016). Semantics and analysis of dmn decision tables. In International Conference on Business Process Management, pages 217–233. Springer.
5. Corea, C. and Delfmann, P. (2018). Supporting business rule management with inconsistency analysis. In Proceedings of the BPM 2018 Industry Track (BPM 2018), Sydney, Australia, September 09-14, 2018.
6. Di Ciccio, C., Maggi, F. M., Montali, M., and Mendling, J. (2017). Resolving inconsistencies and redundancies in declarative process models. *Information Systems*, 64:425–446.
7. Ghose, A. and Koliadis, G. (2007). Auditing business process compliance. In International Conference on Service-Oriented Computing, pages 169–180. Springer.
8. Graham, I. (2007). Business rules management and service oriented architecture: a pattern language. John wiley & sons.
9. Grant, J., & Hunter, A. (2011). Measuring consistency gain and information loss in stepwise inconsistency resolution. In European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty (pp. 362-373). Springer, Berlin, Heidelberg.
10. Grant, J. and Martinez, M. V. (2018). Measuring Inconsistency in Information. College Publications.
11. Hunter, A., Konieczny, S., et al. (2008). Measuring inconsistency through minimal inconsistent sets. *KR*, 8:358–366.
12. Imgrund, F., Malorny, M., and Janiesch, C. (2017). Eine Literaturanalyse zur integration von business rules und business process management. 13. Internationale Tagung Wirtschaftsinformatik, WI 2017, St.Gallen, 2017.
13. Lu, R., Sadiq, S., and Governatori, G. (2008). Measurement of compliance distance in business processes. Information Systems Management, 25(4):344– 355.
14. Polyvyanyy, A., Ouyang, C., Barros, A., and van der Aalst, W. M. (2017). Process querying: Enabling business intelligence through query-based process analytics. *Decision Support Systems* 100 (2017): 41-56.
15. Sadiq, S. and Governatori, G. (2015). Managing regulatory compliance in business processes. In Handbook on Business Process Management 2, pages 265–288. Springer.
16. Thimm, M. (2016). On the compliance of rationality postulates for inconsistency measures: A more or less complete picture. *Künstliche Intelligenz*.
17. van Der Aalst, W. M., Pesic, M., and Schonenberg, H. (2009). Declarative workflows: Balancing between flexibility and support. Computer Science- Research and Development, 23(2):99–113.
18. van der Aalst, W. M. P., La Rosa, M., and Santoro, F. M. (2016). Business process management. Business & Information Systems Engineering, 58(1):1–6.
19. Weidlich, M., Mendling, J., and Weske, M. (2011). Efficient consistency measurement based on behavioral profiles of process models. IEEE Transactions, 37(3):410–429.
20. Weske, M. (2010). Business process management: concepts, languages, architectures. Springer Publishing Company, Incorporated.

7.6 Effects of Quantitative Measures on Understanding Inconsistencies in Business Rules

Source	Effects of Quantitative Measures on Understanding Inconsistencies in Business Rules
Authors	Carl Corea, Sabine Nagel and Patrick Delfmann
Outlet	HICSS 2019 - 52nd Hawaii International Conference on System Sciences, January 9-11, 2019, Maui, USA
Ranking	A
Status	Published

Effects of Quantitative Measures on Understanding Inconsistencies in Business Rules

Sabine Nagel
University of Koblenz-Landau
Institute for IS Research
Germany
snagel@uni-koblenz.de

Carl Corea
University of Koblenz-Landau
Institute for IS Research
Germany
ccorea@uni-koblenz.de

Patrick Delfmann
University of Koblenz-Landau
Institute for IS Research
Germany
delfmann@uni-koblenz.de

Abstract

Business Rules have matured to an important aspect in the development of organizations, encoding company knowledge as declarative constraints, aimed to ensure compliant business. The management of business rules is widely acknowledged as a challenging task. A problem here is a potential inconsistency of business rules, as business rules are often created collaboratively. To support companies in managing inconsistency, many works have suggested that a quantification of inconsistencies could provide valuable insights. However, the actual effects of quantitative insights in business rules management have not yet been evaluated. In this work, we present the results of an empirical experiment using eye-tracking and other performance measures to analyze the effects of quantitative measures on understanding inconsistencies in business rules. Our results indicate that quantitative measures are associated with better understanding accuracy, understanding efficiency and less mental effort in business rules management.

1. Introduction

Business Process Management (BPM) has received widespread adaptation in the development of today's organizations [1]. Here, so-called *business processes* allow representing company activities and their interrelations, which helps organizations to define how the business and its employees should function in order to collaboratively pursue company goals. A central objective in BPM is to warrant efficient and compliant processes [1]–[3]. *Business rules* can support companies through a specification of business logic relative to business processes [2], [4]. Through this formalization of business logic, companies can ensure that processes are aligned towards company goals and regulations.

Utilizing business rules to govern business processes however strongly assumes the consistency of business rules. To clarify, the set of business rules must not contain contradictions, as this impedes a correct usage of business rules. Figure 1 shows an example of such a contradiction. Here, two contradictory business rules determining the creditworthiness of a customer are shown. Due to the inconsistency of these business rules, it is not possible to use these rules for *decision making* in a corresponding process.

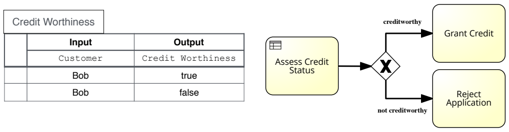


Figure 1: Exemplary process model and corresponding, inconsistent business rules

Recent research on analyzing business rules suggests that inconsistencies can be a problem in collaborative settings [5]–[11]. For instance, Batoulis and Weske [6] report on a recent case study with a large insurance company, where those authors found that 27% of analyzed business rules were erroneous. Such results emphasize the need to support companies with the capacity to manage inconsistencies in business rules [10], [11]. Next to a *detection* of inconsistencies, authors such as Lu et al. [10] or Sadiq and Governatori [11] suggest that companies should be provided with a *quantitative* analysis in the context of business intelligence, in order to promote an *understanding* of inconsistencies and consequently foster inconsistency resolution. The intuition is that some problems may be *more* severe than others and thus a quantification could provide means to assessing and prioritizing inconsistencies.

Despite the suggestion of authors in [10], [11], who explicitly point out that quantitative insights provide added-value to understanding problems in the scope of sustainable business rules management, this proposition has not yet been empirically evaluated.

In this report, we therefore investigate the theoretical foundation of how a quantitative analysis affects the understanding of inconsistencies in business rules. To this aim, we hypothesize the relationship between quantitative measures and understanding inconsistencies in business rules (Section 3). We present the results of an experiment showing that quantitative measures provide added-value to business rules management (Section 5). The underlying experiment design is introduced in Section 4. Our investigation is based on a preliminary discussion of inconsistency measurement in Section 2 and is concluded in Section 6.

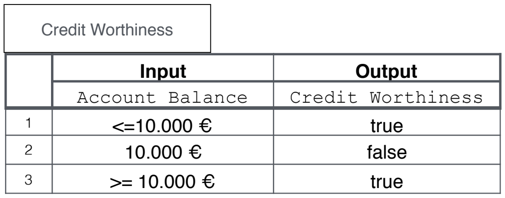
2. Background and Related Work

Business rules can be distinguished into constraints for data objects (structural rules) and rules defining the principles in which business activities should be performed (behavioral rules) [2]. In this work, we focus on behavioral rules. Following Graham [2], behavioral rules are of the general form

$$If\ I_1, \ldots, I_n \rightarrow O, \qquad (1)$$

where $I_1, \ldots, I_n$ is a premise comprising certain inputs, and O is the outcome of the rule which can be concluded if the premise is satisfied. Numerous standards to model business rules following this if/then structure have been introduced and adopted in practice [4]. As a design choice, we consider the Decision Model and Notation (DMN)[1] as a standard to represent business rules, as this can be seen as a current industry standard complementary to the Business Process Model and Notation (BPMN) standard for modeling business processes [6], [8].

DMN allows modeling business rules in decision tables. An example is shown in Figure 2.



Credit Worthiness

	Input	Output
	Account Balance	Credit Worthiness
1	<=10.000 €	true
2	10.000 €	false
3	>= 10.000 €	true

Figure 2: Exemplary DMN table containing inconsistencies

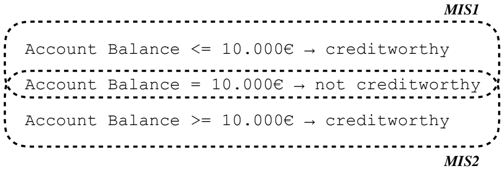
The columns in decision tables are used to denote the in- and outputs of a rule. The rows of decision tables relate to individual business rules. In Figure 2, one can observe three business rules modeling the creditworthiness of customers based on the input of *account balance*. Rule 1 in Figure 2 can be read as „*if* the Account Balance is <= 10.000€, *then* the customer is credit-worthy". The other two rules can be interpreted analogously. As can be seen, the set of business rules is inconsistent, due to an overlap and contradictory conclusions of rule 2 with respect to the other rules.

Following Zhang & Norman [12] and Sadiq & Governatori [11], the modeling of business rules in standards such as DMN is performed in an incremental and collaborative process. Diverging views or understandings on the same domain can yield contradicting rules, making the set of business rules inconsistent.

There have been some works towards a *detection* of inconsistencies in business rules [6], [8], [9], [13]. One approach of inconsistency detection is to consider all business rules as a set, denoted as a rule base. We define inconsistency of a rule base B as logical inconsistency, i.e. there is support for contradictory output „O" and „*not* O" at the same time. This set-theoretic view allows defining minimal inconsistent subsets MIS of B via

$$\mathsf{MIS}(B) = \{B' \subseteq B \mid B' \textit{ is inconsistent and minimal in terms of set inclusion}\}. \qquad (2)$$

This definition of minimal inconsistent subsets can be applied to find inconsistencies in a rule base B. For example, the minimal inconsistent subsets for the rule base in Figure 2 are shown in Figure 3 Here, two minimal inconsistent subsets can be identified, denoted as MIS_1 and MIS_2.



`Account Balance <= 10.000€ → creditworthy`

`Account Balance = 10.000€ → not creditworthy`

`Account Balance >= 10.000€ → creditworthy`

Figure 3: Minimal inconsistent subsets for Figure 2

Next to this detection, there is a broad consensus that a *quantitative* inconsistency analysis in the context of business intelligence could provide added-value to the development and management of business rules [9]–[11], [13]. A scientific field concerned with such a quantitative analysis is the field of inconsistency measurement [14]. Here, a central object of study are

---

[1] https://www.omg.org/spec/DMN/About-DMN/

so-called inconsistency measures, which allow assigning a numerical value to (business) rules, with the informal meaning that a higher value reflects a higher degree of inconsistency [15]. Inconsistency measures subsequently provide the technical means for quantitative analysis of business rule inconsistencies. Let $\boldsymbol{B}$ be the set of all business rule bases, and $\boldsymbol{R}$ the set of all rules that appear in the individual rule bases $\in \boldsymbol{B}$. Then, an inconsistency measure to assess the degree of inconsistency for individual rules is a function

$$I: \boldsymbol{B} \times \boldsymbol{R} \rightarrow [0, \infty), \qquad (3)$$

which assigns a non-negative value to a combination of a rule base and a specific rule. This measure can thus assess the culpability that this rule represents with respect to the inconsistency of a rule base. An example is the so-called $C_{\#}$ measure [16] which assesses the culpability of a rule r for a rule base B, via

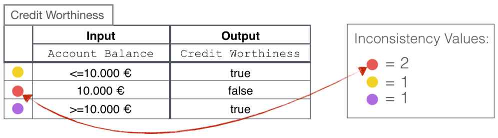
$$C_{\#}(B,r) = |\{M \in \mathsf{MIS}(B) \mid r \in M\}|. \qquad (4)$$

This measure counts the number of minimal inconsistent subsets that a rule r belongs to. Applying this measure to the rule base in Figure 3 results in the following quantification:

$$\begin{aligned} C_{\#}(B, \mathit{Rule}\ 1) &= 2 \\ C_{\#}(B, \mathit{Rule}\ 2) &= 1 \\ C_{\#}(B, \mathit{Rule}\ 3) &= 1 \qquad (5) \end{aligned}$$

Following Thimm [15], a quantification by inconsistency measures such as the $C_{\#}$ measure allows to assess the severity of inconsistency for individual rules. This allows ranking individual rules by their degree of inconsistency, where the rules are sorted by the respective value as quantified by the inconsistency measure. Given the rule base in Figure 3 and the corresponding quantification in (5), *Rule 1* consequently has the highest degree of inconsistency, i.e. the highest amount of blame in the context of the overall inconsistency.

This ranking can be displayed to the user as shown in Figure 4, providing quantitative insights as a basis for an informed inconsistency resolution strategy.



Credit Worthiness

	Input Account Balance	Output Credit Worthiness
🟡	<=10.000 €	true
🔴	10.000 €	false
🟣	>=10.000 €	true

Inconsistency Values:
🔴 = 2
🟡 = 1
🟣 = 1

Figure 4: DMN table with inconsistency values (example)

For a specific quantitative measure, we use inconsistency measures to assess the severity of inconsistency for individual business rules. In the following, we denote the degree of inconsistency for a rule as its inconsistency value.

In theory, an application of quantitative measures for analyzing business rule bases seems intuitive. Yet, the added-value of quantitative business intelligence insights to understanding inconsistencies in business rules has not been evaluated. In the following, we present the results of our experiment empirically assessing these effects.

3. Research Aim

In this work, we follow an Experimental Research methodology as described by Neuman [17]. Experimental Research focuses on causal relations, allowing to isolate and target the impacts of causal variables. Experimental Research is highly suitable for research aimed at investigating the effects of independent variables. To this aim, the researcher controls a condition in which events occur, manipulates the independent variable and analyses the causal effects that occur based on this manipulation. As the aim of this work is to evaluate the effect of quantitative measures, we see this methodology as highly appropriate, as providing a quantification can be seen as an independent variable and thus the causal effects of providing, resp. not providing, a quantification can be assessed.

Following [17], Experimental Research comprises the steps of *formulating a research question, develop hypotheses based on an independent variable, performing the experiment (i.e. data collection),* and *analyzing the results*.

Aligned with the aim of our work as motivated in our introduction, we derive the following research question:

RQ: How do quantitative measures affect the understanding of inconsistencies in business rules?

In order to investigate the causal effect of quantitative measures on the understanding of inconsistencies in business rules, we developed three hypotheses.

The first aim was to investigate the effect of quantitative measures on understanding accuracy, which means how well a user can understand inconsistency related problems in business rules:

Hypothesis 1: Providing a quantification of inconsistency in business rules is associated with better understanding accuracy compared to manual analysis. Following [11], the quantitative insights provided by inconsistency analysis increase the efficiency in which modelers can understand problems in business rule bases. To evaluate this proposition, we investigated understanding efficiency, which relates to how much time it takes a participant to understand the scenario and answer corresponding questions.

Hypothesis 2: Providing a quantification of inconsistency in business rules is associated with better understanding efficiency compared to manual analysis.

Last, we investigated mental effort, as quantitative insights could lower cognitive skills needed for understanding inconsistencies in rule bases:

Hypothesis 3: Providing a quantification of inconsistency in business rules is associated with less mental effort needed for problem understanding compared to manual analysis.

The dimensions of our hypotheses, i.e. understanding accuracy, efficiency, and less mental effort, are based on the experiment design in [18]. To test these hypotheses, we performed controlled experiments with participants. The following section provides insights into our experiment design, as well as the measures used to verify our hypotheses against the data collected in our experiments.

4. Experiment

In this section, we introduce the experiment[2] we conducted as part of this research, including the *experiment design, structure, measurements*, *settings,* and *participants*.

4.1. Experiment Design

In order to test our hypotheses, we conducted an experiment where participants were confronted with questions and scenarios regarding inconsistencies in business rules. The experiment was designed as a single-factor experiment, which is especially suitable to analyze the effects of a single factor on a common response variable following Reijers et al. [19]. The considered factor is the use of quantitative measures, with factor levels "present" and "absent". We used two groups of participants, which we each tested in two separated domains. A domain is defined as a test run, which comprises a block of comprehension questions. Each test run was performed with two different factor configurations, one with and the other one without access to quantitative measures.

The experiment was designed as a balanced single factor experiment with repeated measurement, based on an experiment design from [19]. This means that all participants used all factor levels, which lead to every subject generating data for both domains with respective factor levels. This approach enables the collection of more precise and powerful data as the same number of participants generates twice as much data [20]. In addition, each scenario was only shown once to each participant, to mitigate learning effects.

The design of our experiment is illustrated in Figure 5. In the first run, Group 1 was exposed to tasks 1-4 with access to inconsistency values, while Group 2 started without a respective quantitative insight. In the second run the situation was inverted, i.e. the first group had no access to inconsistency values to work on tasks 5-8, whereas Group 2 was provided with inconsistency values.

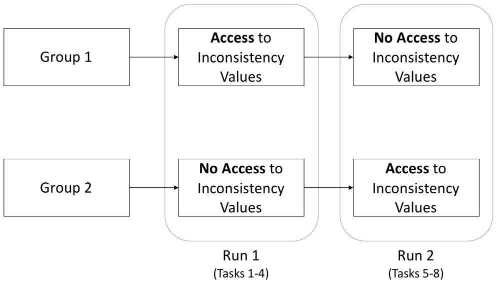


Figure 5: Experiment design

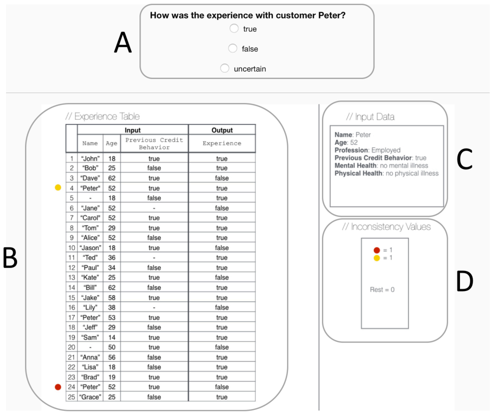
As each participant was tested for all domains and factor levels, the generalizability of the results was increased. A potential distortion due to learning effects was counterbalanced across groups since the order of factor levels was reversed between groups.

The questions were formulated in English in order to enable participants with different native languages to participate in the experiment, as well as to allow for better comprehension in the scope of reproducible research.

Figure 6 shows an exemplary scenario from the second domain.

---

[2] The experiment can be downloaded from: https://cloud.uni-koblenz-landau.de/s/tJ5C8Ky2PEoCt89



How was the experience with customer Peter?

○ true
○ false
○ uncertain

// Experience Table

		Input			Output
		Name	Age	Previous Credit Behavior	Experience
	1	"John"	18	true	true
	2	"Bob"	25	false	true
	3	"Dave"	62	true	false
●	4	"Peter"	52	true	true
	5	-	18	false	true
	6	"Jane"	52	-	false
	7	"Carol"	52	true	true
	8	"Tom"	29	true	true
	9	"Alice"	52	false	true
	10	"Jason"	18	true	false
	11	"Ted"	36	-	true
	12	"Paul"	34	false	true
	13	"Kate"	25	true	false
	14	"Bill"	62	false	true
	15	"Jake"	58	true	true
	16	"Lily"	38	-	false
	17	"Peter"	53	true	true
	18	"Jeff"	29	false	true
	19	"Sam"	14	true	true
	20	-	50	true	false
	21	"Anna"	56	false	true
	22	"Lisa"	18	false	true
	23	"Brad"	19	true	true
●	24	"Peter"	52	true	false
	25	"Grace"	25	false	true

// Input Data

Name: Peter
Age: 52
Profession: Employed
Previous Credit Behavior: true
Mental Health: no mental illness
Physical Health: no physical illness

// Inconsistency Values

● = 1
● = 1

Rest = 0

Figure 6: Exemplary scenario

Every task was divided into four areas. At the top, a (single-choice) question including relevant response options is displayed (A). Below the question, the scenario is shown, which is divided into three areas. On the left, one or more DMN tables (B) are shown. The tables can either be independent or connected to each other. In some cases, the participant is provided with input data (C) that can be used to identify the relevant rules in the DMN table. If a participant has access to inconsistency values, they are shown in the corresponding box (D). Otherwise, the box is empty.

4.2. Experiment Structure

All participants were shown a general introduction to the experiment including a tutorial before being exposed to the scenarios. The tutorial covered the basics of decision management and DMN tables. Additionally, the quantitative measures used in the experiment were introduced.

Each run consisted of four different tasks, containing a scenario and a corresponding question. The scenarios were all independent from each other and were designed to cover common types of inconsistencies in the area of decision management. While some scenarios only contained a single DMN table, others covered inconsistencies across multiple tables.

Across both domains there were three different types of questions:

- Content-related questions, that could be answered with true, false or uncertain (e.g. "Is Dave contractually capable?"). Here, the answer uncertain was correct for questions where no conclusion could be inferred due to an inconsistency of rules.
- Questions that asked for specific rules with the highest/lowest amount of inconsistencies and that could be answered with a particular rule (e.g. "Which rule is in conflict with the highest number of other rules?")
- Questions that asked for the number of inconsistencies of a particular rule (e.g. "How many rules contradict rule 1?")

4.3. Measurements

In our experiment, we distinguish between three types of measurements, referred to as response variables. To measure understanding accuracy, we use the percentage of correctly answered comprehension questions. Next, understanding efficiency was measured by tracking the time from the point that the first question of a run is displayed to the point that the participants selected the answer to the last question of a run. Last, mental effort was measured using eye-fixation duration, which is the period of time where the eyes remain still and focused on a single location. During the eye saccade or movement, vision is suppressed. New information can only be acquired during fixation [21]. The longer the fixation duration, the longer it takes for humans to comprehend respective information. Thus, eye-fixation duration can be used as an objective measure for mental effort [22]. In addition to this objective measure, the participants were asked which run they found easier as a measure of perception of required mental effort.

4.4. Settings

The tasks were implemented as HTML-files and – in combination with the introductory slides – added to the corresponding run using the eye-tracking software Tobii Studio. A pre-test was used to verify that the texts and tables were clearly visible from a distance of over 60 cm, which is the distance required by the eye-tracker (cf. Section 4.5 for details on the pre-test).

The screen was divided into four sections (see Figure 6, section 4.1). The questions were shown at the top and the corresponding DMN tables, input data and inconsistency values were shown below. All sections did neither require nor allow scrolling or zooming. We used Tobii X60, an eye tracker with a 22-inch screen of a resolution of 1680 x 1050. The experiment was set in an IT lab at the University of Koblenz-Landau, Germany. The blinds in front of the windows were closed and the lights on the ceiling were the only light source. The settings were the same for all participants.

4.5. Participants

In advance of the actual experiment, we conducted a pre-test with seven Ph.D. students. The main goal of the pre-test was to ensure understandability, readability and the overall usability of the experiment. After using the results of the pre-test to refine the introductory slides and the resolution and size of the scenarios, the experiment was conducted with 37 Bachelor and Master Students from the Faculty of Computer Science at the University of Koblenz-Landau. Study programs at this faculty range from computer sciences to economic sciences including interdisciplinary courses such as Business Process Management. The experiment required no prior knowledge as all relevant concepts were introduced in the tutorial. However, we took into consideration that all participants came from a business informatics background as they pursue an IT-related degree. The students were randomly assigned to two groups and they participated voluntarily, so no incentive was offered.

5. Results

After data collection through our experiment, we assessed the participant data with the respective measures described in Section 4.3 to evaluate our hypotheses.

First, we checked whether each dependent variable was normally distributed. To this aim, we assumed the data to be normally distributed if the standardized skewness and standardized kurtosis are within the range [-2, +2], following [19]. If the results of both groups in one run were normally distributed, we used Levene's test at a significance level of 0.05 to check whether the variance of both samples was equal. Given equal variance, we applied an independent-sample t-test. If the data was not normally distributed, we ran the Mann-Whitney U test, which can indicate a significant difference between two sample medians of not normally distributed samples [23]. For both the t-test and the Mann-Whitney U test, we assumed the commonly used significance level of 0.05.

5.1. Understanding Accuracy

Figure 7 depicts an overview of test results for the understanding accuracy measurement. The x-axis groups the individual task results. The y-axis shows the mean of the number of correctly answered questions across all participants. In five out of eight tasks, the percentage of correct answers differed by at least 40%, suggesting higher understanding accuracy for test runs with access to inconsistency values.

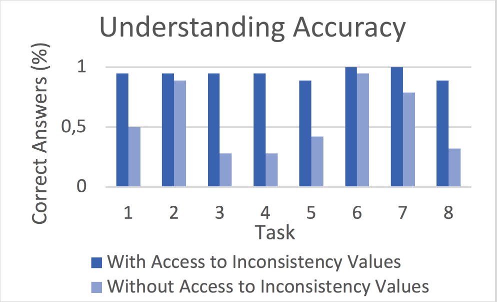


Figure 7: Overview of test result mean for understanding accuracy

Both standardized skewness and standardized kurtosis were not within the range [-2, +2]. Also, due to the fact that we asked four questions per run, there could only be five distinct results for the percentage of correct answers (0, 0.25, 0.5, 0.75, 1), meaning the data was not continuous. Hence, we could not apply other tests to check normal distribution such as the Kolmogorov-Smirnov test, as these tests are not applicable for discrete data. Therefore, the values could not be assumed to be normally distributed. Consequently, we ran the Mann-Whitney U test. The results are shown in Table 1.

Table 1: Test of Hypothesis 1 (understanding accuracy)

	Run 1		**Run 2**	
Group	1	2	1	2
N	19	18	19	18
Mean	0.95	0.49	0.62	0.94
Standard Deviation	0.13	0.21	0.19	0.16
p (1-tailed)	<0.0001		<0.0001	

As both p-values are statistically significant, the understanding accuracy was correlated with the access to inconsistency measures in both runs, which supports Hypothesis 1.

Conclusion 1: Quantitative measures for business rule inconsistencies are associated with an improved understanding accuracy.

5.2. Understanding Efficiency

Figure 8 shows an overview of test results for the understanding efficiency measurement. The results for the individual tasks are grouped along the x-axis. The y-axis indicates the time needed for answering a task in seconds. As can be seen, participants with access to inconsistency values had a lower completion time for all tasks, indicating higher understanding efficiency.

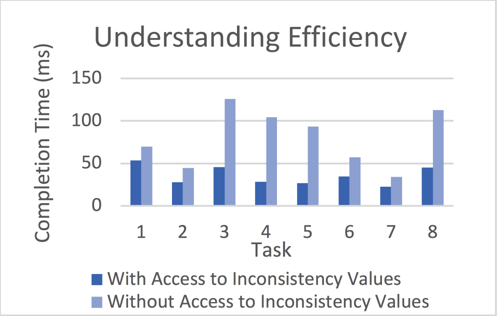


Figure 8: Overview of test result mean for time taken for task completion

The time the participants spent to answer the questions was normally distributed and both samples had equal variances (p values of Levene's test were 0.087 and 0.101, respectively). We consequently ran independent-sample t-tests between the two groups for each run. Table 2 shows the results of this test.

Table 2: Test of Hypothesis 2 (understanding efficiency)

	Run 1		**Run 2**	
Group	1	2	1	2
N	19	18	19	18
Mean	38.89	86.08	74.47	32.32
Standard Deviation	23.72	32.25	31.45	15.25
t	-4.948		5.002	
p (1-tailed)	<0.0001		<0.0001	

The p-values are statistically significant. Thus, the understanding efficiency was correlated with the access of inconsistency measures in both runs, which supports Hypothesis 2.

Conclusion 2: Quantitative measures for business rule inconsistencies are associated with an improved understanding efficiency.

5.3. Mental Effort

In order to measure mental effort, we used the eye-fixation duration. Here, we encountered the problem that the eye-tracking did not work for all 37 participants. Even though we cannot provide a verifiable technical explanation for this, we noticed that this phenomenon exclusively affected participants with glasses. To clarify, this only affected a fraction of participants with glasses. Based on this observation and information provided by the manufacturer of the eye-tracker [24], we assume that the condition of specific glasses and their reflections could be responsible for this. As the eye-tracking results were only needed as a measure for Hypothesis 3, we decided to still include the data of these participants towards evaluating understanding accuracy and efficiency. Accordingly, N was 30 for the eye-tracking measurement.

Figure 9 displays an overview of test results for the objective mental effort measurement. The respective task results are grouped by task on the x-axis. The average eye-fixation duration is shown on the y-axis. For all tasks, participants who had access to inconsistency values displayed a lower fixation duration, meaning that the time for comprehending information during a cognitive load was significantly lower. Following [21], this suggests lower mental effort during task completion.

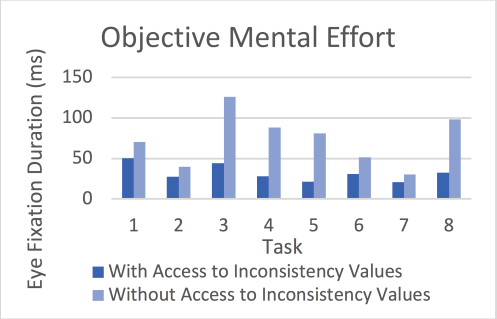


Figure 9: Overview of test result mean for objective mental effort

The eye-fixation durations were normally distributed, and both samples had equal variances (p values of Levene's test were 0.082 and 0.193, respectively), so we ran independent-sample t-tests between the two groups for each run.

Table 3: Test of Hypothesis 3 (objective mental effort)

	Run 1		**Run 2**	
Group	1	2	1	2
N	15	15	15	15
Mean	37.38	81.05	65.18	26.46
Standard Deviation	23.27	33.46	23.39	14.54
t	-4.009		5.261	
p (1-tailed)	0.0002		<0.0001	

The p-values are both statistically significant for the eye-fixation duration, which suggests less mental effort being correlated with the access to inconsistency measures in both runs, supporting Hypothesis 3.

In addition to the objective mental effort, we also asked the participants which run, i.e. the run with or without access to inconsistency values, they found easier or whether they perceived them as equally easy. 36 out of the 37 participants associated the run with access to inconsistency measures with less mental effort and only one person found the run without inconsistency measures easier.

Conclusion 3: Quantitative measures for business rule inconsistencies are associated with reduced mental effort required for understanding and interpreting inconsistencies in business rules.

Figure 10 and Figure 11 show heat maps for a scenario from the first domain without and with access to inconsistency values. Both figures show data from all participants of the corresponding run. Heatmaps display the focus of visual attention and how visual gaze is distributed. When comparing Figure 10 and Figure 11 it is observable that the participants without access to inconsistency values spent a lot more time and effort analyzing the four columns of the table. In Figure 10, participants had a large fixation-duration over the entirety of columns in the DMN tables.

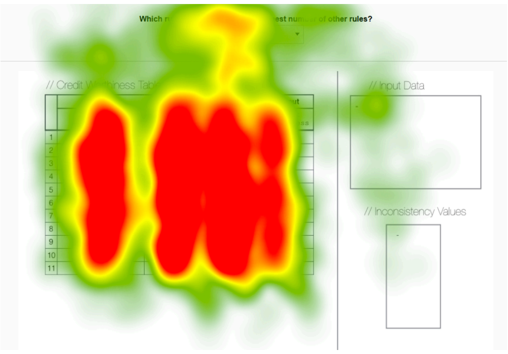


Figure 10: Heat map for task 3 without access to inconsistency values

On the contrary, the participants with access to inconsistency values, however, only briefly looked at the different columns after checking the provided inconsistency values. It is visible that participants who had quantitative insight were able to ignore irrelevant parts of the rule base, allowing for a more efficient task completion. Also, Figure 11 shows that participants actively looked at and used the provided quantitative ranking.

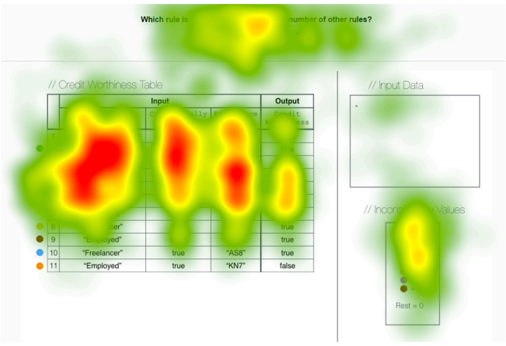


Figure 11: Heat map for task 3 with access to inconsistency values

6. Discussion & Conclusion

In this paper, we presented the results of an experiment investigating the effects of quantitative measures on understanding inconsistencies in business rules.

To verify our hypotheses, our focus was on three measurements: understanding accuracy, understanding efficiency, and mental effort measured via the percentage of correct answers to comprehension questions, the time needed for solving a given task and eye-fixation duration, respectively. The data was accumulated in empirical experiment research.

Due to the experimental nature of our research, potential limitations should be considered.

Our results are based on an experiment with a total number of 37 participants, which could affect the external validity. To put our sample size into perspective we compare our participant size to the number of participants in other related studies. The identified studies include [22] (23 participants), [19] (28 participants), [25] (28 participants) and [18] (50 participants), indicating that our sample size is comparable to those of mentioned works. Furthermore, the results of our t-tests have very low p-values (most below 0.0001), which already indicates an extremely significant effect in the analyzed sample.

It is possible that a change in the structure of the different scenarios or the order of questions might have an effect on the experiment results. Also, English being our language of choice could have had an impact on the participants' performance, as English was not the native language of our subjects and the comprehension of the scenarios therefore depends on the individual language skills. However, we want to point out that the situation was the same for all participants in order to enable internal validity.

The fact that our group of participants only consisted of students as opposed to domain experts, could have an impact on external validity, as the scenarios used in this experiment do not necessarily reflect those that occur in practice. However, since these scenarios represent typical inconsistencies in business rule bases, an application of our results to other domains seems plausible.

Our results support all three hypotheses, which were introduced in section 3. This indicates that quantitative measures are associated with better understanding accuracy, understanding efficiency and less mental effort in business rules management. Due to the applied experimental research methodology, our conclusions are based on the tested group of participants. To conclude, quantitative insights provide added-value to business intelligence by supporting companies in understanding inconsistencies in business rules. Following the suggestions by [10], [11], this work is the first to empirically show these effects.

We identify several possibilities for future work. The inconsistency quantification in our experiments was based on the C# measure. Future work could investigate how other inconsistency measures impact our experiment results. Additionally, the impact of different visualization techniques could also be evaluated.

Based on our results, we can identify several implications for practitioners. First, our results show positive effects of quantitative measures. This validates the findings by [10]. While there has been a plethora of work geared towards automatisms for the qualitative analysis of rule bases (i.e. a detection of errors), automated quantitative analysis has received far less attention. Efforts should therefore be directed towards incorporating quantitative measures in Business Rules Management. This quantification provides a clearly articulated impact assessment of compliance risks and inconsistencies, which can be used as a basis for an informed decision regarding a prioritization of problems and a time-efficient inconsistency resolution by domain experts.

Last, as suggested by [18], we utilized neuro-physiological measurement for industry-related studies. Current devices are becoming more attainable and less intrusive while allowing for in-depth analysis of human behavior in interaction with technology. This form of measurement could provide new opportunities for companies to understand the added-value of BI solutions.

Future research should be geared towards implementing quantitative insights in business intelligence, fostering inconsistency resolution and thus a correct decision-making, and sustainable development of organizations.

Acknowledgements

We thank Wei Wang, Marta Indulska, Shazia Sadiq and Barbara Weber for insights and inspiration to our experiment design.

References

[1] M. Weske, "Business Process Management Architectures," in *Business Process Management*, Springer, Berlin, Heidelberg, 2012, pp. 333–371.

[2] I. Graham, *Business rules management and service oriented architecture: a pattern language*. Chichester, England ; Hoboken, NJ: John Wiley, 2006.

[3] W. M. P. van der Aalst, M. L. Rosa, and F. M. Santoro, "Business Process Management," *Bus. Inf. Syst. Eng.*, vol. 58, no. 1, pp. 1–6, Feb. 2016.

[4] F. Imgrund, M. Malorny, and C. Janiesch, "Eine Literaturanalyse zur Integration von Business Rules und Business Process Management," in *Internationale Tagung der Wirtschaftsinformatik*, St.Gallen, 2017, p. 15.

[5] K. Batoulis and A. Nesterenko, "Decision Management in the Insurance Industry: Standards and Tools," in *BPM 2017 Industry Track co-located with the 15th International Conference on Business Process Management*, Cham, 2017, p. 11.

[6] K. Batoulis and M. Weske, "Soundness of Decision-Aware Business Processes," in *Business Process Management Forum*, Springer, Cham, 2017, pp. 106–124.

[7] D. Calvanese, M. Dumas, F. M. Maggi, and M. Montali, "Semantic DMN: Formalizing Decision Models with Domain Knowledge," in *Rules and Reasoning*, vol. 10364, S. Costantini, E. Franconi, W. Van Woensel, R. Kontchakov, F. Sadri, and D. Roman, Eds. Cham: Springer International Publishing, 2017, pp. 70–86.

[8] D. Calvanese, M. Dumas, Ü. Laurson, F. M. Maggi, M. Montali, and I. Teinemaa, "Semantics and Analysis of DMN Decision Tables," in *International Conference on Business Process Management*, 2016.

[9] C. Di Ciccio, F. M. Maggi, M. Montali, and J. Mendling, "Resolving inconsistencies and redundancies in declarative process models," *Inf. Syst.*, vol. 64, pp. 425–446, Mar. 2017.

[10] R. Lu, S. Sadiq, and G. Governatori, "Measurement of Compliance Distance in Business Processes," *Inf. Syst. Manag.*, vol. 25, no. 4, pp. 344–355, Oct. 2008.

[11] S. Sadiq and G. Governatori, "Managing Regulatory Compliance in Business Processes," in *Handbook on Business Process Management*, vol. 2, 2015, p. 23.

[12] J. Zhang and D. A. Norman, "Representations in distributed cognitive tasks," *Cogn. Sci.*, vol. 18, no. 1, pp. 87–122, Jan. 1994.

[13] J. Grant and A. Hunter, "Measuring Consistency Gain and Information Loss in Stepwise Inconsistency

Resolution," *Symb. Quant. Approaches Reason. Uncertain.*, vol. 6717, no. 1, pp. 362–373, 2011.

[14] J. Grant and M. V. Martinez, *Measuring Inconsistency in Information*. College Publications, 2018.

[15] M. Thimm, "On the Compliance of Rationality Postulates for Inconsistency Measures: A More or Less Complete Picture," *KI - Künstl. Intell.*, vol. 31, no. 1, pp. 31–39, Mar. 2017.

