6-18-2022

Investigating Inconsistency Understanding to Support Interactive Inconsistency Resolution in Declarative Process Models

Sabine Nagel  
_Institute for IS Research_, snagel@uni-koblenz.de

Patrick Delfmann  
_University of Koblenz-Landau_, delfmann@ercis.de

Follow this and additional works at: https://aisel.aisnet.org/ecis2022_rip

Recommended Citation
https://aisel.aisnet.org/ecis2022_rip/59

This material is brought to you by the ECIS 2022 Proceedings at AIS Electronic Library (AISeL). It has been accepted for inclusion in ECIS 2022 Research-in-Progress Papers by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.
INVESTIGATING INCONSISTENCY UNDERSTANDING TO SUPPORT INTERACTIVE INCONSISTENCY RESOLUTION IN DECLARATIVE PROCESS MODELS

Research in Progress

Sabine Nagel, University of Koblenz-Landau, Koblenz, Germany, snagel@uni-koblenz.de
Patrick Delfmann, University of Koblenz-Landau, Koblenz, Germany, delfmann@uni-koblenz.de

Abstract

Handling inconsistencies in business rules is an important part of corporate compliance management. This includes the resolution of inconsistencies, which currently is a fully automated process that might not always be plausible in a real-world scenario. To include human experts and develop interactive resolution approaches, an understanding of inconsistencies is crucial. Thus, we focus on investigating inconsistency understanding in declarative process models by testing the applicability of insights from declarative process model understanding to different inconsistency characteristics. In the future, this will provide the basis for a series of cognitive experiments evaluating the effects of inconsistency characteristics and representation on inconsistency understanding in declarative process models.

Keywords: Inconsistencies, Declarative Process Models, Comprehension, Inconsistency Resolution

1 Introduction

Managing compliance with internal and external regulations is a current challenge for organizations (Hashmi et al., 2018). These regulations are often represented as business rules, which are declarative statements that govern company behavior (Graham, 2006). In this work, we focus on declarative process models (DPM), which are “set[s] of constraints that must all be satisfied during the process run” (Figl et al., 2020, p. 123). DPM can automatically be discovered from event logs (Di Ciccio et al., 2017; Maggi et al., 2011), or modeled manually. However, rules are often modeled collaboratively and incrementally (Batoulis and Weske, 2018), and existing process discovery approaches only focus on the extraction of rules, without taking interrelations between rules into account (Di Ciccio et al., 2017). This can lead to contradictory statements within rule sets being modeled, which are referred to as inconsistencies. In recent years, several approaches for handling inconsistencies in business rules have been introduced, including first approaches for inconsistency resolution (Corea et al., 2019; Di Ciccio et al., 2017). However, there currently only exist (semi-)automated approaches that delete elements from a rule base to achieve consistency. While this is an important first step, it might not always be plausible or applicable in a real-world scenario, as it might lead to erroneous rules being kept, while potentially business-critical rules are deleted instead (Corea et al., 2021). To solve this problem, it is crucial to include human experts in novel and interactive inconsistency resolution approaches. However, this requires humans to be able to understand the problem and identify erroneous rules. This includes understanding that there is a problem and why the interplay between rules leads to an inconsistency. Existing research on DPM understanding has shown that especially combinations of constraints and hidden dependencies pose challenges to human modelers (De Smedt et al., 2016; Haisjackerl et al., 2016). As inconsistencies are subsets of DPM and many challenges from the area of DPM understanding refer to characteristics that are common for inconsistencies, the first research objective (ROI) of this work is to identify these challenges by applying a systematic
literature review. Our second aim (RO2) is to investigate how insights from DPM can be applied to the understanding of inconsistencies by mapping the identified challenges to different characteristics of inconsistencies in DPM. As understanding of inconsistencies in DPM has not been investigated by existing research, the results of this work will provide the basis for a series of future cognitive experiments. By empirically evaluating the effects of inconsistency characteristics and representation on inconsistency understanding we aim to close any gaps identified in the theoretical mapping and specifically focus on characteristics that are unique to inconsistencies, e.g., their type and structure.

In Section 2 we will introduce the preliminaries regarding DPM and inconsistencies. This is followed by a structured literature review on DPM understanding challenges in Section 3, which serves as a basis for a mapping of these challenges and inconsistency characteristics in Section 4. We conclude this research in progress with a discussion of future steps in Section 5.

