Understanding Quality in Declarative Process Modeling Through the Mental Models of Experts

Amine Abbad Andaloussi[1], Christopher J. Davis[2], Andrea Burattin[1], Hugo A. López[3,5], Tijs Slaats[3], and Barbara Weber[4] *

[1] Software and Process Engineering, Technical University of Denmark
2800 Kgs. Lyngby, Denmark – amab@dtu.dk
[2] University of South Florida, Saint Petersburg, FL, USA
[3] Department of Computer Science, University of Copenhagen, Denmark
[4] Institute of Computer Science, University of St. Gallen, Switzerland
[5] DCR Solutions A/S, Denmark

Abstract. Imperative process models have become immensely popular. However, their use is usually limited to rigid and repetitive processes. Considering the inherent flexibility in most processes in the real-world and the increased need for managing knowledge-intensive processes, the adoption of declarative languages becomes more pertinent than ever. While the quality of imperative models has been extensively investigated in the literature, little is known about the dimensions affecting the quality of declarative models. This work takes an advanced stride to investigate the quality of declarative models. Following the theory of Personal Construct Psychology (PCT), our research introduces a novel method within the Business Process Management (BPM) field to explore quality in the eyes of expert modelers. The findings of this work summarize the dimensions defining the quality of declarative models. The outcome shows the potential of PCT as a basis to discover quality dimensions and advances our understanding of quality in declarative process models.

Keywords: Process Model Understandability, Declarative Process Models, Model Quality, Personal Construct Psychology, Repertory Grid

1 Introduction

In the development of process-aware information systems (PAIS), process models are used for enactment and management purposes [4]. Besides their ability to provide a blueprint for process execution, process models are used for requirement elicitation, communication and process improvement. Process models are expressed using languages from either the *imperative* or *declarative* paradigm. While imperative models describe all the process executions explicitly,

* Work supported by the Innovation Fund Denmark project *EcoKnow* (7050-00034A), the Danish Council for Independent Research project *Hybrid Business Process Management Technologies* (DFF-6111-00337), and the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement BehAPI No.778233.

declarative models rather specify the constraints guiding the overall process and allow any execution not violating the given constraints to occur. When dealing with rigid and repetitive processes, imperative languages are the best candidates. However, when it comes to knowledge-intensive processes where flexibly is an inherent requirement, imperative languages become unable to represent processes concisely. Alternatively, the constraint-based approach of declarative languages allows abstracting the details of specific process executions and modeling the general interplay of events. The flexibility of declarative languages comes at the cost of their understandability [16]. Considering the rich semantics of declarative languages and the different ways in which constraints can interact, it becomes hard for the reader to infer the process executions allowed by the model [34].

To support the understandability of declarative models, several hybrid representations extending models with textual annotations and simulations have emerged (review in [3]). Nevertheless, understandability challenges remained apparent [1]. Refining models to improve their quality is an alternative to overcome these limitations. While there is a rich body of literature investigating the quality of imperative models (e.g., [10, 17, 29, 30]), only a few contributions exploring the comprehension of declarative models exist (e.g., [20, 37]). A review by Corradini et al. [10] identified 50 guidelines addressing the quality of process models. However, many are limited to imperative languages and several of their focal constructs (e.g., gateways, pools and lanes, message events) are not relevant to declarative models. Similarly, the use of a single start event and the necessity to minimize concurrency in the model [10] are guidelines common to imperative modeling that counteract the constraint-based approach of declarative languages. Indeed, declarative models can have several entry-points [37]. Likewise, imposing a sequential-flow would need to overconstrain the declarative model, increasing its complexity - and reducing its understandability. In addition, modeling with constraints introduces conceptual challenges (e.g., hidden dependencies [37]), which are absent when modeling imperatively. Nonetheless, guidelines addressing the visual clarity of models (e.g., avoiding overlapping elements and line crossings) can be applied to both language paradigms. As a step towards the development of a more comprehensive framework for assessing the quality of declarative models, we use Personal Construct Theory (PCT) [24] to elicit quality dimensions used by experts when evaluating declarative models. Afterwards, we turn to the literature to discuss the similarities with existing guidelines and mark the key disparities requiring further investigation.

PCT directly fulfills our aim to elicit the criteria used by experts to judge model quality. It postulates that individuals develop a set of personal constructs (i.e., scales) to frame their experiences based on their similarities and differences [24]. In our context, the constructs offer scalar dimensions used by experts to differentiate the qualities of process models. Tapping into these constructs provide a means to articulate each expert's mental model, making the criteria by which model components are judged more tractable. Moreover, grounding our study in PCT overcomes many of the limitations of interpretive studies exploring the quality of process models, in particular those reliant on techniques such as interviews and think-aloud (e.g., [6, 37]). Insights obtained from interviews are usually bound by the interviewer's questions, leaving no chance to discover other relevant aspects beyond the repertoire of questions. As for think-aloud, it helps people to voice their thoughts out-loud and thus reveal their inner thoughts. However, as individuals tend to know more than they can readily articulate [12], part of their thought remains tacit and not readily evident in verbal utterances. PCT overcomes this limitation by removing the bounds of predetermination - the interview structure - offering in its place a framework for a series of comparisons. The similarities and differences between elements (e.g., those of process models) provide the basis for - and scope of - the technique. Through this comparison process, each individual's constructs can be articulated without constraint. Collectively, these benefits motivate our choice of PCT to articulate the constructs undergirding judgements of quality. Following analysis based on grounded theory [8], the constructs articulated are aggregated to propose a multi-dimensional framework for the assessment of declarative model quality.