[16] A. Hunter and S. Konieczny, "Measuring Inconsistency through Minimal Inconsistent Sets," *Knowl. Represent. Reason. KR*, vol. 8, no. 1, p. 9, 2008.

[17] W. L. Neuman, *Basics of social research: Qualitative and quantitative approaches*. PearsonEducation,Inc', 2007.

[18] W. Wang, M. Indulska, S. Sadiq, and B. Weber, "Effect of Linked Rules on Business Process Model Understanding," in *International Conference on Business Process Management*, Cham, 2017, vol. 10445, pp. 200–215.

[19] H. A. Reijers, J. Mendling, and R. M. Dijkman, "Human and automatic modularizations of process models to enhance their comprehension," *Inf. Syst.*, vol. 36, no. 5, pp. 881–897, Jul. 2011.

[20] G. Charness, U. Gneezy, and M. A. Kuhn, "Experimental methods: Between-subject and within-subject design," *J. Econ. Behav. Organ.*, vol. 81, no. 1, pp. 1–8, Jan. 2012.

[21] K. Rayner, "Eye Movements in Reading and Information Processing: 20 Years of Research," *Psychol. Bull.*, vol. 124, no. 3, pp. 372–422, 1998.

[22] R. N. Meghanathan, C. van Leeuwen, and A. R. Nikolaev, "Fixation duration surpasses pupil size as a measure of memory load in free viewing," *Front. Hum. Neurosci.*, vol. 8, Jan. 2015.

[23] D. J. Sheskin, *Handbook of Parametric and Nonparametric Statistical Procedures: Third Edition*. CRC Press, 2003.

[24] Tobii, "Glasses, lenses and eye surgery," *Tobii Eye Tracking Support*. [Online]. Available: http://help.tobii.com/hc/en-us/articles/210249865-Glasses-lenses-and-eye-surgery. [Accessed: 12-Jun-2018].

[25] F. A. Haji, D. Rojas, R. Childs, S. de Ribaupierre, and A. Dubrowski, "Measuring cognitive load: performance, mental effort and simulation task complexity," *Med. Educ.*, vol. 49, no. 8, pp. 815–827, 2015.

7.7 Supporting BRM with Inconsistency Analysis

Source	Supporting Business Rule Management with Inconsistency Analysis
Authors	Carl Corea and Patrick Delfmann
Outlet	BPM 2018 (Industry-Track) - 16th International Conference on Business Process Management, September 9-14, 2018, Sydney, Australia
Ranking	A
Status	Published

Supporting Business Rule Management with Inconsistency Analysis

Carl Corea[1] and Patrick Delfmann[1]

Institute for Information Systems Research, University of Koblenz-Landau, Germany
{ccorea,delfmann}@uni-koblenz.de

Abstract. Business rules have reached considerable attention from todays businesses. Numerous standards such as the Decision Model and Notation (DMN) have been introduced and adapted in practice in order to model company decision logic. However, standards such as DMN often make strong assumptions about respective decision models, e.g. that of complete information. Here, we see a gap between the solutions proposed in research and the actual industry adaptation. As some assumptions in research seem unfeasible in practice, companies currently face the problem of inconsistent business rules and decision models. Here, companies need to be supported in detecting, understanding and resolving inconsistencies. In this work, we report on current problems for Business Rule Management in the field and present an approach to analyze actual process executions and corresponding decisions for inconcistencies.

Keywords: Business Rules · Inconsistency Measurement · Compliance Management

1 Introduction

Business rules (BR) are an important counterpart to Business Process Management (BPM), aimed to ensure that business processes comply to norms and regulations [11, 12]. A multitude of standards have been proposed in the BPM community, cf. Imgrund et al. (2017) for a survey. However, as BPM research is often constrained by assumptions, scientific results may not be plausibly aligned to industral settings. This gap is a potential problem both for companies and academia, as it may not be feasible to implement research results in practice. This is motivated for the DMN standard as follows.

1.1 Problems of BR Research in the Field

DMN[1] allows to represent business rules in so-called decision tables. Here, columns are used to denote the input to a rule, resp. the output which can be concluded. The rows of the decision tables relate to individual business rules. Contrary to the usage and assumptions envisioned in academia, we identify the following major problems for companies currently seeking to implement DMN.

---

[1] https://www.omg.org/spec/DMN/About-DMN/

- **Redundant Information.** Decision models may contain redundant information. This could for instance be duplicate rows or columns, distributed over multiple tables. Based on own experiences gained in industry projects, such redundant rows and columns can in fact occur in collaborative settings, contrary to the guidelines of the DMN standard.
- **Incomplete Information.** DMN models work under the assumption of complete information. However, decisions in practice can often be dependent of underlying domain knowledge [5]. Calvanese et al. (2017) have already identified this peculiarity as an assumption in research that may not be plausible in an industry context, and proposed to extend decision models with domain knowledge.
- **Inconsistent Information.** A potential problem for decision models are inconsistencies, i.e. rules actually contradict each other. Inconsistencies can result from collaborative and incremental modeling, and impede the intended use of decision models, as inconsistent models can not correctly be used to govern compliant process execution [3, 12] .

1.2 Supporting Business Rule Management

There is a broad consensus that the management of above problems is a current issue for BPM [1, 2, 4, 6, 12]. For example, Batoulis and Weske (2017) report on a recent case-study with a large insurance company, where those authors found that 27% of analyzed rules were erroneous. This motivates the need for supporting companies in monitoring correct decision making. This work therefore contributes an approach to *detect and analyze* inconsistencies in actual process executions, based on an application of results from the field of *Inconsistency Measurement* [9] to a unified representation of business rules and domain knowledge. In case of inconsistencies, the company is presented with a careful analysis, identifying problems as well as providing a quantification of inconsistency. To the best of our knowledge, an application of inconsistency measures in Business Rule Management has not yet been investigated.

Our discussion is based on the following main example in Figure 1. Figure 1 shows an exemplary ordering process. We assume that a company uses a process engine to handle this process. A *process instance* is triggered by a new customer input, i.e. *instance data*. This customer input is now processed in the context of the shown decision logic. For the given process instance, i.e. the route of the customer data through the process model, every rule which was used for decision making relative to the resp. instance data is highlighted in red. One can observe that there are multiple errors in this decision logic. The `FreeShipping` table contains contradictory information. Also, the conclusion in the `Eligibility` table contradicts external domain knowledge. Such problems make it impossible for companies to utilize decision logic as intended. Still, it is essential for companies to warrant a correct *process execution*. In this report, we therefore show how our approach helps companies to detect and analyze such inconsistencies, fostering correct and compliant business process execution.

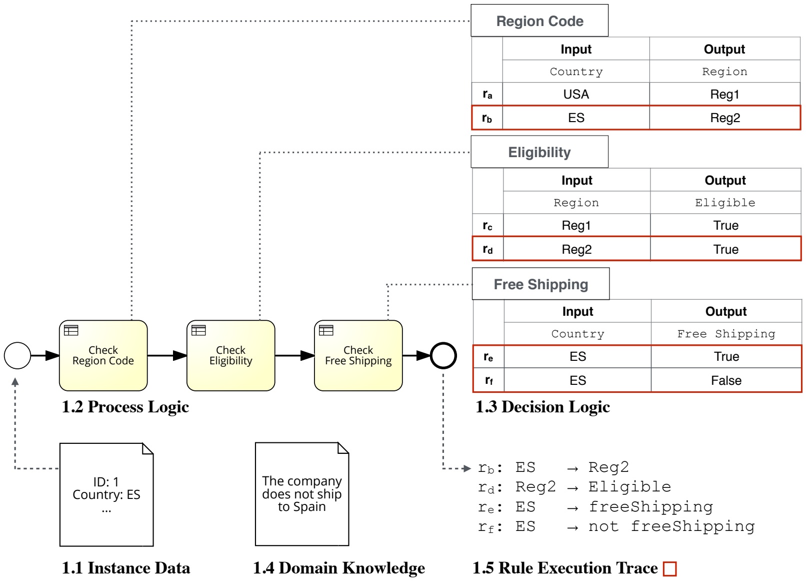


Fig. 1. Main example

2 Extending Business Rules with Domain Knowledge

In order to allow for an analysis of problems such as in Figure 1, a *unified* representation of business rules and domain knowledge is needed. As a design choice, we consider the Formal Contract Language (FCL) [7] as a logical formalism for business rules in this work. FCL allows to capture company knowledge, distinguishing between *facts* and *rules*. Facts capture atomic pieces of information about a domain of interest, e.g. *customer(Mary)*. Rules are of the form

$$r : \quad A_1, \ldots, A_n \to B \tag{1}$$

where $A_1, \ldots, A_n$ is the premise of the rule, B can be concluded given that the premise is satisfied, and r is an identifier. Please note that FCL also allows to model defeasible rules, superiority relations and other normative rules, e.g. to express deontic constraints. For simplicity, we will not revisit the syntax of FCL in greater detail and will continue our discussion on the basis of introduced expressivity. Please see Governatori and Maher (2017) for further description.

FCL can consequently be used to represent business rules [12]. Figure 2 shows an FCL representation of the DMN model in Figure 1.

$r_a : USA \rightarrow Reg1$

$r_b : ES \rightarrow Reg2$

$r_c : Reg1 \rightarrow Eligible$

$r_d : Reg2 \rightarrow Eligible$

$r_e : ES \rightarrow FreeShipping$

$r_f : ES \rightarrow \text{not } FreeShipping$

Fig. 2. FCL Representation of Business Rules in Figure 1

This FCL representation of business rules can subsequently be enriched with domain knowledge. To this aim, background domain knowledge can be captured in FCL, allowing to further define the semantics and interrelations of rules.

$$d_1 : ES \rightarrow \text{not } Eligible$$

Fig. 3. FCL Representation of Domain Knowledge in Figure 1

Figure 3 shows the the external domain knowledge from Figure 1 in an FCL representation. d_1 models the domain knowledge from Subfigure 1.4 as an exception.

The representation of business rules and domain knowledge in FCL provides a logic-based semantics, allowing to capture rules and the relations between rules such as subsumption, negation or datatypes in a unified model. The following section shows how this shared model can be used to analyze process related decisions for inconsistencies.

3 Inconsistency Analysis Approach

A scientific field concerned with the analysis of inconsistent information is the field of *Inconsistency Measurement*, cf. Grant and Martinez (2018). Here, a central object of study are quantitative measures, which allow to assign a numerical value to (elements of) a rule base, with the informal meaning that a higher value reflects a higher degree of inconsistency. These measures foster the possibility to identify inconsistencies in rule bases, i.e. *pinpoint* the exact causes, and *quantify* the amount of blame, that an individual part of a rule base carries in context of the overall inconsistency.

3.1 Approach Architecture

Our approach utilizes these quantitative measures to analyze the consistency of decisions. Here, our proposed application of Inconsistency Measurement results in Business Rules Management provides new forms of quantitative insight for companies. Figure 4 shows the approach architecture.

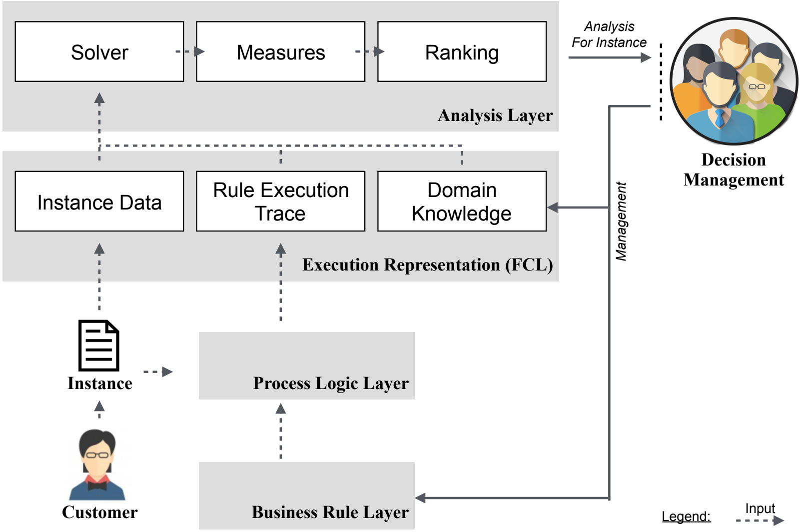


Fig. 4. Approach Architecture

Our approach is geared towards individual *process instances*. The process logic of a given process instance is defined in the *process layer*, manifested by the process model. This process layer in turn relies on a *business rules layer*, governing process execution. The core of our approach is a unified *execution representation*, comprising instance data, domain knowledge as well as all decisions made relative to the process instance. The latter are all business rules which were executed in the context of the respective process instance. These executed rules are stored in a so-called *rule execution trace*. To recall, an example of a rule execution trace is shown in Figure 1.5.

Our approach then allows to analyze this execution representation for inconsistencies. In result, companies are supported in monitoring consistent and compliant business process execution. The inconsistency analysis is based on results from the field of Inconsistency Measurement. Applying these results allows to support companies in detecting and quantifying potential inconsistencies, promoting an understanding of inconsistencies in process execution. The analysis layer comprises one component for *finding* inconsistencies, and a second component *analyzing* and *ranking* the resp. inconsistencies, introduced subsequently.

3.2 Finding Inconsistencies

Let an FCL rule base

$$B = (F, R) \tag{2}$$

where F is a set of facts and R is the set of all rules. Let $L(B)$ be the set of all literals appearing in B. We define inconsistency of a rule base B as logical inconsistency, i.e. there is support for contradictory outcomes A and *not* A at the same time.

Definition 1 (FCL Inconsistency). *An FCL rule base B is inconsistent, if there exists an $l \in L(B)$, s.t. B entails $\{l, \textit{not } l\}$.*

To clarify, an FCL rule base is inconsistent if there is a contradiction between facts or active rules. Then, given a rule base B, the minimal inconsistent subsets MIS of B are defined as

$$\mathsf{MIS}(B) = \{ \; B' \subseteq B \;\; | \;\; B' \text{ is inconsistent and minimal } \}. \tag{3}$$

This definition of inconsistent subsets can be used to find inconsistencies in business rule bases.

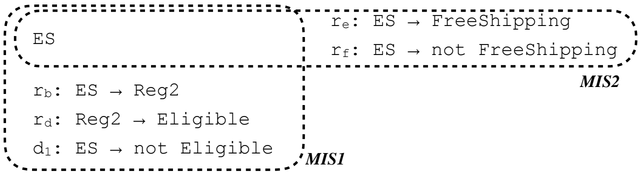


Fig. 5. Minimal Inconsistent Subsets for Figure 1

We recall the example from Figure 1. An analysis of the execution representation for this example yields two minimal inconsistent subsets, visualized in Figure 5 as MIS_1 and MIS_2. To solve for these MIS, existing reasoners such as SPINdle[2] can be utilized. In this way, our approach allows to exploit results from the field of logic programming to detect inconsistencies and support companies in decision management. Next to pinpointing problems, quantitative measures to further analyze these inconsistencies are presented in the following.

3.3 Culpability Measures for Assessing the Causes of Inconsistency

So-called *culpability measures* allow to analyze the FCL rule base from an element perspective [10]. The motivation of culpability measures is to evaluate the responsibility of each element for the overall inconsistency. This is useful for resolving inconsistency in a business rule base, as it allows to identify individual elements that are highly responsible for the inconsistency. Let $\mathfrak{E}$ denote the set of all possible elements, and $\mathfrak{B}$ the set of all business rule bases. Then, a culpability measure $\mathcal{C}$ is a function

$$\mathcal{C} : \mathfrak{B} \times \mathfrak{E} \rightarrow [0, \infty) \tag{4}$$

---

[2] http://spindle.data61.csiro.au/spindle/

which assigns a non-negative number to a mapping of an individual element to a rule base, and can thus assess the culpability that an individual element represents w.r.t. the rule base.

An example is the so-called cardinality based culpability measure $\mathcal{C}_c$ [10] which assesses the culpability of an element α for a rule base B, via

$$\mathcal{C}_c(B,\alpha) = \sum_{M \in \mathsf{MIS}(B) s.t. \alpha \in M} \frac{1}{|M|}. \tag{5}$$

This measure counts the number of minimal inconsistent subsets that an element α belongs to, normalized by the cardinalities of the respective subsets. Applying the $\mathcal{C}_c$ measure for the MIS shown in Figure 5 results in the following quantification:

$$r_b = 0.25 \qquad r_d = 0.25 \qquad r_e = 0.333 \qquad r_f = 0.333 \qquad d_2 = 0.25 \tag{6}$$

Note that we only compute values for rules, as we focus on an assessment of modeling errors and inconsistencies between rules.

An assessment such as in (6) provides a quantification that can be used as a driver for inconsistency resolution [12]. To further guide modelers in inconsistency resolution, we propose a culpability-based ranking. The intuition is that a rule with a higher culpability can be seen as more problematic than others and should be attended to with a higher priority, following [8].

Definition 2 (Culpability Ranking). *Let a rule base B and a culpability measure $\mathcal{C}$, then define the culpability ranking over all rules $r_i \in B$ via $\langle r_1, ..., r_n \rangle$, where $\mathcal{C}(B, r_1) \geq ... \geq \mathcal{C}(B, r_n)$.*

This ranking sorts all rules in B based on their culpability value. Thus, the user can be presented with a prioritized list of which elements to attend to. Given the example in Figure 1 and the respective values computed in (6), this leads to the following culpability ranking:

$$\langle r_e, r_f, r_b, r_d, d_2 \rangle \tag{7}$$

4 Key Learnings

In this report, we presented an approach to analyze the consistency of all decisions made throughout process execution. In case of inconsistent decisions, the company is provided with quantitative insights as a basis for an informed re-modeling strategy.

The first key learning is that the plausibility of assumptions made in BPM research should be carefully examined. The adaptation in industry may be subject to different settings, counteracting a correct implementation. This is supported by a wealth of recent studies analyzing problems in Decision Management [1, 2, 4, 6] and also matches our won experiences gained in industry projects.

A second key learning follows Sadiq and Governatori (2015). Those authors state that businesses need to be aided with systems to provide capacity to manage business rules. As a manual analysis is unfeasible in practice, BPM research needs to further focus towards automated approaches helping companies to understand the causes of problems. To this aim, we showed that measures from the field of Inconsistency Measurement can help companies with such an analysis.

Last, an important key learning gained from this project is the necessity of domain knowledge. Large-scale collaborative modeling is a challenging task for companies. Here, domain experts have to work closely with business rule engineers in order to create plausible decision logic. To this aim, the insights yielded by inconsistency analysis can be used to bridge the gap between these expert groups, fostering business process improvement and sustainable Business Rule Management.

References

1. Batoulis, K., Nesterenko, A., Repitsch, G., Weske, M.: Decision management in the insurance industry: Standards and tools. In: International Conference on Business Process Management. Springer (2017)
2. Batoulis, K., Weske, M.: Soundness of decision-aware business processes. In: International Conference on Business Process Management. Springer (2017)
3. Branco, M.C., Xiong, Y., Czarnecki, K., Küster, J., Völzer, H.: A case study on consistency management of business and it process models in banking. Software & Systems Modeling **13**(3) (2014)
4. Calvanese, D., Dumas, M., Laurson, Ü., Maggi, F.M., Montali, M., Teinemaa, I.: Semantics and analysis of dmn decision tables. In: International Conference on Business Process Management. Springer (2016)
5. Calvanese, D., Dumas, M., Maggi, F.M., Montali, M.: Semantic dmn: Formalizing decision models with domain knowledge. In: International Joint Conference on Rules and Reasoning. Springer (2017)
6. Di Ciccio, C., Maggi, F.M., Montali, M., Mendling, J.: Resolving inconsistencies and redundancies in declarative process models. Information Systems **64** (2017)
7. Governatori, G., Maher, M.J.: Annotated defeasible logic. Theory and Practice of Logic Programming **17**(5-6) (2017)
8. Grant, J., Hunter, A.: Measuring consistency gain and information loss in stepwise inconsistency resolution. In: European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty. Springer (2011)
9. Grant, J., Martinez, M.V. (eds.): Measuring Inconsistency in Information. College Publications (2018)
10. Hunter, A., Konieczny, S., et al.: Measuring inconsistency through minimal inconsistent sets. KR **8** (2008)
11. Imgrund, F., Malorny, M., Janiesch, C.: Eine literaturanalyse zur integration von business rules und business process management. In: Internationale Tagung der Wirtschaftsinformatik (2017)
12. Sadiq, S., Governatori, G.: Managing regulatory compliance in business processes. In: Handbook on Business Process Management 2. Springer (2015)

7.8 A Tool to Monitor Consistent Decision-Making in Business Process Executions

Source	A Tool to Monitor Consistent Decision-Making in Business Process Executions
Authors	Carl Corea and Patrick Delfmann
Outlet	BPM 2018 (Demo-Track) - 16th International Conference on Business Process Management, September 9-14, 2018, Sydney, Australia
Ranking	A
Status	Published

A Tool to Monitor Consistent Decision-Making in Business Process Execution

Carl Corea[1] and Patrick Delfmann[1]

Institute for Information Systems Research, University of Koblenz-Landau, Germany
{ccorea,delfmann}@uni-koblenz.de

Abstract. Workflow Management Systems such as Camunda allow companies to execute business processes. Here, standards such as the Decision Model and Notation (DMN) can be used to model company decision logic, governing how processes are executed. A potential problem here are inconsistencies in company decision logic, as this can lead to erroneous decision-making. However, it is essential to companies to warrant efficient and compliant process execution. In this report, we therefore present a tool which allows to monitor consistent decision-making during business process execution. Our tool detects inconsistencies in automated decisions and provides companies with an inconsistency analysis using quantitative measures.

Keywords: Decision-Making · Inconsistency Measurement · Camunda

1 Introduction

Workflow Management Systems (WMS) have received recent attention for supporting companies in the integration of process- and decision models [10]. Here, business processes and decision logic can be modeled in a shared technical environment, allowing to execute business processes (semi-)automatically, governed by the decision logic [1]. To ensure a correct process execution, a correct decision logic is thus essential. A potential problem here is the problem of inconsistency, i.e. contradictory information within the decision logic. Consider the example in Figure 1. While there are no problems locally, a global perspective yields an inconsistency in decision making for the shown process. Recent works in BPM research suggest that such inconsistencies can occur in decision models, due to the collaborative and incremental development of these artifacts [1, 2, 4].

In result, companies need to be supported in monitoring consistent decision making during process execution, i.e. in a global sense considering all decisions and their interrelations [8]. In this work, we present a tool that allows to detect and analyze inconsistencies of decisions during process execution. In case of inconsistencies, process execution is stopped to warrant that no compliance violations are commited. Furthermore, companies are presented with a careful analysis of inconsistencies so that problems can be resolved in the context of business process improvement. For this analysis, we apply quantitative measures from the scientific field of inconsistency measurement [6]. To the best of our knowledge,

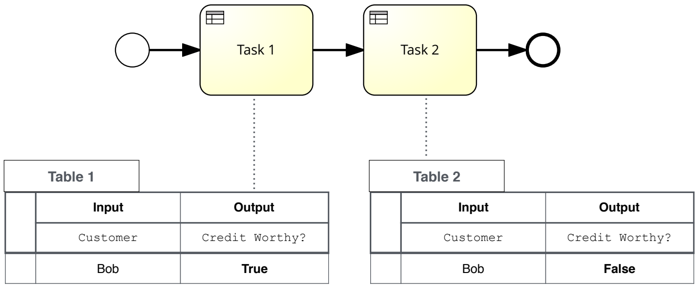


Fig. 1. Exemplary Process Model and Decision Logic

our tool is the first tool to investigate a verification of global consistency of all decisions made during process executions. Also, our tool provides quantitative insights, which can be used as a basis for an informed re-modelling strategy [8]. The following section introduces the tool and provides a usage example.

2 Tool Description

Our tool was developed as a plugin for the browser-based WMS Camunda[1]. A screencast of the tool can be found at https://youtu.be/jus4IkLMOIg.

2.1 Overview

Camunda allows to execute process models as so-called *process instances*. During execution, decisions can be automatedly computed by a rule engine. Our plugin stores every decision made during process execution in a so-called *decision history*. To clarify, this history is incrementally updated during a process execution, storing all respective DMN rules used for decision-making. On every update to this decision history, the tool analyzes the consistency of all decisions made for the current process instance.

The analysis is based on results from the field of Inconsistency Measurement [6]. A central object of study here are so-called *culpability measures*, which allow to assign a numerical value to rules, with the intuition that a higher value reflects a higher degree of inconsistency [9]. We implemented two widely acknowleged measures, namely the $\mathrm{MIV}_{\#}$ and the MIV_{c} measure [7]. For applicability of these measures, we transform the decision logic into a logic-formalism, namely the Formal Contract Language (FCL) [5]. To this aim, we extended the SPINdle library[2] with a functionality to transform DMN rules into an FCL representation. Also, we extended this library with a solver to detect and analyze inconsistencies.

---

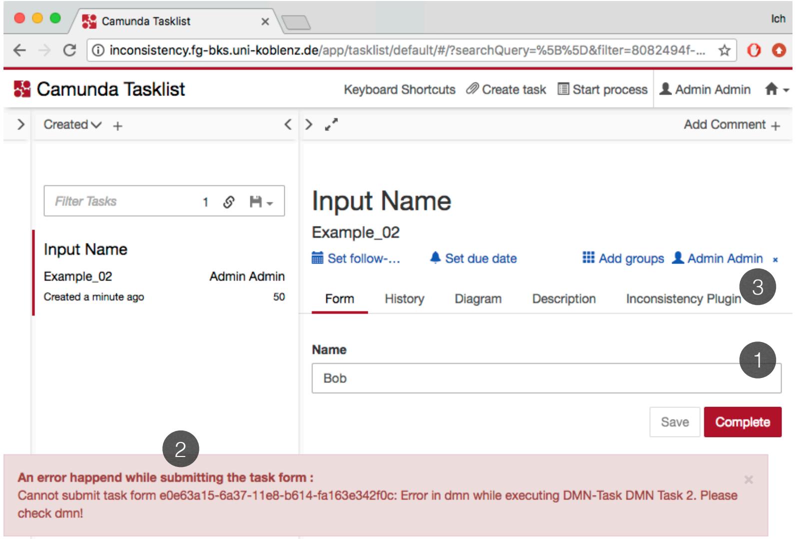
[1] https://camunda.com/

[2] http://spindle.data61.csiro.au/spindle/

To summarize, the implementation of inconsistency measurement in our tool allows to analyze the global consistency of all decisions made during process execution, and to assess the degree of inconsistency for individual rules in order to provide quantitative insights for companies.

2.2 Usage Example

In the following, we apply our tool for the example in Figure 1.



Name: Bob

An error happend while submitting the task form :
Cannot submit task form e0e63a15-6a37-11e8-b614-fa163e342f0c: Error in dmn while executing DMN-Task DMN Task 2. Please check dmn!

Fig. 2. Camunda Tasklist with Inconsistency Plugin

Figure 2 shows the *Camunda Tasklist*. We assume an employee verifies the creditworthiness of customer Bob (1). In *Task 1* of the process, a decision is made via the rule in *Table 1*. After this decision, the decision history contains this single rule and is thus still consistent. Next, *Task 2* is performed and a corresponding decision via the rule in *Table 2* is computed. This updates the decision history. The tool again checks the consistency of decisions for the current process instance. As can be seen, an inconsistency was detected (2). Process execution is automatically stopped to mitigate potential compliance violations. The user can then switch to the *inconsistency plugin tab* (3).

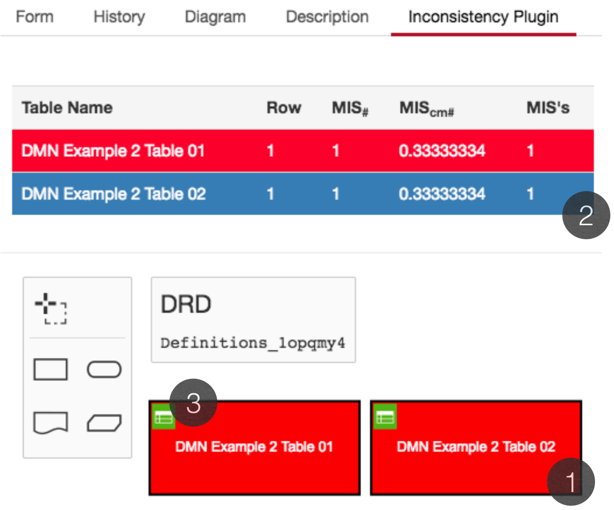


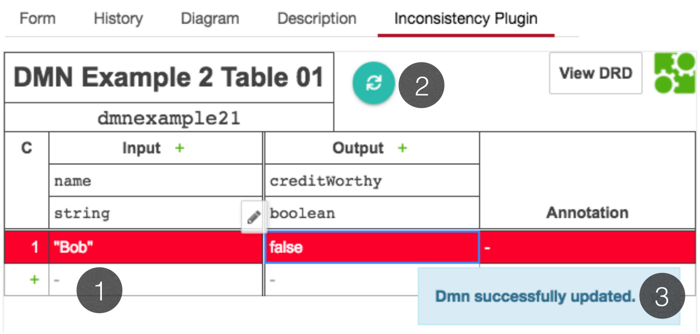
Table Name	Row	$MIS_{\#}$	$MIS_{cm\#}$	MIS's
DMN Example 2 Table 01	1	1	0.33333334	1
DMN Example 2 Table 02	1	1	0.33333334	1

DRD

Definitions_lopqmy4

Fig. 3. Inconsistency Plugin main Overview

Figure 3 shows the overview provided by the plugin. All DMN tables that contain rules which cause the inconsistency for the process instance are highlighted in red (1). Also, (2) shows a quantitative analysis of inconsistencies. In this example, we assume the user decides to alter a rule in *Table 1* (3).



DMN Example 2 Table 01

dmnexample21

C	Input +	Output +	
	name	creditWorthy	
	string	boolean	Annotation
1	"Bob"	false	-
+	-	-	

Dmn successfully updated.

Fig. 4. Resolving Inconsistencies via our Plugin

Figure 4 shows the detail view of *Table 1* in the plugin tab. All inconsistent rows are highlighted to allow the modeler to find the problematic rules (1). We integrated dmn-js[3], a JavaScript library for editing DMN tables. The user can

---

[3] https://bpmn.io/toolkit/dmn-js/

consequently directly edit DMN tables in Camunda. A refresh button allows to upload the changes (2), which automatically deploys the DMN table (3).

Camunda also offers a dashboard for management, entitled the *Camunda Cockpit*. Here, the problems detected by our plugin are seamlessly integrated into the Cockpit by triggering so-called *incidents*. This allows to provide business intelligence for management in the usual Cockpit environment, allowing to quickly be alerted of and view inconsistencies in decision-making which occured during process execution.

3 Conclusion and Outlook

The tool presented in this report allows to monitor consistent decision-making during process execution. *Detecting* inconsistencies supports compliant process execution. Also, inconsistency *analysis* based on inconsistency measurement provides quantitative insights, which can be used as a basis for an informed resolution and re-modelling strategy [8]. Our tool consequently fosters sustainable business rules management.

Our tool is seamlessly integrated into Camunda. In future work, we aim to present case-studies of applying our tool in industrial settings.

References

1. Batoulis, K., Weske, M.: Soundness of decision-aware business processes. In: International Conference on Business Process Management. Springer (2017)
2. Calvanese, D., Dumas, M., Laurson, Ü., Maggi, F.M., Montali, M., Teinemaa, I.: Semantics and analysis of dmn decision tables. In: International Conference on Business Process Management. Springer (2016)
3. Calvanese, D., Dumas, M., Maggi, F.M., Montali, M.: Semantic dmn: Formalizing decision models with domain knowledge. In: International Joint Conference on Rules and Reasoning. Springer (2017)
4. Di Ciccio, C., Maggi, F.M., Montali, M., Mendling, J.: Resolving inconsistencies and redundancies in declarative process models. Information Systems **64** (2017)
5. Governatori, G., Maher, M.J.: Annotated defeasible logic. Theory and Practice of Logic Programming **17**(5-6) (2017)
6. Grant, J., Martinez, M.V. (eds.): Measuring Inconsistency in Information. College Publications (2018)
7. Hunter, A., Konieczny, S.: Measuring inconsistency through minimal inconsistent sets. KR **8** (2008)
8. Sadiq, S., Governatori, G.: Managing regulatory compliance in business processes. In: Handbook on Business Process Management 2. Springer (2015)
9. Thimm, M.: Inconsistency measures for probabilistic logics. Artificial Intelligence **197**, 1–24 (2013)
10. Weske, M.: Business process management architectures. In: Business Process Management, pp. 333–371. Springer (2012)

7.9 A Taxonomy of Business Rule Organizing Approaches in Regard to Business Process Compliance

Source	A Taxonomy of Business Rule Organizing Approaches
Authors	Carl Corea and Patrick Delfmann
Outlet	Enterprise Modelling and Information Systems Architectures (EMISAJ) - International Journal of Conceptual Modeling
Ranking	C
Status	Published

A Taxonomy of Business Rule Organizing Approaches in Regard to Business Process Compliance

Carl Corea[*,a], Patrick Delfmann[a]

[a] University of Koblenz-Landau

Abstract. *In the scope of Business Process Compliance (BPC), business rules are used as a central means to represent regulatory policies and consequently to (automatically) verify, whether business process models abide by respective rules. While there has been a plethora of works regarding this actual verification of process models relative to business rules, we see a strong lack of works regarding the actual creation and maintenance of business rules. More precisely, many works assume sound sets of business rules as a basis for subsequent techniques. However, recent works suggest this assumption cannot be made in practice, and companies actually need to be supported in the scope of managing and organizing business rules, e. g. to remove redundant or contradictory rules. Otherwise, errors in business rules make these rule bases unusable and impede a subsequent verification of process compliance. Yet, the understanding on business rule organization is sparse - especially its relation to BPC. To address this issue on a conceptual level, we develop a taxonomy for business rule organizing approaches. Furthermore, we identify rule organizing approaches from literature based on a systematic literature review and classify these works in the scope of the developed taxonomy. Based on the identified literature, we also identify research gaps and propose a corresponding research agenda.*

Keywords. Business Rule Management • Business Process Compliance • Business Rule Lifecycle

1 Introduction

Business process compliance (BPC) comprises methods and techniques concerned with ensuring the regulatory compliance of company processes (Sackmann et al. 2018). With an increasing amount of laws and regulations that directly affect how a company is allowed to conduct activities, ensuring the regulatory compliance of company processes is an important challenge. Violating policies can otherwise lead to sensitive financial fines, or even criminal prosecution (Hashmi et al. 2018).

Within BPC, business rules are a central artifact used to *represent* regulatory policies. Here, a business rule can be defined as a declarative statement, which guides or constraints company activities (Van der Linden et al. 2019; Weiden et al. 2002). Business rules are used to verify whether company processes, i. e. company activities, are compliant, usually by means such as model query. Here, methods and techniques for this actual verification have been broadly studied, cf. e. g. (Hashmi et al. 2018; Sackmann et al. 2018) for some recent surveys. However, in this paper we want to take a needed step back.

While approaches for business process compliance verification are impressively advancing, they commonly share a central assumption, namely that a sound set of business rules exists which can be used for these approaches. However, numerous recent works on business rules have suggested that

---

[*] Corresponding author.

E-mail. ccorea@uni-koblenz.de

This research is part of the research project "Handling Inconsistencies in Business Process Modeling", which is funded by the German Research Association (reference number: DE1983/9-1).

this cannot simply be assumed in practice (Batoulis et al. 2017; Batoulis and Weske 2018; Calvanese et al. 2018; Corea and Delfmann 2018b; Di Ciccio et al. 2017; Sadiq and Governatori 2015).

A core problem for companies is a phase underlying BPC, namely that of *business rules management*. Business rules management is the discipline of creating and maintaining business rules. Here, relevant rules first need to be identified and then formally authored. Unfortunately, as the authoring process is usually a manual, collaborative and incremental process (Nelson et al. 2008; Sadiq and Governatori 2015), errors can occur frequently (Sadiq and Governatori 2015). Modellers might accidentally make mistakes, or model redundant rules due to a lack of oversight. Worse, modellers with different understandings on the same domain of interest might model business rules in a contradictory manner. As a recent example, Batoulis et al. (2017) reported on a case-study with a large insurance company, where those authors found that 27% of analyzed business rules contained modelling errors. Thus, rules must be assessed after authoring to ensure correctness, denoted as *rule organizing*. In this work, we define business rule organizing as understanding, clustering and selecting rules, with the goal to warrant a sound set of business rules, e. g. free of problems such as redundancies or inconsistencies. That is, rule organizing aims to ensure error-freeness *within* a set of business rules.

Despite the importance of rule organizing, this aspect is not represented in many proposed business rule lifecycle approaches, or even recent BPC surveys, e. g. (Hashmi et al. 2018; Ramezani et al. 2011; Sackmann et al. 2018). Rather, rule organizing is often described as an activity which should be performed by domain experts, cf. e. g. (Nelson et al. 2008). This is unsatisfactory from both an academic as well as a practical perspective. In light of new challenges imposed by an increasing amount, complexity and interdependence of regulations, companies need to be supported with (semi-)automated means in business rule organization in order to implement a sufficient quality management (Sackmann et al. 2018; Smit et al. 2017). Unfortunately, research on specific approaches to support companies in rule organizing is sparse – especially their relation to BPC. Following works such as Smit et al. (2017), there is consequently a need for an overview of rule organization approaches. In this work, we therefore investigate general characteristics of rule organizing approaches and how they can be classified. To this aim, we present a taxonomy of business rule organizing approaches, which conceptualizes important characteristics and allows to classify rule organizing approaches. The actual taxonomy development is conducted based on the approach by Nickerson et al. (2013) and is grounded in a structured literature review as proposed by vom Brocke et al. 2009. Next to the taxonomy development, we also classify the works identified in our literature review using the proposed taxonomy, which provides researchers and practitioners a needed overview of the the current state-of-the-art in business rule organizing approaches. Based on our analysis, we also propose a research agenda in order to leverage research on this important pre-phase of BPC.

The remainder of this work is structured as follows. In Sect. 2, we provide background knowledge on aligning rule organizing with BPC. Sect. 3 presents our research methodology, including the taxonomy development process and a documentation of our literature research. Then, Sect. 4 presents an overview of identified rule organizing approaches in the scope of the presented taxonomy. Here, we also distill a research agenda based on our findings. Last, we conclude in Sect. 5.

2 Background

This section provides preliminary knowledge on business rules, BPC and business rule organizing.