2 Related Work

2.1 Declarative Process Models

In contrast to procedural business process models, which specify each possibility of execution directly in the model, DPM represent circumstantial information (Fahland et al., 2009; Pichler et al., 2012) and define allowed behavior in the form of constraints (Figl et al., 2020). A common modeling language to formalize such constraints is DECLARE (Figl et al., 2020; van der Aalst and Pesic, 2006), which provides a set of constraint templates based on temporal logic (see Table 1). Existence constraints represent cardinality or position restrictions involving single activities, while relation constraints consist of an activation (underlined) that requires or prohibits a target activity. For a detailed overview of DECLARE templates, their graphical notation, and their subsumption hierarchy, we refer to Di Ciccio et al. (2017).

<table>
<thead>
<tr>
<th>Template</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cardinality</td>
<td>EXACTLY</td>
</tr>
<tr>
<td>Position</td>
<td>INIT(a)</td>
</tr>
<tr>
<td>Forward</td>
<td>CHAIN RESPONSE(a,b)</td>
</tr>
<tr>
<td>Backward</td>
<td>CHAIN PRECEDENCE(a,b)</td>
</tr>
<tr>
<td>Coupling</td>
<td>CHAIN SUCCESSION(a,b)</td>
</tr>
<tr>
<td>Negation</td>
<td>NOT RESPONSE(a,b)</td>
</tr>
</tbody>
</table>

Table 1: Overview of DECLARE templates (adapted from Di Ciccio et al. (2017))

2.2 Inconsistencies in Declarative Process Models

A DPM that does not accept any finite execution trace is referred to as unsatisfiable or inconsistent (Corea and Delfmann, 2019; Di Ciccio et al., 2017). In addition to inconsistencies in the classical-logical sense, which can be assessed already at design-time, recent works (Corea and Delfmann, 2019; Corea and Thimm, 2020) have introduced so-called potential inconsistencies that can only be assessed at run-time. If a constraint set contains an inconsistency in the classical-logical sense (e.g., [INIT(a), RESPONSE(a,b), NOT RESPONSE(a,b)]), the language that satisfies the constraint set is empty. In contrast, potential inconsistencies (Corea and Delfmann, 2019) do not contain any information about the activation of rules and, thus, allow no immediate inference. However, they still lead to contradictory conclusions when their shared condition is activated (e.g., [RESPONSE(a,b), NOT RESPONSE(a,b)]). To measure and resolve inconsistencies, we need to identify minimal inconsistent subsets (MIS), which contain all constraints that are always activated together and lead to contradictory conclusions (Corea and Delfmann, 2019). MIS are minimal in terms of set inclusion, i.e., as soon as exactly one constraint is removed from the subset, the inconsistency is resolved (Corea and Delfmann, 2019). For simplicity, we will use the term MIS for both classic and potential inconsistencies in the remainder of the paper.
We distinguish between different inconsistency structures, which can generally be divided into three categories, namely contradictions regarding the relation between two activities, as well as the position or cardinality of a single activity. Table 2 contains eight exemplary constraint sets, each consisting of a statement (comprising a single or multiple constraints) and its contradiction. \( C_1 \) and \( C_2 \) consist of a statement and its direct negation, while the statement being contradicted in \( C_3 \) is a chain of constraints. In \( C_4 \), the contradicted statement contains a hidden dependency, i.e., an “interaction between constraints and their activities that is not made explicit as such in the model itself” (De Smedt et al., 2016, p. 86). Examples of structures that comprise contradictory statements regarding the position of an event are \( C_5 \), which implies that each trace must start with \( a \) and \( b \), and \( C_6 \), which requires both \( b \) and \( c \) to immediately follow \( a \), which can never be satisfied. Furthermore, two statements can either directly contradict each other regarding the number of activities in a trace \( (C_7) \) or a combination of constraints can require a different number of activities than the contradicting constraint implies \( (C_8) \).