Our contribution is threefold. Firstly, we develop a multi-dimensional framework that has the capacity to more comprehensively assess the quality of declarative models. secondly, we demonstrate the potential of PCT in conducting interpretive analysis of process modeling. Our findings enhance the understanding of the dimensions of quality in declarative modeling and promote their use in industry. Moreover, these emergent dimensions of quality have the clear potential to support teaching of declarative modeling, helping students identify pertinent aspects requiring more attention when modeling processes declaratively. Finally, further adoption of PCT in the process modeling field would add to the stream of research exploring the mental models of practitioners. Sect. 2 presents the background, Sect. 3 introduces the related work, Sect. 4 explains the research method, Sect. 5 presents the findings, Sect. 6 discusses the findings and Sect. 7 wraps-up the key contributions and delineates the future work.

2 Background

DCR Graphs. DCR Graphs consist of nodes and edges: the nodes indicate *events*, the edges indicate *relations* between the events. Events can be assigned to *roles*. To maximize flexibility, events that are unconstrained can be executed at any time and any number of times. Events have a state marking, which is a tuple of three Boolean values: *executed*, *included* and *pending*. *Executed* indicates that the event has executed at least once in the past. *Included* indicates whether the event is currently relevant for the process: irrelevant (*excluded*) events cannot be executed, but also cannot constrain the execution of other events. *Pending* indicates that the event must be executed some time in the future, i.e. the event is a requirement that must be fulfilled before we can end the process. Pending events are generally referred as *required events*.

There are five basic relations. A *condition* restricts an event by stating that it cannot be executed before another event has fired at least once. *Milestones* constrain an event by stating that as long as a particular other event is pending, it cannot be executed. The *exclusion* and *inclusion* relations can be used to remove or add back an event from or to the process, effectively toggling event's included state. Finally, the *response* relation indicates that the execution of one event makes another event pending (i.e., required). The last three relations imply a dynamic behavior in the model as they are not constraints in the traditional sense, but rather capture *effects* that some events have on others. Relations and events can be combined together to model specific *behavioral patterns.*

Several extensions complement the core notation above. Hierarchy can be achieved through *nesting* [21], which allows one to group several events together (into a *nest event*), and then add a single relation to or from all of them. It simply acts as a shorthand for having a relation for each individual event and therefore does not add additional semantic meaning. The notion of *multi-instance sub-processes* [13] on the other hand, significantly extends the language by allowing one to model sub-process templates which can be instantiated many times. For example, a funding application round may consist of many individual applications, each application instance having their own unique internal state. Finally one can model the influence of contextual *data* on the process by adding *data expressions* to relations, indicating under what circumstances they should be activated [36]. For example, a response relation between "check expenses report" and "flag report" can be activated only if the amount exceeds a thousand euros.

Mental Models and Personal Construct Psychology. A mental model is an abstract representation of a situation or a system in the individual's mind [18]. Research on mental models addresses two aspects: their structure and change over time. Studies of the structure of mental models contribute to the theory of human reasoning and are used to evaluate individuals' decision making [23]. Change-oriented studies focus on dynamics where the system state changes over time. These studies investigate how individuals' mental models evolve and adapt [19]. In this work, we lean to the former, striving to articulate mental models whose structure reveals experts' judgement of declarative process models. The structure of the mental model - comprised of scalar constructs - provides direct insight into the criteria on which their assessment of quality is based.

To tap into individuals' mental models, we refer to the PCT theory of George Kelly [24]. Kelly assumed that individuals develop unique systems of interrelated personal constructs (i.e., scales), allowing them to understand and predict their surrounding world [24]. These personal constructs emerge from the individuals' past and ongoing experiences. Individuals organize and differentiate their experiences through judgement of similarities and differences, evolving a system of constructs, which they use to frame and predict the consequences of their own actions and interpret those of others [12]. The commonality of a system of constructs enables them to be used as a basis to explain interpersonal relations. This is particularly pertinent to personal experiences that share a cognitive medium or framework. PCT posits that individuals sharing common experiences can develop similar personal constructs [12].

In the view of Kelly, a personal construct is bipolar. It is composed of two ends (e.g., good versus bad). Eliciting constructs is challenging because individuals are generally unable to access the structure of their own cognitive system and verbalize their implicit knowledge [12]. *Repertory Grid* is a knowledge elicitation technique developed to help people identify and articulate their personal constructs [12, 24]. In a nutshell, the approach comprises a series of trials where a participant is asked to identify similarities and differences between different elements – such as process models in DCR Graphs. The result of each comparison is then used to articulate the participant's personal constructs and their meaning. A step-by-step explanation of the Repertory Grid process is provided in Sect. 4.2. Repertory Grid has been used in a wide range of domains (e.g., technology acceptance [12]). However, its potential has not yet been exploited in the field of process modeling. This work builds upon the PCT theory and adapts the Repertory Grid technique to derive a comprehensive framework delineating the dimensions used by experts to evaluate the quality of declarative process models.

Grounded Theory. Grounded Theory adopts a qualitative inductive approach to analyzing and conceptualizing data [8]. A multi-phase process of coding is a central to grounded theory, enabling the phenomena emerging from data to be identified and classified. Three coding techniques – *initial-coding*, *focused-coding* and *axial-coding* – are common [8]. Initial-coding highlights salient aspects in the data; focused coding allows these aspects to be grouped based on similarity of their traits, while axial-coding establishes relationships between the identified codes. Typically, a qualitative analysis starts with initial-coding, followed by focused-coding and finally axial-coding. In model comprehension studies, grounded theory has been used to analyze the verbal utterances of participants when interacting with different representations of process models (e.g., [1, 37]). Building on these works, our analysis uses the coding techniques of grounded theory to analyze the personal constructs verbalized by the experts throughout the different steps of the Repertory Grid.