2.1 Business Rules and their Relation to BPC

Following Graham (2007)[p. 7], a business rule is a *“declarative statement about an aspect of a business”*, which specifies obligations, permissions and restrictions that constrain how company activities should be performed. Business rules are usually divided into structural business rules (which describe constraints in data), and behavioral business rules (which describe how company activities should be conducted) (Graham 2007; Weiden et al. 2002). In this work, we focus on the latter type of behavioral business rules. Behavioral rules are of the general form

$$\textit{if } A_1, \ldots, A_n \textit{ then } B$$

where $A_1, \ldots, A_n$ represents the premise of the rule (condition), and the conclusion B can be entailed, if the premise holds. This representation of conditions and behavioral conclusions allows to model business rules as a basis for BPC.

Example 1 *Numerous regulations such as Sarbanes-Oxley, Basel II or AML/CTF have been introduced to regulate allowed company behavior (Sadiq and Governatori 2015). As an example, a real-life business rule from the Anti-Money Laundering and Counter-Terrorism Financing Act 2006 (AML/CTF) imposes that it is obligatory to check new customers against a company blacklist before accepting the customer application. Process models must adhere to this rule, otherwise compliance breaches could be committed, which in turn could result in sensitive financial fines. Based on the general business rule form above, this rule could be authored with the Formal Contract Language, which is a rule standard to represent deontic constraints, as follows:*

$$NewCustomer \rightarrow O[CheckAgainstBlacklist]$$

This exemplary FCL rule encodes that if *a new customer is registered,* then *an obligation (O) arises to check this customer against a blacklist. Subsequently, the following exemplary process model could be verified against this business rule. Here, results from model query can successfully be applied to ensure that the process model in Fig. 1 adheres to the provided business rule in the scope of Business Process Compliance.*

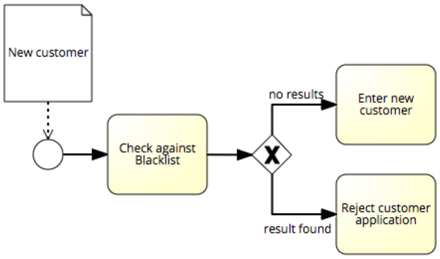


Figure 1: Exemplary customer application process.

Regarding BPC, companies can implement their compliance efforts using different so-called compliance strategies, usually divided into *design-time compliance*, *run-time compliance* and *post-execution compliance* (Hashmi et al. 2018)[1] . Following Hashmi et al. (2018), design-time compliance can be defined as a preventative strategy, with the goal of facilitating compliant-by-design process models. To this aim, results from model checking or model query can be used to verify the compliance of process models against a set of business rules as in the above example. Checking models against business rules at design-time is important for companies, as otherwise illegal (sequences of) activities could be performed following such models. Thus, a sound set of business rules is a mandatory prerequisite for design-time model checking, as otherwise, model checking might not be possible or not correct.

In run-time compliance management, compliance is managed during process execution (Hashmi et al. 2018). To this aim, business rules can be used to govern the compliant execution of company processes. Especially in (semi-) automated process execution, e. g. via workflow management systems, systems and employees rely on business rules as a basis for decision-making. That is, case-dependent information is validated against the set of business rules in order to determine how to proceed with the process. Here,

---

[1] We acknowledge that there are hybrid approaches which combine different strategies, but continue to discuss these three basic strategies for simplicity.

such decision-making also relies on a sound set of business rules. Otherwise, the (automated) decision-making might be flawed, which again can result in non-compliant process execution. Hence, ensuring an error-free set of business rules in the scope of business rule management is a prerequisite for run-time compliance management.

Last, post-execution compliance focuses on the analysis of company activities by domain experts after process execution (Hashmi et al. 2018). Here, the actually observed behavior, e. g. the company activities recorded in event logs during process execution, can be verified for compliance. Again, a sound set of business rules is needed to assess the compliance of observed behavior, as otherwise it might not be possible to detect compliance breaches. Thus, business rules management can be seen as a prerequisite for post-execution compliance.

Regarding which strategy is best, there are intuitively advantages to all approaches. Following Hashmi, *"the increased pressure and threat of possible criminal prosecutions [...] make the auditing method a less attractive compliance reporting strategy"* (Hashmi et al. 2018)[p. 83], which advocates an emphasis on design-time compliance management. Unarguably, compliant-by-design processes are desirable for companies. In this context, works such as (Corea and Delfmann 2018b; Maggi et al. 2011a,b) however point out that some compliance violations might not be detectable a priori, as they might be dependent on case-specific contexts. A further factor to also consider here is that human behavior might not fully be predictable or controllable. Thus, run-time compliance management might be necessary, despite design-time compliance management efforts. Last, companies can benefit from post-execution compliance by the means of a retrospective compliance analysis from a global perspective. This could for example be used to create value through innovation, e. g. streamlining business processes or improving operations.

Regardless of the BPC strategy, we can observe that a sound set of business rules is a necessary prerequisite in all cases and therefore needs to be addressed by companies. Here, ensuring such a sound set of rules is a central goal of business rule organizing, which is embedded in business rule management as follows.

2.2 Business Rule Organizing Capabilities

As a counterpart to business process management and BPC, business rules management is geared towards the creation and long-term maintenance of business rule repositories. In essence, business rule management can be defined as a systematic and controlled approach to support the capturing, authoring and organization of business rules, as well as aligning and implementing rule management within the companies' socio-technical environment (Schlosser et al. 2014; Van der Linden et al. 2019). Adapted from Nelson and Sen (2014), we define a business rule management lifecycle as shown in Fig. 2, containing the components of *strategic alignment*, *creation & maintenance* and *implementation*. In this report, we focus on the creation & maintenance aspect, which contains the phases of *capturing, authoring* and *organizing*.

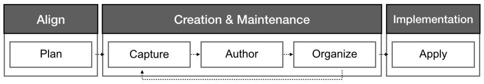


Figure 2: Proposed Business Rule Management lifecycle, adapted from Nelson and Sen (2014) and Schlosser et al. (2014).

In rule capturing, relevant business rules need to be identified. This is often performed by legal experts or using results from rule mining. This is an important step, as companies need to ensure they identify all relevant regulations that affect their domain.

Next, identified rules need to be formalized in the authoring phase. This relates to representing the identified rules in standards and rule languages, such that systems can access business rules. Authoring is usually a manual, collaborative and incremental process. In this setting, errors can occur frequently. For example, modellers might accidentally make mistakes, or model redundant rules due to a lack of oversight. Worse, modellers with different understandings on the same domain of interest might model business rules in a contradictory manner. Thus, the authored rules must be assessed and organized, in order to ensure errorfreeness within the set of business rules. Even in cases where business rules are authored automatedly, e. g. in the scope of rule mining or declarative process discovery, recent works such as (Corea and Delfmann 2019b; Di Ciccio et al. 2016, 2017) show that state-of-the-art algorithms can still yield erroneous rule sets, and thus such results should still be assessed in an organization phase.

Definition 1 (Rule Organizing) In this work, we define business rule organizing as understanding, clustering and selecting rules, with the goal to warrant a sound set of business rules, e. g. free of problems such as redundancies or inconsistencies.

While standards for authoring business rules and means to apply the authored rules have much matured, a wealth of recent research shows that the *organizing* phase still has to be evolved (Batoulis et al. 2017; Calvanese et al. 2016; Corea and Delfmann 2018b; Di Ciccio et al. 2017; Janssens et al. 2016; Lu et al. 2008; Sadiq and Governatori 2015; Weidlich et al. 2011). Despite these recent calls in literature, the understanding on this topicality is still at an early stage. A foundation for understanding and analyzing this domain is therefore needed. In order to address this issue, our research aim is therefore to investigate a classification of rule organizing approaches, explained in the following.

3 Research Approach

3.1 Research Aim and Scope

Our central research aim is to investigate a classification of rule organizing approaches, by means of taxonomy development. Following Nickerson et al. (2013), a taxonomy is a useful artifact to *"provide structure to the knowledge of a field"*, thus allowing researchers and practitioners to study the relations among concepts.

Furthermore, from qualitative research such as (Smit et al. 2017), we see evidence that companies are currently seeking means to organize business rules, as they recognize this business rule management phase as a current challenge (Corea and Delfmann 2018b; Sadiq and Governatori 2015; Smit et al. 2017). To the best of our knowledge, there however currently exists no overview on existing approaches, making it hard for practitioners to grasp the available state-of-the-art. Therefore, our second research aim is to provide an initial overview on this matter. To this aim, we identify state-of-the-art approaches based on a systematic literature review and analyze these approaches using the developed taxonomy. This provides companies and scholars an overview of current approaches based on the proposed classification. Such an overview could for example be used by practitioners to gain a better understanding on existing approaches, or as a basis to make an informed decisions on selecting suitable approaches. Here, we also aim to identify research gaps for current BRO approaches in order to distill a research agenda for using BRO approaches for BPC.

In the following, we present our taxonomy development approach, including a systematic literature review. Then, we present an analysis of the identified works in the scope of our novel taxonomy.

3.2 Taxonomy Development Approach

Following Nickerson et al. (2013), a taxonomy can be described as a system for grouping objects of interest in a domain based on common characteristics. Here, a taxonomy T is defined as a set of n dimensions, each consisting of a set of k characteristics. Based on our research aim, we apply the conceptual-to-empirical taxonomy development approach as proposed by Nickerson et al. (2013), shown in Fig. 3. In the following, we describe our individual research steps.

The first step in the taxonomy development approach is the definition of meta-characteristics. These meta-characteristics are the most comprehensive characteristics and should be aligned with the purpose of the taxonomy (Nickerson et al. 2013). Based on our research aim, we consequently decided to focus on the following aspects: First, to harmonize company efforts in business

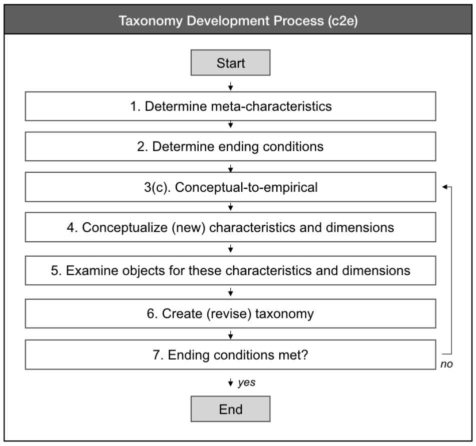


Figure 3: High level overview of the taxonomy development process (conceptual-to-empirical) as proposed by Nickerson et al. (2013).

rule organizing and BPC, we investigate how rule organizing approaches and compliance management strategies are aligned. Also, we investigate the specific capabilities offered by business rule organizing approaches, as well as tool-support, applicability and maturity of solutions. This yields the following meta-characteristics as a basis for our taxonomy:

- **(MC1) BPC Phase.** What are important compliance management phases that need to be addressed by rule organizing approaches?
- **(MC2) Capabilities.** What are important capabilities that need to be addressed by approaches for business rule organizing?
- **(MC3) Applicability.** What are important characteristics impacting the applicability of rule organizing approaches?
- **(MC4) Evaluation.** What are important strategies for evaluating rule organizing approaches?

As a second step in the applied taxonomy development approach, ending conditions need to be defined (Nickerson et al. 2013). This is necessary to assess whether the incremental development process can be ended. Nickerson et al. (2013) provide a series of objective ending requirements which impose formal requirements, e. g., there are no duplicate dimensions in the taxonomy, which were applied in our development process. Furthermore, based on the suggestions by Nickerson et al. (2013), we defined the following subjective ending conditions: The (number of) dimensions of the taxonomy should be *concise* and *comprehensive* as defined in (Nickerson et al. 2013), i. e., it should allow to classify all objects within a domain of interest. Here, this classification should also be *explanatory* as suggested in (Nickerson et al. 2013). Also, the dimensions as well as the characteristics need to be suitable to provide a differentiation between objects (*robustness*). Also, the finished taxonomy should be easily *extendable*, e. g., in future research.

Based on the defined meta-characteristics and ending conditions, we proceeded with the development process. In this work, we follow the conceptual-to-empirical approach as proposed by Nickerson et al. (2013), as it allows to conceptualize dimensions based on (domain) knowledge. This approach was iteratively performed in three cycles to refine the taxonomy. As a basis for our conceptualization, we conducted a structured literature review to consider a larger body of literature as a grounding for the development process. Here, we applied the structured literature review approach as proposed by vom Brocke et al. (Brocke et al. 2009), which consists of the five phases of *defining the review scope, conceptualizing the topic, the literature search, a literature analysis and synthesis* and a subsequent *discussion & agenda*. In the following, we provide details on our literature review as it is a central basis for the conceptualization of our taxonomy dimensions and characteristics.

To define our literature review scope, we use the taxonomy by Cooper (1988) as suggested by vom Brocke et al. 2009. As we aim to identify business rule organizing (BRO) approaches that can aid BPC, our *focus* is the research outcomes and applications of the reviewed works. To align our review with our research aim, the *goal* of our literature review is the integration of central issues. The *organization* of our research result is conceptual, as it is meant as a basis for taxonomy development. To the best of our ability, our results are *presented* in a neutral way. The *audience* of our review and our taxonomy are scholars and practitioners seeking to gain insights of BRO approaches capable of supporting individual company needs. Last, our *coverage* is exhaustive and selective, as our intention is a comprehensive overview of relevant literature by the means of a literature search in established databases, and reviewed works are discussed based on a literature analysis and taxonomy dimensions/characteristics, as opposed to a discussion of all works individually.

For our literature search, we followed the process proposed by vom Brocke et al. (2009), shown in Fig. 4.

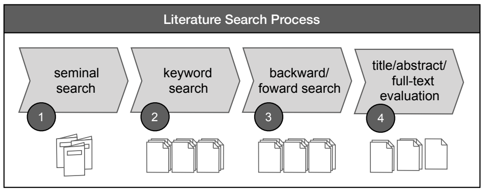


Figure 4: High Level overview of the search process as proposed by vom Brocke et al. (2009).

At first, we conducted brief searches as proposed as a good starting point by (Brocke et al. 2009; Rowley and Slack 2004) and found some initial surveys (Gadi 2015; Hashmi et al. 2018; Imgrund et al. 2017; Sackmann et al. 2018). We worked through these surveys to gain an initial overview of the topic. Then, we applied concept mapping techniques for a topic conceptualization as proposed by (Rowley and Slack 2004; Webster and Watson 2002). This conceptualization was used as a basis to derive suitable keywords. Our concept mapping revealed that the word “maintenance” was often used as a synonym to “organizing” or “long term management”. Through iterative tests, we found that the keywords “business rule(s) maintenance” resulted in a good balance between a feasible amount of results. It is noteworthy that the keyword “business rule(s) management” was not feasible, as this term refers not only to the scope of rule organizing but also to the other phases of the presented business rule management lifecycle. In result, this keyword resulted in too many results which were in many cases also not aligned with the scope of this work. Also, note that the keyword “business rule(s) organizing” returned too little results (the latter results were also subsumed by the selected keyword).

The keyword-based search was then conducted as follows: To warrant for a broad view on the research topicality, we queried 6 pertinent literature databases with the derived keywords, in particular ACM digital library[2] , Springer Link[3] , Emerald Insight[4] , AISel electronic library[5] , Science Direct[6] and IEEE Xplore[7] , respectively. In the scope of the mentioned balance between feasibility and coverage, the query was defined such that the title had to contain the phrase *“business rule(s)”* and any other field had to contain the word *“maintenance”*. In result, our keyword based search yielded a total of 209 results. Fig. 5 shows the searched databases as well as an overview of the selection process, following the standard systematic literature review phases of *identification, screening, eligibility* and *inclusion* (Moher et al. 2009).

Based on the identified search results, we conducted a first review phase (Review Phase I). Here, we removed duplicates. Then, we read the abstracts of the considered search results and determined the potential relevance of the individual works, as suggested by vom Brocke et al. (2015; 2009). In result, we reduced the search results to 52 works.

Then, we conducted a second review phase (Review Phase II). All 52 works were read in full, to determine whether they were relevant in the scope of our research aim. Here, we defined a paper to be relevant if it met the following criteria:

---

[2] https://dl.acm.org/
[3] https://link.springer.com/
[4] https://www.emeraldinsight.com/
[5] https://aisel.aisnet.org/
[6] https://www.sciencedirect.com/
[7] https://ieeexplore.ieee.org/

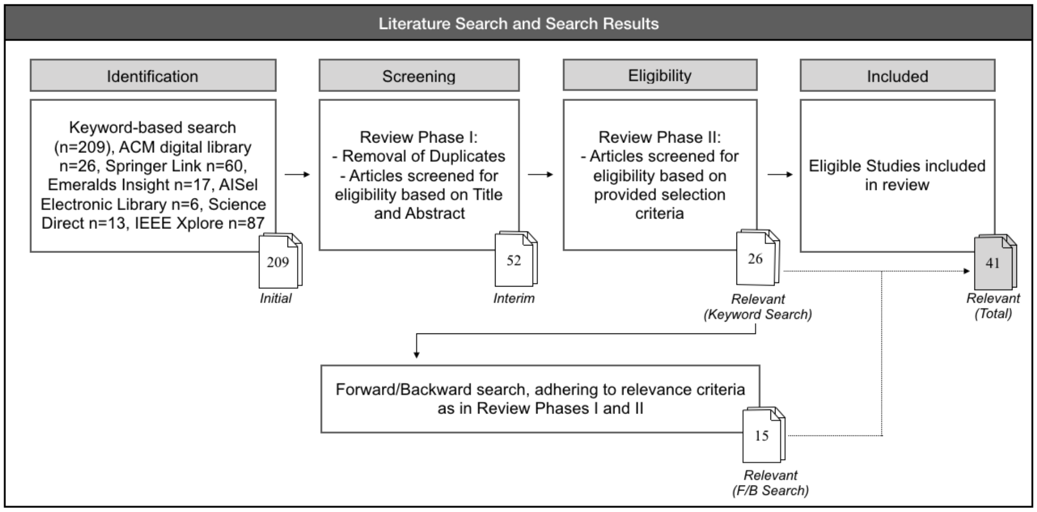


Figure 5: Selection process, including search results of the performed keyword search, backward search and forward search.

1. **Focus on Rule Organizing.** A paper was deemed relevant if its focus was an approach for business rule organizing. That is, following our research aim, the focus had to be on how to understand, cluster and select business rules to ensure the (long-term) error-freeness within a set of business rules.

2. **Focus on Business Rules.** A paper was deemed relevant if its focus was on business rules. That is, following our review scope, the focus had to be on rule formalisms which were applied in an enterprise context, in order to ensure that the identified approaches are applicable to aid companies.

In result, we obtained 26 relevant papers, in the following denoted as relevant (search) results.

As shown in Fig. 5, we then used the relevant results as a basis to retrieve more relevant works via a forward and backward search as proposed by (Brocke et al. 2009; Levy and Ellis 2006). For backward search, we conducted a backward search by references, where we regarded all the sources referenced by the initially found relevant results. Then, we conducted a forward search via Web of Science[8] to identify works that cited the relevant results.

To determine the relevance of the works found via forward- or backward search, we applied the same criteria as stated in the description of Review Phases I and II. Here, we identified 15 additional relevant works. To conclude, we identified a total number of 41 relevant results (26 KW + 15 F/B) by the means of our literature search. The literature search contains results up to January 31st of 2020.

One design-choice we would like to address is that we decided to only include works published from 2000. In our initial seminal search, we had noticed there were many recent publications, rarely any around the 2000s, and the again some publications dating back to the late 1980s to early 1990s. Here, one observation made was that earlier research often did not focus on currently used rule formalisms, as of course some of these standards were not even existant in the 1980s. Therefore, it would require more expertise to apply these results in an enterprise context, as

---

[8] https://apps.webofknowledge.com/

they would need to be adapted to current standards. Also, many recent works explicitly reference limitations of older works and therefore propose how to extend or adapt these results, e. g. (Calvanese et al. 2016). Therefore, we decided to make a cut at the 2000 mark, as many of recent works reference and extend older works, thus information on older works is still incorporated in our review, and the considered works are more accessible to companies based on using recent rule standards.

After our initial literature review, grounded in this body of literature and the acquired domain knowledge, the taxonomy development was iteratively conducted, as mentioned based on the conceptual-to-empirical approach as proposed in (Nickerson et al. 2013). For readability, we will introduce the developed taxonomy at this point and continue to discuss the development of the individual dimensions, respectively characteristics.

3.3 A Taxonomy for Business Rule Organizing Approaches

Fig. 6 shows the developed taxonomy for business rule organizing approaches. The shown dimensions and characteristics are defined as follows.

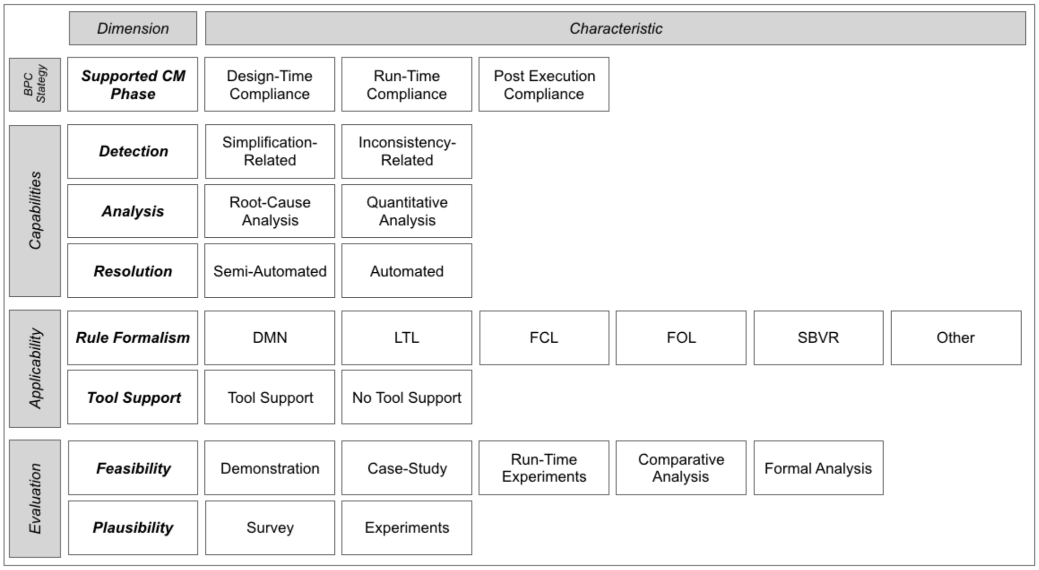
In regard to the meta-characteristic of the BPC phase, we introduced the dimension of a compliance management phase and were able to identify individual characteristics for this dimension. Following works such as (Hashmi et al. 2018; Sackmann et al. 2018), we identified the characteristics of design-time, run-time, and post-execution compliance management[9] , explained as follows:

- **Design-Time Compliance.** Approaches allowing to organize business rules at design-time, i. e. considering a set of business rules during modelling.
- **Run-Time Compliance.** Approaches supporting business rule organization during run-time. Here, these approaches must consider not only the business rules but also instance-dependent facts (Corea and Delfmann 2018b).
- **Post-Execution Compliance.** Approaches that enable to organize business rules in the scope of audits. Here, instance-dependent facts and/or observed process executions should be considered as well.

An important aspect is that rule organizing should ideally be conducted in regard to all three compliance management strategies: During design-time, run-time and post-execution compliance management, the information which can be used as a basis for rule organizing is highly different, and rule organizing has different goals during these perspectives. First, during design-time, only the business rules are known (but not the case-dependent facts that will be evaluated against the set of rules later during run-time). This means, that rule organizing at design-time focuses on finding logical contradictions or flaws *only* within the set of business rules. Eliminating such errors is a minimal prerequisite for deploying the rule set. Then, during run-time, case-dependent facts are known. The specific facts at run-time can yield novel errors in combination with rules (e. g. due to unexpected fact occurrences), which could not be detected at design-time (as it may not be possible to anticipate all possible fact combinations that can occur during run-time). Thus, despite design-time rule organizing, monitoring case-specific facts and organizing business rules during run-time (in regard to the facts) is also important to ensure that the modelled business rules are correctly aligned with reality. Then, during post-execution auditing, errors cannot only be investigated from an individual case perspective, but the interrelations of errors which occurred in different cases can also be used to further understand or prioritize issues in the rule base pertaining to individual cases. Thus, although a business rule and its representation remains the same at design-time, run-time and auditing, these perspectives yield different insights and allow to identify errors which are

---

[9] We decided to allow for objects in the taxonomy to be non-mutually exclusive in favor of conciseness (i. e., instead of specifying all possible combinations) based on the suggestion in (Nickerson et al. 2013).



	Dimension	Characteristic					
BPC Stategy	**Supported CM Phase**	Design-Time Compliance	Run-Time Compliance	Post Execution Compliance			
Capabilities	**Detection**	Simplification-Related	Inconsistency-Related				
	Analysis	Root-Cause Analysis	Quantitative Analysis				
	Resolution	Semi-Automated	Automated				
Applicability	**Rule Formalism**	DMN	LTL	FCL	FOL	SBVR	Other
	Tool Support	Tool Support	No Tool Support				
Evaluation	**Feasibility**	Demonstration	Case-Study	Run-Time Experiments	Comparative Analysis	Formal Analysis	
	Plausibility	Survey	Experiments				

Figure 6: Proposed taxonomy for BRO approaches in regard to BPC.

not detectable from the scope of the other strategies. In result, rule organizing approaches need to address all compliance management strategies.

Next, we conceptualized different dimensions for the meta-characteristic of capabilities. Following Corea and Delfmann (2018b), rule organizing can be divided into three general components, namely the *detection* of errors, the *assessment* of errors and the *resolution* of errors.

As a basis for rule organizing, detection comprises capabilities that allow to identify errors within a set of business rules. A detection of errors is the foundation for other capabilities. Here, many different types of errors in business rules can be defined and detected (we will discuss these error types below as characteristics). Following a detection of errors, recent works have advocated the importance of a (quantitative) analysis of the detected errors (Lu et al. 2008; Nagel et al. 2019; Sadiq and Governatori 2015). This analysis, mostly in the form of a quantitative assessment, can be presented to modellers in order to a) assess the *severity* of the detected errors, and b) provide a *prioritization* in which order rules should be attended to in the scope of re-modelling. A recent study by Nagel et al. (2019) shows that quantitative insights are associated with better understanding accuracy, better understanding efficiency and less mental effort needed for understanding problems in the scope of business rule organizing, as opposed to a manual analysis of errors. As shown in (Corea and Delfmann 2018b), a quantitative assessment can also be used as a driver for an informed re-modelling strategy. Finally, in the scope of ensuring error-freeness within a set of business rules, rule organizing can also include means for the actual resolution of the detected (and analyzed) problems. This can range from semi-automated resolution by the means of recommendation-systems to fully automated resolution algorithms.

We subsequently introduced the dimensions of detection, analysis and resolution under the meta-characteristic “capabilities”.

- **Detection.** Approaches allowing to detect, i. e. identify, errors within a set of business rules.

- **Analysis.** Approaches offering a detailed (quantitative) assessment of the detected errors, e. g. in order to foster an understanding and prioritization of specific errors.

- **Resolution.** Approaches fostering a (semi-) automated resolution of the detected errors, thus resulting in a sound set of business rules.

For these dimensions, we were able to identify several characteristics.

Especially for detection, it is important for companies to have a detailed overview of which types of errors can actually be detected. In this context, many different error classifications have been proposed. For example, Smit et al. (2017) recently proposed the BRM verification capability framework, which defines capabilities needed for business rule organizing. These capabilities include for example identical rules verification (checking if there are duplicate rules), subsumed rule verification (checking whether a rule is irrelevant), or interdeterminism verification (checking whether there are rules that yield contradictory conclusions). While those authors present a comprehensive framework, there exist too many different capabilities to identify a concise set of taxonomy characteristics.

We therefore propose to generally group error types in business rules and the corresponding detection capabilities into two main groups, namely *simplification-related* error types and *inconsistency-related* error types.

Simplification-related errors refer to multiple business rules which should be merged, or reduced. For example, if one would detect that two rules are identical, one can simply delete one of the two. Hence, for all simplification approaches, the resolution of the error is trivial, or at least undisputed.

Inconsistency-related errors are generally defined as business rules that yield logically contradictory conclusions. Here, handling inconsistency-related errors is not trivial. For example, two modelers with different views on the same domain of interest might have entered two contradictory rules, such as the conclusions *creditWorthy* and *not creditWorthy* for the same condition. In this case, it is not clear how to resolve the issue, as two contradictory pieces of information exist, which requires careful analysis by experts.

Not only is inconsistency in business rules more difficult to resolve, but it can also have much more dramatic effect on business process compliance. Consider again the example of a rule base containing two identical rules. While this is unarguably undesirable and can lead to problems in data maintenance, two identical rules do not necessarily impose a problem with regard to business process compliance (e. g., the worst case is that a compliance check would be conducted twice). However, in case of inconsistent business rules, the inconsistency makes it *impossible* to use the business rules for their intended purpose of governing compliant business process execution. Hence, while simplification should not be neglected, handling inconsistencies in business rules is an important challenge to address for companies. Consequently, we identify these two general error groups as characteristics for our taxonomy.

Regarding the dimension of analysis, the characteristics of root-cause analysis and quantitative analysis were identified.

- **Root-cause Analysis.** Means to identify the exact causes of the detected problems, e. g., pin-pointing specific rules which are highly problematic.

- **Quantitative Analysis.** Means to assess the severity of problems, e. g., by providing a quantitative assessment, such as numerical values, indicating the degree of the problem.

Last, for the dimension of resolution, we identify the characteristics of semi-automated resolution, e. g. by the means of recommendations, and fully-automated resolution, e. g. algorithms to automatedly resolve errors within sets of business rules based on our findings.

Continuing, the third meta-characteristic of applicability is aimed to capture characteristics impacting the adaptation or usage of methods and techniques, e. g. in an enterprise context. From our literature review, we observed that there was no clear consensus on which rule formalism to use. If a company has already invested efforts into modelling business rules, the support of specific rule standards may be a constraint for adaptation. Therefore, we introduced the dimension of "rule formalism". Based on our literature review, we could also identify some seminal rule standards, shown as characteristics in Fig. 6. We explicitly included a characteristic "other" as a design choice to warrant a higher flexibility of our taxonomy. Furthermore, regarding the applicability of individual approaches, it might be an important factor whether there exists an implementation that can be used "out-of-the-box", as opposed to having to invest efforts into implementing an approach. We therefore introduced the dimension "tool support".

Finally, the meta-characteristic of "evaluation" is meant to capture indicators for the maturity of approaches. Here, we distinguish between the evaluation dimensions of performance and plausibility. Performance evaluation is geared to show the general feasibility of the developed tools. Plausibility analysis refers to an analysis of approaches in the scope of surveys or experiments with human participants. In the identified literature, a multitude of different evaluation strategies were applied, and added to the taxonomy as characteristics. Please see Sect. 4.1.4 for a further discussion.

After three iterations, the applied taxonomy development approach was concluded. Next to the objective ending conditions, all subjective ending conditions were met. The taxonomy has 8 dimensions with a maximum of 6 characteristics, and is therefore concise and robust. By applying the conceptual-to-empirical approach following Nickerson et al. (2013), as well as the literature review following vom Brocke et al. (2009), the resulting taxonomy can be seen as comprehensive. It can be easily extended in future work, e. g. by adding more characteristics such as detection capabilities. Additionally, the taxonomy is explanatory, which will be further discussed in the next Section.

The proposed taxonomy extends the descriptive knowledge on business rule management and allows researchers and practitioners to classify rule organizing approaches. Intuitively, a limitation of our proposed taxonomy is the conceptualization by the researcher as dictated by the applied conceptual-to-empirical approach. This is however completely in line with the goal of developing taxonomies as defined by Nickerson et al. (2013), which is to provide a "useful" taxonomy, as opposed to the "best" or "correct" one (the latter of which is often intractible). As Iivari puts it: *"Conceptual knowledge, including taxonomies, does not have a truth value but is relevant input for the development of theories representing forms of descriptive knowledge"* (Iivari 2007)[p.5]. Here, our taxonomy provides a useful foundation for theory building and can guide future research.

Next to our research aim of developing such an initial taxonomy, based on calls in academia that an overview of specific rule organizing approaches is missing as well, our second main research aim is to provide such a needed overview. This is important for scholars and practitioners in order to gain insights into the state-of-the-art, and further allows to identify research gaps. In the following, we classify the works identified in our literature review using the proposed taxonomy in order to provide an overview of business rule organizing approaches.

4 An Overview of Business Rule Organizing Approaches

Based on the presented taxonomy, Tab. 1 shows a classification of rule organizing approaches identified in our literature search. Note that the characteristics regarding error detection are shown in Tab. 2 following the framework by (Smit et al. 2017)[10] .

---

[10] Interdeterminsm relates to the characteristic of inconsistency-related errors as defined in the proposed taxonomy. Tab. 2 is simply meant as a more detailed view, as "detection" is the most researched dimension.

	BPC Strategy			**Capabilities**			**Applicability**		**Evaluation**	
Literature	Design-Time Compliance	Run-time Compliance	Post-Execution Compliance	Detection*	Quantitative Analysis	Resolution	Rule Formalism	Tool	Performance	Plausibility
Kardasis et al. (2004)	X						Text			
Fu et al. (2004)	X			X			n/a	X	run-time (syn), complexity	
Lin et al. (2005)	X						FOL		case-study	
Bajec et al. (2005)	X						Text	X	case-study	
Hicks (2007)	X			X			Text	X		
Cheng et al. (2009)	X			X			n/a		run-time (syn)	
Governatori et al. (2009)	X			X		A	FCL			
Governatori et al. (2010)	X			X		A	FCL			
Levy et al. (2010)	X			X			RIF	X		
Maggi et al. (2011a)		X		X	X		LTL		case-study	
Maggi et al. (2011b)		X		X	X		LTL		demonstration	
Decker et al. (2013)	X			X	X		FOL			
da Silva et al. (2013)	X						DRL	X	case-study	
Maggi et al. (2013)	X			X		A	LTL	X	run-time (r)	
Berstell-Silva et al. (2014)	X			X		A	FOL			
Cuzzocrea et al. (2014)	X			X	X		FOL		formal analysis (p)	
Santos et al. (2014)	X			X		A	SBVR	X		
Zhang al. (2014)	X			X		A	n/a	X	run-time (syn), comparative	
Cemus et al. (2015)	X						n/a			
Gomez et al. (2016)		X		X			n/a			
Olivieri et al. (2015)	X			X		A	FCL			
Agli et al. (2016)	X			X		A	ILOG	X	Demonstration	
Burgstaller et al. (2016)	X						SBVR			
Calvanese et al. (2016)	X			X			DMN	X	run-time (r)	
De Smedt et al. (2016)	X			X			LTL		Demonstration	N=95, M=(UA, Q)
Di Ciccio et al. (2016)	X			X		A	LTL	X	run-time (r)	
Gomez et al. (2016)		X		X			n/a	X	Demonstration	
Houari et al. (2016)	X			X			n/a	X	run-time (syn)	N=10, M=Q)
Batoulis et al. (2017)	X			X			DMN		Demonstration, Case-Study	
Calvanese et al. (2017)	X			X			DMN			
Di Ciccio et al. (2017)	X			X		A	LTL	X	run-time(r), complexity analysis	
Anand et al. (2018)	X			X			SBVR	X	Case-Study	
Batoulis et al. (2018)	X			X		A	DMN	X	run-time(syn)	
Calvanese et al. (2018)	X			X		A	DMN	X	run-time, comparative	
Corea et al. (2018)		X	o	X	X	SA	FCL			
Corea et al. (2018)		X	o	X	X	SA	DMN	X	Demonstration	N=37, M=(UA, UE, OME, Q)
De Smedt et al. (2018)	X			X			LTL		Demonstration	N=146, M=(UA, UE, Q)
Ezekiel et al. (2018)	X			X			DRL	X	Demonstration	
Corea et al. (2019b)	X			X	X	A	LTL	X	run-time (r)	
Corea et al. (2019)	X			X			DMN	X	run-time (syn)	
Corea et al. (2019a)	X			X	X		LTL	X	run-time (r)	

Detection:* Please refer to Tab. 2 for an overview of sub-capabilities.
Resolution: SA (Semi-Automated), A (Automated).
Rule Formalisms: Text (textual description/natural language), n/a (Non-standard or individual rule formalisms, e. g. if-then-structures), FOL (First-order logic), FCL (Formal Contract Language), RIF (Rule Interchange Format), DRL (Drools Rule Language), LTL (Linear Temporal Logic, respectively Declare), SBVR (Semantics of Business Vocabulary and Business Rules), Ilog (IBM Ilog), DMN (Decision Model and Notation).
Performance Evaluation: cf. the description of evaluation techniques. r (Real-Life dataset), syn (Synthetic dataset).
Plausibility Evaluation: N (number of participants), M (measures used), UA (understanding *a*ccuracy), UE (understanding *e*fficiency), OME (*o*bjective *m*ental *e*ffort), Q (*q*uestionnaire)

Table 1: Overview of approaches for Business Rule Organizing, classified into the dimensions of BPC Strategy, Capabilities, Applicability and Evaluation.

4.1 Discussion

Our results provide an overview of 41 business rule organizing approaches. Fig. 7 shows the number of identified approaches per year. As can be seen, there is an increase in works since 2013, and especially since 2016.

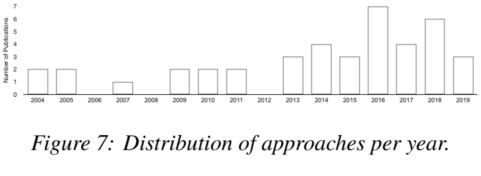


Figure 7: Distribution of approaches per year.

In the following, we discuss our results divided into the individual meta-characteristics, and then provide implications for practice and future research.