<table>
<thead>
<tr>
<th>Statement</th>
<th>Contradiction</th>
<th>Structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>( C_1 ) RESPONSE( (a,b) )</td>
<td>NOTRESPONSE( (a,b) )</td>
<td>Direct opposites (same template)</td>
</tr>
<tr>
<td>( C_2 ) RESPONSE( (a,b) )</td>
<td>NOTPRECEDE( (a,b) )</td>
<td>Direct opposites (different template)</td>
</tr>
<tr>
<td>( C_3 ) RESPONSE( (a,b) ), RESPONSE( (b,c) )</td>
<td>NOTRESPONSE( (a,c) )</td>
<td>Chained opposites</td>
</tr>
<tr>
<td>( C_4 ) RESPONSE( (a,b) ), CHAINRESPONSE( (a,c) )</td>
<td>NOTPRECEDE( (c,b) )</td>
<td>Opposites with hidden dependency</td>
</tr>
<tr>
<td>( C_5 ) INIT( (a) )</td>
<td>INIT( (b) )</td>
<td>Contradicting relative position</td>
</tr>
<tr>
<td>( C_6 ) CHAINRESPONSE( (a,b) )</td>
<td>CHAINRESPONSE( (a,c) )</td>
<td>Contradicting relative position</td>
</tr>
<tr>
<td>( C_7 ) ATMOST( ONE( (a) )</td>
<td>EXACTLY( TWO( (a) )</td>
<td>Explicit cardinality contradiction</td>
</tr>
<tr>
<td>( C_8 ) EXACTLY( ONE( (b) ), CHAINRESPONSE( (a,b) )</td>
<td>EXACTLY( TWO( (a) )</td>
<td>Implicit cardinality contradiction</td>
</tr>
</tbody>
</table>

Table 2: Examples of different inconsistency structures

An additional structure that does not accept any finite trace are loops \( (C_9 = \{\text{RESPONSE}(a,b), \text{RESPONSE}(b,a)\}) \). Please note that self-loops (e.g., \( \{\text{RESPONSE}(a,a)\} \)) might also occur. Here, the problem lies in the constraint itself as opposed to the interplay between constraints, so the issue can only be resolved by removing the constraint, which does not require any human-in-the-loop intervention.

In recent years, several approaches for the measurement of both classic (Corea and Delfmann, 2018; Thimm, 2019) and potential inconsistencies (Corea and Delfmann, 2019; Corea and Thimm, 2020) have been introduced, that not only allow quantifying the culpability of a constraint but can also serve as the basis for inconsistency resolution approaches, as explained in the following section.

2.3 Inconsistency Resolution

To date, several works have introduced fully automated approaches (Corea et al., 2019; Di Ciccio et al., 2017; Maggi et al., 2012) to support inconsistency resolution by deleting elements from a rule base to achieve consistency. A common approach is to delete elements in order of their culpability, i.e., to start with the element that is responsible for the highest number of inconsistencies (Corea et al., 2019; Maggi et al., 2012). However, these approaches cannot guarantee minimal information loss w.r.t. the number of deleted constraints, as they can run into local optima. In the exemplary DECLARE model in Figure 1, three constraints \( (\text{CHAINRESPONSE}(a,c), \text{CHAINRESPONSE}(a,b), \text{SUCCESSION}(d,e)) \) share the highest culpability value of 2, so one of these constraints would be deleted first. However, deleting \( \text{CHAINRESPONSE}(a,c) \) first would result in a total number of three required deletions, while consistency can generally be achieved by deleting only two constraints. To solve this issue and guarantee minimal deletions of constraints, Corea et al. (2021) have proposed an approach that returns the cardinality smallest correction set, i.e., the cardinality smallest set of elements that have to be deleted to achieve consistency \( (\{\text{CHAINRESPONSE}(a,b), \text{SUCCESSION}(d,e)\}) \).

Generally, automated approaches might not always be plausible in a real-world scenario, as they might lead to erroneous rules being kept, while potentially business-critical rules are deleted instead. Consider the constraints in MIS, with \( d = \text{receive payment} \) and \( e = \text{close order} \). In that case, \( \text{SUCCESSION}(d,e) \) would imply that an order will only be closed after payment has been received and
**Inconsistency Understanding in Declarative Process Models**

**NOT PRECEDENCE**\((d,e)\) indicates that closing an order must never be preceded by its payment. Intuitively, **NOT PRECEDENCE**\((d,e)\) seems to be the erroneous rule, as companies need to ensure payment before the corresponding order is being closed. However, minimal resolution approaches would delete **SUCCcession**\((d,e)\) instead, which could lead to financial loss for the company.