3 Related Work

Model quality frameworks have emerged in different contexts. In conceptual modeling, guidelines addressing the use of graphical notations and the overall quality of conceptual models have emerged (e.g., [26,28,31]). In process modeling, a large body of literature focusing on the quality of imperative models exists (for an overview see the literature reviews in [10, 17]). In addition, a set of guidelines have been proposed on how to create process models of good quality (e.g., [27, 29, 30, 35]). However, when it comes to declarative languages, only a very limited number of studies exploring specific aspects of declarative models have emerged. Namely, the authors in [20] suggested that the comprehension of declarative models could be affected by the layout and the complexity of the used constraints. As for [37], the author suggested that modularization could support the comprehension of declarative models when solving a particular type of tasks.

Our study differs from the earlier works in several aspects. As opposed to [26, 28, 31] where guidelines are generic to any model-based representation, our work emphasizes declarative models, in particular those in DCR graphs, providing a closer examination of the quality dimensions relevant for that matter. With regards to [10, 17, 27, 29, 30, 35], many of the proposed guidelines either do not apply to declarative models or need further investigation to ensure their applicability (cf. Sect. 1). Alternatively, our research bases its analysis on declarative models and compares to related work to highlight the similarities and disparities between imperative and declarative guidelines (cf. Sect. 6). When it comes to studies looking into declarative process models, we argue that model quality was not well emphasized. Instead, the focus was on exploring the use of declarative models [20] or assessing the impact of modularization [37] on the performance of users. Conversely, our work emphasizes the quality of declarative process models and aims at providing a multi-dimensional quality framework to further promote their use in practice. Besides, our study design (based on PCT, cf. Sect. 4) differs from the existing qualitative designs as explained in Sect. 1.

4 Research Method

This section introduces our research method including the research question (cf. Sect 4.1), data collection (cf. Sect. 4.2) and analysis procedures (cf. Sect. 4.3).

4.1 Research Question

This work addresses the need for a comprehensive framework allowing to evaluate the quality of declarative process models, particularly DCR Graphs. Our research question is formulated as follows: **Which quality dimensions are used by experts when comparing DCR Graphs?**

4.2 Data Collection

Data was collected using a step-wise approach underpinned by PCT. The following sections explain our data collection process in detail, introduce the research setting, and describe the materials used in the study.

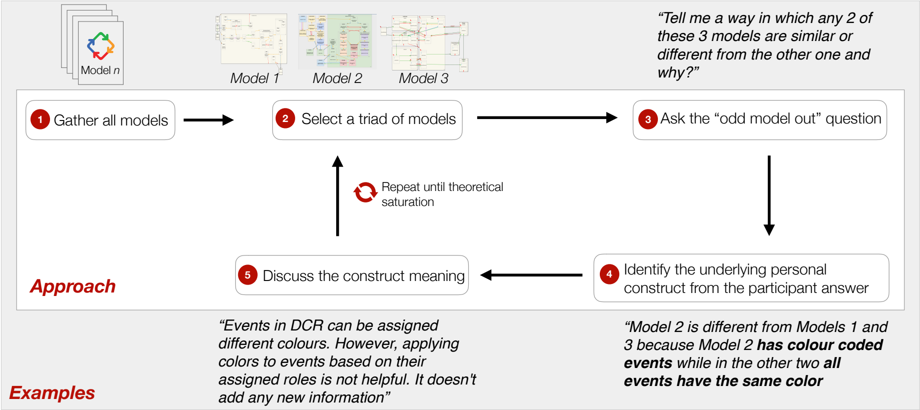
Approach. Following the theoretic position set out in Sect. 2, we use the Repertory Grid to identify the constructs used by experts to evaluate the quality of DCR Graphs. The elicitation process is initiated by the *selection of a set of elements* referring to different instances of a universe of discourse [24]. Repertory Grid studies use different types of elements. In clinical contexts, elements are usually represented as roles (i.e., people); however, in other studies, elements are represented as working tasks [12]. In our study, we consider the elements as models provided by modelers with different levels of expertise. *Collecting the models* representing the elements of the grid is, then, the first phase of our data collection. To this end, we have shared a process description with a set of participants and asked them to design the corresponding model in DCR Graphs. The resulting models are available in our online repository [2].

Once the models defining the elements of the grid have been collected, we move to the second phase of our data collection, where participants recruited for their expertise evaluate the quality of the collected models. This phase begins by *eliciting of personal constructs.* Through a series of trials, the participant is given a triad (i.e., set of three) of models and asked, following the minimum context form described by Kelly [12, 24]), to *(1)* identify the "*odd model out*" (i.e., the model that differs from the other two models of the triad) *(2)* and explain "*why*", that is to say, what –in her terms – makes it odd. This articulates one dimension of the scale used to differentiate the models (elements). The participant is then asked what –if anything– makes the remaining (non-odd) elements similar. Often, this is a simple negation: for instance, a triad composed of 3 process models might be differentiated because one model *has color coded events*, while in the other two *all events have the same color.* In this sense, the construct defined with the poles *has color coded events* versus *all events have the same color* is an example of a participant's personal construct. A construct is thus articulated as two distinct poles drawn from the difference between the odd model and the similarity of the other two models.