4.1.1 MC1: Supported Compliance Strategies

Fig. 8 shows the distribution of works supporting the individual BPC strategies. The majority of works (81%) are geared towards design-time compliance. This is in line with works such as (Hashmi et al. 2018; Olivieri et al. 2015; Sadiq and Governatori 2015), which strongly advocate the so-called compliance-by-design principle. In turn,

	Detection capabilities (detailed view)						
Literature	Identical Rules	Equivalent Rules	Subsumed Rules	Unnecessary Facts	Contradicting Conclusions	Overlapping Conditions	Missing Rules
Fu et al. (2004)	X		X			X	
Hicks (2007)					X		X
Cheng et al. (2009)	X		X		X		X
Governatori et al. (2009)	X		X			X	
Governatori et al. (2010)	X		X			X	
Levy et al. (2010)	X	X		X			
Maggi et al. (2011a)					X		
Maggi et al. (2011b)					X		
Decker et al. (2013)					X		
Maggi et al. (2013)			X			X	
Berstell-Silva et al. (2014)					X	X	X
Cuzzocrea et al. (2014)					X		
Santos et al. (2014)	X		X		X	X	
Zhang al. (2014)					X		
Gomez et al. (2016)				X		X	X
Olivieri et al. (2015)	X		X			X	
Agli et al. (2016)					X		
Calvanese et al. (2016)	X		X			X	X
De Smedt et al. (2016)					X		
Di Ciccio et al. (2016)	X		X		X	X	
Gomez et al. (2016)				X		X	X
Houari et al. (2016)	X		X	X	X	X	
Batoulis et al. (2017)	X		X			X	X
Batoulis et al. (2017)	X		X			X	X
Calvanese et al. (2017)	X		X			X	X
Di Ciccio et al. (2017)	X		X		X	X	
Anand et al. (2018)	X	X			X		
Batoulis et al. (2018)	X		X			X	
Calvanese et al. (2018)	X		X			X	X
Corea et al. (2018)	X				X	X	
Corea et al. (2018)	X				X	X	
De Smedt et al. (2018)					X		
Ezekiel et al. (2018)			X			X	
Corea et al. (2019b)	X				X	X	
Corea et al. (2019a)	X				X	X	
Corea et al. (2019)	X	X	X	X	X	X	X

Table 2: Detailed view of which types of errors can be detected by the individual approaches, following the capability classification by (Smit et al. 2017).

there is a variety of approaches that can be used to organize business rules at design-time. Only a few approaches support run-time compliance - even fewer support post-execution compliance. However, as pointed out in (Corea and Delfmann 2018b) or (Maggi et al. 2011a,b), instance dependent case input can reveal errors in the decision logic that cannot be identified at design-time. An inadequacy of rules might also only be observable considering the actual activities at run-time, due to unexpected or unpredicted behavior. Therefore, it is necessary to conduct run-time compliance management in order to detect potential inconsistencies in decisions during run-time, cf. e. g. (Corea and Delfmann 2018a) or (Maggi et al. 2011b). Furthermore, a holistic a posteriori analysis of business rules in the context of the observed behavior, e. g. event logs, can also provide valuable insights that cannot be inferred during design-time (Burattin et al. 2012). Therefore, future work is needed on BRO approaches that support run-time and post-execution compliance. Due to the complementary effects of using multiple compliance management strategies as described in (Hashmi et al. 2018), it would also be desirable to have holistic approaches covering e. g. both design-time compliance and run-time compliance, as there are currently no holistic approaches that can be used during all compliance stages.

While it would theoretically be possible to combine multiple techniques (that each only address one BPC strategy) in order to gain coverage, a unified solution might be more beneficial due to the additional insights that can be gained by not only considering phases individually, but from a holistic perspective. Also, trying to combine different techniques is currently limited by the existing approaches and tools: There exist no sufficient means for rule-organizing during post-execution compliance.

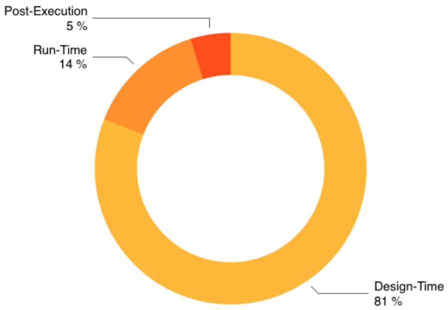


Figure 8: Distribution of BPC strategies supported by the individual approaches.

4.1.2 MC2: Capabilities

Tab. 1 classifies the capabilities of the considered approaches into the dimensions of detection, analysis and resolution. Fig. 9 shows the number of works that support these different high-level capabilities.

Detection Capabilities. In regard to the considered business rule management lifecycle, error detection within the organizing phase is usually performed *after* the rule authoring phase. That is, a given set of business rules needs to be analyzed in order to detect errors. As shown in Tab. 1, 84% of all works fall into this traditional use-case of rule organizing and can be used to detect some form of error within a given set of business rules. Interestingly, there are a few approaches which did not focus on the detection of errors, but instead on preventing errors during modeling. To clarify, such approaches are closely intertwined with the rule authoring phase, and aim to provide means to author business rules in such a way, that errors are proactively counteracted. Most prominently, this is performed by linking business rules to predefined rule schemas (Kardasis and Loucopoulos 2004; Lin et al. 2005; Santos Guimarães et al. 2014; Silva et al. 2013), meta-models (Bajec and Krisper 2005; Burgstaller et al. 2016) or semantic annotations (Ezekiel et al. 2018; Lévy et al. 2010) during rule authoring. In result, advanced reasoning capabilities can be used to support modellers in creating sound sets of business rules. Houari et al. (2016) even propose a procedure model, i. e. an actual guideline for modelers how to create business rules.

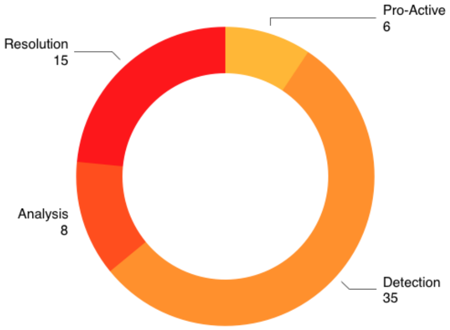


Figure 9: Number of approaches supporting the individual dimensions.

Yet, as mentioned the majority of approaches follow the more sequential business rule lifecycle model, i. e. a given set of business rules is assessed. Tab. 2 provides a detailed overview of the specific decision logic level verification capabilities offered by the individual approaches, outlined in the following. Note that "interdeterminism" as defined by those authors relates to inconsistency-related errors as defined in this work, and all other capabilities relate to simplification-related errors.

Simplification of business rules. A baseline approach is to simplify a set of business rules, e. g. by identifying identical rules, subsumed rules or overlapping conditions. When such errors are detected, they can be removed to reduce the size and complexity of the rule set. There are a substantial number of approaches geared towards this goal, available for various rule standards such as the Formal Contract Language (Governatori 2010; Governatori and Rotolo 2010), the Semantics of Business Vocabulary and Business rules (Anand et al. 2018), the Decision Model and Notation (Batoulis and Weske 2018; Calvanese et al. 2016, 2018, 2017; Corea et al. 2019), or Declare (Di Ciccio et al. 2017; Maggi et al. 2013). All these approaches present some form of algorithm geared towards identifying different errors within a set of business rules as shown in Tab. 2. Regarding these simplification approaches, there are only a few which support the detection of equivalent rules, e. g. logically equivalent rules (Anand et al. 2018; Calvanese et al. 2017) or a semantically equivalent rule vocabulary (Corea et al. 2019). Contrary to a simplification of rule sets, some approaches are also geared towards enhancing the rule set by identifying missing rules, e. g. gaps in rule conditions (Calvanese et al. 2016, 2017) or inadequacy of rules relative to business processes (Batoulis and Weske 2017).

On the concept of contradictory rules. A second interesting line of detection approaches are those concerned with contradictory rules. In essence, these approaches try to detect sets of rules that are logically contradictory or inconsistent. Other than with simplification or with missing rules, it is often not clear how to resolve this type of errors, as multiple, contradictory pieces of information exist and need to be resolved by experts. Regarding the concept of contradictory business rules, it is noteworthy that this notion is used very differently and not easily characterizable.

Di Ciccio et al. (2016; 2017) utilize automata representations of business rules in order to find inconsistent (sub)sets of business rules. Here, inconsistency is defined as a set of rules that cannot be satisfied, e. g. there cannot exist a sequence of activities that satisfies the set of constraints – hence the automata product is empty. Corea et al. (2019) extend this concept and introduce the notion of quasi-inconsistency, which relates to cases where a set of rules *can* become inconsistent, *should* certain rules be activated together. In such cases, there can in fact exist activity sequences that satisfy the rule set – however some particular sequences will render the model inconsistent. The approach by Corea et al. (2019) can thus be used to detect those activities that yield an inconsistency. Interestingly, de Smedt et al. (2016; 2018) propose a similar notion of hidden dependencies, which describes hidden relations of activities that can block each other in case of certain sequences of activities. For example, the occurrence of certain activities might block the execution of certain rules (however this is not made explicit from the rule set, hence the term hidden dependency). The approach by de Smedt et al. (2016; 2018) is similar to that in Corea et al. (2019), however where the former focuses more on possible (executable) sequences of activities, the latter focuses more on contradictions in business rules that can arise from particular activities.

Next to these mentioned approaches to detect contradictions at design-time, there are works geared towards detecting contradictions during run-time, e. g. inconsistent rule conclusions made during different points in time of running processes (Corea and Delfmann 2018a,b), inconsistencies in fact values during run-time (Gómez-López et al. 2016), or contradictions in business rules relative to observed (unexpected) behavior (Maggi et al. 2011a,b). Almost all works on run-time approaches provide examples where run-time errors occurred that could not be detected at design-time. This strongly advocates implementing multiple compliance management strategies.

Last, we can observe that there exist virtually no approaches supporting business rule organizing during post-execution compliance. The approach by Corea et al. (2018) can be used to analyze rules and facts that occured in a process *after* the actual process runtime, however this approach does not consider the interrelations of different process executions, i. e. no process log traces are analyzed or compared. Due to the increased availability of event logs (Burattin et al. 2012), future works should therefore focus on developing new means for a posteriori business rule organizations, as considering the relations of different traces could reveal even new problems which cannot be understood from a local run-time (i. e. individual process instance) perspective.

To summarize, regarding detection capabilities, we can identify four main capabilities which are commonly supported, namely *identifying identical rules, identifying redundant rules, identifying contradictions and inconsistencies*, and *identifying missing rules*. Only a few works support the *identification of semantically equivalent rules* or *unnecessary facts*. Detecting contradictions is supported by some works, though this area is still under development and the notion of contradiction and inconsistency are not yet clearly defined in literature. However, as contradictions in business rules are harder to solve than for example redundancies (as the former type of errors usually needs to be resolved by domain experts and it is not trivial how to resolve the problem), future work should continue to investigate the notion of inconsistency and contradiction in business rules.

Some approaches go beyond error detection. For example, some of the run-time approaches also allow companies to continue monitoring even after some inconsistencies were encountered during run-time (Maggi et al. 2011a), or support companies by automatically stopping process execution in the case inconsistencies were found, so that no compliance violations are committed (Corea and Delfmann 2018a). Also, we see some initial works aiming to apply results from knowledge representation such as inconsistency-tolerant reasoning (Agli et al. 2016; Cheng and Huang 2009; Cuzzocrea et al. 2014; Decker and Muñoz-Escoı 2013) or inconsistency measurement (Corea and Delfmann 2018b, 2019a,b; Cuzzocrea et al. 2014) to business rule organizing. This line of investigation should be continued in future work due to the strong alignment of goals.

What can be observed is that there is virtually no approach which covers all capabilities, except one approach for DMN decision tables (Corea et al. 2019). Here, future work needs to be directed towards developing new approaches that support a broader variety of detection capabilities for other standards. Also more approaches for post-execution should be investigated, as this could yield a further understanding of errors that cannot be detected during design-time or run-time.

Analysis Capabilities. Regarding the dimension of analysis, only very few approaches (13%) offer any form of quantitative insight. In general, quantitative insights for business rule organizing were divided into an overall assessment of the entire rule base, and the assessment of individual business rules.

Regarding *overall* rule base assessment, results from inconsistency measurement were primarily used to quantify the degree of inconsistency in business rule bases (Corea and Delfmann 2019a; Cuzzocrea et al. 2014; Decker and Muñoz-Escoı 2013). These results allow companies to understand the severity of overall problems in their rule bases. Future work should investigate further metrics for other error types in sets of business rules, see e. g. (Hasic et al. 2017) for a recent work in this direction.

Regarding the assessment of *individual* business rules, Corea et al. (2018; 2019) investigate culpability measures, which are quantitative measures which assign a numerical value to *individual* business rules. The intuition is that a higher value reflects a higher degree of blame, that an individual business rule carries in the overall context of inconsistency. In this way, highly problematic business rules can be pin-pointed. Furthermore, the modeler can be presented with a prioritization as a basis for an informed decision regarding a re-modelling strategy. Following Lu et al. (2008; 2015), such a prioritization of individual rules can be an important driver for experts during error resolution, as a resolution based on a "simple" detection might be unfeasible in practice. Also, in a recent study, Nagel et al. (2019) could in fact show the positive cognitive effects of such quantitative insights on understanding errors in business rules, where those authors found that culpability measures are associated with better understanding accuracy, better understanding efficiency and less mental effort needed for understanding contradictions in business rules. Subsequently, future work should focus on approaches which allow to assess individual business rules, to help companies to understand and prioritize errors.

Resolution Capabilities. Regarding the dimension of resolution, we see that a decent amount of approaches (40%) offer means for a (semi-) automated resolution. The high density of recent works which offer resolution mechanisms can be seen as positive because companies can be aided in efficiently resolving detected errors.

As discussed, the detection capabilities can be roughly divided into those who investigate a simplification of business rules, and those who investigate contradictory subsets of business rules. For the former, resolution is usually undisputed, e. g. redundant rules can be removed, or missing rules can be added. Consequently, many approaches that detect such errors also offer algorithms or fully automated means to resolve such errors (Calvanese et al. 2018; Di Ciccio et al. 2016, 2017; Governatori 2010; Governatori and Rotolo 2010; Maggi et al. 2013), or even for restructuring/improving the set of business rules (Batoulis and Weske 2018; Calvanese et al. 2016, 2018; Corea et al. 2019). For the latter type of approaches that detect contradictory sets of business rules, a resolution is not trivial and can depend on an assessment by domain experts. Di Ciccio et al. (2016; 2017) provide means for a fully automated resolution, however those authors state that there might be a chance certain information is deleted by the algorithm due to a trade-off between efficiency and considering only local optima. As deleting information may be a task which should be strongly supervised by domain experts, semi-automated approaches such as recommender-systems should also be investigated in order to guide modelers in resolving errors in a step-by-step manner, see e. g. (Corea and Delfmann 2019b) for a recent work in this direction.

4.1.3 MC3: Applicability

As companies might have already invested efforts in business rules elicitation and authoring, the rule formalism of approaches is an important criterion for the applicability in a company context, as business rules and the approach have to be compatible. Fig. 10 shows the percental distribution of which rule formalisms are used in the surveyed approaches. Declare is supported by the highest number of approaches, followed by the Decision Model and Notation. As can be seen in Fig. 7, there has been an increase in approaches since 2013. To gain a better understanding of recent approaches and relevant rule formalisms, we therefore investigated the development of rule formalisms support since 2013. Fig. 11 shows the percental distribution of formalisms supported by the respective approaches over time since 2013. As can be seen, from 2013-2015, there were a lot of approaches that used some non-standard formalisms or only supported business rules that followed a simple “if-then”-structure. Since around 2016, the evolution of standards is reflected in the proposed approaches, as approaches increasingly support maturing standards such as the Decision Model and Notation. Although it is interesting to see that these recent standards are being adapted in approaches, future work has to conclude the discussion on rule standards, such that there is a shared consensus and thus the approaches can be better aligned to support companies.

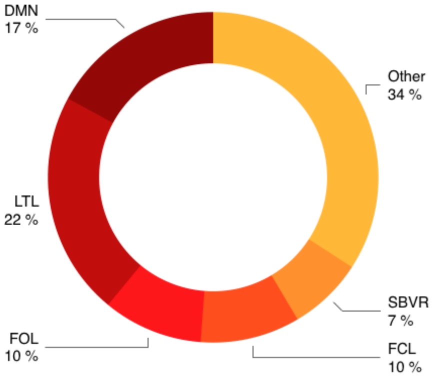


Figure 10: Distribution of rule formalisms supported by the identified approaches.

4.1.4 MC4: Evaluation of current approaches

The evaluation of proposed approaches is an essential component of design science research, allowing companies and scholars to assess and compare the maturity and quality of respective approaches. While the evaluation overview provided in Tab. 1 provides an indication towards feasibility and plausibility of approaches, a definite comparison of approaches is difficult at this point. In the following, we discuss the evaluation for those 26

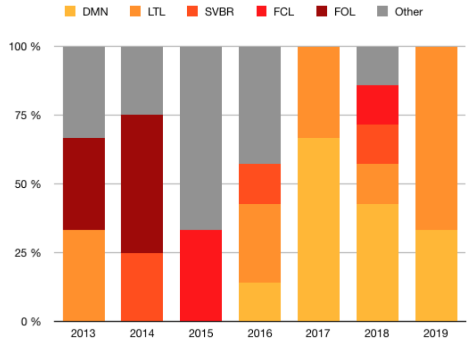


Figure 11: Percental distribution of supported rule formalisms over time.

out of 41 approaches that have been implemented (denoted as "tools"), as the evaluation of unimplemented approaches did not go beyond simple demonstrations.

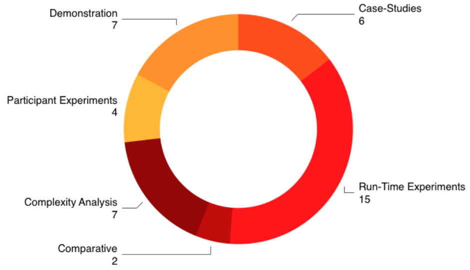


Figure 12: Number of works performing the individual evaluation techniques.

Fig. 12 shows an overview of evaluation techniques performed in the considered implemented tools. For the 26 (out of 41) implemented approaches, many authors only provide a demonstration, i. e. a short exemplary application of the proposed approach. While this is good for comprehension, the external validity of such toy examples is questionable. Here, roughly 20% of implemented works report on some case-studies, e. g. studies conducted with the tool and industrial partners. Such case-studies showcase the general feasibility of the proposed tools and make interesting cases for business rule organizing. More case-studies should be conducted to further motivate thee need for BRO approaches in relation to BPC, or to investigate the adoption of specific approaches. Furthermore, out of the 26 tools, 13 works presented some form of run-time evaluation. This can be seen as positive, as run-time experiments can showcase the feasibility for the approaches on synthetic or real-life data. What we would like to point out, is that it is still difficult to compare run-times, as often different logs were used. Future works should focus more on comparing own results to other works, such as in (Calvanese et al. 2018; Zhang et al. 2014), or in general comparative meta-studies, e. g. as in (Cemus et al. 2015; Olivieri et al. 2015). Ideally, (real-life) data sets or data generators should be made available. A BRO competition for certain verification capabilities could also be fruitful. Next to a feasibility analysis by the means of run-time experiments, an evaluation aspect which needs to be investigated further is the formal analysis of proposed algorithms and approaches. We found only two works who provide a complexity analysis of the proposed algorithms. As complexity analysis allows to abstract compared to run-time experiments, a formal analysis should be emphasized more in future work.

Next, some authors also investigate the plausibility of their approaches. That is, such works investigate the usability of their approaches, and whether their proposed approaches can actually have positive (cognitive) effects for business rule organizing. As a baseline approach, new approaches can be shown to participants, who can then try the tool and provide feedback on usability, as in (De Smedt et al. 2016, 2018; Houari and Taghezout 2016; Nagel et al. 2019). Moreover, De Smedt et al. (2016; 2018) and Nagel et al. (2019) also performed controlled experiments to assess the cognitive effects of their approaches. To this aim, those authors applied between-subject design experiments, where participants where split into groups – some groups of which had access to an individual tool, and some groups of which did not.

Then, those authors asked participants to solve general BRO tasks – as mentioned some groups with tool support. Then, different objective metrics were measured in order to identify significant (cognitive) effects of the proposed tools. Regarding these metrics, two baseline metrics seem to be understanding accuracy, i. e. the number of correct answers, and understanding efficiency, i. e. the speed in which the tasks were solved. Furthermore, Nagel et al. (2019) also used eye fixation duration using eye-tracking in order to measure the objective mental effort. As pointed out by those authors, instruments for neuro-physiological measurement such as eye-trackers or heart rate monitors are becoming more attainable, thus future works should consider to utilize these instruments in order to measure cognitive load and objective mental effort. Also, the evaluation of plausibility should in general be emphasized more in future works.

Interim Result 1 (Guideline for evaluation)
As an interim result, we propose the following guideline for rigorous evaluation of BRO approaches, based on the state-of-the-art evaluation techniques identified in our literature review.

- **Demonstration.** A demonstration of proposed approaches, e. g. by the means of examples, should be viewed as a minimum requirement.
- **Case-Studies.** If possible, approaches should be applied in (industrial) case-studies, to show general feasibility and make the case for BRO approaches.
- **Run-time Experiments.** Implemented approaches should be comprehensibly tested in run-time experiments. Ideally, data-sets should be made available and re-used by other others if applicable.
- **Comparison.** Proposed approaches should be compared to other approaches based on objective measures, e. g. run-times.
- **Complexity Analysis.** Next to run-time experiments, complexity of proposed algorithms should be formally analyzed.
- **Plausibility/Experiments.** The usability of proposed tools should be investigated. As a baseline approach, participants can use the tool and provide qualitative feedback. Furthermore, experiment design research should ideally be applied to measure significant effects of proposed approaches in terms of improved usability or understandability. Also, cognitive effects should be measured.

4.1.5 Summary

Our distilled overview shown in Tab. 1 compares and classifies existing business rule organizing approaches using our developed taxonomy. Here, our discussion illustrates the state-of-the art and can be used to derive research gaps. Our overview consequently reveals two results regarding a novel overview of BRO approaches and needed future research.

Result 1 (Overview)
The distilled results in Tab. 1 extend the current body of knowledge in business rule management research. Here, existing approaches are classified relatively to the considered meta-characteristics of their compliance phase, capabilities, applicability and evaluation. This overview can be used by scholars and practitioners to gain insights about the current state-of-the-art. In a company context, our results can be used as a basis for informed decisions: Here, we envisage two scenarios, a) a company already has a business rule base and is seeking for rule organizing means, or b) a company is seeking to newly implement the business rules management approach (and rule organizing as a part of it).

For the first case, we argue it is essential that rule organizing approaches fit the existing rule management environment of the company, in order to allow for a seamless integration. Here, our results can be used to verify if there are suitable approaches from a company's perspective. For example, if a company verifies the compliance of their process models via model query at design-time, a corresponding BRO approach should be aligned to this phase, hence the company can verify this in the presented overview (e. g. column 1). Moreover, Tab. 2 provides detailed insights into the actual errors which can be detected. A company can thus utilize the overview to ensure that certain error types are analyzed. Also, information about tool support and maturity (evaluation) can be used as a basis for an informed decision as to which approach to adapt. Here, especially the information on supported rule formalisms helps to identify approaches that fit the rule standard applied by the company.

For the second case (a company is starting to implement business rules management), the presented state-of-the-art as well as the presented taxonomy allow to derive some preliminary guidelines:

- **Embracing rule organizing.** In the field of rule management, it is often assumed that a sound set of business rules exist. Much efforts are therefore directed towards rule standards or elicitation techniques, neglecting the organizing phase. However, we see recent evidence from the field that companies are having problems in the rule authoring phase. Therefore, instead of putting pressure on modellers, companies should embrace the chances of implementing a rule organizing phase as part of their business rule management lifecycle. Time and resources should be dedicated to this task, in order to iteratively improve business rules and foster sustainable business rule management.
- **Taking a holistic perspective.** Rule organizing research has predominantly focused on design-time approaches. However, we see evidence that certain errors cannot be detected a priori, as they may arise due to case-dependent facts. Here, next to striving for compliant-by-design rule bases, companies should invest efforts into implementing compliance monitoring and reporting. This allows to react to rule errors occuring at run-time, and allows to further analyze errors in the scope of auditing. Ideally, this process should also be iterative, i. e., results from auditing should be considered in new design phases and so forth.
- **Current support of rule standards.** The choice of the actual rule formalism is important, as this strongly affects how legal regulations can be encoded and used. While there are many different types of business rules (Van der Linden et al. 2019), some rule standards have evolved that can all be broadly used within BPC. Still, there are some factors that should be considered in the selection. First, companies should carefully consider the skillset of the experts involved in rule management or BPC. That is, while some formalisms such as FCL may allow for a higher expressiveness, the complexity might be overwhelming in such cases. Here, the DMN standard which uses graphical representation seems to be evolving into a highly acknowledged standard with a good balance between complexity and expressiveness, as well as a well-defined semantics (Calvanese et al. 2016). Second, the actual formalisms that are supported by rule organizing approaches might also be interesting to consider. Here, we see that the three rule standards of Declare, FCL and DMN have a good tool support.

Result 2 (Research Agenda)

Our literature analysis also identified research gaps, cf. also the above discussion.

What stands out is that there currently exists no approach which supports all three compliance management phases. While works such as (Hashmi et al. 2018) strongly advocate using multiple compliance strategies, there are only very few approaches for run-time and post-execution compliance (and as mentioned no approach allowing for a holistic support of all phases). The key focus of future research should therefore be to understand the interconnections of these three phases in relation to business rule organizing, in order to develop suitable approaches.

Furthermore, the supported detection capabilities are rather scattered. Future works should seek to combine different verification capabilities in order to provide companies with unified solutions, as opposed to companies having to use

Dimension	Research needed on...
1 - *(CM Phase)*	• Approaches supporting run-time and post-execution compliance • Holistic BRO approaches adressing multiple phases
2 - *(Detection)*	• Pro-active procedure models to guide business rules authoring • Further research on error verification capabilities
3 - *(Analysis & Resolution)*	• More approaches facilitating quantitative assessment • Foundations of how to quantify inconsistencies or errors, including postulates/properties for quantitative measures • Means to guide modelers in remodelling (Recommendations) • Means to anticipate possible case-data and corresponding errors in business rules at design-time (Preemptive Diagnostics)
4 - *(Other)*	• More case-studies to motivate BRO • Identification of company needs to further extend the presented taxonomy • Strong need for comparative studies • More rigorous evaluation, including plausibility analysis (cf. Interim Result 1) • *Further conclusion on rule standards (part of a much higher scope)*

Table 3: Research Agenda: Overview of potential research avenues for future research on business rule organizing.

multiple tools. In this context, procedure models for guiding business rule authoring should also be investigated in order to guide modelers and proactively counteract potential problems in collaborative rule modeling.

Future works should also focus on a quantification of detected errors in order to provide quantitative insights for companies. Recent studies (Nagel et al. 2019) show that metrics on errors in business rules help employees to resolve cases with a better accuracy, in a smaller amount of time, and with less mental effort needed. Prioritizing errors in the form of recommender systems can thus create value for companies by facilitating innovation, e. g. speeding up re-modelling.

On a meta-level, more case-studies or research on maturity models would be beneficial to further motivate the case for BRO and study the adoption of BRO approaches. Also, the evaluation of BRO approaches could be more extensive and comparative, in order to better understand the advantages of newly proposed means.

5 Conclusion

The pressure on business process compliance has increased in recent years. While approaches to verify the compliance of process models relative to business rules have advanced rapidly, research on the preceding question on how to ensure a sound set of business rules as a valid input for such approaches has been neglected. Here, organizing business rules is a mandatory prerequisite to BPC and must therefore be implemented by companies. In this work, we presented a taxonomy for rule organizing approaches. Furthermore, we provided an overview of current rule organizing approaches based on a systematic literature review. Our results support scholars and organizations in classifying (existing) rule organizing approaches. Also, we identified challenges and points of interest, which should be addressed in future work to support companies in business rules management.

Intuitively, a limitation of our research approach is that the presented artifacts are based on the conceptualization by the researcher. Here, we applied well-established procedural approaches and suggestions by works such as (Brocke et al. 2015, 2009; Moher et al. 2009; Nickerson et al. 2013; Webster and Watson 2002) to warrant a rigorous research process.

Our results show that BRO approaches supporting the design-time phase are dominant, and holistic approaches aligned with all BPC phases need to be addressed. Also, our results indicate that error detection capabilities need to be unified in order to counteract the current fragmented landscape of verification capabilities.

We see a strong need for future research regarding the following two aspects. First, the interrelation of rule authoring and rule organizing should be studied more closely. Here, an understanding of how errors occur in business rules authoring could foster the development of pro-active approaches. Second, research on recommender-systems that are able to guide modelers in error resolution should be investigated. Providing prioritization or quantitative insights could help to support modelers in understanding problems. Investigating preemptive diagnostics, e. g. anticipating potential run-time problems based on possible case-data during business rule design, could also be beneficial to uncover unexpected behavior and unveil further hidden relations between business rules that were unintended by the modeler. In future work, our taxonomy should also be extended by conducting qualitative research with key practitioners in order to refine the taxonomy dimensions and characteristics.

The motivation of this work is to raise awareness for the need of business rule organizing as a prerequisite to BPC, as well as the challenges related to business rule organizing. Many approaches embedded in later phases of business rule management or business process compliance assume a correct set of business rules as an input. However, we actually see evidence that this can currently not be sufficiently ensured in practice (Batoulis et al. 2017; Calvanese et al. 2018; Corea and Delfmann 2018b; Di Ciccio et al. 2017; Sadiq and Governatori 2015; Smit et al. 2017). Companies thus need to be supported with means for business rules organizing. Here, our work contributes a needed foundation for future work on business rule organizing.

References

Agli H., Bonnard P., Gonzales C., Wuillemin P.-H. (2016) Business Rules Uncertainty Management with Probabilistic Relational Models. In: International Symposium on Rules and Rule Markup Languages for the Semantic Web. Springer

Anand K., Chittimalli P. K., Naik R. (2018) An Automated Detection of Inconsistencies in SBVR-based Business Rules Using Many-sorted Logic. In: International Symposium on Practical Aspects of Declarative Languages. Springer

Bajec M., Krisper M. (2005) A methodology and tool support for managing business rules in organisations. In: Information Systems 30(6)

Batoulis K., Nesterenko A., Repitsch G., Weske M. (2017) Decision Management in the Insurance Industry: Standards and Tools. In: Proceedings of the BPM 2017 Industry Track co-located with the 15th International Conference on Business Process Management

Batoulis K., Weske M. (2017) A Tool for Checking Soundness of Decision-Aware Business Processes. In: Proceedings of the BPM Demo Track and BPM Dissertation Award co-located with 15th International Conference on Business Process Modeling

Batoulis K., Weske M. (2018) Disambiguation of DMN Decision Tables. In: Business Information Systems - 21st International Conference. Lecture Notes in Business Information Processing Vol. 320. Springer, pp. 236–249

Berstel-Da Silva B. (2014) Verification of business rules programs. Springer

vom Brocke J., Simons A., Riemer K., Niehaves B., Plattfaut R., Cleven A. (2015) Standing on the Shoulders of Giants: Challenges and Recommendations of Literature Search in Information Systems Research.. In: Communications of the Association for Information Systems 37

vom Brocke J., Simons A., Niehaves B., Riemer K., Plattfaut R., Cleven A., et al. (2009) Reconstructing the giant: On the importance of rigour in documenting the literature search process.. In: European Conference on Information Systems

Burattin A., Maggi F. M., van der Aalst W. M., Sperduti A. (2012) Techniques for a posteriori analysis of declarative processes. In: 2012 IEEE 16th International Enterprise Distributed Object Computing Conference. IEEE, pp. 41–50

Burgstaller F., Steiner D., Schrefl M. (2016) Modeling context for business rule management. In: 18th Conference on Business Informatics (CBI). IEEE

Calvanese D., Dumas M., Laurson Ü., Maggi F. M., Montali M., Teinemaa I. (2016) Semantics and analysis of DMN decision tables. In: International Conference on Business Process Management. Springer

Calvanese D., Dumas M., Laurson Ü., Maggi F. M., Montali M., Teinemaa I. (2018) Semantics, analysis and simplification of DMN decision tables. In: Information Systems 78

Calvanese D., Dumas M., Maggi F. M., Montali M. (2017) Semantic DMN: Formalizing decision models with domain knowledge. In: International Joint Conference on Rules and Reasoning. Springer

Cemus K., Cerny T., Donahoo M. J. (2015) Evaluation of approaches to business rules maintenance in enterprise information systems. In: Proceedings of the 2015 Conference on research in adaptive and convergent systems. ACM

Cheng M.-Y., Huang C.-J. (2009) A Novel Approach for Treating Uncertain Rule-based Knowledge Conflicts.. In: Journal of Information Science & Engineering 25(2)

Cooper H. M. (1988) Organizing knowledge syntheses: A taxonomy of literature reviews. In: Knowledge in society 1(1), p. 104

Corea C., Blatt J., Delfmann P. (2019) A Tool for Decision Logic Verification in DMN Decision Tables. In: Proceedings of the Dissertation Award, Doctoral Consortium, and Demonstration Track at BPM 2019 co-located with 17th International Conference on Business Process Management, BPM 2019, Vienna, Austria, September 1-6, 2019.

Corea C., Delfmann P. (2018a) A Tool to Monitor Consistent Decision-Making in Business Process Execution. In: Proceedings of the BPM 2018 Demo Track co-located with the 16th International Conference on Business Process Management

Corea C., Delfmann P. (2018b) Supporting Business Rule Management with Inconsistency Analysis. In: Proceedings of the BPM 2018 Industry Track co-located with the 16th International Conference on Business Process Management

Corea C., Delfmann P. (2019a) Quasi-Inconsistency in Declarative Process Models. In: Business Process Management Forum - BPM Forum 2019, Vienna, Austria, September 1-6, 2019, Proceedings, pp. 20–35

Corea C., Delfmann P. (2019b) Resolving Inconsistencies in Declarative Process Models based on Culpability Measurement. In: 14th International Conference On Wirtschaftsinformatik

Cuzzocrea A., Decker H., Muñoz-Escoı F. D. (2014) Scalable Uncertainty-Tolerant Business Rules. In: International Conference on Hybrid Artificial Intelligence Systems. Springer

De Smedt J., De Weerdt J., Serral E., Vanthienen J. (2016) Improving understandability of declarative process models by revealing hidden dependencies. In: International Conference on Advanced Information Systems Engineering. Springer, pp. 83–98

De Smedt J., De Weerdt J., Serral E., Vanthienen J. (2018) Discovering hidden dependencies in constraint-based declarative process models for improving understandability. In: Information Systems 74, pp. 40–52

Decker H., Muñoz-Escoı F. D. (2013) Inconsistency-tolerant Business Rules in Distributed Information Systems. In: OTM Confederated International Conferences" On the Move to Meaningful Internet Systems". Springer

Di Ciccio C., Maggi F. M., Montali M., Mendling J. (2016) Ensuring model consistency in declarative process discovery. In: International Conference on Business Process Management. Springer

Di Ciccio C., Maggi F. M., Montali M., Mendling J. (2017) Resolving inconsistencies and redundancies in declarative process models. In: Information Systems 64

Ezekiel K., Vassilev V., Ouazzane K., Patel Y. (2018) Adaptive business rules framework for workflow management. In: Business Process Management Journal

Fu G., Shao J., Embury S. M., Gray W. A. (2004) Algorithms for analysing related constraint business rules. In: Data & knowledge engineering 50(2)

Gadi T. (2015) Methods of Detection and Resolution of Anomalies in Business Rule: Comparative Study. In: International Journal of Computer Applications 975

Gómez-López M. T., Gasca R. M., Pérez-Álvarez J. M. (2016) Compliance validation and diagnosis of business data constraints in business processes at runtime. In: Information Systems 48

Governatori G. (2010) Law, logic and business processes. In: Proceedings of the Third International Workshop on Requirements Engineering and Law. IEEE

Governatori G., Rotolo A. (2010) Norm compliance in business process modeling. In: Dean M., Hall J., Rotolo A., Tabet S. (eds.) Proceedings of the 4th International Web Rule Symposium. Springer

Governatori G., Sadiq S. (2009) The journey to business process compliance. In: Handbook of Research on Business Process Modeling. IGI Global

Graham I. (2007) Business rules management and service oriented architecture: a pattern language. John wiley & sons

Hashmi M., Governatori G., Lam H.-P., Wynn M. T. (2018) Are we done with business process compliance: state of the art and challenges ahead. In: Knowledge and Information Systems

Hasic F., De Smedt J., Vanthienen J. (2017) Towards Assessing the Theoretical Complexity of the Decision Model and Notation (DMN).. In: RADAR+ EMISA@ CAiSE, pp. 64–71

Hicks R. C. (2007) The no inference engine theory - Performing conflict resolution during development. In: Decision Support Systems 43(2)

Houari N. S., Taghezout N. (2016) A novel collaborative approach for business rules consistency management. In: International Conference on Decision Support System Technology. Springer

Iivari J. (2007) A Paradigmatic Analysis of Information Systems As a Design Science. In: Scandinavian J. Inf. Systems 19(2), p. 5

Imgrund F., Malorny M., Janiesch C. (2017) Eine Literaturanalyse zur Integration von Business Rules und Business Process Management. In: Towards Thought Leadership in Digital Transformation: 13. Internationale Tagung Wirtschaftsinformatik

Janssens L., Bazhenova E., De Smedt J., Vanthienen J., Denecker M. (2016) Consistent integration of decision (DMN) and process (BPMN) models. In: Proceedings of the CAiSE'16 Forum, at the 28th International Conference on Advanced Information Systems Engineering (CAiSE 2016) Vol. 1612. CEUR-WS. org, pp. 121–128

Kardasis P., Loucopoulos P. (2004) Expressing and organising business rules. In: Information and Software Technology 46(11)

Lévy F., Guissé A., Nazarenko A., Omrane N., Szulman S. (2010) An environment for the joint management of written policies and business rules. In: Tools with Artificial Intelligence (ICTAI). IEEE

Levy Y., Ellis T. J. (2006) A systems approach to conduct an effective literature review in support of information systems research.. In: Informing Science 9

Lin L., Embury S. M., Warboys B. C. (2005) Facilitating the implementation and evolution of business rules. In: 21st IEEE International Conference on Software Maintenance (ICSM'05). IEEE

Lu R., Sadiq S., Governatori G. (2008) Measurement of compliance distance in business processes. In: Information Systems Management 25(4)

Maggi, Bose R. J. C., van der Aalst W. M. (2013) A knowledge-based integrated approach for discovering and repairing declare maps. In: International Conference on Advanced Information Systems Engineering. Springer

Maggi, Montali M., Westergaard M., Van Der Aalst W. M. (2011a) Monitoring business constraints with linear temporal logic: An approach based on colored automata. In: International Conference on Business Process Management. Springer, pp. 132–147

Maggi, Westergaard M., Montali M., van der Aalst W. M. (2011b) Runtime verification of LTL-based declarative process models. In: International Conference on Runtime Verification. Springer, pp. 131–146

Moher D., Liberati A., Tetzlaff J., Altman D. G. (2009) Preferred reporting items for systematic reviews and meta-analyses: the PRISMA statement. In: Annals of internal medicine 151(4), pp. 264–269

Nagel S., Corea C., Delfmann P. (2019) Effects of Quantitative Measures on Understanding Inconsistencies in Business Rules. In: Hawaii International Conference on System Sciences. IEEE

Nelson M. L., Rariden R. L., Sen R. (2008) A lifecycle approach towards business rules management. In: Hawaii International Conference on System Sciences. IEEE, pp. 113–113

Nelson M. L., Sen R. (2014) Business rules management in healthcare: A lifecycle approach. In: Decision Support Systems 57, pp. 387–394

Nickerson R. C., Varshney U., Muntermann J. (2013) A method for taxonomy development and its application in information systems. In: European Journal of Information Systems 22(3), pp. 336–359

Olivieri F., Governatori G., van Beest N., Ghooshchi N. G. (2015) Declarative Approaches for Compliance by Design. In: Service Research and Innovation. Springer

Ramezani E., Fahland D., van der Werf J. M., Mattheis P. (2011) Separating compliance management and business process management. In: International Conference on Business Process Management. Springer

Rowley J., Slack F. (2004) Conducting a literature review. In: Management research news 27(6)

Sackmann S., Kuehnel S., Seyffarth T. (2018) Using Business Process Compliance Approaches for Compliance Management with Regard to Digitization: Evidence from a Systematic Literature Review. In: International Conference on Business Process Management. Springer

Sadiq S., Governatori G. (2015) Managing regulatory compliance in business processes. In: Handbook on Business Process Management 2. Springer

dos Santos Guimarães D., Lima P., Correa A. L., Schmitz E. A., Alencar A. J. (2014) A method for verifying the consistency of business rules using alloy. In: Proceedings of the Twenty-Sixth International Conference on Software Engineering & Knowledge Engineering

Schlosser S., Baghi E., Otto B., Oesterle H. (2014) Toward a functional reference model for business rules management. In: Hawaii International Conference on System Sciences. IEEE, pp. 3837–3846

da Silva C. F., Baptista L., da Cunha P. R., Melo P. (2013) Empowering end-users to manage business rules: the case of a graphical environment built for a telco. In: Concurrent Engineering Approaches for Sustainable Product Development in a Multi-Disciplinary Environment. Springer

Smit K., Zoet M., Berkhout M. (2017) Verification capabilities for business rules management in the Dutch governmental context. In: Research and Innovation in Information Systems (ICRIIS), 2017 International Conference on. IEEE

Van der Linden E., Smit K., Berkhout M. (2019) The Business Rule Type Jungle: An Explorative Analysis. In: Bled eConference

Webster J., Watson R. T. (2002) Analyzing the past to prepare for the future: Writing a literature review. In: MIS quarterly

Weiden M., Hermans L., Schreiber G., van der Zee S. (2002) Classification and representation of business rules. In: Proceedings of European Business Rules Conference, June Vol. 26

Weidlich M., Mendling J., Weske M. (2011) Efficient consistency measurement based on behavioral profiles of process models. In: IEEE Transactions on Software Engineering 37(3), pp. 410–429

Zhang Q., Zeng G., Xiao C., Yue Y. (2014) A rule conflict resolution method based on Vague set. In: Soft Computing 18(3)

7.10 On Quasi-Inconsistency and its Complexity

Source	On Quasi-Inconsistency and its Complexity
Authors	Carl Corea and Matthias Thimm
Outlet	Artificial Intelligence (AIJ)
Ranking	A*
Status	Published

On Quasi-Inconsistency and its Complexity

Carl Corea, Matthias Thimm
University of Koblenz-Landau

Abstract

We address the issue of analyzing *potential* inconsistencies in knowledge bases. This refers to knowledge bases that contain rules which will always be activated together, and the knowledge base will become inconsistent, should these rules be activated. We investigate this problem in the context of the industrial use-case of business rule management, where it is often required that sets of (only) rules are analyzed for potential inconsistencies, e. g., during business rule modelling. To this aim, we introduce the notion of quasi-inconsistency, which is a formalization of the above-mentioned problem of potential inconsistencies. We put a specific focus on the analysis of computational complexity of some involved problems and show that many of them are intractable.