![Diagram](image)

**Figure 1**: Exemplary DECLARE model (1) and corresponding inconsistencies (2)

Thus, Corea et al. (2021) have introduced an interactive repair, where a user is presented with minimal correction sets, but has the possibility to prevent certain constraints from being deleted or receive quantitative or visual decision-making support. However, this would require the user to check up to nine correction sets without seeing the contained constraints in the context of their inconsistencies. Therefore, we aim to focus on a stepwise and interactive inconsistency resolution by confronting users with one inconsistency (MIS) at a time and asking them to resolve it. The goal is to restore consistency in an iterative manner while keeping the required time and mental effort to a minimum. The idea is to identify inconsistencies that require little mental effort to understand and present them to the user first. That way, assuming that constraints are commonly part of multiple inconsistencies, consistency can be restored without having to be exposed to the more complex and mentally challenging inconsistencies.

## 3 Declarative Process Model Understanding

Understanding how humans comprehend process models has been of significant interest to scholars and practitioners. This includes both procedural (Figl, 2017) and declarative process models (Figl et al., 2020; Haisjackl et al., 2016). As we aim to investigate understanding of inconsistencies, which are subsets of DPM, we will now identify and classify common problems with DPM understanding.

### 3.1 Literature Search Process

We conducted a systematic literature review following vom Brocke et al. (2015, 2009) and Webster and Watson (2002). First, we performed a keyword search in five scientific databases (see Figure 2), which were selected to cover a broad range of scientific publications relevant to our domain. We limited our search to the title, abstract, and keywords of a publication and considered journal articles and conference papers. To identify relevant sources, we employed the search string \("(declarative process model\*" OR "declarative business process model\*) AND ("understand\*" OR "comprehend\*" OR "perce\*" OR "cognitive\")\). Here we combined DPM with synonyms of understanding to ensure exhaustiveness of the search process, while still only considering relevant papers.

As shown in Figure 2, the keywords search yielded 291 results, of which 30 were considered relevant after title and abstract screening. After removing 4 duplicates, we screened the full text of the remaining 26 papers. We generally identified relevant papers based on the following selection criteria:

1. As the focus of this work is solely on DPM, we excluded all papers covering understandability of procedural or hybrid process models. For an exhaustive literature review on the comprehension of procedural process models, we refer to Figl et al. (2020).
2. We only considered empirical studies measuring understanding of existing DPM and theoretical discussions (e.g., literature-based research models or propositions) with an appropriate level of detail. Papers that only discuss DPM understanding in their related work sections were not included in the final set.
Inconsistency Understanding in Declarative Process Models

![Diagram](Figure 2: Literature Search Process)

The resulting initial set of 17 papers was then used as the starting point for backward and forward snowballing (Wohlin, 2014). First, we scanned the references of all papers within the initial set and found one additional paper, which was not covered by the applied search string but was still considered relevant after full-text screening. After completing the backward snowballing process by scanning the references of the added paper, we performed one cycle of forward snowballing. Here we found one more paper that was published within a thesis (i.e., was not covered by the database search) but still contained highly relevant concepts, so we added the paper to our final set of 19 papers.

### 3.2 Declarative Process Model Understanding Challenges

Based on the 19 sources identified in the previous step, we extracted all DPM understanding challenges identified in these papers. Many sources already provided categories for the challenges they studied, so we adopted these categories in our initial version. Any individual challenges were then either assigned to an existing category or combined to form a new category. To refine and organize these categories, we created a concept matrix following Webster and Watson (2002). Table 3 provides an overview of the 19 identified papers mapped to the extracted and categorized DPM understanding challenges. As each category of challenges comprises multiple factors, we will provide a more detailed discussion of each category and its related challenges in the remainder of this section.

<table>
<thead>
<tr>
<th>ID</th>
<th>Category</th>
<th>(Andaloussi et al., 2019a)</th>
<th>(Andaloussi et al., 2019b)</th>
<th>(Andaloussi et al., 2020a)</th>
<th>(Andaloussi et al., 2021b)</th>
<th>(De Smedt et al., 2016)</th>
<th>(De Smedt et al., 2018)</th>
<th>(Fahland et al., 2009a)</th>
<th>(Fahland et al., 2009b)</th>
<th>(Haisjacker et al., 2012a)</th>
<th>(Haisjacker et al., 2012b)</th>
<th>(Haisjacker et al., 2012c)</th>
<th>(Haisjacker et al., 2013)</th>
<th>(Haisjacker et al., 2014)</th>
<th>(Zugal et al., 2011)</th>
<th>(Zugal et al., 2012a)</th>
<th>(Zugal et al., 2012b)</th>
<th>(Zugal et al., 2012c)</th>
<th>(Zugal et al., 2012d)</th>
<th>(Zugal et al., 2015)</th>
</tr>
</thead>
<tbody>
<tr>
<td>UC1</td>
<td>Complexity</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>UC2</td>
<td>Individual Constraints</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UC3</td>
<td>Constraint Combinations</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UC4</td>
<td>Representation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UC5</td>
<td>Reading Order</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UC6</td>
<td>Background &amp; Experience</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UC7</td>
<td>Task Type</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3: Overview of DPM understanding challenge categories (* = theoretical discussions)