The identification of personal constructs is usually complemented by a discussion of the meaning of the constructs to the participant. The discussion is moderated using *laddering up* and *laddering down* techniques used respectively to elaborate or abstract the insights offered by the participant, further articulating their relevance [12]. The same triad approach is repeated until a theoretical saturation of constructs is reached. Rather than data saturation, where all possible triads should be visited, we follow a theoretical saturation approach, striving to provide the participant with new triads until no more new constructs emerge. On average, most constructs were articulated after 7 triads, which falls within the same range of triads generally used to identify the most salient constructs [11]. Fig. 1a summarizes the process of eliciting personal constructs.

Following the identification of constructs, the participant is given a grid where columns represent the collected models and rows show the identified constructs. During this process, the participant is allowed to review and edit her constructs before being asked to *rate each of the models based on the identified constructs.* The literature discusses different rating methods [12], in our study, we use a five-point scale following the insights in [12]. As the constructs usually emerge from comparisons within triads of models, some constructs might not apply to all models. In such a case the participant is told to skip these particular grid cells. Analysis of the numeric ratings enables the grid to illustrate underlying but unseen associations between elements and constructs and thus their meaning using concrete terminology drawn from the participants 'world', which in turn supports the analysis of these personal constructs. A fragment of a Grid is illustrated in Fig. 1b. The collected grids are available in our online repository [2].

The *talkback interview* is the last step. It aims at reflecting the overall process and scrutinize the personal constructs based on the obtained qualitative insights and the grid ratings. While some studies conduct further statistical analyses to investigate the correlations between constructs and elements, our


Model n
Model 1
Model 2
Model 3
"Tell me a way in which any 2 of these 3 models are similar or different from the other one and why?"
1 Gather all models
2 Select a triad of models
3 Ask the "odd model out" question
Repeat until theoretical saturation
4 Identify the underlying personal construct from the participant answer
5 Discuss the construct meaning
Approach
"Events in DCR can be assigned different colours. However, applying colors to events based on their assigned roles is not helpful. It doesn't add any new information"
"Model 2 is different from Models 1 and 3 because Model 2 has colour coded events while in the other two all events have the same color
Examples

(a) The process of eliciting the participants' personal constructs

Personal Construct Poles	Elements						Personal Construct Poles
	Model 1	*Model 2*	*Model 3*	*Model 4*	*Model 5*	*Model 6*	
Has color coded events	5	4	2	5	3	1	*All Events have the same color*
...	3		2		5	1	....

(b) Fragment of a Repertory Grid

Fig. 1: Illustrations of the different steps of the Repertory Grid approach

work rather focuses on the insights obtained throughout the different steps of the Repertory Grid and analyzes them following grounded theory. To keep track of these insights, the conversations with the participants were fully recorded.

Participants. To collect the models representing the elements of the grid, we have recruited 13 participants with different levels of expertise in DCR Graphs. Novice participants (3 students) have taken a BPM course where they have been introduced to process modeling in general. Intermediate participants (4 students) have been familiarized with DCR Graphs for at least one semester. Whereas expert participants (2 professors, 2 postdocs and 2 industry practitioners) are more deeply immersed through their use of and research into DCR Graphs. The heterogeneity of participants enabled us to explore the range of model complexities and reflect different modeling practices employed by users with different levels of expertise. This heterogeneity also provides the basis to allow differences between novices and experts - and novice models and expert models - to emerge.

To evaluate the models, we used 4 experts among the pool of participants in the first phase. Each expert was exposed to the 12 models collected from the other participants and her model as well. Including experts' own models in the comparison gave them the opportunity to reflect on their models (compared to others) which in turn enriched the analysis. Overall, 94 bi-polar personal constructs were elicited from the models [2].

Material. The process description used to collect the models representing the elements of the grid is inspired by a real-world use-case study presented in [15]. The process description (cf. online repository [2]) was shared with 13 participants, who were asked to design the corresponding process model in DCR Graphs.

4.3 Data Analysis

The analysis started by listening to the audio recordings of the repertory grid procedure, timestamping the periods where each of the constructs was discussed, and then taking notes of the collected insights. Here, the verbal utterances provided by each participant were related directly to the ratings of the relevant model in the repertory grid providing concrete, context-specific articulations of the participant's insights. Afterwards, we turned to grounded theory to investigate the participants' constructs and their meanings. To reduce subjectivity during the coding process, we recruited two coders. We followed the code-confirming strategy [25] to distribute the tasks between the primary and the secondary coders. The primary coder was responsible for conducting the first round of coding, while the secondary coder was recruited to critically scrutinize the codes and trigger discussions to improve the coding. Both coders are researchers within the BPM field. For each grid, the primary coder conducted the first round of initial-coding to the participant's constructs [7] based on the constructs' poles. In case the poles were not clear, the primary coder referred to the collected notes. Afterwards, the secondary coder reviewed the initial-coding and performed the second round of coding, which was, in turn, discussed by both coders to reach an agreement. Next, the constructs obtained from all the participants were combined and subjected to focused-coding [7] grouping repeating and overlapping initial codes to identify the commonality or focus among the concepts articulated. The resulting codes reveal the different *dimensions* used by the participants to evaluate the quality of declarative process models. The relationships between the revealed dimensions were elaborated using axial coding [7]. Here, the revealed dimensions were organized according to recurrent themes and then categorized. This phase was conducted in 2 rounds by both coders, followed by a discussion where the final codes were agreed. An excel sheet illustrating this process is available as part of our online repository [2]. The resulting categories, themes and dimensions are presented in Sect. 5.