1. Introduction

Inconsistency is a core problem in knowledge representation and reasoning, and usually refers to a knowledge base containing multiple pieces of information which cannot hold at the same time. For example, consider the following logic program K_1 (we will formalize syntax and semantics later)

$$K_1 = \{a; \quad b \leftarrow a; \quad \neg b \leftarrow a\}.$$

K_1 is inconsistent in the classic-logical sense, as the conclusions $b, \neg b$ cannot hold at the same time. Handling such inconsistencies is not only subject of research within the wider area of Knowledge Representation and Reasoning (KR), but also a problem often faced in practice, for example, in the field of *Business Rule Management* (BRM), cf. [1, 2] for an overview or [3, 4, 5] for some recent works. Informally speaking, business rules are declarative

---

[1] {ccorea,thimm}@uni-koblenz.de

statements about a domain of interest, which are used to model (external) regulations in order to govern compliant company activities. Inconsistencies in business rule bases are a challenge within BRM, as they impede using business rules for their intended purpose of reasoning about allowed company behavior [6].

Consider the exemplary rule base $\mathcal{B}_1$ defined via

$$\mathcal{B}_1 = \{\mathit{creditworthy} \leftarrow \mathit{newCustomer};\ \ \neg\mathit{creditworthy} \leftarrow \mathit{newCustomer}\}.$$

In contrast to $\mathcal{K}_1$, we see that this business rule base contains only rules, but no facts (such as a in $\mathcal{K}_1$). This is very common in a BRM setting, as business rules are modelled at design-time (and case-dependent facts will only be known during process execution). Yet, the company has to warrant a "consistent" rule-base during modelling. In this regard, we see that $\mathcal{B}_1$ is *consistent* in the classic-logical sense, as no non-trivial inference is possible without facts. Existing approaches to inconsistency-tolerant reasoning [7] and inconsistency measurement [8] would therefore not indicate any issue in this case. Yet, from a business rule management perspective, it does not make sense to have two rules which will a) always be activated together, but b) are contradictory, should they be activated. Intuitively, the rules in $\mathcal{B}_1$ would become useless for compliance reasoning, given we would encounter a fact *newCustomer* during run-time. Thus, methods are needed to analyze such *potential* inconsistencies in knowledge bases. This could be useful for companies, in order to handle modelling errors in the business logic and facilitate correct compliance reasoning.

In order to address the scenario discussed above, we introduce the notion of *quasi-inconsistency*. The intuition is that a rule base, i. e., a knowledge base containing only rules, is quasi-inconsistent, if there are rules that will always be activated together but yield inconsistent conclusions. This notion is a generalisation of the notion of *incoherence* from description logics and similar formalisms [9, 10, 11]. In this work, however, we consider a general rule-based knowledge representation formalism in order to theoretically investigate this notion in a broad manner (Section 2). We then define the notion of quasi-inconsistency (Section 3) and examine the complexity of several computational tasks involving quasi-inconsistency (Section 4). We conclude with a discussion in Section 5 where we also point out the relationship to the notion of incoherence.

2. Preliminaries

We consider a general but simple monotonic rule-based knowledge representation formalism, similar to logic programs with classical negation but without default negation [12]. Let $\mathcal{A}$ be a set of propositional atoms and $\mathcal{L}$ the corresponding set of literals, i. e., $\mathcal{L} = \{a, \neg a \mid a \in \mathcal{A}\}$ where $\neg$ is interpreted as classical negation. We abbreviate $\overline{\neg a} = a$ and $\overline{a} = \neg a$ for an atom a.

A rule r has the form

$$r: \quad l_0 \leftarrow l_1, \ldots, l_m. \tag{1}$$

with $l_0, \ldots, l_m \in \mathcal{L}$ and $m > 1$ (note that we require rules to have a non-empty premise on purpose). We abbreviate *head*$(r) = l_0$ and *body*$(r) = \{l_1, \ldots, l_m\}$. Let $\mathcal{R}_{\mathcal{L}}$ denote the set of all rules. A *rule base* $\mathcal{B}$ is a set of rules, i. e., $\mathcal{B} \subseteq \mathcal{R}_{\mathcal{L}}$, and a set of facts (=literals) $F \subseteq \mathcal{L}$ is also called *fact base*. For X being a rule base or fact base, let $\mathcal{A}(X)$ denote the set of atoms appearing in X.

An important fragment of $\mathcal{R}_{\mathcal{L}}$ is the set of *acyclic* rules bases, i. e., rule bases without any cycles in the rules. More concretely, the *dependency graph* $G_{\mathcal{B}}$ of a rule base $\mathcal{B}$ is a directed graph $G_{\mathcal{B}} = (\mathcal{B}, E_{\mathcal{B}})$ where $(r_1, r_2) \in E_{\mathcal{B}}$ for $r_1, r_2 \in \mathcal{B}$ iff *head*$(r_1) \in$ *body*(r_2). Let $\mathcal{R}_{\mathcal{L}}^{\text{acyclic}} \subseteq \mathcal{R}_{\mathcal{L}}$ be the set of those rule bases $\mathcal{B} \in \mathcal{R}_{\mathcal{L}}$ where $G_{\mathcal{B}}$ is acyclic. Computationally, rule bases in $\mathcal{R}^{\text{acyclic}}$ are easier to analyse than general rule bases, as we will see in the remainder of the paper. However, we will also consider the case of rule bases that may indeed contain cycles.

Example 1. *Consider the rule base $\mathcal{B}_2$ defined via*

$$\mathcal{B}_2 = \{b \leftarrow a; \quad c \leftarrow b; \quad d \leftarrow c\}.$$

$\mathcal{B}_2$ is acyclic, where $G_{\mathcal{B}_2} = (\mathcal{B}_2, E_{\mathcal{B}_2})$ with

$$\begin{aligned} E_{\mathcal{B}_2} = \{&(b \leftarrow a; \quad c \leftarrow b), \\ &(c \leftarrow b; \quad d \leftarrow c)\} \end{aligned}$$

A set $M \subseteq \mathcal{L}$ of literals is *closed* wrt. a rule base $\mathcal{B}$ and a fact base F, iff $F \subseteq M$ and for every rule of the form in (1), if $l_1, \ldots, l_m \in M$ then $l_0 \in M$. The *F-minimal model* M of a rule base $\mathcal{B}$, denoted by $\min_F(\mathcal{B})$ is the smallest (wrt. set inclusion) closed set of literals. A set M of literals is called *consistent* if it does not contain both a and $\neg a$ for an atom a. A rule base $\mathcal{B}$ is called *F-consistent* if its F-minimal model is consistent. If not, we call $\mathcal{B}$ F-inconsistent. Some obvious properties of F-consistency are summarised in the following result (given without proof).

Proposition 1. *Let $\mathcal{B}$ be a rule base and F a set of facts.*

1. *$\mathcal{B}$ is $\emptyset$-consistent.*
2. *$\mathcal{B}$ is $\mathcal{L}$-inconsistent.*
3. *If F is inconsistent then $\mathcal{B}$ is F-inconsistent.*
4. *If $\mathcal{B}$ is F-inconsistent then $\mathcal{B}$ is F'-inconsistent for every $F \subseteq F'$.*
5. *If $\mathcal{B}$ is F-consistent then $\mathcal{B}$ is F'-consistent for every $F' \subseteq F$.*

Due to property 3.) and 4.) from above we strengthen the notion of F-inconsistency as follows. We say that a rule base $\mathcal{B}$ is *minimally F-inconsistent* if 1.) F is consistent, 2.) $\mathcal{B}$ is *F-inconsistent*, and 3.) for every $F' \subsetneq F$, $\mathcal{B}$ is *F'-consistent.*

Example 2. *Consider the rule base $\mathcal{B}_3$ and fact base F_1, defined via*

$$\begin{aligned}\mathcal{B}_3 &= \{b \leftarrow a;\quad c \leftarrow b;\quad \neg c \leftarrow a\}\\ F_1 &= \{a\}\end{aligned}$$

Then we have F_1 is a consistent set of facts, $\mathcal{B}_3$ is $\{a\}$-inconsistent, and $\mathcal{B}_3$ is $\emptyset$-consistent, thus, $\mathcal{B}_3$ is minimally $\{a\}$-inconsistent.

3. Quasi-Inconsistency

When considering knowledge bases consisting only of rules, we see from Proposition 1 that every rule base is consistent in the classic-logical sense, i. e., $\emptyset$-consistent. In this work, we are, however, interested in cases where there exists a set of facts F, s.t. a rule base becomes F-inconsistent. That is, we are interested in cases where a rule base will become inconsistent, should certain facts be introduced. In industrial application scenarios such as business rule management, addressing this problem is of high interest, as the modelling of business rules happens at an earlier point in time, independent of facts, i. e., it cannot be foreseen which combination of (case-dependent) facts will occur "later" during run-time. Here, it becomes important for companies to analyze whether there exist *potential* inconsistencies in the rule base, such that experts can improve operations and counteract potential compliance breaches during process execution.

Example 3. *Consider the rule base $\mathcal{B}_4$ defined via*

$$\mathcal{B}_4 = \{e \leftarrow a;\quad \neg e \leftarrow c;\quad c \leftarrow a;\quad e \leftarrow b\}$$

Observe that $\mathcal{B}_4$ *is minimally* $\{a\}$*-inconsistent and minimally* $\{b, c\}$*-inconsistent. However, the* $\{a\}$*-, resp.* $\{b, c\}$*-inconsistency entail different types of problems:*

The $\{b, c\}$*-inconsistency says that whenever* b *and* c *are added to the rule base, we activate both* $\neg e \leftarrow c$ *and* $e \leftarrow b$*, which yields an inconsistent conclusion. However, there can of course be cases where only* b*, or only* c *occurs, thus it is possible to activate the rules individually and draw meaningful conclusions from the rule base. It may even be the case that* b *and* c *will never appear together during run-time, because there is some extrinsic constraint not allowing the two to appear together (for example* b *could be "The client is under-age" and* c *could be "The client is older than 60 years"; although these two statements are not direct complements of each other, they cannot be true at the same time). We therefore denote the* $\{b, c\}$*-inconsistency as a* potential issue *and we do not consider dealing with these in this paper.*

On the other hand, the $\{a\}$*-inconsistency says that whenever the rule* $e \leftarrow a$ *is activated, we automatically derive both* e *and* $\neg e$ *(because of the rules* $c \leftarrow a$ *and* $\neg e \leftarrow c$*). To clarify, the rules* $e \leftarrow a$ *and* $c \leftarrow a$ *can either not be activated at all, or, they will always be activated together but in this case yield inconsistent conclusions. Thus, these two rules cannot be used for any meaningful reasoning. We therefore denote the* $\{a\}$*-inconsistency as an* (actual) issue*.*

Intuitively, it is important to resolve *(actual) issues* in the scope of business rule management, as they clearly indicate a modelling error in the set of business rules. For that purpose, we will now introduce the notion of *quasi-inconsistency*. Informally, we say that a rule base is *quasi-inconsistent* if it contains rules that will always be activated together and yield inconsistent conclusions. For that we need some further notation.

Definition 1 (Rule Set activation)**.** *A set of facts* X activates *a finite set of rules* R *iff there is a sequence* $< r_1, \ldots, r_n >$ *with* $\{r_1, \ldots, r_n\} = R$ *such that*

1. $body(r_1) \subseteq X$
2. *for all* $i = 2, \ldots, n$ *we have* $body(r_i) \subseteq \{head(r_1), \ldots, head(r_{i-1})\} \cup X$

A set of facts X minimally (w.r.t. set-inclusion) activates *a set of rules* R *iff* X *activates* R *and there is no proper subset of* X *that activates* R*. If* X *(minimally) activates* R *we also say that* X *is a* (minimal) activation set *of* R*. Let* $\mathsf{ActSets}(R)$ *be the set of minimal activation sets of* R*.*

Intuitively, X is a set of facts sufficient for deriving all conclusions of rules in R. We summarise some obvious properties of activation sets in the following result (given without proof).

Proposition 2. *Let $\mathcal{B}$ be a rule base and X, X' sets of facts.*

1. *If X activates $\mathcal{B}$ and $X \subseteq X'$ then X' activates $\mathcal{B}$.*
2. *If $\mathcal{B} \in \mathcal{R}_{\mathcal{L}}^{acyclic}$ then $\mathsf{ActSets}(\mathcal{B}) = \{X_{\mathcal{B}}\}$ with*

$$X_{\mathcal{B}} = \bigcup_{r \in \mathcal{B}} body(r) \setminus \bigcup_{r \in \mathcal{B}} \{head(r)\}$$

In particular, note that 2.) means that acyclic rule bases have uniquely determined activation sets that have a simple characterisation (and can be computed in polynomial time).

Example 4. *We recall the rule base $\mathcal{B}_2$*

$$\mathcal{B}_2 = \{b \leftarrow a; \quad c \leftarrow b; \quad d \leftarrow c\}.$$

For each individual rule, its activation set consists simply of the body of the rule, i. e., $\{a\}$ is an activation set of $\{b \leftarrow a\}$. Furthermore, the set $\{a\}$ also activates the entire set $\mathcal{B}_2$.

Observe that a cyclic rule base may have multiple (minimal) activation sets.

Example 5. *Consider the rule base $\mathcal{B}_c = \{a \leftarrow b; \quad b \leftarrow a\}$. Then both $\{a\}$ and $\{b\}$ minimally activate $\mathcal{B}_c$.*

In general, $\mathsf{ActSets}(R)$ may become exponential in size. Consider, e. g., $R_i = \{a_1 \leftarrow b_1; \quad b_1 \leftarrow a_1; \quad \dots; \quad a_i \leftarrow b_i; \quad b_i \leftarrow a_i\}$ with $|\mathsf{ActSets}(R_i)| = 2^i$ for $i > 0$.

We are now ready to define *quasi-inconsistency* as follows.

Definition 2 (Quasi-Inconsistency)**.** *Let $R_1, R_2 \subseteq \mathcal{R}_{\mathcal{L}}$ be rule bases and X_1, X_2 be consistent sets of literals. A tuple (R_1, X_1, R_2, X_2) is called an* issue *iff*

1. *$X_1 \subseteq X_2$.*
2. *X_1 minimally activates R_1.*
3. *X_2 minimally activates R_2.*
4. *R_1 is X_1-consistent and R_2 is X_2-consistent.*
5. *$R_1 \cup R_2$ is X_2-inconsistent.*

A tuple (R_1, X_1, R_2, X_2) *is called a* minimal issue *iff there are no* R_1', R_2', X_1', X_2' *with* $R_1' \subseteq R_1$ *and* $R_2' \subseteq R_2$ *(one of these set inclusions being proper) such that* (R_1', X_1', R_2', X_2') *is an issue.*

A rule base $\mathcal{B}$ *is* quasi-inconsistent *iff there is an issue* (R_1, X_1, R_2, X_2) *with* $R_1, R_2 \subseteq \mathcal{B}$. *Then we also say that* (R_1, X_1, R_2, X_2) *is an issue of* $\mathcal{B}$. *Let* $\mathsf{Issues}(\mathcal{B})$, $\mathsf{MinIssues}(\mathcal{B})$ *be the set of all (minimal) issues of* $\mathcal{B}$, *respectively.*

In other words, an issue (R_1, X_1, R_2, X_2) describes a case where the activation of one set of rules R_1 implies the activation of a second set of rules R_2 and both sets together derive an inconsistency (while being consistent on their own). Note that for acyclic rule bases we can write an issue (R_1, X_1, R_2, X_2) simply as (R_1, R_2) as the minimal activation sets X_1 and X_2 are uniquely determined, cf. Proposition 2.

Example 6. *Consider the following rule bases* $\mathcal{B}_5$*–*$\mathcal{B}_7$*, defined via*

$$\begin{aligned}
\mathcal{B}_5 &= \{c \leftarrow a; \quad \neg c \leftarrow a, b\} \\
\mathcal{B}_6 &= \{c \leftarrow a, b; \quad \neg c \leftarrow a, d; \quad d \leftarrow b\} \\
\mathcal{B}_7 &= \{c \leftarrow a, f; \quad \neg c \leftarrow h, d; \quad d \leftarrow b; \quad f \leftarrow b; \quad h \leftarrow a\}
\end{aligned}$$

Then for

$$\begin{aligned}
t_1 &= (\{c \leftarrow a\}, \{a\}, \{\neg c \leftarrow a, b\}, \{a, b\}) \\
t_2 &= (\{c \leftarrow a, b\}, \{a, b\}, \{d \leftarrow b; \neg c \leftarrow a, d\}, \{a, b\}) \\
t_3 &= (\{f \leftarrow b; \quad c \leftarrow a, f\}, \{a, b\}, \\
&\qquad \{d \leftarrow b; \quad h \leftarrow a; \quad \neg c \leftarrow h, d\}, \{a, b\})
\end{aligned}$$

we have that t_1 *is an issue of* $\mathcal{B}_5$, t_2 *is an issue of* $\mathcal{B}_6$, *and* t_3 *is an issue of* $\mathcal{B}_7$. *Hence, all these rule bases are quasi-inconsistent (while being classically consistent).*

Example 7. *We recall the business rule base* $\mathcal{B}_1$ *from the introduction*

$$\begin{aligned}
\mathcal{B}_1 =\{&\mathit{creditworthy} \leftarrow \mathit{newCustomer}; \\
&\neg \mathit{creditworthy} \leftarrow \mathit{newCustomer}\}.
\end{aligned}$$

Then we have that

$$\begin{aligned}
t_4 =(&\{\mathit{creditworthy} \leftarrow \mathit{newCustomer}\}, \{\mathit{newCustomer}\}, \\
&\{\neg \mathit{creditworthy} \leftarrow \mathit{newCustomer}\}, \{\mathit{newCustomer}\})
\end{aligned}$$

is a minimal issue of $\mathcal{B}_1$, *i. e.,* $\mathcal{B}_1$ *is quasi-inconsistent (despite being classically consistent).*

We conclude this section with a result summarising some general properties of quasi-inconsistency (proofs are straightforward and omitted).

Proposition 3. *Let $\mathcal{B}_1, \mathcal{B}_2$ be rule bases.*

1. *If $\mathcal{B}_1$ is quasi-inconsistent and $\mathcal{B}_1 \subseteq \mathcal{B}_2$ then $\mathcal{B}_2$ is quasi-inconsistent.*
2. *If $\mathcal{B}_1 \subseteq \mathcal{B}_2$ then $\mathsf{Issues}(\mathcal{B}_1) \subseteq \mathsf{Issues}(\mathcal{B}_2)$.*
3. *If $\mathcal{B}_1 \subseteq \mathcal{B}_2$ then $\mathsf{MinIssues}(\mathcal{B}_1) \subseteq \mathsf{MinIssues}(\mathcal{B}_2)$.*

The means to detect quasi-inconsistency proposed in this work heavily rely on the notion of (minimal) issues. Especially as we envisage to apply our results to support an industrial use-case, the actual computation of issues needs to be addressed. In the following, we therefore investigate the computational complexity of various problems related to quasi-inconsistency.

4. Computational complexity

We assume familiarity with basic concepts of computational complexity and basic complexity classes such as P and NP, see [13] for an introduction.

We start with analysing the complexity of verification tasks pertaining to issues.

Lemma 1. *Let $\mathcal{B}$ be a rule base, $R_1, R_2 \subseteq \mathcal{B}$, and X_1, X_2 consistent sets of literals. Checking whether (R_1, X_1, R_2, X_2) is an issue can be done in polynomial time.*

Proof. We go through the properties of an issues step by step (compare with Definition 2):

1. Checking $X_1 \subseteq X_2$ is obviously polynomial.
2. Checking that X_1 activates R_1 is simple forward propagation (check every rule whether it can be activated with X_1 alone; if yes add the head of that rule to X_1 and continue). In order to check that X_1 minimally activates R_1 it suffices to check whether $X_1' \subseteq X_1$ where X_1' has exactly one fact less than X_1 does not activate R_1 (for all such X_1', which are exactly $|X_1|$ many)
3. Checking that X_2 activates R_2 is analogous.
4. Checking $X_1 \cup R_1 \not\models \bot$ is simple forward propagation and checking whether the set of derived literals is consistent; analogous for $X_2 \cup R_2 \not\models \bot$.
5. Checking $X_1 \cup R_1 \cup X_2 \cup R_2 \models \bot$ is analogous. □

A bit surprisingly maybe, even verifying minimal issues can be done in polynomial time.

Lemma 2. *Let $\mathcal{B}$ be a rule base, $R_1, R_2 \subseteq \mathcal{B}$, and X_1, X_2 consistent sets of literals. Checking whether (R_1, X_1, R_2, X_2) is a minimal issue can be done in polynomial time.*

Proof. Checking whether (R_1, X_1, R_2, X_2) is an issue can be done as in Lemma 1. For minimality, we check for each $R_1' \subseteq R_1, R_2' \subseteq R_2$, where exactly one of R_1', R_2' contains one rule less, whether $X_1 \cup R_1' \cup X_2 \cup R_2' \models \perp$. If that is the case, (R_1, X_1, R_2, X_2) cannot be a minimal issue. Note that there are only polynomially many tuples $(|R_1| + |R_2|)$ to check, each check being polynomial, cf. the proof of Lemma 1. □

Let us now turn to our central notion of quasi-inconsistency and the task to check whether a rule base is quasi-inconsistent:

Dec-QI	**Input**:	rule base $\mathcal{B}$
	Output:	TRUE iff $\mathcal{B}$ is quasi-inconsistent

It turns out that Dec-QI is intractable even for acyclic rule bases.

Proposition 4. Dec-QI *is* NP*-complete. This remains true even for* $\mathcal{R}_{\mathcal{L}}^{acyclic}$.

Proof. In order to show NP-membership consider the following non-deterministic algorithm. On input $\mathcal{B}$, first guess sets of rules R_1, R_2 with $R_1, R_2 \subseteq \mathcal{B}$ and consistent sets of literals X_1, X_2. If $t = (R_1, X_1, R_2, X_2)$ is an issue of $\mathcal{B}$, return TRUE, otherwise return FALSE. Observe that this check can be done in polynomial time according to Lemma 1. This shows Dec-QI $\in$ NP.

In order to show NP-hardness we reduce the classical satisfiability problem Sat to Dec-QI. Let Φ be an instance of Sat over the signature A (=set of atoms), i. e., $\Phi = C_1 \wedge \ldots \wedge C_n$ where each $C_i = l_{i,1} \vee \ldots \vee l_{i,n(i)}$ with literals $l_{i,1}, \ldots, l_{i,n(i)}$ over A for $i = 1, \ldots, n$ (recall that a literal is either an atom $a \in A$ or its negation $\neg a$). The question is whether Φ is satisfiable, i. e., whether we can find an interpretation $I : A \to \{\mathsf{T}, \mathsf{F}\}$ s. t. for all $i = 1, \ldots, n$ there is a $k \in \{1, \ldots, n(i)\}$ with $I(l) = \mathsf{T}$ (if $l = l_{i,k}$ is an atom) or $I(l') = \mathsf{F}$ (if $\neg l' = l_{i,k}$ is a negated atom). On input Φ we construct a rule base $\mathcal{B}_\Phi$ as follows. For each clause C_i, $i = 1, \ldots, n$, we create two new atoms α_i, α_i' (both with the informal meaning that α_i/α_i' is derivable in $\mathcal{B}_\Phi$ if C_i is satisfied). For each clause $C_i = l_{1,1} \vee \ldots \vee l_{1,n(i)}$ we construct $2n(i)$ rules of the form

$$\mathcal{B}_i = \{\alpha_i \leftarrow l_{i,1}; \quad \ldots \quad ; \quad \alpha_i \leftarrow l_{i,n(i)}; \quad \alpha_i' \leftarrow l_{i,1}; \quad \ldots \quad ; \quad \alpha_i' \leftarrow l_{i,n(i)}\}$$

Then we create yet another new atom π and define $\mathcal{B}_\Phi$ to be composed of the above rules and two further rules:

$$\mathcal{B}_\Phi = \mathcal{B}_1 \cup \ldots \cup \mathcal{B}_n \cup \{\pi \leftarrow \alpha_1, \ldots, \alpha_n; \quad \neg\pi \leftarrow \alpha'_1, \ldots, \alpha'_n\}$$

We claim that Φ is satisfiable iff $\mathcal{B}_\Phi$ is quasi-inconsistent. We first show that satisfiability of Φ implies quasi-inconsistency of $\mathcal{B}_\Phi$. Let I be an interpretation that satisfies Φ. Without loss of generality, for each clause C_i let $l_{1,i}$ be the literal that is satisfied by I (can be easily achieved by reordering the literals in each clause). Define $X = \{l_{1,1}, \ldots, l_{1,n}\}$ to be the set of all these literals. Furthermore, define

$$\begin{aligned} R &= \{\alpha_1 \leftarrow l_{1,1}; \quad \ldots; \quad \alpha_n \leftarrow l_{1,n}\} \\ R' &= \{\alpha'_1 \leftarrow l_{1,1}; \quad \ldots; \quad \alpha'_n \leftarrow l_{1,n}\} \\ R_1 &= R \cup \{\pi \leftarrow \alpha_1, \ldots, \alpha_n\} \\ R_2 &= R' \cup \{\neg\pi \leftarrow \alpha'_1, \ldots, \alpha'_n\} \end{aligned}$$

By construction, X activates R and R'. Let $X' \subseteq X$ s.t. X' minimally activates R resp R'. Then X' also minimally activates both R_1 and R_2. It follows that (R_1, X_1, R_2, X_1) is an issue of $\mathcal{B}_\Phi$ showing that the latter is quasi-inconsistent. For the other direction, assume that $\mathcal{B}_\Phi$ is quasi-inconsistent and let (R_1, X_1, R_2, X_1) be an issue of $\mathcal{B}_\Phi$. Observe that the only rules able to derive contradictory claims in $\mathcal{B}_\Phi$ are the two rules with heads π and $\neg\pi$, respectively. So one of these rules must be in R_1 and the other in R_2 (if they both would be in one of them this would violate condition 4 of Definition 2). Assume $\pi \leftarrow \alpha_1, \ldots, \alpha_n \in R_1$, in order to activate this rule, one rule for each α_i, $i = 1, \ldots, n$ needs to be present in R_1. Moreover, for each such rule, one of the literals of the corresponding clause must be present in an activation set X_1. From these literals, an interpretation I can be constructed satisfying all clauses C_i, $i = 1, \ldots, n$ in analogy to the reverse direction before (note that this interpretation is partial, not all propositions need to occur in $X_1 \cup X_2$; however, the truth value of the remaining propositions is irrelevant and can be set arbitrary).

Finally, observe that $\mathcal{B}_\Phi$ is of polynomial size wrt. Φ. This gives a polynomial-time reduction from SAT to DEC-QI, showing that the latter is NP-hard. The reader can easily verify that the construction above yields an acyclic rule base, also showing NP-hardness for this special case. □

Moreover, Given a rule $r \in \mathcal{B}$, checking whether r is contained in at least one minimal issue is also intractable.

Lemma 3. *Let $\mathcal{B}$ be a rule base and $r \in \mathcal{B}$. Checking whether there is a minimal issue (R_1, X_1, R_2, X_2) with $r \in R_1 \cup R_2$ is NP-complete.*

Proof. For NP-membership, we guess a tuple (R_1, X_1, R_2, X_2) with $r \in R_1 \cup R_2$ and check in polynomial time using Lemma 2 whether (R_1, X_1, R_2, X_2) is a minimal issue.

For NP-hardness, we use the exact same reduction as in the proof of Proposition 4 and ask whether the rule $\pi \leftarrow \alpha_1, \dots, \alpha_n$ is contained in a minimal issue. This is exactly the case iff the input CNF formula Φ is satisfiable. $\square$

We now consider some counting problems related to issues:

#ISSUES	**Input**:	rule base $\mathcal{B}$
	Output:	$\lvert\mathsf{Issues}(\mathcal{B})\rvert$
#MINISSUES	**Input**:	rule base $\mathcal{B}$
	Output:	$\lvert\mathsf{MinIssues}(\mathcal{B})\rvert$

Using a similar reduction as in the proof of Proposition 4 we can show the following results.

Proposition 5. #ISSUES *and* #MINISSUES *are* $\#\mathsf{P}$*-complete.*[2]

Proof. For $\#\mathsf{P}$-membership, Lemmas 1 and 2 already showed that checking whether a given tuple (R_1, X_1, R_2, X_2) is a (minimal) issue can be decided in polynomial time. It follows that #ISSUES and #MINISSUES are in $\#\mathsf{P}$.

For showing hardness, we reduce the problem #1-3-SAT to our problems that has been shown to be $\#\mathsf{P}$-complete in [14]. Given a formula Φ over $A = \{a_1, \dots, a_m\}$ in 3-CNF, i. e., a formula of the form $\Phi = (l_{1,1} \vee l_{1,2} \vee l_{1,3}) \wedge \dots \wedge (l_{n,1} \vee l_{n,2} \vee l_{n,3})$ (with exactly 3 literals per clause), we ask for the number of those interpretations $I : A \to \{\mathsf{T}, \mathsf{F}\}$ s. t. for all $i = 1, \dots, n$ there is *exactly one* $k \in \{1, \dots, n(i)\}$ with $I(l) = \mathsf{T}$ (if $l = l_{i,k}$ is an atom) or $I(l') = \mathsf{F}$ (if $\neg l' = l_{i,k}$ is a negated atom). We call an interpretation satisfying this condition *1-3-model* of Φ. On input Φ we construct a rule base $\mathcal{B}_\Phi$ as follows. For each clause C_i, $i = 1, \dots, n$, we create two new atoms α_i, α_i' (both with the informal meaning that α_i/α_i' is derivable in $\mathcal{B}_\Phi$ if C_i is satisfied). For each clause $C_i = l_{i,1} \vee l_{i,2} \vee l_{i,3}$ we construct six rules

---

[2] $\#\mathsf{P}$ is the complexity class of counting problems where the problem of deciding whether a particular element has to be counted is in P.

of the form

$$\begin{aligned}\mathcal{B}_i = \{&\alpha_i \leftarrow l_{i,1}, \overline{l_{i,2}}, \overline{l_{i,3}};\\ &\alpha_i' \leftarrow l_{i,1}, \overline{l_{i,2}}, \overline{l_{i,3}};\\ &\alpha_i \leftarrow \overline{l_{i,1}}, l_{i,2}, \overline{l_{i,3}};\\ &\alpha_i' \leftarrow \overline{l_{i,1}}, l_{i,2}, \overline{l_{i,3}};\\ &\alpha_i \leftarrow \overline{l_{i,1}}, \overline{l_{i,2}}, l_{i,3};\\ &\alpha_i' \leftarrow \overline{l_{i,1}}, \overline{l_{i,2}}, l_{i,3}\}\end{aligned}$$

Moreover, we create new atoms $\delta_1, \ldots, \delta_m$ and construct the following rules

$$\mathcal{B}_A = \{\delta_i \leftarrow a_i;\ \ \delta_i \leftarrow \neg a_i \mid i = 1, \ldots, m\}$$

Then we create yet another new atom π and define $\mathcal{B}_\Phi$ to be composed of the above rules and two further rules:

$$\begin{aligned}\mathcal{B}_\Phi = \mathcal{B}_1 \cup \ldots \cup \mathcal{B}_n \cup \mathcal{B}_A \cup \{&\pi \leftarrow \alpha_1, \ldots, \alpha_n, \delta_1, \ldots, \delta_m;\\ &\neg\pi \leftarrow \alpha_1', \ldots, \alpha_n', \delta_1, \ldots, \delta_m\}\end{aligned}$$

We now claim that the number of 1-3-models of Φ is exactly the number of issues of $\mathcal{B}_\Phi$, which is exactly the number of minimal issues of $\mathcal{B}_\Phi$.

Let I be a 1-3-model if Φ. Define (X, R_1, X, R_2) via

$$\begin{aligned}X &= \{a \mid a \in \mathcal{A}, I(a) = \mathsf{T}\} \cup \{\neg a \mid a \in \mathcal{A}, I(a) = \mathsf{F}\}\\ R_A &= \{\delta_i \leftarrow a_i \mid a_i \in X\} \cup \{\delta_i \leftarrow \neg a_i \mid \neg a_i \in X\}\\ R_1 &= R_A \cup \{\pi \leftarrow \alpha_1, \ldots, \alpha_n, \delta_1, \ldots, \delta_m\}\\ &\quad \cup \{\alpha_i \leftarrow l_{i,1}, \overline{l_{i,2}}, \overline{l_{i,3}} \mid I(l_{i,1}) = \mathsf{T}, I(l_{i,2}) = \mathsf{F}, I(l_{i,3}) = \mathsf{F}, i = 1, \ldots, n\}\\ &\quad \cup \{\alpha_i \leftarrow \overline{l_{i,1}}, l_{i,2}, \overline{l_{i,3}} \mid I(l_{i,1}) = \mathsf{F}, I(l_{i,2}) = \mathsf{T}, I(l_{i,3}) = \mathsf{F}, i = 1, \ldots, n\}\\ &\quad \cup \{\alpha_i \leftarrow \overline{l_{i,1}}, \overline{l_{i,2}}, l_{i,3} \mid I(l_{i,1}) = \mathsf{F}, I(l_{i,2}) = \mathsf{F}, I(l_{i,3}) = \mathsf{T}, i = 1, \ldots, n\}\\ R_2 &= R_A \cup \{\neg\pi \leftarrow \alpha_1', \ldots, \alpha_n', \delta_1, \ldots, \delta_m\}\\ &\quad \cup \{\alpha_i' \leftarrow l_{i,1}, \overline{l_{i,2}}, \overline{l_{i,3}} \mid I(l_{i,1}) = \mathsf{T}, I(l_{i,2}) = \mathsf{F}, I(l_{i,3}) = \mathsf{F}, i = 1, \ldots, n\}\\ &\quad \cup \{\alpha_i' \leftarrow \overline{l_{i,1}}, l_{i,2}, \overline{l_{i,3}} \mid I(l_{i,1}) = \mathsf{F}, I(l_{i,2}) = \mathsf{T}, I(l_{i,3}) = \mathsf{F}, i = 1, \ldots, n\}\\ &\quad \cup \{\alpha_i' \leftarrow \overline{l_{i,1}}, \overline{l_{i,2}}, l_{i,3} \mid I(l_{i,1}) = \mathsf{F}, I(l_{i,2}) = \mathsf{F}, I(l_{i,3}) = \mathsf{T}, i = 1, \ldots, n\}\end{aligned}$$

Observe that for each clause C_i, $i = 1, \ldots, n$ both R_1 and R_2 contain exactly one rule with head α_i and α_1', respectively, and that exact rule is activated by X. It follows that X minimally activates both R_1 and R_2. Furthermore,

$X \cup R_1 \not\models \perp$, $X \cup R_2 \not\models \perp$, and $X \cup R_1 \cup R_2 \models \perp$ and therefore (X, R_1, X, R_2) is an issue of B_Φ. (X, R_1, X, R_2) is also a minimal issue as every rule in R_1 and R_2 is needed to entail π and $\neg\pi$, respectively.