**Complexity** can be measured using a variety of metrics. The size of a DPM can be measured using the number of activities and constraints (Haisjacker and Zugal, 2014) or a combination of both (Andaloussi et al., 2021a). It is agreed upon that an increased size negatively affects understanding (Haisjacker et al., 2016; Zugal et al., 2012b), i.e., difficulty, comprehension accuracy, and reading time (Andaloussi et al., 2021a). The ratio of activities and constraints is referred to as density. Here, Andaloussi et al. (2021a) propose that a “high coupling between the activities […] requires more checks to evaluate the influence of each activity on the rest of activities” and later prove that an increased density indeed negatively influences difficulty and time to understand. A DPM can consist of multiple subgraphs that are not connected by any relations, which can be measured using the degree of connectivity.
(Andaloussi et al., 2021a). As “sets of activities that are within the same weakly connected component can be executed without considering the rest of the graph” (Andaloussi et al., 2021a), a lower number of maximal subgraphs seems to increase the perception of difficulty and time when trying to make sense of such models. Furthermore, variability, i.e., the number of different constraint types in a DPM, can also negatively affect perceived difficulty, comprehension accuracy, and comprehension time (Andaloussi et al., 2021a). One last factor is the use of modularization and hierarchy to reduce the complexity of DPM. According to Zugal et al. (2015, p. 1091) “using hierarchy means to abstract certain parts of a declarative process model by the means of sub-processes”. Here, the effects on understandability are diverse and highly depend on additional factors. More specifically, abstracting information through the use of sub-processes “decreases mental effort by hiding information” (Zugal et al., 2012c, p. 9), but might also increase mental effort by “forcing the reader to switch attention between fragments and integrating information from fragments” (Zugal et al., 2012c, p. 10).

**Individual constraints** seem to cause significantly fewer problems than combinations (Zugal et al., 2015). The most common problems include misunderstanding or mixing up certain templates (e.g., PRECEDENCE) (De Smedt et al., 2018; Haisjackl et al., 2016; Haisjackl and Zugal, 2014). Haisjackl et al. (2016, p. 331) also observed that activities are often named explicitly and connections implicitly when describing a DPM, while existence constraints were often not mentioned at all.

**Combinations of constraints** pose significant understanding challenges (Haisjackl et al., 2016; Zugal et al., 2015). This includes making sense of pairs of constraints (i.e., constraints that define conditions for the same pair of activities) (Haisjackl et al., 2016; Haisjackl and Zugal, 2014) and having to combine the semantics of multiple constraints to understand their interplay (Zugal et al., 2015). The latter becomes even more problematic when hidden dependencies are present, as understanding the underlying model requires analyzing all constraints for implicit dependencies between activities as opposed to just relying on explicit information (De Smedt et al., 2018, 2016). Haisjackl et al. (2016) and De Smedt et al. (2018) found that humans tend to understand hidden dependencies in less complex models, but fail to in more complex examples. Also, De Smedt et al. (2018, 2016) were able to show that further explanations and visualizations of hidden dependencies can increase understanding.

**Representation** has also been investigated regarding its effects on understanding. While visual representations of DPM can help humans to understand the interplay between constraints (Andaloussi et al., 2021b, 2019a), it was empirically shown that providing graphical DPM in addition to textual rules rather increases mental effort and leads to more reasoning mistakes being made (De Smedt et al., 2018; Figl et al., 2020; Haisjackl and Zugal, 2014). This could be explained by the presence of the split-attention effect, which “occurs when information from different sources has to be integrated and is known to increase mental effort” (Haisjackl and Zugal, 2014, p. 9) or the similarity in the graphical representations of procedural and DPM despite the semantics being different (Haisjackl and Zugal, 2014). Textual representations of DPM can either be in the form of natural language or a domain-specific language, such as law text (Andaloussi et al., 2019a; Figl et al., 2020), which allows to increase understandability, especially for participants with different backgrounds and levels of experience. Lastly, providing possibilities for DPM simulation, e.g., in the form of test cases, valid execution traces, or assertions, can reduce mental effort as it allows humans to better understand dependencies between constraints (Andaloussi et al., 2019a; Zugal et al., 2012a, 2012b, 2011). As all forms of representations have advantages and disadvantages and seem to be highly dependent on other factors, Andaloussi et al. (2019a) also propose the idea of using hybrid representation approaches.