5 Findings

The analysis of the constructs allowed the identification of seven themes organized into 2 categories. Sections 5.1 and 5.2 present the themes associated with the semantic qualities and pragmatic qualities of process models respectively.

5.1 Semantic Qualities

Semantics denotes the ability of the model to make true statements about the way the business process operates in the real-world [35]. The semantics of a model is a relative indicator of quality as the model behavior is subjective to the process specifications. The analysis of experts' personal constructs, drawn from their interpretation of the models, gave rise to 4 themes overarching a number of dimensions capable of assessing the semantic quality of DCR Graphs: these are *modeling behavior, modeling patterns, modeling events* and *modeling data.*

Modeling Behavior. Within this theme, several dimensions have emerged. *Comprehensiveness of behavior* is identified throughout our analysis of personal constructs. Here, experts used this dimension to evaluate the completeness of the model. When it comes to the alignment between the process specifications and the model behavior, the experts elicited the *presence of behavioral errors* dimension to assess the validity of the behavior supported by the model.

Flow-based versus *declarative modeling* is a relevant dimension used by the experts to evaluate flexibility. They identified a spectrum of modeling behaviors ranging from very flexible to over-restricted ones. Overall, the experts asserted that declarative models should support parallel behavior and avoid being restrictive. Nevertheless, they also advised avoiding both extremes (being too flexible or too restrictive) and advised to rather comply to the *process specifications.*

Modeling of required events is another relevant dimension identified by the experts. This dimension evaluates the modeling of events that must eventually be executed in the process. These events are regarded as goals which must be fulfilled in any execution [22]. Identifying these events in the specifications and modeling them correctly are important criteria to model behavior consistently. In DCR Graphs, required events can be modelled by assigning a specific marking to events at design time or by using the response relation (cf. Sect. 2).

The experts identified the dimension *Modeling of end-events* to assess whether the model allows termination. In DCR Graphs, end-events refer to events whose execution disable the rest of events in the model from executing. While some experts recommended to model termination, other experts asserted that one cannot generalize that all processes should incorporate termination. In some cases, process specifications require processes to be *suspended* rather than terminated, leaving the possibility to resume them at any point in time. For such processes, only the no-longer relevant events should be removed from the process before suspension. Similarly, the experts identified the dimension *Modeling of start-events* (i.e., events initiating the process) to assess whether the models identify the process start-events appropriately. In this respect, some experts advised using a unique start-event, while other experts affirmed that this depends on the process specifications. Nonetheless, experts advised checking whether the non-constrained events in the model are good candidates for being start-events to the process, if not, then these events must be constrained by others to prevent their occurrence when the process is first initiated.

Additionally, the *Multi-instance processing* dimension emerged to compare the extent to which multi-instance sub-processes are supported (cf. Sect. 2). From this perspective, the experts noticed that most of the models do not comply with the given process specifications as they do not offer the possibility to indicate the parts which can be executed multiple times concurrently.

The *Modeling against IT silliness* dimension addresses the experts' felt need to assess the flexibility of the models in tackling failures that prevent occurred events from being registered by the PAIS. In this context, the distinction between *unlawful behavior* (i.e., the behavior violating the constraints of the process) and *impossible behavior* (i.e., a behavior which would never occur in the real-world) has emerged. While the former is crucial to avoid, the latter can be tolerated assuming that the PAIS might fail to register some non-value adding events at their occurrence (e.g., granting a loan without signing the contract must never be allowed, whereas, signing the contract without receiving it, could be tolerated by the model assuming that the PAIS failed to register that event).

The *purpose of the model* is a dimension used by the experts to evaluate the granularity of the models. Accordingly, the level of detail exposed by the scope or bounds of the business process can be adjusted to fit the intended purpose (e.g., enactment, management). The identification of the model purpose is a crucial aspect because it goes beyond the semantic qualities of the model also to affect the pragmatics of the model. In that sense, a model intended for enactment can be hard to interpret if used for management purposes.

Modeling Patterns. Modeling patterns denote the set of mechanisms used to represent specific behaviors when modeling processes. The elicited insights focused on the *use of standard patterns*, which encompass the conventional modeling patterns advised for modeling different behaviors. For experts, standard patterns provide a clear representation of the intended model behavior. The use of standard patterns also reoccurred while inspecting the way modelers represented common behavior, exceptional behavior, and termination.

The dimension *Condition-response* versus *Include-exclude patterns* emerged when comparing the common behavior represented in the models. The *condition* and *response* relations can be used together to model a wide range of specifications. However, a similar behavior can be achieved using *exclude* and *include* relations, which was recurrent in many models. During the discussion, the experts advised adhering to the *condition-response* pattern when modeling common behavior for the following reasons: *(1)* The dynamic behavior of the *include* and *exclude* relations (cf. Sect. 2) is more likely to create hidden dependencies between events, adding unnecessary complexity to the model. *(2)* The *include* and *exclude* relations are rather used for modeling exceptions and termination.

The dimension *Treatment of exception pattern* assesses whether the modeler uses the appropriate pattern to treat exceptions clearly. For the experts, exceptional events are not part of the main process and thus they should initially be excluded in the model and included (using the include relation) only when exceptions occur. Likewise, the dimension *Use of termination pattern* addresses whether termination is modeled using the appropriate pattern. Here, the experts recommended grouping events into a nest event (cf. Sect. 2) and add one exclude relation from the end-event to the nest event.

Modeling Events. The experts used the *Role assignment* dimension to check the assignment of roles to events and asserted that it is crucial for clarifying “who is doing what?”, which in turn supports better traceability and access control.