Conversely, let (X_1, R_1, X_1, R_2) be any issue. As the only derivable conflict in R_Φ is between π and $\neg\pi$, the corresponding rules must be present in R_1 and R_2, respectively. Assume $\pi \leftarrow \alpha_1, \ldots, \alpha_n, \delta_1, \ldots, \delta_m \in R_1$ then there has to be at least one rule from $\{\delta_i \leftarrow a_i;\ \ \delta_i \leftarrow \neg a_i\}$ for each $i = 1, \ldots, m$ in R_1. As X must be consistent not both rules can be activated, so there is exactly one of these rules in R_1. It also follows that X_1 is exactly the union of the premises of that rules. Due to $X_1 \subseteq X_2$ and X_2 must be consistent it follows $X_1 = X_2$ (the addition of any literal makes X_1 inconsistent). For each α_i there must be at least one of the three rules with head α_i in R_1, otherwise α_i could not be derived. As at most one of these rules can be activated by X_1, there is exactly one of the three rules in R_1 (which is also activated, otherwise (X_1, R_1, X_1, R_2) would not be an issue). The same applies to R_2 and the rules with head α'. Now observe that (X_1, R_1, X_1, R_2) is also a minimal issue as every rule in R_1 and R_2 is needed to entail π and $\neg\pi$, respectively. Moreover, define now an interpretation I via $I(a) = \mathsf{T}$ if $a \in X_1$ and $I(a) = \mathsf{F}$ if $\neg a \in X_1$. As X_1 activates each rule with head α_i for $i = 1, \ldots, n$, I satisfies exactly one literal of each clause C_i. It follows that I is a 1-3-model of Φ.

It follows that each 1-3-model of Φ corresponds exactly to one (minimal) issue of $\mathcal{B}_\Phi$. Therefore, their number is exactly the same. As $\mathcal{B}_\Phi$ is of polynomial size wrt. Φ we have shown that both #Issues and #MinIssues are #P-hard. $\square$

5. Discussion and Conclusion

In this paper, we addressed the problem of potential inconsistencies in rule bases. As motivated in the introduction, this use-case is very common in domains such as business rules management, where rules that will always be activated together (but yield inconsistent conclusions) relate to actual modelling errors and need to be resolved by experts. To this aim, we introduced the notion of quasi-inconsistency and discussed various aspects related to the computational complexity on this matter.

Our notion of quasi-inconsistency is a generalisation of incoherence [9, 10, 11]. The notion of incoherence refers to the problem of unsatisfiable concepts in, e. g., description logics. For example, consider the description logic statements $A \sqsubseteq B$ (every A is a B), $A \sqsubseteq C$ (every A is a C) and $B \sqcap C \sqsubseteq \perp$ (there is nothing that is both B and C). Here, the concept A

is unsatisfiable, because in the presence of an individual that is of concept A, we derive an inconsistency. Phrased in our (propositional) language, an unsatisfiable concept corresponds to a (minimal) issue (R_1, X_1, R_2, X_2) where $X_1 = X_2$ is a singleton set. Therefore, quasi-inconsistency covers a wider spectrum of phenomena than that of incoherence.

Our results can be used to provide companies with an initial analysis of potential inconsistencies in business rule bases, and thus promotes inconsistency handling in the scope of business rule management. While our investigation on computational complexity showed that many problems are intractable (in the worst case), it has to be noted that this in line with results from classical inconsistency measurement [15], which investigates similar problems as we do. Actually, there are many classical inconsistency measures where decisions variants of computing their value, are higher up the polynomial hierarchy than "merely“ NP. Indeed, the NP-completeness result of Proposition 4 and the #P-result of Proposition 5 allows us to use mature satisfiability solving and model counting techniques [16] for our problems. While this work focuses on an initial detection of potential inconsistencies, due to the close relation to the field of inconsistency measurement, it seems promising to also investigate means for assessing the *severity* of potential inconsistencies, which is part of future work.

Acknowledgements

This research is part of the research project "Handling Inconsistencies in Business Process Modeling“, which is funded by the German Research Association (reference number: DE1983/9-1).

References

[1] I. Graham, Business rules management and service oriented architecture: a pattern language, John wiley & sons, 2007.

[2] M. L. Nelson, R. Sen, Business rules management in healthcare: A lifecycle approach, Decision Support Systems 57 (2014) 387–394.

[3] K. Batoulis, M. Weske, Disambiguation of DMN decision tables, in: Business Information Systems - 21st International Conference, 2018, pp. 236–249.

[4] K. Smit, M. Zoet, M. Berkhout, A verification framework for business rules management in the dutch government context, International Journal On Advances in Systems and Measurements 12 (2019) 101–112.

[5] C. Corea, P. Delfmann, Quasi-inconsistency in declarative process models, in: International Conference on Business Process Management, Springer, 2019, pp. 20–35.

[6] M. Hashmi, G. Governatori, H.-P. Lam, M. T. Wynn, Are we done with business process compliance: state of the art and challenges ahead, Knowledge and Information Systems 57 (1) (2018) 79–133.

[7] A. Hunter, L. Bertossi, T. Schaub (Eds.), Inconsistency Tolerance, Springer, 2005.

[8] M. Thimm, Inconsistency measurement, in: Proceedings of the 13th International Conference on Scalable Uncertainty Management (SUM'19), 2019, pp. 9–23.

[9] C. A. D. Deagustini, M. V. Martinez, M. A. Falappa, G. R. Simari, How does incoherence affect inconsistency-tolerant semantics for datalog?, Annals of Mathematics and Artificial Intelligence 82 (2018) 43–68.

[10] G. Flouris, Z. Huang, J. Z. Pan, D. Plexousakis, H. Wache, Inconsistencies, negations and changes in ontologies, in: Proceedings of the 21st National Conference on Artificial Intelligence (AAAI'20), AAAI Press, 2006, pp. 1295–1300.

[11] G. Qi, A. Hunter, Measuring incoherence in description logic-based ontologies, in: Proceedings of the 6th international The semantic web and 2nd Asian conference on Asian semantic web conference, ISWC'07/ASWC'07, Springer-Verlag, Berlin, Heidelberg, 2007, pp. 381–394.

[12] M. Gelfond, V. Lifschitz, Classical negation in logic programs and disjunctive databases, New Generation Computing 9 (1991) 365–385.

[13] C. Papadimitriou, Computational Complexity, Addison-Wesley, 1994.

[14] N. Creignou, M. Hermann, On #p-completeness of some counting problems, Tech. rep., INRIA (1993).

[15] M. Thimm, J. P. Wallner, On the complexity of inconsistency measurement, Artificial Intelligence 275 (2019) 411–456.

[16] A. Biere, M. Heule, H. van Maaren, T. Walsh (Eds.), Handbook of Satisfiability, Vol. 185 of Frontiers in Artificial Intelligence and Applications, IOS Press, 2009.

[17] J. Grant, A. Hunter, Measuring consistency gain and information loss in stepwise inconsistency resolution, in: European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty, Springer, 2011, pp. 362–373.

[18] M. Thimm, J. P. Wallner, Some complexity results on inconsistency measurement, in: Proceedings of the 15th International Conference on Principles of Knowledge Representation and Reasoning (KR'16), 2016, pp. 114–123.

Appendix

Some initial measures for assessing the "degree" of quasi-inconcistency can be defined using similar principles as "traditional" measures (cf. [17, 8]), e.g., the drastic inconsistency measure or measures built on minimal inconsistent subsets (which in our case correspond to minimal issues).

Definition 3 (The drastic quasi-inconsistency measure)**.** *Define the* drastic quasi-inconsistency measure *via*

$$\mathcal{I}_d^Q(\mathcal{B}) = \begin{cases} 1 & \textit{if } \mathcal{B} \textit{ is quasi-inconsistent} \\ 0 & \textit{otherwise} \end{cases}$$

Next, a second approach using the set of minimal issues is to take its cardinality.

Definition 4 (MI-quasi-inconsistency measure)**.** *Define the* MI-quasi-inconsistency measure *via*

$$\mathcal{I}_{\mathsf{MI}}^Q(\mathcal{B}) = |\mathit{MinIssues}(\mathcal{B})|$$

As a third approach, we adapt the *problematic measure* [17]. In our setting, we define it as the number of rules that contribute to at least one issue of the rule base.

Definition 5 (Problematic quasi-inconsistency measure)**.** *Define the* problematic quasi-inconsistency measure *via*

$$\mathcal{I}_p^Q(\mathcal{B}) = |\bigcup_{(R_1,X_1\, R_2,X_2) \in \mathit{MinIssues}(\mathcal{B}))} R_1 \cup R_2|$$

We now turn to the computational complexity of these inconsistency measures proposed. We focus on the natural function problem $\textsc{Value}_{\mathcal{I}}$, see also [18, 15], defined via

$\textsc{Value}_{\mathcal{I}}$	**Input**:	rule base $\mathcal{B}$
	Output:	The value of $\mathcal{I}(\mathcal{K})$

We start with the drastic inconsistency measure $\mathcal{I}_d^Q$, which can be easily analysed building on the general results on quasi-inconsistency we established so far.

Corollary 1. $\textsc{Value}_{\mathcal{I}_d^Q}$ *is in* FNP*-complete.*[3]

Proof. Proposition 4 established that it is NP-complete to check whether a rule base $\mathcal{B}$ is quasi-inconsistent. Therefore, a single NP-oracle call suffices to determine whether $\mathcal{I}_d^Q(\mathcal{B}) = 0$ or $\mathcal{I}_d^Q(\mathcal{B}) = 1$ for any rule base $\mathcal{B}$, showing that $\textsc{Value}_{\mathcal{I}_d^Q}$ is in FNP.

For FNP-hardness, we can utilise the fact that $\textsc{FSat}$, i. e., the problem of determining a satisfying interpretation of a CNF formula Φ is FNP-complete. Using the same reduction as in the proof of Proposition 4 we can reduce $\textsc{FSat}$ to a variant of $\textsc{Value}_{\mathcal{I}_d^Q}$ that also returns an issue. $\square$

We now turn to $\mathcal{I}_{\mathsf{MI}}^Q$. Observe that $\#\textsc{MinIssues}$ is equivalent to $\textsc{Value}_{\mathcal{I}_{\mathsf{MI}}^Q}$ and due to Proposition 5 we get:

Corollary 2. $\textsc{Value}_{\mathcal{I}_{\mathsf{MI}}^Q}$ *is* $\#\mathsf{P}$*-complete.*

We conclude this investigation of computational complexity with considering the final inconsistency measure $\mathcal{I}_p^Q$.

Proposition 6. $\textsc{Value}_{\mathcal{I}_p^Q}$ *is in* $\mathsf{FP}^{\mathsf{NP}[n]}$.[4]

Proof. Lemma 3 established that checking for a single rule $r \in \mathcal{B}$ whether it is contained in a minimal issue, is NP-complete. In order to determine $\mathcal{I}_p^Q(\mathcal{B})$ we simply perform this check for every rule $r \in \mathcal{B}$, needing polynomial time with $|\mathcal{B}|$ many NP-oracle calls. $\square$

We believe the previous result can be strengthened to $\mathsf{FP}^{\mathsf{NP}[\log n]}$, but we will leave a deeper investigation for future work.

---

[3] FNP is the class of function problems that can be solved by a non-deterministic Turing machine in polynomial time.

[4] $\mathsf{FP}^{\mathsf{NP}[n]}$ is the class of functional problems that can be solved by a deterministic Turing machine in polynomial time that can issue at most a linear number of NP-oracle calls.

7.11 DMN Change Patterns for Dynamic System Evolution

Source	Decision Model Change Patterns for Dynamic System Evolution
Authors	Faruk Hasic, Carl Corea, Jonas Blatt, Patrick Delfmann and Estefania Serral
Outlet	Knowledge and Information Systems Journal (KAIS)
Ranking	B
Status	Published

Decision Model Change Patterns for Dynamic System Evolution

Faruk Hasić[a,*], Carl Corea[b], Jonas Blatt[b], Patrick Delfmann[b], Estefanía Serral[a]

[a] *Research Centre for Information Systems Engineering (LIRIS)*
Warmoesberg 26, 1000 Brussels, Belgium
[b] *University of Koblenz-Landau, Institute for IS Research, Koblenz, Germany*

Abstract

In the modern digital era, information systems must operate in increasingly interconnected and dynamic environments, which force them to be changeable yet consistent. Such modern information systems are usually decision- and knowledge-intensive. A recently introduced standard, the Decision Model and Notation (DMN), has been adopted in both industry and academia as a suitable method for modelling decisions and decision rules. Noteworthy is that, despite the dynamic nature of modern knowledge-intensive systems, DMN was only studied and implemented in a static fashion, as decision schema change patterns have not received any attention so far. This paper identifies and analyses the change patterns that can occur in a DMN decision model. A change in the decision model can require the triggering of other changes in order to safeguard consistency. As such, this paper will also investigate for each change pattern which further changes should be performed to ensure model consistency. The patterns presented in this paper will not only facilitate the understanding of decision change management and within-model consistency, but can also be capitalised on for developing and implementing flexible decision management systems. To illustrate this, we present a modelling environment prototype that provides modelling support when applying the proposed change patterns.

Keywords: DMN, Decision Model and Notation, Model Evolution, Change Patterns.

---

[*] Corresponding author
Email address: `faruk.hasic@kuleuven.be` (Faruk Hasić)

1. Introduction

Decision Model and Notation (DMN) is a recently introduced decision modelling standard that has enjoyed significant interest in literature [1–8]. DMN consists of two levels that are to be used in conjunction. First, the decision requirement level represented by the Decision Requirement Diagram (DRD) which depicts the requirements of decisions and the dependencies between elements involved in the decision model. Second, the decision logic level, which presents ways to specify the underlying decision logic. The DMN standard employs rectangles to depict decisions and subdecisions, ovals to represent data input, corner-cut rectangles for business knowledge models, and curved rectangles to represent knowledge sources. A schematic example of a DRD is provided in Figure 1. The decision logic level is usually represented in the form of decision tables.

DMN has mainly been adopted in business process management (BPM) literature, which moves towards accommodating decision management into the paradigms of Separation of Concerns (SoC) and Service-Oriented Architecture (SOA), by externalising decisions and encapsulating them into separate decision models [2, 9–13]. Hence, the decisions are implemented as externalised services. This externalisation of decisions from processes, provides a plethora of advantages regarding maintainability and flexibility for the process [2, 5, 14–16]. However, all current works approach DMN from a static perspective, i.e., no attention has been given to changing or adaptable decision models. Nevertheless, the dynamic nature of modern knowledge-intensive systems demands a degree of flexibility to cope with the changing requirements. In this paper we approach this research gap by identifying and defining the change patterns that can be applied to DMN decision models. Additionally, we asses the effect of each applied change pattern on decision model consistency and we investigate and determine the change propagation that is needed to be triggered to keep such consistency. These findings can aid in developing and implementing dynamic decision management tools and systems that meet the requirements that the digital era demands.

The contribution of this paper is fourfold:

1. We are first to identify and define change patterns for DMN decision models.
2. We analyse the influence of each decision model change pattern on the within-model consistency.
3. We investigate the propagation of decision model change patterns throughout the decision model, i.e., we discuss the chain of change patterns that can be set in motion to restore within-model consistency after applying a single change pattern on the decision model.
4. We develop a modelling environment prototype which provides modelling support when applying the change patterns proposed in this paper.

This paper is structured as follows. Section 2 discusses related work and Section 3 provides preliminary formalisation needed for the construction of the change patterns. In Section 4 we introduce a running example of a DMN decision model which will be used to illustrate the change patterns. Section 5 introduces a set of DMN decision model change patterns, while Section 6 discusses the inconsistencies and change propagation induced by the change patterns. In Section 7 we present the modelling environment prototype which provides modelling support for change pattern administration in DMN decision models. Finally, Section 8 provides conclusions and directions for future research.

2. Related Work

DMN was first introduced by the Object Management Group (OMG) in 2015. Since then, several works on DMN have emerged. Most works have focused on the integration of process and decision models from a modelling point of view e.g., [2, 15, 17–23]. Others focus on the automatic discovery of decision model from enriched process event logs [9, 24]. Works on soundness of DMN models have been proposed as well [25, 26]. Furthermore, literature provides a set of tools for modelling DMN models [5, 11, 12, 27]. Before the introduction of DMN, The Decision Model (TDM) was introduced for modelling business logic and as a new requirements artifact for decision-aware business processes [28, 29]. Furthermore, works on ontologies for knowledge-intensive business processes have been suggested [30, 31]. These consider additional relevant aspects in decision making which DMN does not cover or which cannot always be adequately represented in a decision table, such as advantages, disadvantages, risks, facts, evidence and feelings. Nonetheless, in certain knowledge-intensive settings, these aspects can cause changes that affect the consistency of a decision model.

The ability of a knowledge-based system to efficiently deal with decision rule changes is considered of paramount importance in literature [32–36]. This way, business rigidity is avoided and the ability to transform the underlying business rules to new realities is facilitated. However, the lack of research on process models at runtime is emphasised in [37], and existing DMN decision model literature addresses decision modelling from a static perspective where decision models are built and used, without any form of model evolution. Nevertheless, adaptable models are considered of paramount importance [37–39]. Change patterns are often used to define the possible evolution of models. These change patterns rely on the elementary edit operations that can be applied on the model elements, i.e., insertion and deletion, as well as substitution, which in essence is a combination of insertion and deletion [40]. Furthermore, change patterns can help facilitate the understanding of model change management as they provide a guide for implementing changes to models while maintaining model consistency.

Change propagation throughout the model is an important aspect in hierarchical structures where referential integrity needs to be upheld. Typically, such structures are connected by tables, such as the relational database structure. The changes in relational tables are propagated by rules such as on delete cascade or on update cascade [41, 42]. Decision tables that are used in decision models are of a similar hierarchical structure, and hence, similar change propagation can be expected. However, instead of cascading an update or delete action throughout the whole model, the change propagation can also be captured by applying other change patterns. Nonetheless, change patterns for decision models were not addressed yet in literature and, to the best of our knowledge, this is the first paper on change pattern propagation for DMN decision models.

3. Preliminaries

In this section, we also provide a formalisation for key DMN concepts needed for the development of the change patterns that will be discussed in the following sections. For the formalisation, we rely on [2, 43]. The backbone of a DMN decision model is formed by rectangles that depict decisions and subdecisions and ovals that represent data input. Additionally, business knowledge models and knowledge sources can be defined. The underlying decision logic is usually represented in decision table form.

Definition 1 (Decision requirement diagram (DRD)). A DRD is a tuple (D_{dm}, ID, BK, KS, IR, KR, AR) consisting of a finite non-empty set of decision nodes D_{dm}, a finite non-empty set of input data nodes ID, a finite non-empty set of business knowledge model nodes BK, and a finite non-empty set of knowledge source nodes KS. These nodes are connected by requirements into a directed acyclic graph (DAG): a finite non-empty set of directed edges IR representing the information requirements such that $IR \subseteq (D_{dm} \cup ID) \times D_{dm}$, a finite non-empty set of knowledge requirements KR such that $KR \subseteq BK \times (D_{dm} \cup BK)$, and a finite non-empty set of authority requirements AR such that $AR \subseteq (D_{dm} \cup ID \cup KS) \times (D_{dm} \cup BK \cup KS)$.

A schematic DRD is represented in Figure 1. According to the DMN standard, a decision requirement graph can be an incomplete or partial representation of the decision requirements in a decision model. The set of all DRDs in the decision model constitutes the exhaustive set of requirements. The information contained in this set can be combined into a single DRD representing the decision requirements level as a whole. The DMN standard refers to such a DRD as a decision requirement graph (DRG). We expand the notion of a DRG, in such a way that a DRG is a DRD which is self-contained, i.e. for every decision in the diagram all its requirements are also represented in the diagram.

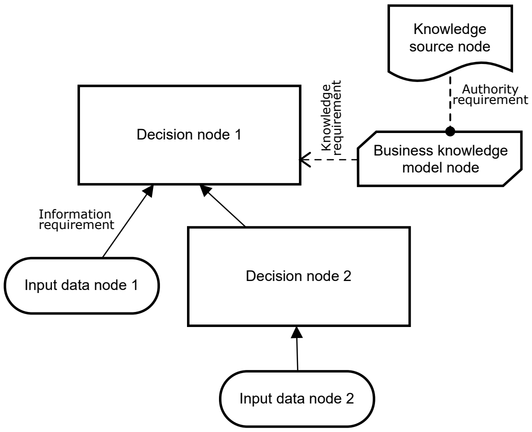


Figure 1: A schematic overview of DRD model elements.

Definition 2 (DRG). A decision requirement diagram DRD is a decision requirement graph DRG if and only if for every decision in the diagram all its requirements are also represented in the diagram.

The term *decision* can have a number of meanings. According to the DMN specification a decision is the logic used to determine an output from a given input. Meanwhile, in process modelling an activity can be a decision task, e.g., the business rule task in the Business Process Model and Notation (BPMN) standard [44] in which a decision is made by applying decision logic [29].

Another common meaning is that a decision is the actual result, which we call the output of a decision, or simply the decision result. We define a decision as follows:

Definition 3 (Decision). A decision $d \in D_{dm}$ is a tuple (I_d, O_d, L), where $I \subseteq ID$ is a set of input symbols, O a set of output or result symbols and L the decision logic defining the relation between symbols in I_d and symbols in O_d.

In case of decision tables, I and O contain the variables of the input and output elements respectively, and L is the table itself, i.e. the set of decision rules present in the table.

In DRDs these decisions d_i are contained by the decision nodes $D_i \in D_{dm}$. One can state that d_i and D_i are equivalent views at different levels of granularity: d_i looks at a decision on its own, while D_i places d_i in the hierarchy of the DRD. We will use D to refer to both a decision and its representing node in a DRD. From the definition of DRGs, it is clear that every decision D in a DMN model has a unique decision requirement graph DRG_D with D as its single top-level decision. A DRG contains all information requirements of its top level decisions. Hence, only one DRG exists with D as its single top-level decision, i.e., DRG_D. Furthermore, all the decisions in a DRG_D, except D, are consequently subdecisions of D. In other words, the top-level decision *requires* these lower level subdecisions.

4. Running Example of a DMN Decision Model

Consider a patient health monitoring system for a person diagnosed with the Chronic Obstructive Pulmonary Disease (COPD). COPD is a disease that obstructs the lungs and obstructs the airflow and breathing of the patient. Acute attacks of the disease can happen. In that case the patient can experience uncomfortable complications such as fast breathing, a fast heart rate, hyperactive use of muscles, and a cold skin [45]. In scientific literature it has been recognised as well that a patient monitoring system can help increase the life quality of the patient and decrease the risks that are inherent to the disease [45]. Multiple sensors and wearable technologies exist that can collect patient data relevant for the patient monitoring process [45]:

- Electrocardiogram (ECG) sensors monitor the heart.
- Respiratory sensors check the breathing rate.
- Skin temperature sensors monitor the skin temperature.
- Muscular Electromyography (EMG) sensors monitor the muscle activity.

All these sensors collect measurements on the patient's health. Note that a single sensor or even a few sensors combined are not enough to capture the COPD. For instance, the patient might take a walk outside in the winter and a sensor registers a low skin temperature. In that case, the patient is not necessarily suffering from COPD at that moment. However, an expert can build patient-specific decision rules to capture COPD in such a monitoring system. For instance, if the sensors register a low skin temperature, a short and fast breathing rate, together with high blood pressure, the monitoring process might decide that the patient is suffering an attack and running out of oxygen.

A suitable way of modelling such decisions in complex environments is through the Decision Model and Notation (DMN) standard, since the standard provides maintainability and understandability of decisions [1, 46, 47]. Figure 2 gives an example decision requirements graph of a DMN model for COPD severeness based on data gathered by the sensors. Figure 3 provides the top-level COPD decision table.

5. Decision Model Change Patterns

To accommodate changes, designers should be able to evolve the decision models. A number of changes can occur in the decision model. The core elements of a decision model are depicted

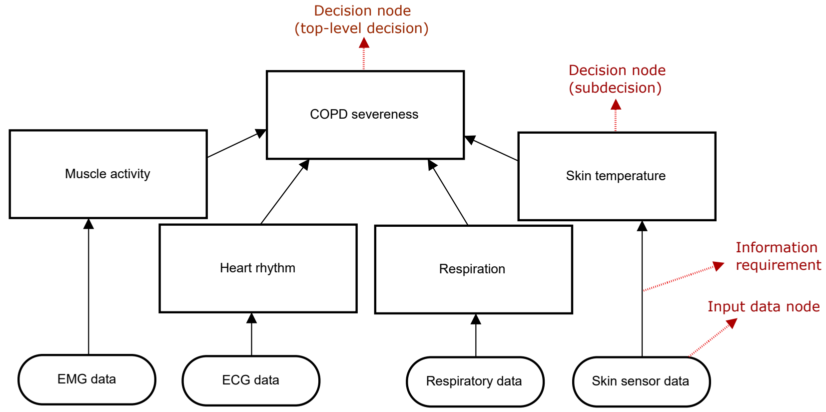
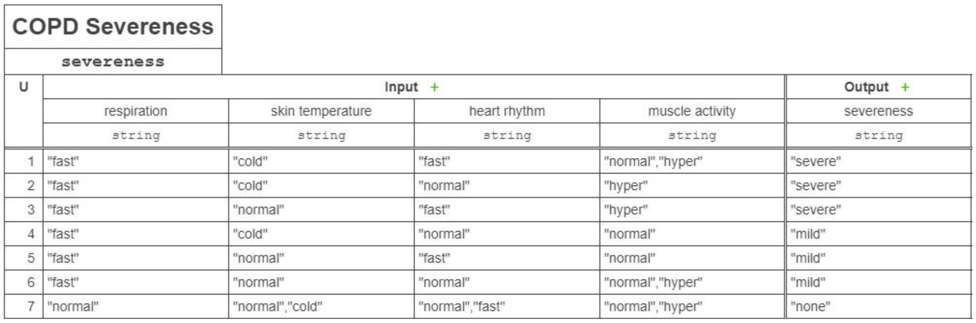


Figure 2: A DRD model for COPD severeness based on data from the sensors.



COPD Severeness

severeness

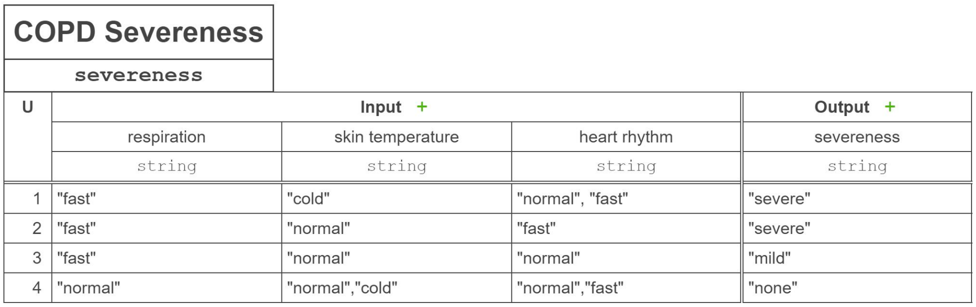
U	Input				Output
	respiration	skin temperature	heart rhythm	muscle activity	severeness
	string	string	string	string	string
1	"fast"	"cold"	"fast"	"normal","hyper"	"severe"
2	"fast"	"cold"	"normal"	"hyper"	"severe"
3	"fast"	"normal"	"fast"	"hyper"	"severe"
4	"fast"	"cold"	"normal"	"normal"	"mild"
5	"fast"	"normal"	"fast"	"normal"	"mild"
6	"fast"	"normal"	"normal"	"normal","hyper"	"mild"
7	"normal"	"normal","cold"	"normal","fast"	"normal","hyper"	"none"

Figure 3: A decision table for COPD severeness.

in Figure 2, i.e. the input data and the decision nodes within a DRD, connected via information requirements arrows. The logic encapsulated in a decision node is usually modelled with decision tables, such as shown in Figure 3. To determine the change patterns, we investigate the changes that can manifest themselves on core DMN elements, provided in the metamodel of the DMN specification [1], at different levels of granularity. An overview of the full DMN metamodel is provided in Figure 17 in the Appendix of this paper (Section 9), followed by more detailed meta models for the distinctive parts of DMN. First, we assess the change patterns within a single decision rule, i.e., changing the inputs and outcomes of a single rule. Next, we look at change patterns for a decision rule in its entirety, i.e., adding or deleting decision rules from a decision table. These change patterns all pertain to a single node of the DRD, i.e., a single decision table, according to the DMN decision table metamodel (see Figure 18). Furthermore, we investigate change patterns on the topological structure of the DRD itself, i.e., the addition and deletion of core DRD elements (decision nodes and data input nodes), as specified in the metamodels in Figures 19 and 20. The change patterns are derived from the formalisation of core decision model elements in Section 3 and the elementary edit operations that can be applied on the elements, i.e., insertion and deletion, as well as substitution, which in essence is a combination of insertion and deletion [40]. We illustrate each change pattern on the running example introduced in Section 4. Table 1 provides an overview of the change patterns directly relating to core DMN elements.

5.1. Decision table change patterns

For changing decision rules, we can distinguish a plethora of change patterns. We denote a change pattern with $\Delta\Pi$, Π refers to a pattern, while Δ stands for change, e.g., ΔI_d denotes a change in the input variables of a decision node. In essence, three elements in the decision table can undergo changes, as derived from Definition 3: the inputs I_d, the outputs O_d and the logic L mapping the inputs to the outputs, i.e. the decision rules. We define these changes in what follows.



COPD Severeness

severeness

U	Input +			Output +
	respiration	skin temperature	heart rhythm	severeness
	string	string	string	string
1	"fast"	"cold"	"normal", "fast"	"severe"
2	"fast"	"normal"	"fast"	"severe"
3	"fast"	"normal"	"normal"	"mild"
4	"normal"	"normal","cold"	"normal","fast"	"none"

Figure 4: Decision table with changed inputs.

5.1.1. Changes within decision rules

Changing inputs

ΔΠ 1 (Decision input exclusion). *A change* ΔI_{d-} *indicates a change in the input set* I_d *of a decision D as follows: an input variable, or an existing value of a variable, can be deleted from a decision table.*

ε1 (Example). Suppose that in the top-level decision table represented in Figure 3, the input of *muscle activity* is considered irrelevant for the decision on COPD severeness. Hence, this input variable is entirely deleted from the decision table. Figure 4 presents the decision table after change pattern ΔΠ1 was applied. Note that this decision table has been refactored to avoid overlapping and missing rules, since simply deleting an input variable can render the decision table to be incomplete and incorrect [48].

ΔΠ 2 (Decision input inclusion). *A change* ΔI_{d+} *indicates a change in the input set* I_d *of a decision D as follows: opposite to ΔΠ1, an input variable, or a new value for an existing variable, can be added to a decision table.*

ε2 (Example). Adding a new input variable results in the exact opposite changes as in the previous change pattern: if the input variable of *muscle activity* were to be added again to the decision table in Figure 4, that would result in the decision table in Figure 3.

Changing outcomes

ΔΠ 3 (Decision output inclusion). *A change* ΔO_{d+} *indicates a change in the output set* O_d *of a decision* D *as follows: a new output value can be added to a decision table.*

ε3 (Example). Suppose that in the table presented in Figure 3, the COPD severeness decision outcome of the first decision rule changes from *severe* to *lethal*. This new table is given in Figure 5. Notice that the outcome value *lethal* is a new outcome that was not represented in the decision table before.

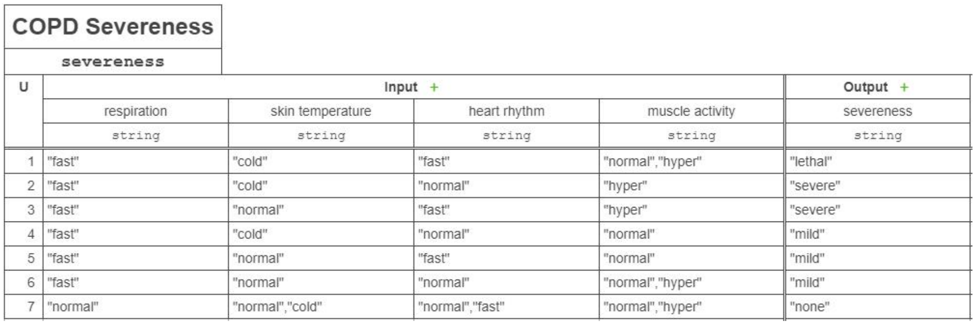
ΔΠ 4 (Decision output exclusion). *A change* ΔO_{d-} *indicates a change in the output set* O_d *of a decision* D *as follows: an existing output value can be deleted from a decision table.*

ε4 (Example). Suppose that in the table presented in Figure 5, the COPD severeness decision outcome of the first decision rule changes from *lethal* to *severe*, i.e., the outcome of *lethal* is deleted from the decision table. The table in Figure 5 is consequently reverted to the situation in Figure 3.

Changing decision logic

ΔΠ 5 (Decision rule logic change). *A change* ΔL *indicates a change in the logic* L *of a decision* D*, i.e., a change in relating the existing input symbols* I_d *to the existing output symbols* O_d *within the decision table.*

ε5 (Example). Suppose that in the table presented in Figure 3, the logic in the fourth decision rule changes. Instead of mapping the input values to an output of *mild*, the inputs are now considered to be of a *severe* nature. This change is exemplified in the decision table in Figure 6.



COPD Severeness

severeness

U	Input +				Output +
	respiration	skin temperature	heart rhythm	muscle activity	severeness
	string	string	string	string	string
1	"fast"	"cold"	"fast"	"normal","hyper"	"lethal"
2	"fast"	"cold"	"normal"	"hyper"	"severe"
3	"fast"	"normal"	"fast"	"hyper"	"severe"
4	"fast"	"cold"	"normal"	"normal"	"mild"
5	"fast"	"normal"	"fast"	"normal"	"mild"
6	"fast"	"normal"	"normal"	"normal","hyper"	"mild"
7	"normal"	"normal","cold"	"normal","fast"	"normal","hyper"	"none"

Figure 5: Decision table with changed outcomes.

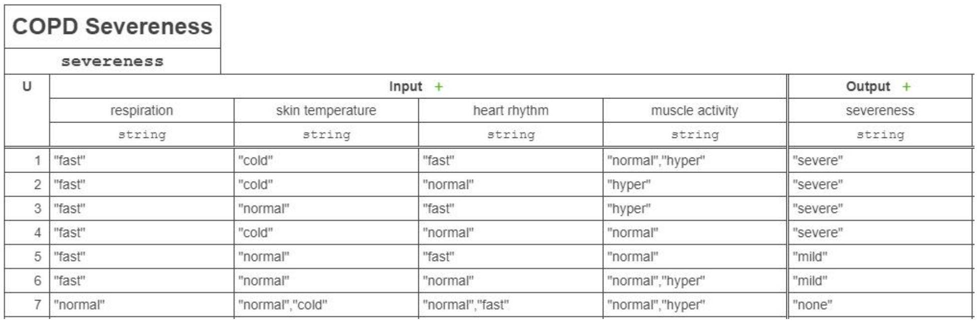
5.1.2. Changes on decision rules in their entirety

ΔΠ 6 (Decision rule exclusion). *If a decision rule $i_d \in I_d \xrightarrow{L} o_d \in O_d$ is deemed irrelevant at a certain point in time, it can be deleted in its entirety from a decision table.*

ε6 (Example). Suppose that decision rule 7 is entirely deleted from the decision table in Figure 3, rendering the decision table depicted in Figure 7. Notice that the decision table does not anymore contain the output *none*, and that by deleting the decision rule, the decision table is not complete anymore, i.e., the combination of input values that was present in rule 7 in Figure 3 can perhaps still manifest itself in real life. However, the decision table is not able to return an outcome for this input combination. This could lead to a deadlock in the system. To avoid this, either the decision table should be completed and the input values at hand should be mapped to other existing decision outcomes, or the system should be redesigned to capture the possibility of no decision outcome being returned.

ΔΠ 7 (Decision rule inclusion). *If a new decision rule $i_d \in I_d \xrightarrow{L} o_d \in O_d$ is deemed relevant at a certain point in time, it can be added in its entirety to an existing decision table.*

ε7 (Example). Suppose that the decision rule deleted in the previous change pattern is reintroduced again to the decision table. Thus, to the decision table in Figure 7, one rule is added, rendering the decision table in Figure 3.

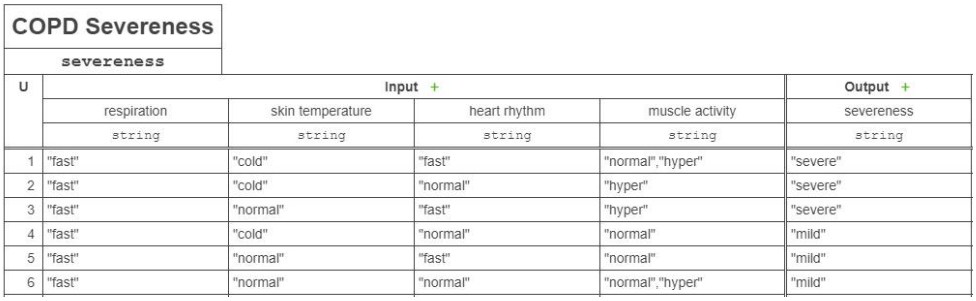


COPD Severeness

severeness

U	Input +				Output +
	respiration	skin temperature	heart rhythm	muscle activity	severeness
	string	string	string	string	string
1	"fast"	"cold"	"fast"	"normal","hyper"	"severe"
2	"fast"	"cold"	"normal"	"hyper"	"severe"
3	"fast"	"normal"	"fast"	"hyper"	"severe"
4	"fast"	"cold"	"normal"	"normal"	"severe"
5	"fast"	"normal"	"fast"	"normal"	"mild"
6	"fast"	"normal"	"normal"	"normal","hyper"	"mild"
7	"normal"	"normal","cold"	"normal","fast"	"normal","hyper"	"none"

Figure 6: Decision table with changed logic.