**Reading order** of activities and constraints is not only dependent on the order of representation artifacts (Andaloussi et al., 2019a) and the order within one artifact (e.g., the graphical layout) (Zugal et al., 2012b) but also on the order in which the user tends to read the model when trying to understand the underlying process (Haisjackl et al., 2016). Regarding the latter, it has been found that humans tend to read DPM in a sequential way, despite the circumstantial nature of DPM (Haisjackl et al., 2016; Zugal et al., 2015). More specifically, subjects tried identifying an entry point, the order of activities, and the endpoint of the model (Haisjackl et al., 2016; Zugal et al., 2015). Furthermore, Haisjackl et al. (2016, p. 339) were able to show that “starting with the init constraint was more important to the subject than the prevalent reading direction of our culture”.

**Inconsistency Understanding in Declarative Process Models**
Background and experience of users seem to strongly influence not only DPM understanding but also the effects of other factors. For example, Andaloussi et al. (2019a) found that caseworkers tend to prefer law texts while academics prefer graphical representation, which can be traced back to their different backgrounds and proficiency. Also, a general lack of modeling experience can affect DPM understanding and, in particular, dealing with combinations of constraints (Haisjackson and Zugal, 2014; Zugal et al., 2015). However, previous experience with procedural process models can also negatively affect understanding, especially due to the similarity in the graphical representations of procedural and DPM, despite the semantics being different (Haisjackson et al., 2013; Haisjackson and Zugal, 2014).

The type of task a human is asked to execute based on a given DPM can also influence DPM understanding. Andaloussi et al. (2019a) discuss the effects of questions about individual constraints, contextual information, or allowed model behavior in general on the preferred type of DPM representation. Furthermore, Fahland et al. (2010, 2009) propose that, in contrast to procedural process models, DPM rather enable establishing and maintaining circumstantial information as opposed to sequential information, which is empirically confirmed by Pichler et al. (2012). Lastly, Haisjackson et al. (2016) look at the error distribution when asking users to name minimal, valid, and invalid traces and found that naming invalid traces was rather unproblematic as they “can be constructed by selecting a single constraint that can be violated“ (Haisjackson et al., 2016, p. 341) while naming minimal traces seemed to be considerably more difficult. Similarly, Zugal et al. (2012c, p. 10) explain that “when checking whether a certain execution trace is supported by a process model, activities that are not contained in the trace are irrelevant for answering the question”, which might reduce mental effort.

4 Inconsistency Understanding

As inconsistencies in DPM are always subsets of the overall model, we can apply the previously identified challenges (Table 3) to gain first insights into potential effects on inconsistency understanding. In the following, we will provide a theoretical discussion of these potential effects.

As part of an interactive and stepwise inconsistency resolution approach, we aim to show users one MIS at a time, so users are only shown relevant constraints for solving the current problem. By only providing a subset, the original model is automatically modularized, which lowers complexity (UC1) and is expected to decrease mental effort, as irrelevant information is hidden and attention only has to be focused on the current fragment (Zugal et al., 2015). However, inconsistencies always comprise combinations of constraints over the same activities (UC3), which has been found to pose significant understanding challenges. This highlights the need for providing interactive decision-making support and further investigating the trade-off between different understanding challenges.

Generally, inconsistencies consist of at least two constraints. In contrast to potential inconsistencies, classic inconsistencies always contain an activation (i.e., an existence constraint), so they usually contain one more element than their potential inconsistency counterpart (UC1). However, activations always represent a “starting point” when trying to make sense of the subset, which users tend to look for (Haisjackson et al., 2016) (UC5). For potential inconsistencies, the activation is not explicit and must be identified, which will likely decrease understanding.