Use of intermediate events is a pertinent dimension. In DCR, intermediate events denote the events used to enforce specific behaviors, without being explicitly mentioned in the process specifications. Intermediate events can be used to automate some actions or to model decisions. For the experts, although their use might be necessary (e.g., for implementation), intermediate events can hinder the understandability of the model and should be avoided whenever possible.

Besides, the *implicitness of events* dimension was introduced to evaluate whether all the events mentioned in the process specifications are explicitly represented in the model. Indeed, some modelers merged several events into one. For the experts, modelers should ensure a one-to-one correspondence between the events of the process specifications and those represented in the model.

Modeling Data. The dimension *Encoding decisions explicitly or using data expressions* was used by the experts to evaluate whether decisions are encoded using intermediate events or using data expressions. As mentioned in Sect. 2, data events allow assigning values to variables, which in turn are used in the evaluation of data expressions. Following the experts, the activation of the DCR relations in a model can be controlled by assigning them data expressions. At run-time, if the expression evaluates to *true*, then the semantics of the relation applies in the model, otherwise, it doesn't. Data expressions can be difficult to interpret. However if used purposefully for modeling decisions, they can reduce the complexity of the model (e.g., by removing intermediate decision events).

Besides, the experts identified the dimension *Appropriate choice of data types for data variables* to indicate cases where the data types of variables were not correctly chosen. Here, the experts highlighted the necessity of choosing a data type which infers meaning about the use of the variable it represents.

The *Local/global effect of data variables* dimension emerged to describe whether a data variable is evaluated immediately after being assigned a value (using a data event), or postponed to a later stage of processing. On that matter, the experts recommended evaluating data variables immediately after assigning them values, making the correspondence between the data event and its subsequent evaluation clearer. However, depending on the process specifications, an immediate evaluation of data variables is not always feasible. In this case, the experts advised a consistent naming of data events and data variables, making the correspondence between both easily perceived (cf. Sect. 5.2).

5.2 Pragmatic Qualities

Pragmatics denotes the correspondence between the model and the reader's understanding of it [35]. The pragmatic qualities of a model do not formally affect its behavior. However, they might have direct consequences on the use of the model as a communication artifact. The experts' meanings revealed 3 themes related to pragmatic qualities: *Model Layout, Event Layout, Data Layout.*

Model Layout. The experts used the dimension *Alignment and positioning of elements* to appraise the way models are laid out. They highlighted the extraneous visual complexity raising from models where elements (i.e., events, relations) overlap, and advised a careful alignment and spacing of events. Here, two strategies were used: the former evaluates whether the events assigned to the same role are aligned along the same vertical axis, while the second strategy assesses whether the events are aligned following their likely order of occurrence during execution. For the experts, these strategies could improve the pragmatic quality of the model. In addition, the experts looked into the way models were oriented and suggested a left-to-right or top-to-bottom orientation, indicating that start-events should be positioned at the left-most top-most part of the model.

The *grouping of events* dimension evaluates the way events are grouped in the model. Nest events (cf. Sect. 2) allow gathering events belonging to the same phase or assigned to the same role. With a preference for phase-based nesting, the experts associated the use of nesting with an enhanced understandability of the model. In the same vein, multi-level hierarchy was raised by experts to emphasize the benefits of going beyond a single level of nesting.

Visual conciseness focuses on the overall clarity of the model. This dimension was defined by the previously mentioned aspects e.g., alignment and grouping of events, but also in relation to the optimized use of constraints and the absence of intermediate events. These characteristics embrace both pragmatic and semantic qualities, showing that the themes and dimensions emerging within both categories influence the experts' perception of visual conciseness.

Event layout. The experts emphasized particularly the internal pragmatics of events. The dimension *Meaningful naming of events* was used to assess the meaningfulness of events' names. For experts, events should be assigned comprehensible names which can be easily traced back to the process specifications.

Furthermore, the experts used the dimension *Verb-object* versus *noun-based naming of events* to evaluate the phrasing of the events' names. Here, they recommended a verb-object phrasing, except for the intermediate events used for modeling decisions, where a noun-based format could be acceptable.

Color coding was another identified dimension. Although, DCR allows assigning colors to events, some experts were confused by the meaning of these colors, and asserted that they are hard to interpret when no explicit legend is provided. Hence, several experts suggested avoiding to color events.

Data Layout. The dimension *Correspondence between variable names and data events' names* was used by the experts to evaluate whether the data event altering the value of a data variable can be easily recognized in the model. For experts, data events and data variables should be assigned the same name because data variables might not be evaluated immediately after being assigned a value. Hence, with the lack of a clear matching between a variable name and its corresponding data event's name, it becomes hard for the reader to infer the variable's value when being evaluated in a data expression as all the previously executed data events could presumably change the value of that data variable.

6 Discussion

The dimensions identified by the experts share many similarities with the existing imperative process modeling guidelines. For instance, comprehensiveness of behavior and presence of behavioral errors (two of the identified semantic qualities) relate to the notions of completeness (i.e., the coverage of the relevant statements of a particular domain) and validity (i.e., the correctness of the statements in the model) discussed in [28]. Moreover, the importance of designing models fitting their intended purpose (i.e., enactment, management) both in terms of granularity and target audience was not only recognized by our experts, but also emphasized in [28]. In terms of pragmatic qualities, the insights about the alignment and positioning of elements intersect with the findings in [10,30], while the recommendations about assigning meaningful names to events and phrasing them following a verb-object format have been discussed in [10,30]. Regarding the use of colors to mark events, there was no agreement between experts. This concurs with literature on the usage of color in the context of imperative processes which is also inconclusive [5,10].