COPD Severeness

severeness

U	Input +				Output +
	respiration	skin temperature	heart rhythm	muscle activity	severeness
	string	string	string	string	string
1	"fast"	"cold"	"fast"	"normal","hyper"	"severe"
2	"fast"	"cold"	"normal"	"hyper"	"severe"
3	"fast"	"normal"	"fast"	"hyper"	"severe"
4	"fast"	"cold"	"normal"	"normal"	"mild"
5	"fast"	"normal"	"fast"	"normal"	"mild"
6	"fast"	"normal"	"normal"	"normal","hyper"	"mild"

Figure 7: Decision table with a deleted decision rule.

5.2. DRD change patterns

This subsection deals with the deletion or addition of entire elements in the DRD model. Here, we focus on the core elements of a DRD model, i.e., decision nodes and input data nodes. By deleting the input data and decision nodes from the DRD, the information requirement arrows that connect them are deleted as well.

5.2.1. Decision node changes

$\Delta\Pi$ 8 (Decision node exclusion). *A decision node $D \in D_{dm}$ can be deleted from the DRD. This corresponds to deleting all decision rules ($I_d \xrightarrow{L} O_d$) from a decision node D. Hence, this change pattern is an aggregation of multiple $\Delta\Pi 6$ changes. Note that deleting a decision node D also deletes all its incoming and outgoing information requirements arrows from the set IR.*

ε8 (Example). Suppose that the subdecision `Muscle activity` in the DRD in Figure 2 is excluded from the model. The updated decision model is then given in Figure 8.

$\Delta\Pi$ 9 (Decision node inclusion). *A new decision node D can be added to the set of decision nodes D_{dm}. This corresponds to adding a new decision table, and thus, adding multiple decision rules ($I_d \xrightarrow{L} O_d$) encapsulated in the decision node D. Hence, this change pattern is in essence an aggregation of multiple $\Delta\Pi 7$ changes. Note that adding a decision node D also adds the necessary incoming and outgoing information requirements arrows to the set IR.*

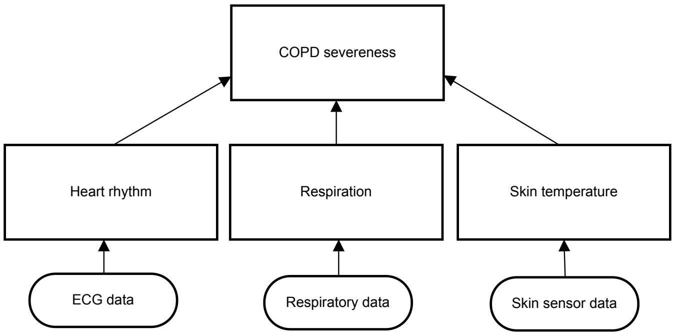


Figure 8: DRD adapted to changed inputs in the top-level decision.

ε9 (Example). Suppose that a decision node `Muscle activity` is added to the decision requirements diagram in Figure 8, effectively producing the DRD presented in Figure 2, as the *EMG data* input is added to the DRD as well.

5.2.2. Input data node changes

ΔΠ 10 (Input data node inclusion). *A new data input node can be added to the set of data input nodes* ID*. By adding a data input node, its necessary input requirement arrows are also added to the set of* IR *and connected to the relevant decision nodes in* D_{dm}*. Notice that this change pattern on the DRD level corresponds to adding a new input variable to the decision table that requires the newly added data input node. Hence, this change pattern is equivalent to* ΔΠ*2.*

ε10 (Example). Consider the decision model provided in Figure 8. Assume that an additional

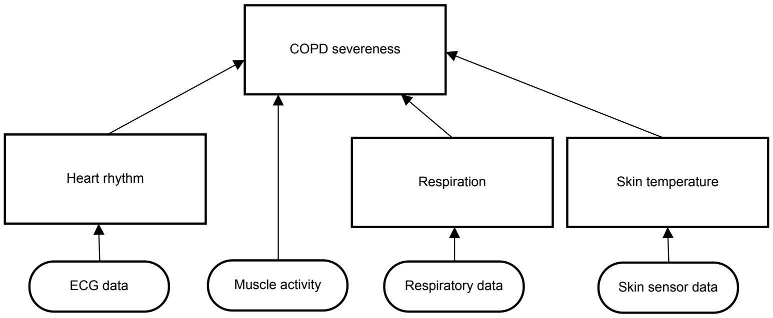


Figure 9: A DRD with an added input data node.

input data node is added that provides information on muscle activity. The new decision model is then given in Figure 9.

ΔΠ 11 (Input data node exclusion). *A data input node can be deleted from the set of data input nodes ID. By deleting a data input node, all its input requirement arrows are also deleted from to the set of IR. Notice that this change pattern on the DRD level corresponds to deleting an input variable from the decision table that required the input data node. Hence, this change pattern is again equivalent to* $\Delta\Pi$*1.*

ε11 (Example). Consider the decision model provided in Figure 9. Assume that the `Muscle activity` input data node is deleted from the requirements diagram since this information is now deemed invalid to make the top-level `COPD severeness` decision. The newly obtained DRD is then given in Figure 8.

5.3. Change patterns on non-core DMN elements

So far, we have enumerated change patterns that have a considerable impact on the core DMN elements according to the metamodel presented in the standard specification [1] (see Section 9). These are: the input data nodes, the decision nodes, and the underlying decision tables and their components. However, changes related to non-core elements of the model are possible as well. We give an overview of those changes. First, we enumerate the changes that are in essence the same as the core changes presented in this section, since the elements that undergo the changes are either composed out of the elements that have already been discussed, or they connect the previously discussed core elements.

- Business knowledge models (BKM) and associated knowledge requirements can be introduced in a DRD and, therefore, they can undergo changes as well. The encapsulated decision logic of business knowledge models is often represented in the form of decision tables as well (see the metamodel in Figure 21). Hence, the change patterns the BKMs can go through are analogous to those of the decision nodes and their related decision tables.

Table 1: Overview of decision model change patterns.

Decision table change patterns	
Changes within decision rules	
$\Delta\Pi 1$	Decision input exclusion.
$\Delta\Pi 2$	Decision input inclusion.
$\Delta\Pi 3$	Decision output inclusion.
$\Delta\Pi 4$	Decision output exclusion.
$\Delta\Pi 5$	Decision rule logic change.
Changes on decision rules	
$\Delta\Pi 6$	Decision rule exclusion.
$\Delta\Pi 7$	Decision rule inclusion.
Decision requirements diagram change patterns	
Decision node changes	
$\Delta\Pi 8$	Decision node exclusion.
$\Delta\Pi 9$	Decision node inclusion.
Input data node changes	
$\Delta\Pi 10$	Input data node inclusion.
$\Delta\Pi 11$	Input data node exclusion.

- Decision services are built up of core elements of the DRD, as shown in the metamodel in Figure 22. Hence, changing the decision services corresponds to changing the individual core elements within the DRD that are encapsulated in the decision services. As such, these changes have been addressed above. Note that according to the latest DMN standard specification [1], there is an element which can be used in the DRD to represent a decision service. This decision service element encapsulates a subset of the core elements of a DRD, much like a subprocess task in BPMN encapsulates a subset of logically related process elements of a master process.
- The requirement connections between DRD elements can change, i.e., the requirement arrows can be added or deleted. This has possible impact on both the begin and end nodes associated to the requirement arrows, as shown in the decision metamodel in Figure 19. If the requirement arrow is deleted and the begin node has no other requirements associated to it, it needs to be deleted as well. If a new requirement arrow is added, the relevant information passed through that arrow should be incorporated in the designated end node.

Next to these non-core DMN elements that are closely related to the core DMN elements, other DMN components exist as well, as specified in the metamodels in Section 9, and, therefore, they too can be subject to change.

- The data types of the input and/or outputs can be changed (see Figure 18).
- The names of all the elements within the decision model can be changed (see Figure 17).
- Knowledge sources and associated authority requirements can be added to the DRD, as shown in the metamodels in Figures 19 and 21, and they can undergo change as well. However, the knowledge source and authority requirements in DMN are mainly used for documentation purposes and do not affect the rest of the model.

- The hit policies of decision tables can be altered as well, as shown in the decision table metamodel in Figure 18. This usually leads to refactoring and remodelling the decision table [48], however, it has no further influence on the decision model elements from the input side or the output side.

- Finally, text annotations and associations can be added, deleted, or changed to provide additional textual information on elements within the DRD, as depicted in the metamodel in Figure 23. These are meant for documentation and clarification purposes and do not affect the decision model in any way if they are subject to change.

5.4. Refactoring and simplification of decision tables

Modelling logic with decision tables has been discussed in previous works, in particular when it comes to inconsistencies in the decision logic itself. More specifically, the works have focused on finding overlapping and missing rules in decision tables, and on refactoring the decision tables to remedy these issues [49, 50]. Furthermore, decision table simplification was researched as well [48, 51]. Here, the goal is to represent the decision logic in a decision table in an unambiguous and understandable way by for instance contracting the inputs of decision rules which lead to the same decision outcome. Another form of refactoring is induced when the hit policy of the decision table is changed, as the decision rules will often need to undergo reordering to fit the newly assigned hit policy. However, researchers have argued to stick with unique hit decision tables and to avoid hit policies such as first hit and priority hit in order to avoid ambiguity and overlapping rules as a design principle [14, 52]. As such, the notions of decision table refactoring and simplification are well researched and they have led to tools that support the automatic refactoring and simplification of decision tables.

6. Change Propagation

Applying a single change pattern to a decision model can have repercussions across multiple elements of a DRD. More precisely, an applied change pattern can render the decision model inconsistent. To restore consistency other change patterns may need to be triggered. In what follows, we investigate how each change pattern should trigger a chain of change patterns within the decision model. For each change pattern, we state which inconsistencies the change may induce into the decision model. Next, we indicate which change patterns need to be triggered and propagated to resolve the inconsistencies. We provide an overview of change pattern propagation in Table 2.

6.1. $\Delta\Pi 1$*-induced change pattern propagation*

6.1.1. Induced inconsistencies

Deleting an input from a decision rule ($\Delta\Pi 1$) corresponds to deleting the input requirements of that decision table. The input of a decision consists of its input data elements and/or the outcomes of its subdecisions. Hence, deleting the input of the decision table without deleting its predecessors that supply the inputs, renders the decision model inconsistent.

6.1.2. Resolving inconsistencies

Given the hierarchical topology of a DRD, $\Delta\Pi 1$ implies changing the elements that are required by the decision whose inputs undergo changes:

- If the deleted input is obtained from a data input node, this manifests itself by deleting input data nodes ($\Delta\Pi 11$).
- If the deleted input was obtained from another decision node, i.e., a subdecision, the propagated change patterns to the subdecisions can be manifold.

- An existing decision outcome in the subdecision table can be deleted ($\Delta\Pi4$). Note that this only applies if the subdecision outcome is not used as an input in another decision table. Otherwise, other decision tables that require the subdecision outcome are rendered inconsistent.
- Another option is to exclude a whole decision rule in the subdecision whose decision outcome feeds into the decision table with the changed input ($\Delta\Pi6$).
- If an entire input variable is deleted, rather than just an existing value of an existing input variable, the entire subdecision node that feeds its outcome into the higher-level decision where the input is deleted in the first place ($\Delta\Pi8$).

Thus, applying $\Delta\Pi1$ can on its turn trigger change patterns $\Delta\Pi4$, $\Delta\Pi6$, $\Delta\Pi8$, and $\Delta\Pi11$. Take for instance the deletion of the variable *muscle activity* from the inputs in the decision table in Figure 3, rendering decision table 4. This also affects the decision hierarchy in the DRD in Figure 2. The subdecision `Muscle activity`, together with its input data, i.e., *EMG data*, are not part of the top-level decision anymore. As such, they should be discarded from the decision requirements diagram. In this case, $\Delta\Pi1$ also triggers change patterns $\Delta\Pi8$ and $\Delta\Pi11$. The updated decision requirements model is provided in Figure 8.

6.2. $\Delta\Pi2$*-induced change pattern propagation*

6.2.1. Induced inconsistencies

Adding an input to a decision rule ($\Delta\Pi2$) corresponds to changing the input requirements of that decision table. The input of a decision consists of its input data elements and/or the outcomes of its subdecisions. Hence, adding an input to the decision table without changing its predecessors that supply the inputs, renders the decision model inconsistent.

6.2.2. Resolving inconsistencies

Given the hierarchical topology of a DRD, $\Delta\Pi 2$ implies changing the elements that are required by the decision whose inputs undergo changes:

- If the added input is obtained from a data input node, this manifests itself by adding input data nodes ($\Delta\Pi 10$).
- If the added input is obtained from another decision node, i.e., a subdecision, the propagated change patterns to the subdecisions can be manifold.
  - A new decision outcome in the subdecision table can be added ($\Delta\Pi 3$).
  - Another option is to include a whole new decision rule in the subdecision whose decision outcome feeds into the decision table with the changed input ($\Delta\Pi 7$).
  - If a new input variable is introduced, rather than just a new value for existing input variables, a whole new decision node that feeds its outcome into the higher-level decision where the input is added can be included($\Delta\Pi 9$).

Thus, applying $\Delta\Pi 2$ can on its turn trigger change patterns $\Delta\Pi 3$, $\Delta\Pi 7$, $\Delta\Pi 9$, and $\Delta\Pi 10$. Take for instance decision table 4 and suppose that the *muscle activity* variable is added again. The DRD should reintroduce the subdecision `Muscle activity` and its required *EMG data*. Thus, this change pattern version would return the models to the initial state as shown in Figures 2 and 3, by initiating change patterns $\Delta\Pi 9$ and $\Delta\Pi 10$ after $\Delta\Pi 2$.

6.3. $\Delta\Pi 3$*-induced change pattern propagation*

6.3.1. Induced inconsistencies

Given the hierarchical topology of a DRD, adding a decision outcome to a decision rule ($\Delta\Pi 2$) also affects the input requirements, i.e. the inputs, of the higher level decision table that requires the added decision outcome, assuming that the added decision outcome does not belong to the top-level decision node of the DRD. Simply changing the outcome of a decision without adapting the higher-level decision tables that require that outcome, leads to an inconsistent decision model.

6.3.2. Resolving inconsistencies

Applying $\Delta\Pi3$ can trigger change pattern $\Delta\Pi2$ in all higher-level decision tables that require the decisions affected by $\Delta\Pi3$. Note that newly added outcome values can be captured in higher-level decision tables by introducing new decision rules ($\Delta\Pi7$), or even new decision nodes ($\Delta\Pi9$) if an entirely new decision outcome variable is introduced. Take for instance the decision table represented in Figure 3, the COPD severeness decision outcome of the first decision rule changes from *severe* to *lethal*. This new table is given in Figure 5. Notice that in this case the DRD in Figure 2 does not undergo any change, since the changed decision outcome belongs to the top-level decision of the DRD. Hence, the outcome does not affect any decision constructs that are higher in the decision hierarchy.

6.4. $\Delta\Pi4$*-induced change pattern propagation*

6.4.1. Induced inconsistencies

Given the hierarchical topology of a DRD, deleting a decision outcome to a decision rule ($\Delta\Pi4$) also affects the input requirements, i.e. the inputs, of the higher level decision table that requires the deleted decision outcome, assuming that the deleted decision outcome does not belong to the top-level decision node of the DRD. Here too, simply changing the outcome of a decision without adapting the higher-level decision tables that require that outcome, leads to an inconsistent decision model.

6.4.2. Resolving inconsistencies

Applying $\Delta\Pi4$ should trigger change pattern $\Delta\Pi1$ in all higher-level decision tables that require the decisions affected by $\Delta\Pi4$. Similar to the previous change pattern, deleted outcome values can be resolved by deleting decision rules that require the outcome values in higher-level decision tables (($\Delta\Pi6$), or deleting entire higher-level decision nodes if the outcome variable is deleted in its entirety ($\Delta\Pi8$) It is worthwhile to notice the exact opposite changes in the model occur as in the previous change pattern. If in the decision table of Figure 5, the *lethal* outcome were again to be replaced by the *severe* outcome, then the decision table would undergo redesign to revert back to the model presented in the running example, i.e., Figure 3.

6.5. $\Delta\Pi5$*-induced change pattern propagation*

Since $\Delta\Pi5$ is only concerned with the logic within one decision node of the decision model, no additional change patterns are propagated as a result of applying $\Delta\Pi5$. This is due to the fact that the input and outcome sets of the decision table do not change, but merely the logic that maps the inputs onto the outcomes. The inputs and outcomes are linking elements for the information requirements that construct the decision hierarchy. Since this change pattern does not alter the inputs and outcomes, no change propagation to other elements in the decision model can occur.

6.6. $\Delta\Pi6$*-induced change pattern propagation*

6.6.1. Induced inconsistencies

Excluding a decision rule ($\Delta\Pi6$) is in essence a combination of deleting inputs ($\Delta\Pi1$), deleting decision outcomes ($\Delta\Pi4$), and the logic that connects them ($\Delta\Pi5$). Hence, this change pattern can possibly induce the same model inconsistencies as patterns $\Delta\Pi1$ and $\Delta\Pi4$.

6.6.2. Resolving inconsistencies

Thus, applying $\Delta\Pi6$ can on its turn trigger change patterns $\Delta\Pi1$, $\Delta\Pi4$, $\Delta\Pi5$ and $\Delta\Pi6$ again. Notice that if no input variables or rule outcomes are deleted when $\Delta\Pi6$ occurs, then this pattern corresponds to $\Delta\Pi3$, i.e., a mere change in underlying logic, without any further change propagation.

When applying $\Delta\Pi6$ to decision rule 7 in the decision table in Figure 3, obtaining the decision table depicted in Figure 7, a unique decision output value, i.e., *none*, is deleted. However, this does not affect the rest of the decision model, since the affected decision table corresponds to the top-level decision. As such, these change patterns cannot propagate any patterns affecting the higher levels in the decision hierarchy. Furthermore, the lower-level elements in the decision hierarchy are not affected either, since the deleted input values are not unique, and no input variable was deleted in its entirety. Thus, in this case, $\Delta\Pi6$ only triggers a change in underlying logic ($\Delta\Pi5$) without affecting the hierarchy of the DRD. However, the decision table in Figure 7 is not longer complete and should undergo redesign according to [48] to capture all possible input combinations.

6.7. $\Delta\Pi7$*-induced change pattern propagation*

6.7.1. Induced inconsistencies

Including a decision rule ($\Delta\Pi7$) is, in analogy with $\Delta\Pi6$ propagation, a combination of $\Delta\Pi2$, $\Delta\Pi3$, $\Delta\Pi5$ in the case of deleting existing values of existing input variables or outcomes. Hence, this change pattern can possibly trigger the same inconsistencies as patterns $\Delta\Pi2$ and $\Delta\Pi3$ when inputs or outcomes are added to a decision rule, assuming new values of existing variables are introduced.

6.7.2. Resolving inconsistencies

Thus, applying $\Delta\Pi7$ can on its turn trigger change patterns $\Delta\Pi2$, $\Delta\Pi3$, $\Delta\Pi5$ and $\Delta\Pi7$ again. Notice that if no unique values of input variables and rule outcomes are introduced when $\Delta\Pi7$ occurs, then this pattern corresponds to $\Delta\Pi5$, i.e., a mere change in underlying logic, without any further change propagation.

When applying $\Delta\Pi7$ by decision rule 7 in the decision table in Figure 7, obtaining the decision table depicted in Figure 3, a unique decision output, i.e., *none*, is added. As such the table is complete. However, in analogy with $\Delta\Pi6$, no other elements in the decision hierarchy undergo any changes in this example.

6.8. $\Delta\Pi8$*-induced change pattern propagation*

6.8.1. Induced inconsistencies

Deleting an entire decision node ($\Delta\Pi8$) corresponds to applying $\Delta\Pi6$ on every decision rule contained within the decision table of that decision node, i.e. deleting the decision table related to that node. As such, the same chain of inconsistencies is established. All elements that are lower in the decision hierarchy and that are only connected to the deleted decision node, become obsolete as well. The elements that are higher in the decision hierarchy and that require the outcome of the deleted decision node, render the model inconsistent, since the required inputs can no longer be obtained due to the deletion of the lower-level decision node.

6.8.2. Resolving inconsistencies

All input nodes, i.e., input data nodes and subdecisions, that feed into the deleted decision node and are only connected to that decision node, need to be deleted, effectively triggering $\Delta\Pi8$ and $\Delta\Pi11$. This applies to decision model elements that are lower in the decision hierarchy than the deleted decision node. For elements that are higher in the decision hierarchy, change pattern

$\Delta\Pi 1$ should be triggered as well, i.e., whole input variables are deleted, since the required decision node that provides those variables is deleted as well. As such, the underlying decision logic of the decision model can be subject to changes as well $\Delta\Pi 5$.

Suppose that the subdecision `Muscle activity` in the DRD in Figure 2 is excluded from the model. This also deletes the subdecision's input data, i.e., *EMG data* ($\Delta\Pi 11$). The updated decision model is then given in Figure 8. Since the deleted decision node is connected to a higher-level decision node, i.e., `COPD severeness`, which requires the deleted decision node as input, inconsistencies arise in the decision model. Namely, the decision table in Figure 3 has a *muscle activity* input that can no longer be obtained due to the deletion of the `Muscle activity` decision node. As such, $\Delta\Pi 8$ also triggers $\Delta\Pi 1$ in this example, effectively transforming the decision table in Figure 3 into the decision table given in Figure 4 by deleting the input variable *muscle activity*. As such, the decision model is rendered consistent again.

6.9. $\Delta\Pi 9$-induced change pattern propagation

6.9.1. Induced inconsistencies

Inserting an entire decision node ($\Delta\Pi 9$) corresponds to adding an entirely new decision table, i.e., applying $\Delta\Pi 7$ on every decision rule contained within the decision table of that decision node. Hence, the same inconsistencies arise as in $\Delta\Pi 7$.

6.9.2. Resolving inconsistencies

For all decision elements that require the inserted decision node, a new variable will need to be introduced in the inputs ($\Delta\Pi 2$). For the inserted decision node itself, the relevant input requirements need to be provided, either by linking existing decision nodes and input data nodes to the inserted decision node, or by introducing new decision nodes and input data nodes ($\Delta\Pi 9$ and $\Delta\Pi 10$). This also affects the decision logic ($\Delta\Pi 5$).

Suppose that the subdecision `Muscle activity` is added to the DRD in Figure 8, obtaining the DRD in Figure 2. This also adds the subdecision's input data, i.e., *EMG data* ($\Delta\Pi 10$) to the DRD model in Figure 2. Furthermore, the decision table in Figure 4 corresponding to the `COPD severeness` decision needs to be altered as well to incorporate the addition of the new decision node. As such the *muscle activity* variable is added to reflect the information requirement between the `Muscle activity` subdecision and the `COPD severeness` decision ($\Delta\Pi 2$). This way, the decision table in Figure 3 is obtained. As such, the consistency between the different decision model elements is restored.

6.10. $\Delta\Pi 10$*-induced change pattern propagation*

6.10.1. Induced inconsistencies

Including a new input data node ($\Delta\Pi 10$) in a DRD renders the decision model inconsistent if the decision table to which the input data node is connected, is not updated as well to consider that input node.

6.10.2. Resolving inconsistencies

Including a new input data node ($\Delta\Pi 10$) also corresponds to changing the inputs of all decision rules in all decision nodes that require the added input data node ($\Delta\Pi 2$). This also changes the decision logic ($\Delta\Pi 5$). Hence, all decision nodes that require the input data node need to introduce the variable concerning the newly added input data node to their inputs.

Consider the decision model provided in Figure 8 with its corresponding top-level decision table in Figure 4. Assume that an additional input data node is added that provides information on muscle activity. The new DRD model is then given in Figure 9. However the newly introduced *Muscle activity* node induces a change in the decision table in Figure 4 as well ($\Delta\Pi 2$). More precisely, the inputs of that decision table need to change to incorporate the *muscle activity* variable, rendering the decision table in Figure 3. As such, the consistency between the decision model elements is restored.

6.11. $\Delta\Pi11$-*induced change pattern propagation*

6.11.1. Induced inconsistencies

Excluding an input data node ($\Delta\Pi11$) from a DRD renders the decision model inconsistent if the decision table to which the input data node was connected, is not updated as well.

6.11.2. Resolving inconsistencies

Excluding an input data node ($\Delta\Pi11$) corresponds to changing the inputs of all decision rules in all decision nodes that require the deleted input data node ($\Delta\Pi1$). This also changes the decision logic ($\Delta\Pi5$). Hence, the variable derived from the deleted input data node is deleted from all the decision nodes that required the input data node. Since the input data nodes are by definition (see Definition 1) the lowest elements in the decision hierarchy, they can only propagate their changes to higher-level decision nodes. This holds both for change propagation induced by $\Delta\Pi11$, as well as $\Delta\Pi10$.

Consider the decision model provided in Figure 9 with its corresponding top-level decision table in Figure 3. Assume that the `Muscle activity` input data node is deleted. The new DRD model is then given in Figure 8. However, the deleted *Muscle activity* node should induce a change in the decision table in Figure 3 as well ($\Delta\Pi1$). More precisely, the inputs of that decision table need to change to incorporate the deletion of the *muscle activity* variable, rendering the decision table in Figure 4. As such, the consistency between the decision model elements is restored.

6.12. Overview of induced change pattern propagation

Notice that all the applied change patterns can also induce a change in the decision logic ($\Delta\Pi5$), i.e., the mapping between input and output elements. When a change pattern is applied to the

Table 2: Change propagation: an overview of decision model change patterns that may need to be triggered in other elements of the decision model to restore consistency.

Applied $\Delta\Pi$	**Propagated $\Delta\Pi$ possibly triggered by initial $\Delta\Pi$**
$\Delta\Pi\mathit{1}$	$\Delta\Pi\mathit{4}$, $\Delta\Pi\mathit{5}$, $\Delta\Pi\mathit{6}$, $\Delta\Pi\mathit{8}$, $\Delta\Pi\mathit{11}$
$\Delta\Pi\mathit{2}$	$\Delta\Pi\mathit{3}$, $\Delta\Pi\mathit{5}$, $\Delta\Pi\mathit{7}$, $\Delta\Pi\mathit{9}$, $\Delta\Pi\mathit{10}$
$\Delta\Pi\mathit{3}$	$\Delta\Pi\mathit{2}$, $\Delta\Pi\mathit{5}$, $\Delta\Pi\mathit{7}$, $\Delta\Pi\mathit{9}$
$\Delta\Pi\mathit{4}$	$\Delta\Pi\mathit{1}$, $\Delta\Pi\mathit{5}$, $\Delta\Pi\mathit{6}$, $\Delta\Pi\mathit{8}$
$\Delta\Pi\mathit{5}$	
$\Delta\Pi\mathit{6}$	$\Delta\Pi\mathit{1}$, $\Delta\Pi\mathit{4}$, $\Delta\Pi\mathit{5}$, $\Delta\Pi\mathit{6}$
$\Delta\Pi\mathit{7}$	$\Delta\Pi\mathit{2}$, $\Delta\Pi\mathit{3}$, $\Delta\Pi\mathit{5}$, $\Delta\Pi\mathit{7}$
$\Delta\Pi\mathit{8}$	$\Delta\Pi\mathit{1}$, $\Delta\Pi\mathit{5}$, $\Delta\Pi\mathit{8}$, $\Delta\Pi\mathit{11}$
$\Delta\Pi\mathit{9}$	$\Delta\Pi\mathit{2}$, $\Delta\Pi\mathit{5}$, $\Delta\Pi\mathit{9}$, $\Delta\Pi\mathit{10}$
$\Delta\Pi\mathit{10}$	$\Delta\Pi\mathit{2}$, $\Delta\Pi\mathit{5}$
$\Delta\Pi\mathit{11}$	$\Delta\Pi\mathit{1}$, $\Delta\Pi\mathit{5}$

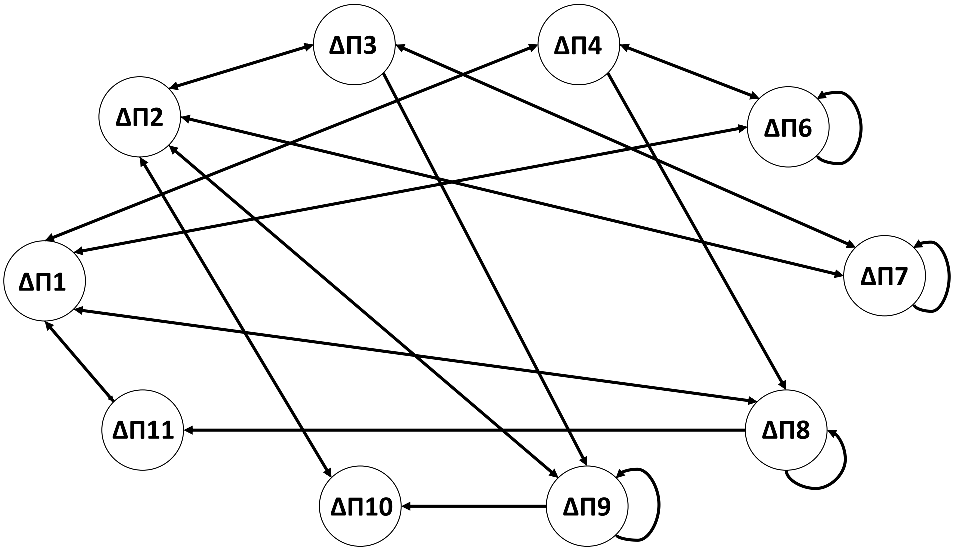
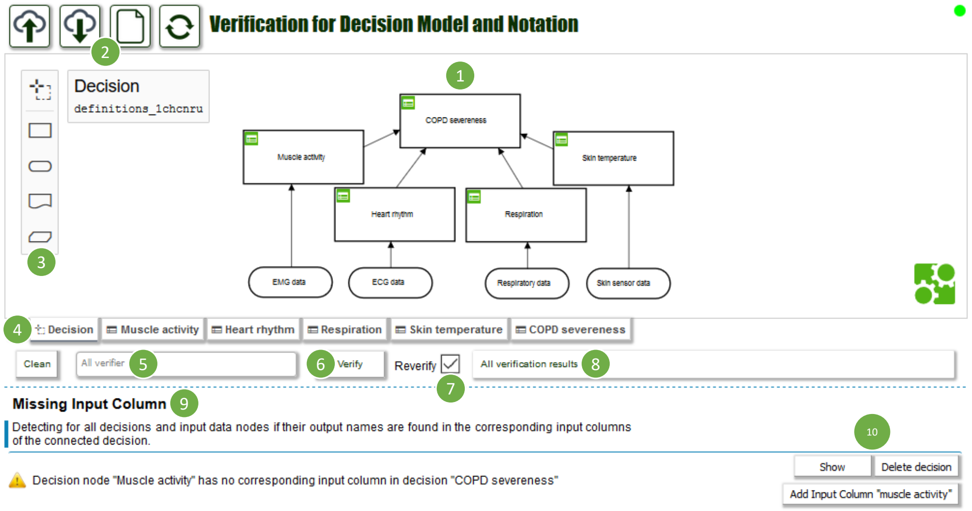


Figure 10: Overview of change propagation: change patterns that can possibly follow each other. Note that ΔΠ5 should be considered as a consequence of applying any change pattern, but it is not included in the graph in order to provide a more comprehensible overview.

decision model, it can trigger another change pattern. That triggered change pattern can in turn trigger yet another change pattern. Thus, a chain of triggered patterns may need to be propagated through the decision model until the system stabilises. Table 2 gives an overview of how this change pattern propagation chain can come into existence. Figure 10 provides a depiction of the change propagation graph. For instance, when pattern $\Delta\Pi1$ is applied, $\Delta\Pi8$ may need to be triggered. Change pattern $\Delta\Pi8$ can in turn trigger $\Delta\Pi11$ after which the propagation of changes stop and the system returns to a stable state. Note that, as indicated in Table 2, the change patterns can also induce a change in decision logic ($\Delta\Pi5$). However, we have excluded $\Delta\Pi5$ from the change propagation graph in order to render the graph more comprehensible.

7. A Modelling Environment Prototype for the Evolution of DMN Models

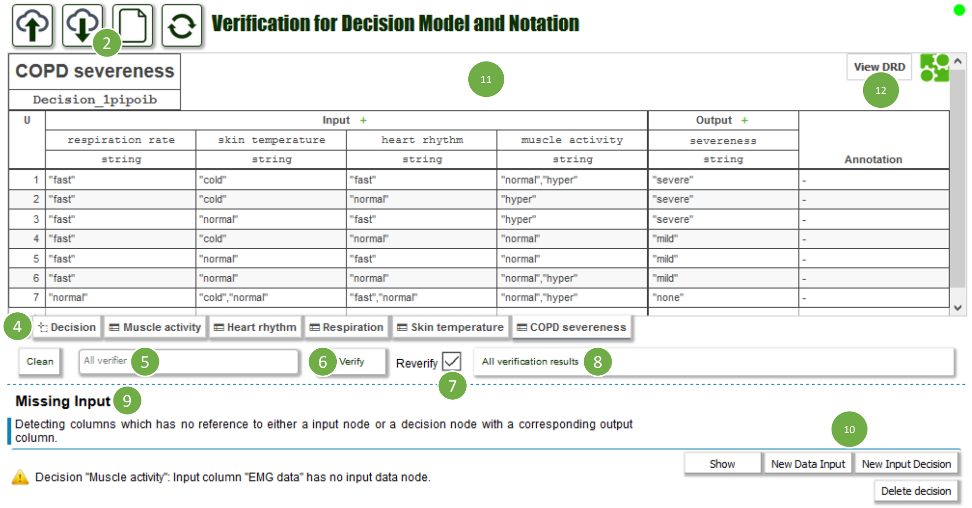
This section presents a modelling environment prototype which provides modelling support for the evolution of DMN decision models. The modelling environment is based on the open source Camunda[1] modeller which we have advanced with verification capabilities. The modeller can apply any of the change patterns discussed in this paper. After doing so, the modelling environment checks for consistency errors and displays error messages. The modelling environment can highlight the errors and suggests actions to remedy the inconsistencies. After the modeller selects an action, the modelling environment automatically performs it and checks for errors again. This way, the DMN can be evolved iteratively in a consistent way. In what follows, we explain the interface of the modelling environment and we show its verification capabilities with an example.



Detecting for all decisions and input data nodes if their output names are found in the corresponding input columns of the connected decision.

⚠ Decision node "Muscle activity" has no corresponding input column in decision "COPD severeness"

Figure 11: The DRD view of the DMN verification tool.



U	Input +				Output +	
	respiration rate	skin temperature	heart rhythm	muscle activity	severeness	
	string	string	string	string	string	Annotation
1	"fast"	"cold"	"fast"	"normal","hyper"	"severe"	-
2	"fast"	"cold"	"normal"	"hyper"	"severe"	-
3	"fast"	"normal"	"fast"	"hyper"	"severe"	-
4	"fast"	"cold"	"normal"	"normal"	"mild"	-
5	"fast"	"normal"	"fast"	"normal"	"mild"	-
6	"fast"	"normal"	"normal"	"normal","hyper"	"mild"	-
7	"normal"	"cold","normal"	"fast","normal"	"normal","hyper"	"none"	-

Detecting columns which has no reference to either a input node or a decision node with a corresponding output column.

⚠ Decision "Muscle activity": Input column "EMG data" has no input data node.

Figure 12: The decision table view of the DMN verification tool.

7.1. The modelling environment

Figures 11 and 12 show the interface of the environment, with numbers indicating the different parts of the tool:

1. The DRD model is displayed here.
2. Buttons to upload a DMN model into the environment, to download the model from the environment, and to generate an empty DMN model are provided.
3. DMN elements can be selected by the modeller to add to the DMN model.
4. The modeller can switch views, i.e., by selecting a decision, the tool switches from the DRD view in Figure 11 to the decision table view in Figure 12.
5. The modeller can select the verifiers which are relevant, i.e., the errors that the modelling environment should check for. Alternatively, instead of selecting a single verifier, all verifiers can be selected as to display all errors that the modelling environment discovers.
6. By clicking on the *Verify* button, the modelling environment checks for errors of in the current DMN model.
7. By checking the *Reverify* box the modelling tool will perform a verification of the model every time the modeller performs one of the change patterns on the model. Unchecking this box deactivates this feature, as an experienced modeller will not need to check the consistency after every action performed. However, after applying a whole chain of changes, the modeller can again re-enable the verification feature of the tool.
8. After the verification happens, the modeller can select which error, i.e., which verification result to display. Alternatively, the modeller can op to display all verification results.

---

[1] https://camunda.com/download/modeler/

9. Here, the error messages as a result of the verification mechanism of the modelling environment are displayed. The error messages are explained as well, both in general terms, as well as in specific terms, i.e., indicating the names of the DMN elements that are affected.

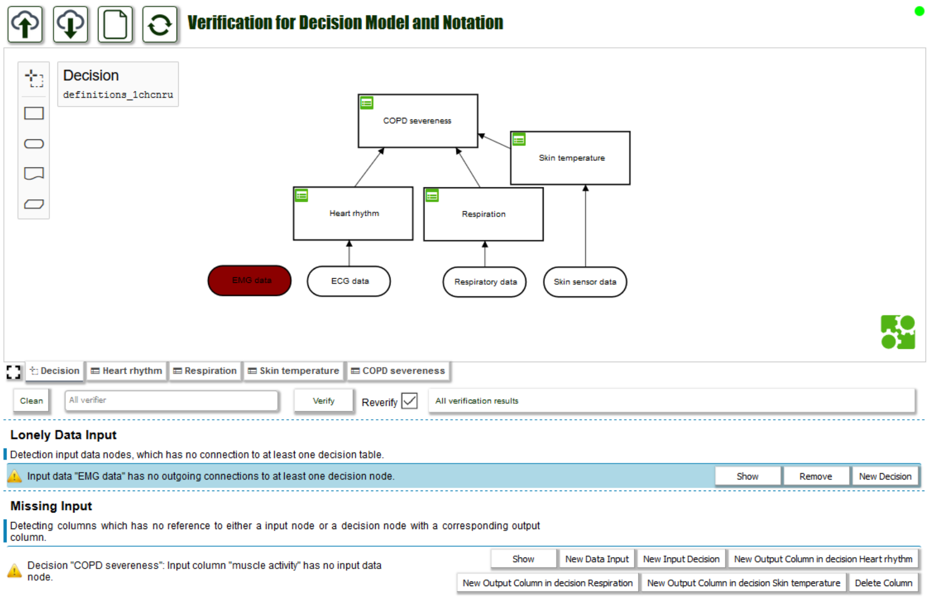
10. Next to the specific error messages, actions are suggested by the modelling environment that can resolve the errors. By clicking the *Show* button, the modelling environment highlights the errors in the DMN model in a red colour. By clicking any of the other suggested action buttons, the modelling environment automatically performs the selected action.