Depending on the inconsistency structure, the minimal number of involved activities (UC1) ranges from one (explicit cardinality contradiction) to three (statements containing a hidden dependency and contradictions about relative positions), with most structures having a minimum of two involved activities (direct opposites, contradictions about fixed positions, implicit cardinality contradictions and loops). While chains and loops can grow infinitely large in terms of their involved constraints and activities, other structures are limited (e.g., contradictory statements about positions). Generally, a constraint can contribute to an inconsistency explicitly or implicitly (Ribeiro and Thimm, 2021), i.e., it can either directly contradict another constraint (explicit) or contradict a statement comprising multiple constraints (implicit). Thus, the overall degree of implicit of an MIS is influenced by the size (UC1) of the statement being contradicted. The degree of implicit is also influenced by the usage of templates (UC1/UC2) and hidden dependencies (UC3). The former can potentially be restricted, depending on the structure of an inconsistency. This applies to the templates themselves (UC2), as
well as the number of different templates (UC1) across one inconsistency (e.g., explicit cardinality restrictions can only comprise exactly two cardinality constraints). When looking at C1-C4 in Table 2, we can observe that the inconsistencies intuitively have increasing degrees of implicitness. While the inconsistency in C1 can be detected right away due to the use of a single template and its exact negation, C2 requires some basic understanding of DECLARE templates and their semantics due to the use of different, but still contradicting templates. C3 also makes use of a single template, but the statement being contradicted is hidden within a chain of two constraints. In C4, the degree of implicitness is even higher due to the presence of a hidden dependency within the statement being contradicted. Thus, to understand this inconsistency, the hidden dependency must be detected and resolved first.

The choice of representation (UC4) generally remains a problem for displaying MIS. Although visualization of DECLARE constraints has been found to be rather confusing for humans, it is important to consider it in studies on inconsistency understanding, as the representation might have different effects on understanding, depending on the characteristics of an inconsistency. For example, a graphical notation as shown in Figure 1 might intuitively help humans to identify loops or chains (MIS3), while inconsistencies involving hidden dependencies (MIS2) might benefit from a textual description. Also, some templates might be easier to visualize than others. MIS are also likely to have a higher degree of connectivity compared to the full model, as all constraints within one MIS are automatically related to each other and are only visually separated in case of additional existence constraints. Thus, a visual representation might aid users in better understanding problematic interrelations, which highlights the need for further investigations of inconsistency visualization.

As this work in progress is the first step towards the development of novel and interactive inconsistency resolution approaches, the type of task (UC7) required from users mainly deals with understanding the problem at hand (i.e., understanding why exactly a particular subset is inconsistent) and – as a next step – being able to adapt the constraints to make the model consistent, while still only allowing the originally desired behavior. Thus, understanding and being able to identify valid and invalid traces is of utmost importance, especially for potential inconsistencies. The remaining identified challenge, namely the background and experience of users (UC6), seems to be a general issue that is not necessarily dependent on the characteristics of an MIS.

5 Conclusion and Future Work

This work deals with the application of insights from DPM understanding to the understanding of inconsistencies. Our results provide the starting point for further analyses, which include a series of experiments analyzing the effects of inconsistency characteristics on understanding accuracy, efficiency, as well as both objective and subjective mental effort. While objective mental effort will be measured using fixation duration obtained from eye-tracking (Meghanathan et al., 2015), subjective mental effort will be measured using post-experiment questions. Additionally, we will make use of think-aloud protocols during the study to gain additional insights into the cognitive processes of the subjects (Ericsson and Simon, 1980). Our planned studies aim to focus on the effects of the – currently rather informally defined – degree of implicitity of MIS on inconsistency understanding. Here it is especially important to investigate the trade-off between size and degree of implicitness, as size has generally been found to increase mental effort; however, large, and rather explicit models might still be easier to understand than small, implicit ones. Furthermore, the type of inconsistency, the contained templates, and their subsumption might affect inconsistency understanding, which is why these aspects will also be empirically evaluated. While the effects of inconsistency metrics and visualizations on inconsistency understanding have been studied in the context of decision tables (Djurica et al., 2020; Nagel et al., 2020, 2019), we are currently not aware of any research on the use of decision support technologies to improve inconsistency understanding in DPM. Thus, we also plan to investigate inconsistency representation, i.e., describing or visualizing the contradiction and its structure in a follow-up experiment. As being able to define and measure understanding is a crucial prerequisite for ranking MIS by the expected comprehension time and mental effort, we are hoping that this work will provide a valuable foundation for novel and interactive inconsistency resolution approaches in DPM.
References


Inconsistency Understanding in Declarative Process Models