The use of standard patterns is among the pertinent dimensions, which experts argued it enhances the understandability of the model. While catalogues of patterns showing how to model certain re-occurring problems exist for imperative models [14], we cannot currently rely on such resources when modeling declaratively. Additional research is needed to elicit a catalogue for DCR Graphs and to empirically evaluate its impact on model quality. Our findings show that the general idea of using decomposition to reduce process model complexity is shared with imperative models [37]. However, additional guidelines – on when and how to decompose declarative models – are missing. Decomposition in imperative models involves identifying particular points in the flow where a complex behaviour can be abstracted into an individual step with a single entry and exit point. This is not as easy in declarative modeling, where different parts of the model may interact in different ways, making it challenging to find clear distinctions between the entangling constraints of the model. There is also a need for empirical research on the impact of modularization on the quality of declarative models. Existing research [37] suggests that modularization enables abstraction and information hiding, which in turn supports the comprehension of the model. Contrarily, modularization also risks fragmentation, giving rise to split-attention effects and a need for integration between different parts of the model.

Existing guidelines on the usage of gateways for modeling decisions are not applicable to declarative models, including DCR graphs. Experts mentioned the modeling of decisions using either intermediate events or data expressions. The use of events to model decisions would lead to construct overload as a single notational element is being used to represent multiple concepts (i.e., actions and decisions). Existing research states that construct overload impacts the understandability of the model negatively [31]. Alternatively, experts suggested modeling decisions using data expressions. However, the implications of using data expressions on the understandability of declarative models are questionable and require additional research. Regarding the modeling of start-events, existing guidelines [30] advise use of a single start-event. While some experts agreed, others questioned the general applicability of this guideline and suggested that it depends on the process. Due to the constraint-based approach of declarative languages, any non-constrained event is a possible entry-point to the process. This makes modeling of start-events in declarative languages more complex than imperative languages since in declarative modelers one must check all non-constrained events to ensure that they are good candidate start-events for the process or constrain them to prevent their occurrence when the process is first initiated.

While several insights agree with the literature on imperative process models, our study identified some contradictions. For instance, our findings promote the concurrency of behavior in declarative models, whereas existing guidelines [10] advise minimizing concurrency when modeling imperatively. Moreover, existing guidelines [32] assume that processes should eventually terminate. Conversely, our insights relax this assumption by evoking the possibility of suspension instead of termination. However, little is known about when to use what, which necessitates detailed guidelines. Moreover, while the use of single end-events is recommended to ensure understandable models [30], the impact of modeling processes without explicit end-events is yet to be explored.

The results of this study have impacts on research, education and practice. The insights obtained advance our understanding of quality in declarative models. While several of the findings concur with prior research on imperative modeling, our study also revealed several dimensions where further investigation is required. The positive effects of standard patterns on both quality and comprehension of declarative models suggest a potential hypothesis worthy of test in the light of the existing theory. A further hypothesis might address effects of applying modularization on the understandability of declarative models. Moreover, the applicability of PCT in process modeling paves the path for new studies exploring the mental models of practitioners when dealing with different aspects of process models. With regards to education, our findings support the teaching of declarative process modeling (particularly in DCR Graphs) by providing a set of dimensions allowing students to focus their attention on the pertinent quality aspects to improve their design of declarative models. Our findings also have implications for practice. Several of the identified semantic qualities (relating to modeling of events and data) and pragmatic qualities (related to model, event and data layouts) can be automatically inferred from the model and thus could be implemented by tool vendors to assess the quality of process models at design time offering the potential of customized tool-support for modelers.

Limitations. Our research has some limitations. Our sample is relatively small: however, in common with other Repertory Grid studies (e.g., [9,33]) the scale and richness of the elicitation process gave rise to over 400 numeric data points, highlighting both the cognitive focus and demand of the approach, which required some 4-5 hours per session. Another limitation might arise through bias during the coding procedure. To minimize this risk, we recruited a secondary coder who was purposefully critical of the coding of the primary coder. Finally, our results do not address syntactical qualities since the models were all designed using a tool (i.e., `dcrgraphs.net`) which automatically resolves syntax-dependent errors.

7 Conclusion and Future Work

This work investigates the quality of declarative process models. The results present a set of quality dimensions identified by experts in DCR Graphs. Similarities with existing guidelines highlight qualities shared with imperative models – while clear differences identify candidate aspects worthy of further investigation. Future work could subject the different qualities to further theoretical and empirical investigation. Several hypotheses have already emerged, as noted above. Moreover, our data could be used to investigate how different quality dimensions affect each other. The models provided by the different groups of participants could be further analyzed to discern patterns characterizing the modeling of novices, intermediates and experts, which in turn could guide the profiling of modelers at run-time and optimizing tool support. Our approach also offers sound potential to contribute to studies that explore the mental models of practitioners and their interaction with process models.