11. In the decision table view, instead of the DRD model, a decision table that has been selected is displayed in the central view of the tool, as shown in Figure 12.

12. By clicking the *View DRD* button, the modeller can switch back from the decision table view to the DRD view in the modelling environment.

7.2. An example of change propagation and modelling support

In this subsection we provide a short example of the workings of the DMN verification tool. The DMN model presented in Figures 2 and 3 is uploaded into the modelling environment. Suppose that instead of assessing muscle activity with data obtained from the EMG sensor, the physician wishes to investigate the patient and manually enter the muscle activity classification. Hence, the *Muscle Activity* subdecision from Figure 2 is obsolete and the modeller decides to delete it (i.e., the modeller applies decision node exclusion). Figure 13 shows the situation after applying this change pattern. Two errors are raised by the modelling environment: an input data node which is not connected to any decision node and an input column in the top-level *COPD Severeness* decision which does not have any input data node or subdecision node providing the required input. Figure 13 highlights the lonely input data node after clicking on the *Show* button.



Lonely Data Input

Detection input data nodes, which has no connection to at least one decision table.

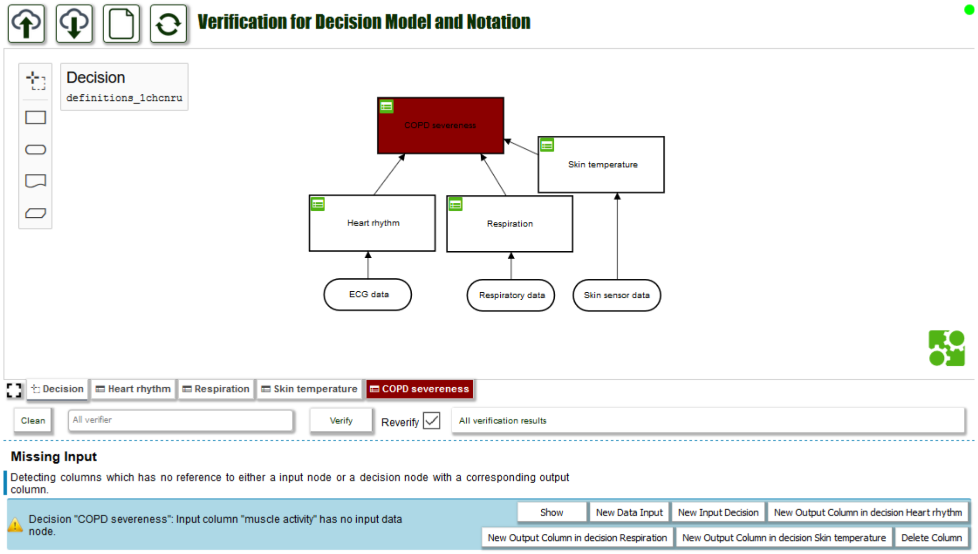
Input data "EMG data" has no outgoing connections to at least one decision node.

Missing Input

Detecting columns which has no reference to either a input node or a decision node with a corresponding output column.

Decision "COPD severeness": Input column "muscle activity" has no input data node.

Figure 13: Situation after applying decision node exclusion.

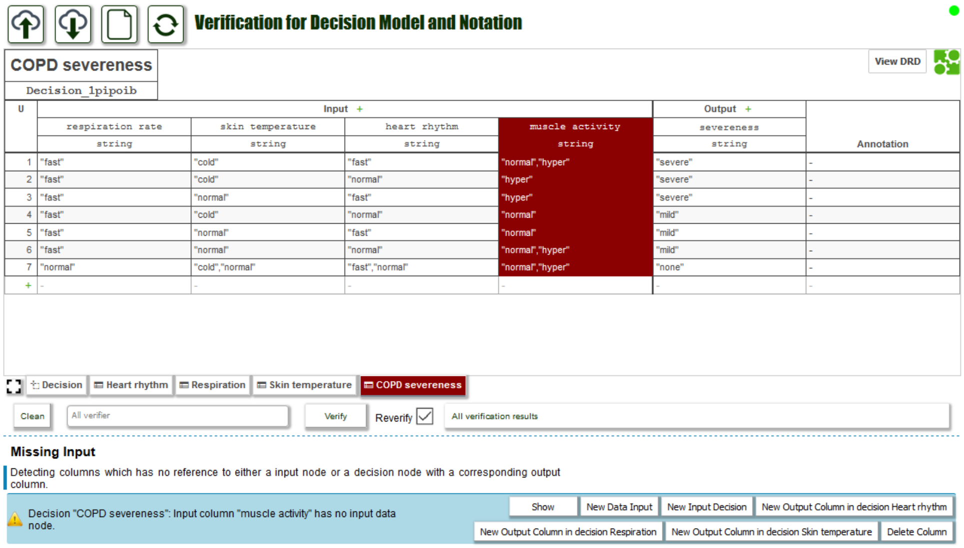


Missing Input

Detecting columns which has no reference to either a input node or a decision node with a corresponding output column.

Decision "COPD severeness": Input column "muscle activity" has no input data node.

Figure 14: Situation after applying input data node exclusion (DRD view).



COPD severeness

Decision_1pipoib

U	Input +				Output +	
	respiration rate	skin temperature	heart rhythm	muscle activity	severeness	
	string	string	string	string	string	Annotation
1	"fast"	"cold"	"fast"	"normal","hyper"	"severe"	-
2	"fast"	"cold"	"normal"	"hyper"	"severe"	-
3	"fast"	"normal"	"fast"	"hyper"	"severe"	-
4	"fast"	"cold"	"normal"	"normal"	"mild"	-
5	"fast"	"normal"	"fast"	"normal"	"mild"	-
6	"fast"	"normal"	"normal"	"normal","hyper"	"mild"	-
7	"normal"	"cold","normal"	"fast","normal"	"normal","hyper"	"none"	-
+	-	-	-	-	-	-

Missing Input

Detecting columns which has no reference to either a input node or a decision node with a corresponding output column.

⚠ Decision "COPD severeness": Input column "muscle activity" has no input data node.

Figure 15: Situation after applying input data node exclusion (decision table view).

By clicking the *Remove* button next to the input data node error, the change pattern of input data node exclusion is automatically performed by the modelling environment, leaving only the missing input error relating to the top-level *COPD Severeness* decision. Figure 14 highlights this error in the DRD view of the modelling environment, while Figure 15 highlights the error in the decision table view.

As shown in Figures 14 and 15, the modelling environment suggests a number of actions to remedy the issues surrounding the missing input in the top-level decision. Inputs can be secured in three different ways: adding a new data input (i.e., input data inclusion), adding a new input decision (i.e., decision node inclusion), or adding outputs to existing subdecision nodes (i.e. decision output inclusion). Alternatively, the situation can be remedied by deleting the input column which does not have the required inputs (i.e., decision input exclusion). After deciding that a physician will manually provide the information needed as input in the *COPD Severeness* decision table, the

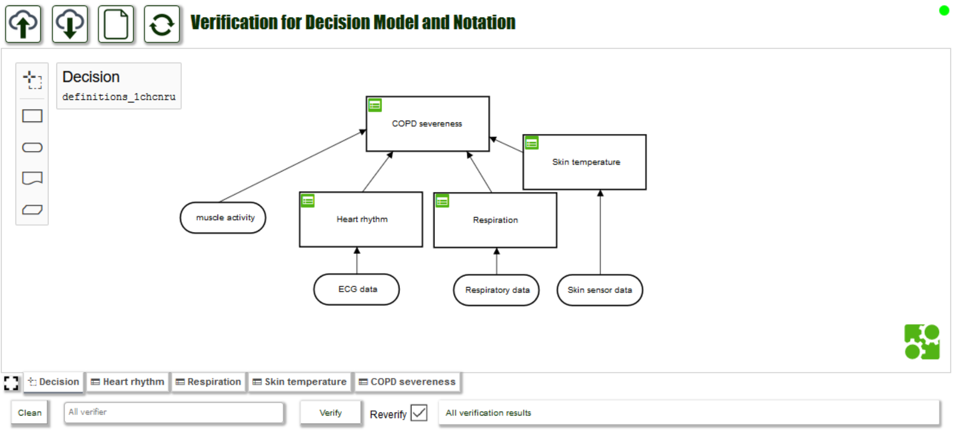


Figure 16: Stable situation after applying input data node inclusion.

action of adding new data input is selected and the modelling environment automatically introduces a matching input data node, as show in Figure 16.

8. Conclusion and Future Work

This paper investigates and defines the possible decision model change patterns for the evolution of DMN decision models. Through a running example we have illustrated that the adaptation of DMN decision model can lead to within-model inconsistencies. In order to remedy these inconsistencies, additional change patterns may need to be propagated throughout the entire decision model, effectively triggering other change patterns. This way, consistency between the different decision model elements is upheld. The proposed change patterns illustrate how decision models can evolve and which inconsistency concerns rise up when evolving the DMN models. Furthermore, the change patterns can aid in developing and implementing flexible and dynamic decision management systems in order to move away from the static approaches present both in literature and in industry. We illustrate the feasibility of the approach by providing a modelling environment prototype which provides modelling support and automation for change pattern administration on DMN decision models.

In future work, we will investigate how changing decision models impact other systems and models that rely on the logic encapsulated in the decision models. More specifically, we will examine how the proposed change patterns influence process and decision model consistency in an integrated process (BPMN) and decision (DMN) model environment. Changing the underlying decisions of a process can lead to a construction of a new process variant. Additionally, in the case of knowledge-intensive processes, instead of BPMN, change patterns can be considered for Case Management Model and Notation (CMMN) models, as they allow to model activities that can be performed in an unpredictable order by knowledge workers. Furthermore, it should be investigated whether DMN can be extended with a decision ontology to better cope with knowledge-intensive aspects that cannot adequately be represented in a decision table. Moreover, flexible decision models are of particular interest to Internet-of-Things (IoT) process settings [53, 54], as IoT process are inherently subject to a dynamic and changeable environment.

9. Appendix

This appendix gives an overview of the DMN metamodel, as specified in the latest DMN standard specification [1].

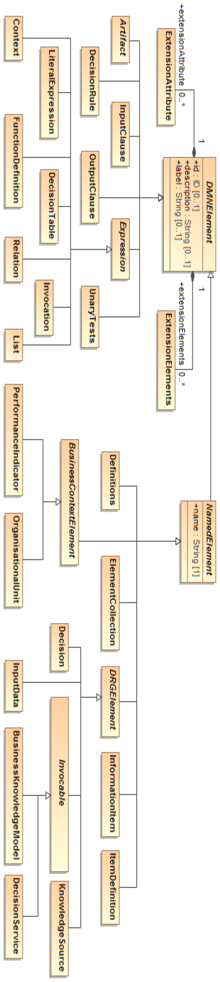


Figure 17: Overview of the DMN meta model.

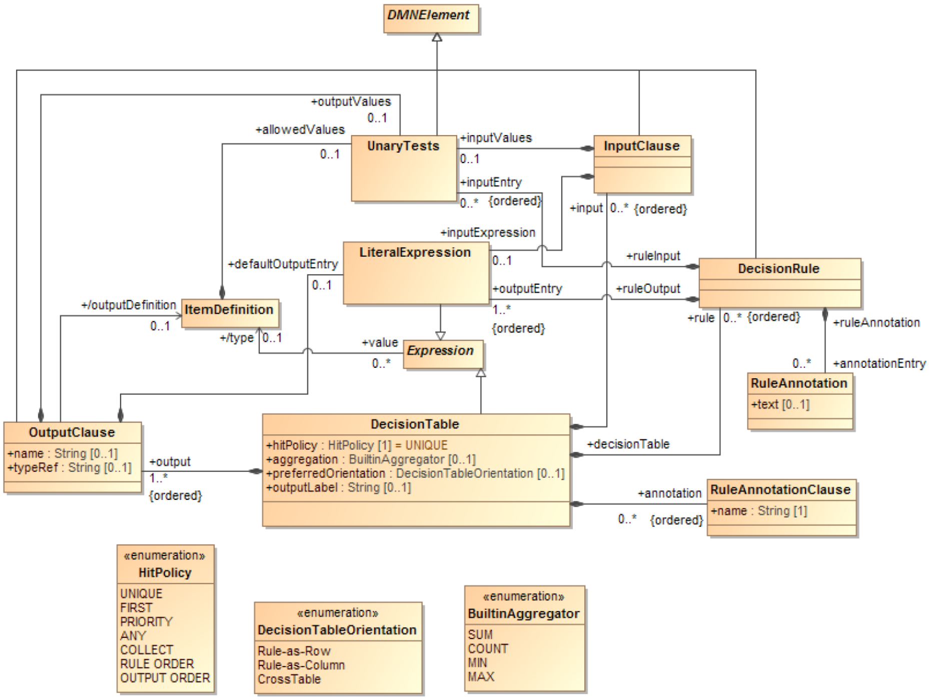


Figure 18: Decision table meta model.

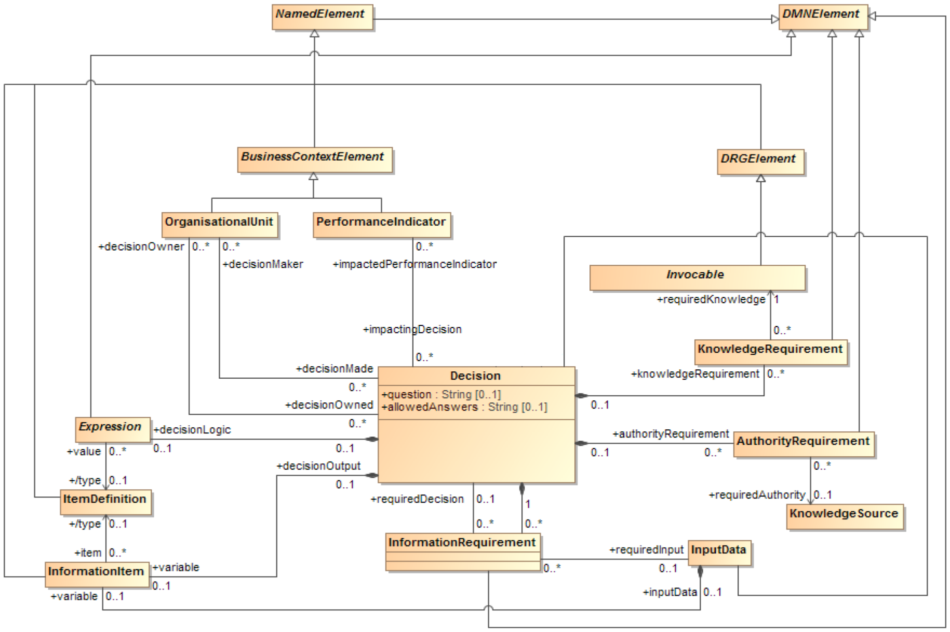


Figure 19: Decision meta model.

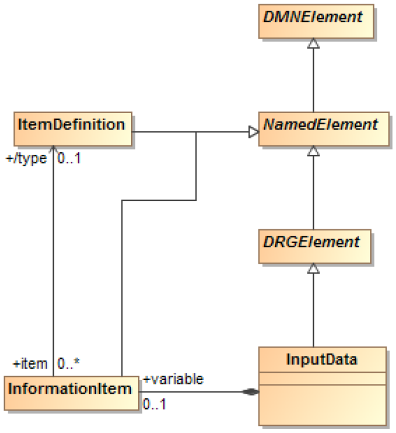


Figure 20: Input data meta model.

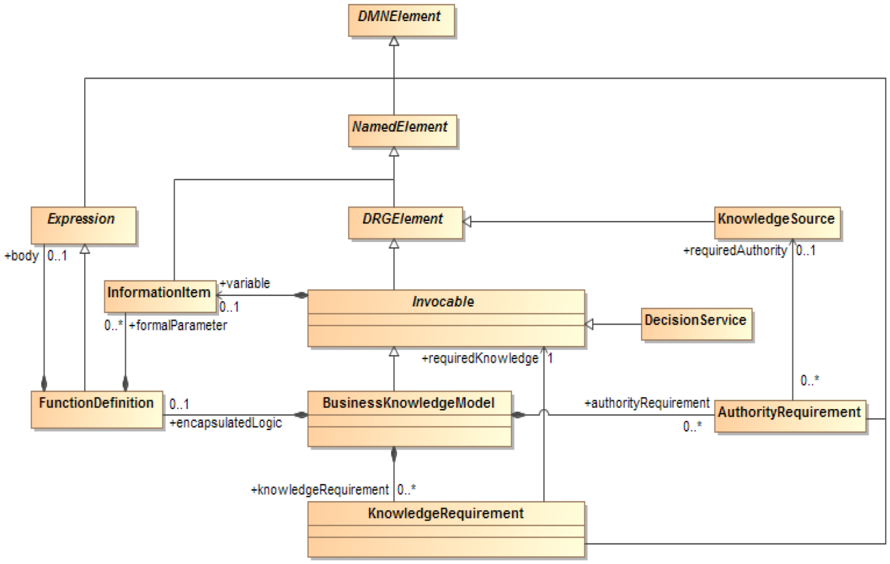


Figure 21: Business knowledge model meta model.

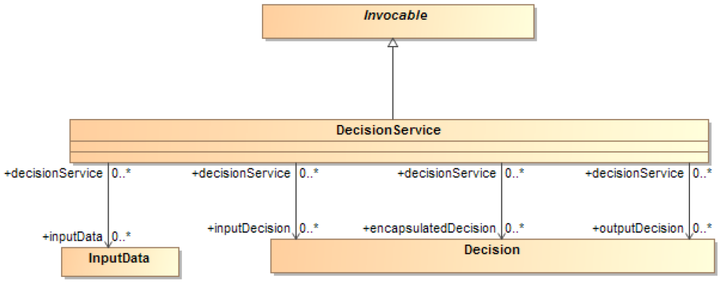


Figure 22: Decision service meta model.

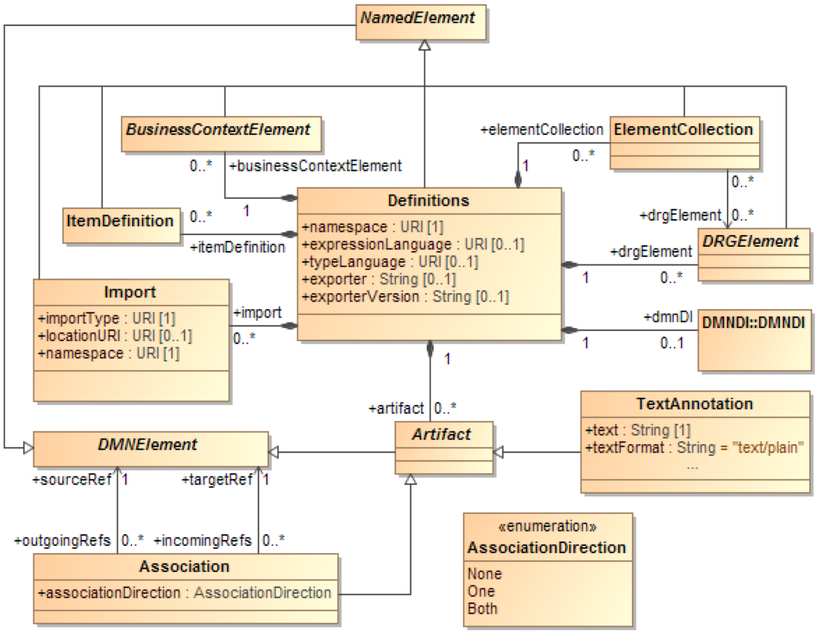


Figure 23: Definitions meta model.

Acknowledgements

This work was in part supported by the Deutsche Forschungsgemeinschaft (grant DE 1983/9-1).

References

[1] OMG, Decision Model and Notation (DMN) 1.2 (2018).

[2] F. Hasić, J. De Smedt, J. Vanthienen, Augmenting processes with decision intelligence: Principles for integrated modelling, Decision Support Systems 107 (2018) 1 – 12.

[3] D. Bork, D. Karagiannis, B. Pittl, Systematic analysis and evaluation of visual conceptual modeling language notations, in: 2018 12th International Conference on Research Challenges in Information Science (RCIS), IEEE, 2018, pp. 1–11.

[4] F. Hasi, J. Vanthienen, Complexity metrics for dmn decision models, Computer Standards & Interfaces 65 (2019) 15 – 37. `doi:https://doi.org/10.1016/j.csi.2019.01.006`.
URL `http://www.sciencedirect.com/science/article/pii/S0920548918303647`

[5] K. Hinkelmann, A. Pierfranceschi, E. Laurenzi, The knowledge work designer-modelling process logic and business logic., in: Modellierung (Workshops), 2016, pp. 135–140.

[6] M. Deryck, F. Hasić, J. Vanthienen, J. Vennekens, A case-based inquiry into the decision model and notation (dmn) and the knowledge base (kb) paradigm, in: C. Benzmüller, F. Ricca, X. Parent, D. Roman (Eds.), Rules and Reasoning, Springer International Publishing, Cham, 2018, pp. 248–263.

[7] D. Bork, R. Buchmann, D. Karagiannis, M. Lee, E.-T. Miron, An open platform for modeling method conceptualization: The omilab digital ecosystem, Communications of the Association

for Information Systems 44 (2019) 673–697. doi:10.17705/1CAIS.04432.
URL http://eprints.cs.univie.ac.at/5462/

[8] F. Hasić, J. Vanthienen, From decision knowledge to e-government expert systems: the case of income taxation for foreign artists in belgium, Knowledge and Information Systemsdoi:10.1007/s10115-019-01416-4.
URL https://doi.org/10.1007/s10115-019-01416-4

[9] J. Campos, P. Richetti, F. A. Baião, F. M. Santoro, Discovering business rules in knowledge-intensive processes through decision mining: an experimental study, in: International Conference on Business Process Management, Springer, 2017, pp. 556–567.

[10] F. M. Santoro, F. A. Baião, Knowledge-intensive process: A research framework, in: International Conference on Business Process Management, Springer, 2017, pp. 460–468.

[11] C. Corea, P. Delfmann, A tool to monitor consistent decision-making in business process execution, Proceedings of the Dissertation Award, Demonstration, and Industrial Track at BPM (2018) 9–14.

[12] K. Hinkelmann, Business process flexibility and decision-aware modeling-the knowledge work designer, in: Domain-Specific Conceptual Modeling, Springer, 2016, pp. 397–414.

[13] S. Nagel, C. Corea, P. Delfmann, Effects of quantitative measures on understanding inconsistencies in business rules, in: Proceedings of the 52nd Hawaii International Conference on System Sciences, 2019.

[14] K. Figl, J. Mendling, G. Tokdemir, J. Vanthienen, What we know and what we do not know about dmn, Enterprise Modelling and Information Systems Architectures (EMISAJ) 13 (2018) 2–1.

[15] F. Hasić, J. De Smedt, J. Vanthienen, An illustration of five principles for integrated process and decision modelling (5pdm), Available at SSRN 3082752.

[16] J. Hu, G. Aghakhani, F. Hasić, E. Serral, An evaluation framework for design-time context-adaptation of process modelling languages, in: IFIP Working Conference on The Practice of Enterprise Modeling, Springer, 2017, pp. 112–125.

[17] F. Hasić, J. De Smedt, J. Vanthienen, Developing a modelling and mining framework for integrated processes and decisions, in: C. Debruyne, H. Panetto, G. Weichhart, P. Bollen, I. Ciuciu, M.-E. Vidal, R. Meersman (Eds.), On the Move to Meaningful Internet Systems. OTM 2017 Workshops, Springer International Publishing, Cham, 2018, pp. 259–269.

[18] F. Hasić, J. De Smedt, J. Vanthienen, Redesigning processes for decision-awareness: Strategies for integrated modelling, in: International Conference on the Quality of Information and Communications Technology, 2018.

[19] F. Hasić, L. Vanwijck, J. Vanthienen, Integrating processes, cases, and decisions for knowledge-intensive process modelling, in: International Workshop on Practicing Open Enterprise Modeling, CEUR, 2017.

[20] T. Biard, J.-P. Bourey, M. Bigand, J.-C. Bocquet, Modélisation des prises de décisions dans les processus métier grâce à dmn (decision model and notation), in: 12ème Congrès International de Génie Industriel 2017, 2017.

[21] T. Biard, J.-P. Bourey, M. Bigand, Dmn (decision model and notation): De la modélisation à l'automatisation des décisions, in: INFORSID 2017, 2017.

[22] T. Biard, A. Le Mauff, M. Bigand, J.-P. Bourey, Separation of decision modeling from business

process modeling using new decision model and notation(dmn) for automating operational decision-making, in: Working Conference on Virtual Enterprises, Springer, 2015, pp. 489–496.

[23] K. Batoulis, A. Baumgraß, N. Herzberg, M. Weske, Enabling dynamic decision making in business processes with dmn, in: International Conference on Business Process Management, Springer, 2015, pp. 418–431.

[24] J. De Smedt, F. Hasi, S. vanden Broucke, J. Vanthienen, Holistic discovery of decision models from process execution data, Knowledge-Based Systems 183 (2019) 104866. `doi:https://doi.org/10.1016/j.knosys.2019.104866`.
URL `http://www.sciencedirect.com/science/article/pii/S0950705119303405`

[25] K. Batoulis, M. Weske, Soundness of decision-aware business processes, Working Paper. URL goo.gl/nhtCZ7.
URL `goo.gl/nhtCZ7`

[26] K. Batoulis, S. Haarmann, M. Weske, Various notions of soundness for decision-aware business processes, in: International Conference on Conceptual Modeling, Springer, 2017, pp. 403–418.

[27] K. Hinkelmann, K. Kritikos, S. Kurjakovic, B. Lammel, R. Woitsch, A modelling environment for business process as a service, in: International Conference on Advanced Information Systems Engineering, Springer, 2016, pp. 181–192.

[28] B. von Halle, L. Goldberg, The decision model: A business logic framework linking business and technology, Taylor and Francis, LLC.

[29] B. von Halle, L. Goldberg, The decision model# 2: Improving process models and the requirements process, Knowledge Partners International, LLC.

[30] J. B. dos Santos França, J. M. Netto, J. do ES Carvalho, F. M. Santoro, F. A. Baião, M. Pimentel, Kipo: the knowledge-intensive process ontology, Software & Systems Modeling 14 (3) (2015) 1127–1157.

[31] F. Santoro, T. Slaats, T. T. Hildebrandt, F. Baiao, Dcr-kipn a hybrid modeling approach for knowledge-intensive processes, in: International Conference on Conceptual Modeling, Springer, 2019, pp. 153–161.

[32] Z. Bellahsene, Schema evolution in data warehouses, Knowledge and Information Systems 4 (3) (2002) 283–304.

[33] N. F. Noy, M. Klein, Ontology evolution: Not the same as schema evolution, Knowledge and information systems 6 (4) (2004) 428–440.

[34] W. M. W. Kadir, P. Loucopoulos, Linking and propagating business rule changes to is design, in: Information Systems Development, Springer, 2005, pp. 253–264.

[35] J. Boyer, H. Mili, Agile business rule development, in: Agile Business Rule Development, Springer, 2011, pp. 49–71.

[36] F. Corradini, A. Polzonetti, O. Riganelli, Business rules in e-government applications, arXiv preprint arXiv:1802.08484.

[37] M. Szvetits, U. Zdun, Systematic literature review of the objectives, techniques, kinds, and architectures of models at runtime, Software & Systems Modeling 15 (1) (2016) 31–69.

[38] G. Blair, N. Bencomo, R. B. France, Models@ run. time, Computer 42 (10) (2009) 22–27.

[39] S. Loukil, S. Kallel, M. Jmaiel, An approach based on runtime models for developing dynamically adaptive systems, Future Generation Computer Systems 68 (2017) 365–375.

[40] A. Wombacher, M. Rozie, Evaluation of workflow similarity measures in service discovery., Service Oriented Electronic Commerce 80 (2006) 51–71.

[41] C. Calero, M. Piattini, M. Genero, Empirical validation of referential integrity metrics, Information and Software Technology 43 (15) (2001) 949 – 957. `doi:https://doi.org/10.1016/S0950-5849(01)00202-6`.
URL `http://www.sciencedirect.com/science/article/pii/S0950584901002026`

[42] C. Ordonez, J. Garca-Garca, Referential integrity quality metrics, Decision Support Systems 44 (2) (2008) 495 – 508. `doi:https://doi.org/10.1016/j.dss.2007.06.004`.
URL `http://www.sciencedirect.com/science/article/pii/S0167923607000887`

[43] E. Bazhenova, F. Zerbato, B. Oliboni, M. Weske, From bpmn process models to dmn decision models, Information Systems 83 (2019) 69–88.

[44] OMG, Business process model and notation (BPMN) 2.0 (2011).

[45] M. Hassanalieragh, A. Page, T. Soyata, G. Sharma, M. Aktas, G. Mateos, B. Kantarci, S. Andreescu, Health monitoring and management using Internet-of-Things (IoT) sensing with cloud-based processing: Opportunities and challenges, in: 2015 IEEE international conference on services computing (SCC), IEEE, 2015, pp. 285–292.

[46] F. E. A. Horita, D. Link, J. P. de Albuquerque, B. Hellingrath, odmn: an integrated model to connect decision-making needs to emerging data sources in disaster management, in: System Sciences (HICSS), 2016 49th Hawaii International Conference on, IEEE, 2016, pp. 2882–2891.

[47] F. E. Horita, J. P. de Albuquerque, V. Marchezini, E. M. Mendiondo, Bridging the gap between decision-making and emerging big data sources: an application of a model-based framework to disaster management in brazil, Decision Support Systems 97 (2017) 12–22.

[48] D. Calvanese, M. Dumas, Ü. Laurson, F. M. Maggi, M. Montali, I. Teinemaa, Semantics, analysis and simplification of dmn decision tables, Information Systems 78 (2018) 112–125.

[49] Ü. Laurson, F. M. Maggi, A tool for the analysis of dmn decision tables., in: BPM (Demos), 2016, pp. 56–60.

[50] K. Batoulis, M. Weske, A tool for checking soundness of decision-aware business processes, 2016.

[51] C. Corea, J. Blatt, P. Delfmann, A tool for decision logic verification in dmn decision tables, in: Proceedings of the Dissertation Award, Doctoral Consortium, and Demonstration Track at BPM 2019 co-located with 17th International Conference on Business Process Management, BPM 2019, Vienna, Austria, September 1-6, 2019, 2019.

[52] G. Wets, J. Vanthienen, H. Timmermans, Modelling decision tables from data, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, 1998, pp. 412–413.

[53] F. Hasić, E. Serral Asensio, Executing IoT processes in BPMN 2.0: Current support and remaining challenges, IEEE RCIS 2019 proceedings.

[54] C. Janiesch, A. Koschmider, M. Mecella, B. Weber, A. Burattin, C. D. Ciccio, A. Gal, U. Kannengiesser, F. Mannhardt, J. Mendling, A. Oberweis, M. Reichert, S. Rinderle-Ma, W. Song, J. Su, V. Torres, M. Weidlich, M. Weske, L. Zhang, The internet-of-things meets business process management: Mutual benefits and challenges, CoRR abs/1709.03628. `arXiv:1709.03628`. URL `http://arxiv.org/abs/1709.03628`

Appendix A

Library Usage Instructions

The general library can be downloaded at `https://gitlab.uni-koblenz.de/fg-bks/inconsistencylibrary`. Also, browser-based interfaces using the library can be found at:

- Error Verification Tool for DMN [27]:
  `http://inconsistency.fg-bks.uni-koblenz.de:8090/`
- Run-Time Monitoring Plugin for Camunda [30]:
  `http://inconsistency.fg-bks.uni-koblenz.de/`

Screencast for these tools can be found at `https://www.youtube.com/watch?v=yTXTKi3s6LM`, resp. `https://www.youtube.com/watch?v=jus4IkLMOIg`

The library is written in Java and allows to compute (quasi-) inconsistencies in business rule bases. Table A.1 shows the important packages of the library.

Formalism	**Root**	**Notes/Dependencies**
FCL	`spindle-fork/`	Embedded in SPINdle library[a]
Declare	`declare-lang/`	Guava[b]
DMN	`dmn/`	Camunda-dmn[c]

[a] `http://spindle.data61.csiro.au/spindle/`, [b] `https://github.com/google/guava/`,
[c] `https://github.com/camunda/camunda-engine-dmn`

Table A.1: Library Overview

The .jar file *"inconLib.jar"* can be used via the command line, via the following syntax:

```
Defeasible Logic:
        java -jar inconLib.jar fcl <Filename>
Declare:
        java -jar inconLib.jar declare <Filename>
DMN:
        java -jar inconLib.jar dmn <Filename>
```

These commands return all minimal inconsistent subsets (where applicable) and actual issues ("quasi-inconsistencies"). Exemplary test-files can be found in the downloadable Java project.

Furthermore, the library can be integrated into projects for future work. Figure A.1 shows an example for computing all quasi-inconsistent subsets in a Declare rule base.

```
File path = new File("src/main/resources/example.txt");
ParseDeclareInput testParser = new ParseDeclareInput();
DeclareRuleBase rb = testParser.getDeclareRuleBase(path);
DeclareSolver solver = new DeclareSolver(rb);


solver.getGraphSolver().getResultQMIS().forEach(list->System.out.println(list));
```

Figure A.1: Exemplary Code snippet for the basic computation of actual issues in Declare

Appendix B

Curriculum Vitae

Carl Corea is a research assistant at the Institute for Information Systems Research (at the University of Koblenz-Landau). He received his bachelor's degree in Information Management and master's degree in Web Science, both at the University of Koblenz-Landau. His research interest lies on the intersection between business informatics (especially Business Process Management) and a more formal, theoretical informatics.

Personal data

Name:	Carl Corea
Date of Birth:	07.04.1990
Citizenship:	German/Canadian

Education

2010-2014	B.Sc. Information Management, University of Koblenz-Landau
2014-2016	M.Sc. Web Science, University of Koblenz-Landau

Employment

2013-2016	Web-Developer, 247GRAD GmbH
2016-current	Research assistant, University of Koblenz-Landau

Scientific Activities

- Invited talk on *"Intelligent Chatbot Systems"* at the Science Dialogue of the 8th "Zukunftskongress (Staat & Verwaltung)", 2020. Berlin.
- Teaching Assignment (Lehrauftrag) - Business Process Management (Master), 2019. University of Koblenz-Landau
- Teaching Assignment (Lehrauftrag) - Project Management (Bachelor), 2019. University of Applied Sciences Südwestfalen.
- Teaching Assignment (Lehrauftrag) - Introduction to Business Informatics (Bachelor), 2017. University of Applied Sciences Bonn-Rhein-Sieg.
- Doctoral Consortium - 15th International Conference on Business Process Management (BPM 2017), 2017.

Full list of Publications

- Carl Corea, Matthias Thimm. On Quasi-Inconsistency and its Complexity. In *Artificial Intelligence* (AIJ). 2020.
- Carl Corea, Patrick Delfmann. A Taxonomy of Business Rule Organizing Approaches in Regard to Business Process Compliance. In *International Journal of Conceptual Modeling* (EMISAJ). 2020
- Faruk Hasic, Carl Corea, Jonas Blatt, Patrick Delfmann and Estefania Serral. *A Tool for the Verification of Decision Model and Notation (DMN) Models.* In Proceedings of the 14th International Conference on Research Challenges in Information Science (RCIS 2020). Limassol, 2020.
- Faruk Hasic, Carl Corea, Jonas Blatt, Patrick Delfmann and Estefania Serral. Decision Model Change Patterns for Dynamic System Evolution. In *Knowledge and Information Systems* (KAIS). 2020
- Carl Corea, Matthias Thimm. *Towards Inconsistency Measurement in Business Rule Bases.* In Proceedings of the 24th European Conference on Artificial Intelligence (ECAI 2020). Santiago de Compostela, 2020.
- Carl Corea, Patrick Delfmann, Sabine Nagel. *Towards Intelligent Chatbots for Customer Care - Practice-Based Requirements for a Research Agenda.* In Proceedings of the 53rd Hawaii International Conference on System Sciences (HICSS 2020). Maui, 2020.
- Carl Corea, Sabine Nagel, Patrick Delfmann. *Effects of Visualization Techniques on Understanding Inconsistencies in Automated Decision-Making.* In Proceedings of the 53rd Hawaii International Conference on System Sciences (HICSS 2020). Maui, 2020.

- Carl Corea, Jonas Blatt, Patrick Delfmann. *A Tool for Decision-Logic Level Verification in DMN Decision Tables.* In Proceedings of the Demonstration Track at BPM 2019 co-located with the 17th International Conference on Business Process Management (BPM 2019). Wien, 2019.
- Carl Corea, Patrick Delfmann. *Quasi-Inconsistency in Declarative Process Models.* In Business Process Management Forum co-located with the 17th International Conference on Business Process Management (BPM 2019). Wien, 2019.
- Patrick Delfmann, Dennis M. Riehle, Steffen Höhenberger, Carl Corea, Christoph Drodt. *The Diagramed Model Query Language 2.0: Design, Implementation, and Evaluation.* In Polyvyanyy, A. (Ed.), Process Querying Methods. 2019.
- Carl Corea, Matthias Deisen, Patrick Delfmann. *Resolving Inconsistencies in Declarative Process Models based on Culpability Measurement.* In Proceedings der 14. Internationalen Tagung der Wirtschaftsinformatik (WI 2019). Siegen, 2019.
- Sabine Nagel, Carl Corea, Patrick Delfmann. *Effects of Quantitative Measures on Understanding Inconsistencies in Business Rules.* In Proceedings of the 52nd Hawaii International Conference on System Sciences (HICSS 2019). Hawaii, 2019.
- Carl Corea, Patrick Delfmann. *Supporting Business Rule Management with Inconsistency Analysis.* In Proceedings of the Industrial Track at BPM 2018 co-located with the 16th International Conference on Business Process Management (BPM 2018). Sydney, 2018.
- Carl Corea, Patrick Delfmann. *A Tool to Monitor Consistent Decision-Making in Business Process Execution.* In Proceedings of the Demonstration Track at BPM 2018 co-located with the 16th International Conference on Business Process Management (BPM 2018). Sydney, 2018.
- Carl Corea, Patrick Delfmann. *Detecting Compliance with Business Rules in Ontology-Based Process Modeling.* In Proceedings der 13. Internationalen Tagung der Wirtschaftsinformatik (WI 2017). St.Gallen, 2017.
- Carl Corea, Matthias Thimm. *Using Matrix Exponentials for Abstract Argumentation.* In Proceedings of the First Workshop on Systems and Applications of Formal Argumentation (SAFA 2016). Postdam, 2016.