References

1. Abbad Andaloussi, A., Burattin, A., Slaats, T., Petersen, C.A., Hildebrandt, T.T., Weber, B.: Exploring the Understandability of a Hybrid Process Design Artifact Based on DCR Graphs. In: BPMDS (2019)
2. Abbad Andaloussi, A., Davis, C.J., Burattin, A., López, H.A., Slaats, T., Weber, B.: Online repository: used material, collected data and full analysis (March 2020), http://doi.org/10.5281/zenodo.3724609
3. Andaloussi, A.A., Burattin, A., Slaats, T., Kindler, E., Weber, B.: On the declarative paradigm in hybrid business process representations: A conceptual framework and a systematic literature study. Information Systems 91, 101505 (2020)
4. Becker, J., Rosemann, M., von Uthmann, C.: Guidelines of Business Process Modeling pp. 30–49 (2000)
5. Bera, P., Soffer, P., Parsons, J.: Using eye tracking to expose cognitive processes in understanding conceptual models. MIS Quarterly 43(4), 1105–1126 (2019)
6. Bernstein, V., Soffer, P.: How does it look? exploring meaningful layout features of process models. In: International Conference on Advanced Information Systems Engineering. pp. 81–86. Springer (2015)
7. Bryant, A., Charmaz, K.: The SAGE Handbook of Current Developments in Grounded Theory. SAGE Publications (2019)
8. Charmaz, K.: Constructing Grounded Theory. Introducing Qualitative Methods series, SAGE Publications (2014)
9. Christie, D.F., Menmuir, J.G.: The repertory grid as a tool for reflection in the professional development of practitioners in early education. Teacher Development 1(2), 205–218 (1997)
10. Corradini, F., Ferrari, A., Fornari, F., Gnesi, S., Polini, A., Re, B., Spagnolo, G.O.: A guidelines framework for understandable bpmn models. Data & Know Engineering 113, 129–154 (2018)
11. Curtis, A.M., Wells, T.M., Higbee, T., Lowry, P.B.: An overview and tutorial of the repertory grid technique in information systems research. Communications of the Association for Information Systems 23(3), 37–62 (2008)
12. Davis, C.J., Hufnagel, E.M.: Through the eyes of experts: A socio-cognitive perspective on the automation of fingerprint work. MIS Quarterly 31(4) (2007)

13. Debois, S., Hildebrandt, T.T., Slaats, T.: Replication, refinement & reachability: complexity in dynamic condition-response graphs. Acta Informatica 55(6) (2018)
14. van Der Aalst, W.M., Ter Hofstede, A.H., Kiepuszewski, B., Barros, A.P.: Workflow patterns. Distributed and parallel databases 14(1), 5–51 (2003)
15. Elgammal, A., Turetken, O., van den Heuvel, W.J., Papazoglou, M.: Formalizing and appling compliance patterns for business process compliance. Software & Systems Modeling 15(1), 119–146 (2016)
16. Fahland, D., Lübke, D., Mendling, J., Reijers, H., Weber, B., Weidlich, M., Zugal, S.: Declarative versus imperative process modeling languages: The issue of understandability. In: Proceedings of EMMSAD. pp. 353–366 (2009)
17. Figl, K.: Comprehension of procedural visual business process models. Business & Information Systems Engineering 59(1) (2017)
18. Forrester, J.W.: Industrial dynamics. 1961. Pegasus Communications (1961)
19. Groesser, S.N., Schaffernicht, M.: Mental models of dynamic systems: taking stock and looking ahead. System dynamics review 28(1), 46–68 (2012)
20. Haisjackl, C., Barba, I., Zugal, S., Soffer, P., Hadar, I., Reichert, M., Pinggera, J., Weber, B.: Understanding declare models: strategies, pitfalls, empirical results. Software and Systems Modeling 15(2), 325–352 (2016)
21. Hildebrandt, T., Mukkamala, R.R., Slaats, T.: Nested dynamic condition response graphs. In: Proceedings of Fundamentals of Software Engineering (2011)
22. Hildebrandt, T.T., Mukkamala, R.R.: Declarative Event-Based Workflow as Distributed Dynamic Condition Response Graphs. EPTCS 69, 59–73 (2011)
23. Johnson-Laird, P.N.: Mental models and deduction. Trends in cognitive sciences 5(10), 434–442 (2001)
24. Kelly, G.A.: The psychology of personal constructs norton. New York (1955)
25. King, N., Horrocks, C., Brooks, J.: Interviews in Qualitative Research. SAGE Publications (2018)
26. Krogstie, J.: Model-based development and evolution of information systems: A Quality Approach. Springer Science & Business Media (2012)
27. Krogstie, J.: Quality in Business Process Modeling. Springer, Cham (2016)
28. Lindland, O.I., Sindre, G., Solvberg, A.: Understanding quality in conceptual modeling. IEEE software 11(2), 42–49 (1994)
29. Mendling, J.: Metrics for process models: empirical foundations of verification, error prediction, and guidelines for correctness. Springer (2008)
30. Mendling, J., Reijers, H.A., van der Aalst, W.M.: Seven process modeling guidelines (7pmg). Information and Software Technology 52(2), 127–136 (2010)
31. Moody, D.: The "physics" of notations: toward a scientific basis for constructing visual notations in software engineering. IEEE Transactions on SE (2009)
32. Moreno-Montes de Oca, I., Snoeck, M.: Pragmatic guidelines for business process modeling. Available at SSRN 2592983 (2014)
33. P, Gary John, K.M.: Developing an expert support system for tender enquiry evaluation: A case study. European Journal of Operational Research 56 (1992)
34. Pesic, M.: Constraint-based workflow management systems : shifting control to users. Ph.D. thesis, TU Eindhoven (2008)
35. Reijers, H.A., Mendling, J., Recker, J.: Business process quality management. In: Handbook on Business Process Management 1, pp. 167–185. Springer (2010)
36. Slaats, T., Mukkamala, R.R., Hildebrandt, T., Marquard, M.: Exformatics declarative case management workflows as dcr graphs. In: BPM proceedings (2013)
37. Zugal, S.: Applying Cognitive Psychology for Improving the Creation, Understanding and Maintenance of Business Process Models. Ph.D. thesis, University of Innsbruck (2013)